# MODELO RELACIÓN DE ENTIDADES



* Supuesto:
  + “. . . Asignar uno o más empleados a un departamento . . .”
  + “. . . Algunos departamentos aún no tienen empleados asignados. . .”

**número nombre ubicación**

**\* o**

**compuesto por**

**nombre puesto**

**DEPARTMENT**

**asignado a**

**número #\***

**EMPLOYEE**

**#\***

**\* o**

* Crear un diagrama de entidad/relación a partir de narrativas o especificaciones de negocio:

**Modelo de Relación de Entidades**

En un sistema eficaz, los datos se dividen en entidades o categorías discretas. Un modelo de relación de entidades (ER) es una ilustración de varias entidades de un negocio y las relaciones entre ellas. Un modelo de ER se deriva de las especificaciones o descripciones de negocio y se crea durante la fase de análisis del ciclo de vida de desarrollo del sistema. Los modelos de ER separan la información que necesita un negocio a partir de las actividades realizadas en el mismo. Aunque los negocios pueden cambiar sus actividades, el tipo de información suele permanecer constante. Por lo tanto, las estructuras de datos también tienden a ser constantes.

**Ventajas del Modelado de ER:**

* + Recopila información para la organización en un formato claro y preciso.
  + Proporciona una imagen clara del ámbito de los requisitos de la información.
  + Proporciona una asignación gráfica de fácil compresión para el diseño de la base de datos.
  + Ofrece un marco efectivo para integrar varias aplicaciones.



Identificador único (UID) Primario marcado con “#” Secundario marcado con “(#)”

**número nombre ubicación**

**\* o**

**compuesto por**

**nombre puesto**

**DEPARTMENT**

**asignado a**

**número #\***

**EMPLOYEE**

**#\***

**\* o**

Atributo:

* Nombre en singular
* Minúsculas
* Atributo obligatorio marcado con \*
* Atributo opcional marcado con “o”
* Nombre único en singular
* Mayúsculas
* Recuadro editable
* Sinónimo entre paréntesis

Entidad:

**Convenciones de Modelado de Relación de Entidades**

**Componentes Clave**

* + **Entidad:** aspecto significativo sobre qué información se debe conocer. Algunos ejemplos pueden ser los departamentos, empleados y pedidos.
  + **Atributo:** algo que describe o cualifica una entidad. Por ejemplo, para la entidad de empleados, los atributos serían el número de empleado, el nombre, cargo, fecha de contratación, número de departamento, etc. Cada uno de los atributos puede ser necesario u opcional. Este estado se denomina *opciones*.
  + **Relación:** asociación con nombre entre las entidades que muestran opciones y grados. Algunos ejemplos pueden ser los

empleados y departamentos, pedidos y elementos.

**Entidades**

Para representar una entidad, utilice las siguientes convenciones:

* + Nombre de entidad único en singular
  + Nombre de entidad en mayúscula
  + Recuadro editable
  + Nombres de sinónimos opcionales en mayúsculas entre paréntesis: ( )

**Atributos**

Para representar un atributo en un modelo utilice las siguientes convenciones:

* + Nombre en singular y en minúscula
  + Etiqueta de asterisco (\*) para los atributos obligatorios (es decir, valores que se *deben* conocer)
  + Etiqueta de letra “o” para los atributos opcionales (es decir, los valores que se *pueden* conocer)

**Relaciones**

|  |  |
| --- | --- |
| **Símbolo** | **Descripción** |
| Línea discontinua | Elemento opcional que indica “quizás”. |
| Línea continua | Elemento obligatorio que indica “debe ser”. |
| Pata de gallo | Elemento de grado que indica “uno o más”. |
| Una sola línea | Elemento de grado que indica “único”. |

Cada dirección de la relación contiene:

* + **Una etiqueta:** por ejemplo, *enseñado por* o *asignado a*
  + **Una opción:** *debe ser* o *quizás*
  + **Un grado:** *único* o *uno o más*

**Nota:** el término *cardinalidad* es un sinónimo del término *grado*.

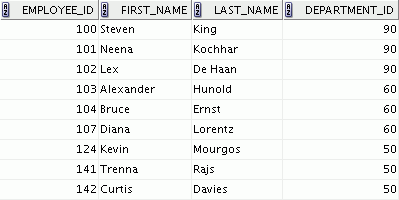
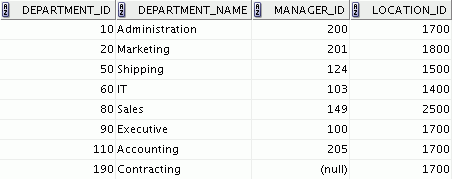
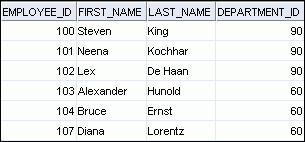
Cada entidad de origen {puede ser | debe ser} nombre de relación {único | uno o más} entidad de destino.

**Nota:** la convención se lee hacia la izquierda.

**Identificadores Únicos**

Un identificador único (UID) es cualquier combinación de atributos o relaciones, o de ambos, que sirve para distinguir las incidencias de una entidad. Cada incidencia de entidad se debe identificar de forma única.

* + Etiquetar cada atributo que forma parte del UID con un signo de almohadilla “#”.
  + Etiquetar los UID secundarios con un signo de almohadilla entre paréntesis (#).



**Clave ajena**

**Clave primaria**

**Clave primaria**

**Nombre de la tabla: EMPLOYEES**

* Cada fila de datos de una tabla se identifica como única mediante una clave primaria.
* Puede relacionar de forma lógica desde varias tablas mediante claves ajenas.

**Nombre de la tabla: DEPARTMENTS**

**Relación de Varias Tablas**

**Relación de Varias Tablas**

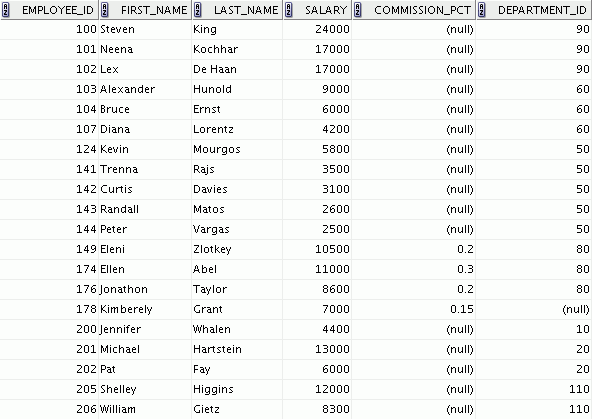
Cada tabla contiene datos que describen exactamente una entidad. Por ejemplo, la tabla EMPLOYEES contiene información sobre los empleados. Las categorías de datos se muestran en la parte superior de cada tabla y los casos individuales se muestran a continuación. Si utiliza un formato de tabla, puede visualizar, comprender y utilizar la información de inmediato.

Debido a que los datos sobre las diferentes entidades se almacenan en diferentes tablas, puede que tenga que combinar dos o más tablas para responder a una pregunta determinada. Por ejemplo, puede que desee conocer la ubicación del departamento en el que trabaja un empleado. En este supuesto, necesita información de la tabla EMPLOYEES (que contiene datos sobre los empleados) y la tabla DEPARTMENTS (que contiene información sobre los departamentos). Con RDBMS, puede relacionar los datos de una tabla con los datos de otra utilizando las claves ajenas. Una clave ajena es una columna (o juego de columnas) que hace referencia a una clave primaria en la misma tabla o en otra tabla.

Tiene la posibilidad de relacionar datos de una tabla con datos de otra tabla para organizar la información en unidades separadas y gestionables. Los datos de empleado se pueden mantener, de forma lógica, separados de los datos de departamento almacenándolos en una tabla independiente.

**Instrucciones para Claves Primarias y Ajenas**

* + No puede utilizar valores duplicados en una clave primaria.
  + Generalmente, las claves primarias no se pueden cambiar.
  + Las claves ajenas se basan en valores de datos y son punteros puramente lógicos (no físicos).
  + Un valor de clave ajena debe coincidir con un valor de clave primaria o de clave única existente o, de lo contrario, debe ser nulo.
  + Una clave ajena debe hacer referencia a una columna de clave primaria o única.



**Terminología de Bases de Datos Relacionales**

**3**

**2**

**4**

**5**

**1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **6** |  |
|  |
|  |
|  |
|  |
|  |  |  |  |  |
|  |  |  |  |  |

**Terminología de Bases de Datos Relacionales**

Una base de datos relacional puede contener una o más tablas. Una *tabla* es la estructura básica de almacenamiento de RDBMS. Una tabla contiene todos los datos necesarios sobre algo del mundo real, como empleados, facturas o clientes.

La diapositiva muestra el contenido de la tabla o relación EMPLOYEES. Los números indican lo siguiente

1. Una única *fila* (o *tupla*)*,* que representa todos los datos necesarios para un empleado concreto. Cada fila de una tabla se debe identificar por una clave primaria, que no permite duplicar filas. El orden de las filas no es importante; especifique el orden de la fila cuando se recuperen los datos.
2. Una *columna* o atributo*,* que contiene el número de empleado. El número de empleado identifica un *único* empleado en la tabla EMPLOYEES. En este ejemplo, la columna del número de empleado está designada como *clave primaria*. Una clave primaria debe contener un valor y el valor debe ser único.
3. Una columna que no es un valor clave. Una columna representa un tipo de dato en una tabla; en este ejemplo, los datos son los salarios de todos los empleados. El orden de las columnas no es importante al almacenar datos; especifique el orden de la columna al recuperar los datos.
4. Una columna que contiene el número de departamento, que también es una *clave ajena*. Una clave ajena es una columna que define cómo se relacionan las tablas sí. Una clave ajena hace referencia a una clave primaria o única en la misma tabla o en otra tabla. En el ejemplo, DEPARTMENT\_ID identifica de forma única un departamento en la tabla DEPARTMENTS.
5. Un *campo* está en la intersección de una fila y una columna. Sólo puede contener un valor.
6. Un campo puede no contener ningún valor. Esto se denomina valor nulo. En la tabla EMPLOYEES solo aquellos empleados que tienen el rol de vendedor tienen un valor en el campo COMMISSION\_PCT (comisiones).

# TIPOS DE DATOS

|  |  |
| --- | --- |
| Tipo de Dato | Descripción |
| VARCHAR2(*size*) | Datos de caracteres de longitud variable |
| CHAR(*size*) | Datos de caracteres de longitud fija |
| NUMBER(*p*,*s)* | Datos numéricos de longitud variable |
| DATE | Valores de fecha y hora |
| LONG | Datos de caracteres de longitud variable (hasta 2 GB). |
| CLOB | Datos de caracteres (hasta 4 GB). |
| RAW and LONG RAW | Datos binarios raw |
| BLOB | Datos binarios (hasta 4 GB) |
| BFILE | Datos binarios almacenados en un archivo externo (hasta 4 GB) |
| ROWID | Sistema numérico de base -64 que representa la dirección única de una fila en su tabla correspondiente. |



**Tipos de Dato**

Al identificar una columna para una tabla, debe proporcionar un tipo de dato para la columna. Hay varios tipos de dato disponibles:

|  |  |  |
| --- | --- | --- |
| **Tipo de Dato** | **Descripción** | |
| **VARCHAR2(*size*)** | Datos de caracteres de longitud variable (Se debe especificar un tamaño máximo: el tamaño mínimo es 1 y el máximo es 4.000). | |
| **CHAR [(*size*)]** | Datos de caracteres de longitud fija de tamaño de bytes (El tamaño por defecto y mínimo es 1; el tamaño máximo es 2.000). | |
| **NUMBER [(*p,s*)]** | Número con la precisión *p* y la escala *s* (La precisión es el número total de cifras decimales y la escala el número de dígitos a la derecha del punto decimal; la precisión puede oscilar entre 1 y 38 y la escala entre –84 y 127). | |
| **DATE** | Valores de fecha y hora hasta el segundo más cercano entre el 1 de enero, 4712 a. C., y el 31 de diciembre, 9999 d. C. | |
| **LONG** | Datos de caracteres de longitud variable (hasta 2 GB). | |
| **CLOB** | Datos de caracteres (hasta 4 GB). | |
| **RAW(*size*)** | | Datos binarios raw de tamaño de longitud (Se debe especificar un tamaño máximo: El tamaño máximo es 2.000.) |
| **LONG RAW** | | Datos binarios raw de longitud variable (hasta 2 GB). |
| **BLOB** | | Datos binarios (hasta 4 GB) |
| **BFILE** | | Datos binarios almacenados en un archivo externo (hasta 4 GB). |
| **ROWID** | | Sistema numérico de base 64 que representa la dirección única de una fila en su tabla correspondiente. |

**Instrucciones**

* + Las columnas LONG no se copian al crear una tabla mediante una subconsulta.
  + Las columnas LONG no se pueden incluir en una cláusula GROUP BY u ORDER BY.
  + Sólo se puede utilizar una columna LONG por tabla.
  + No se pueden definir restricciones en las columnas LONG.
  + Puede que desee utilizar una columna CLOB en lugar de una columna LONG.

|  |  |
| --- | --- |
| **Tipo de Dato** | **Descripción** |
| TIMESTAMP | Fecha con segundos fraccionarios |
| INTERVAL YEAR TO MONTH | Almacenados como un intervalo de años y meses |
| INTERVAL DAY TO SECOND | Almacenado como un intervalo de días, horas, minutos y segundos |

## Tipos de Dato de Fecha y Hora

|  |  |
| --- | --- |
| **Tipo de Dato** | **Descripción** |
| **TIMESTAMP** | Permite almacenar los datos de tiempo como fecha con segundos fraccionarios. Almacena los valores de año, mes, día, hora, y segundos del tipo de dato DATE, así como el valor para segundos fraccionarios. Hay distintas variantes de este tipo de dato, entre otras, WITH  TIMEZONE, WITH LOCALTIMEZONE. |
| **INTERVAL YEAR TO MONTH** | Permite almacenar el tiempo como un intervalo de años y meses.  Utilizado para representar la diferencia entre dos valores de fecha y hora en los que las únicas partes significativas son el año y el mes. |
| **INTERVAL DAY TO SECOND** | Permite almacenar el tiempo como un intervalo de días, horas, minutos y  segundos. Utilizado para representar la diferencia exacta entre dos valores de fecha y hora. |

## Tipo de datos BLOB

El tipo de datos BLOB almacena grandes objetos binarios no estructurados. Los objetos BLOB pueden considerarse como secuencias de bits sin semántica de conjuntos de caracteres.  los objetos BLOB pueden almacenar datos binarios hasta (4 gigabytes -1) \* (el valor del  parámetro de almacenamiento CHUNK LOB). Si los espacios de tablas en su base de datos son de tamaño de bloque estándar, y si ha usado el valor predeterminado del  parámetro de almacenamiento CHUNK de LOB al crear una columna LOB, esto es equivalente a (4 gigabytes - 1) \* (tamaño de bloque de base de datos).

Los objetos BLOB tienen soporte transaccional completo. Los cambios realizados a través de SQL, el paquete DBMS\_LOB o la interfaz de llamada de Oracle (OCI) participan completamente en la transacción.  Las manipulaciones de valor BLOB pueden ser cometidas y revertidas. Sin embargo, no puede guardar un localizador BLOB en una variable PL / SQL u OCI en una transacción y luego usarlo en otra transacción o sesión.

El tipo CLOB de datos almacena datos de caracteres de un solo byte y multibyte. Se admiten los juegos de caracteres de ancho fijo y de ancho variable, y ambos usan el juego de caracteres de la base de datos. los objetos CLOB pueden almacenar hasta (4 gigabytes -1) \* (el valor del parámetro CHUNK de almacenamiento LOB) de datos de caracteres. Si los espacios de tablas en su base de datos son de tamaño de bloque estándar, y si ha usado el valor predeterminado del parámetro CHUNK de almacenamiento de LOB al crear una columna LOB, esto es equivalente a (4 gigabytes - 1) \* (tamaño de bloque de base de datos).

Los objetos CLOB tienen soporte transaccional completo. Los cambios realizados a través de SQL, el paquete DBMS\_LOB o la interfaz de llamada de Oracle (OCI) participan completamente en la transacción. Las manipulaciones CLOB de valor pueden ser cometidas y revertidas. Sin embargo, no puede guardar un localizador CLOB en una variable PL / SQL u OCI en una transacción y luego usarlo en otra transacción o sesión.

## Tipo de datos NCLOB

El tipo de datos NCLOB almacena datos Unicode. Se admiten los juegos de caracteres de ancho fijo y de ancho variable, y ambos usan el juego de caracteres nacional. los objetos NCLOB pueden almacenar hasta (4 gigabytes -1) \* (el valor del parámetro CHUNK de almacenamiento LOB) de datos de texto de caracteres. Si los espacios de tablas en su base de datos son de tamaño de bloque estándar, y si ha usado el valor predeterminado del parámetro CHUNK de almacenamiento de LOB al crear una columna LOB, esto es equivalente a (4 gigabytes - 1) \* (tamaño de bloque de base de datos).

Los objetos NCLOB tienen soporte transaccional completo. Los cambios realizados a través de SQL, el paquete DBMS\_LOB u OCI participan completamente en la transacción.  Las manipulaciones de valor NCLOB pueden ser cometidas y revertidas. Sin embargo, no puede guardar un localizador NCLOB en una variable PL / SQL u OCI en una transacción y luego usarlo en otra transacción o sesión.

## Tipo de datos BFILE

El BFILEtipo de datos permite el acceso a los LOB de archivos binarios que se almacenan en sistemas de archivos fuera de la base de datos Oracle. Una columnas BFILE o atributo almacena un localizador BFILE, que sirve como un puntero a un archivo binario en el sistema de archivos del servidor. El localizador mantiene el nombre del directorio y el nombre del archivo.

Puede cambiar el nombre de archivo y la ruta de acceso de a BFILE sin afectar la tabla base mediante la función BFILENAME.

Los LOB de archivos binarios no participan en las transacciones y no son recuperables. Más bien, el sistema operativo subyacente proporciona integridad y durabilidad de los archivos. los datos BFILE pueden ser de hasta 2 64 -1 bytes, aunque su sistema operativo puede imponer restricciones a este máximo.

El administrador de la base de datos debe asegurarse de que exista el archivo externo y que los procesos de Oracle tengan permisos de lectura del sistema operativo en el archivo.

El tipo de datos BFILE permite el soporte de solo lectura de archivos binarios grandes. No puede modificar o replicar tal archivo. Oracle proporciona API para acceder a los datos de archivos. Las interfaces principales que utiliza para acceder a los datos del archivo son el paquete DBMS\_LOB y la Interfaz de llamada de Oracle (OCI).

## Tipo de datos ROWID

Cada fila en la base de datos tiene una dirección. Las secciones que siguen describen las dos formas de dirección de fila en una base de datos Oracle.

Las filas de las tablas organizadas en heap que son nativas de la base de datos Oracle tienen direcciones de fila llamadas **rowids** . Puede examinar una dirección de fila de rowid consultando la pseudocolumna ROWID. Los valores de esta pseudocolumna son cadenas que representan la dirección de cada fila. Estas cadenas tienen el tipo de datos ROWID. También puede crear tablas y grupos que contengan columnas reales que tengan el tipo de datos ROWID. La base de datos de Oracle no garantiza que los valores de dichas columnas sean filas válidas.

**Nota:**

A partir de Oracle8, Oracle SQL incorporó un formato extendido para que los rowids admitieran de manera eficiente tablas e índices particionados y direcciones de bloque de datos relativas al espacio de tabla sin ambigüedad. Si está ejecutando la versión 7 de la base de datos y tiene la intención de actualizar, use el paquete DBMS\_ROWID para migrar los rowids de sus datos al formato extendido.

Los ROWIDS contienen la siguiente información:

* El **bloque** de datos del archivo de datos que contiene la fila. La longitud de esta cadena depende de su sistema operativo.
* La **fila** en el bloque de datos.
* El **archivo de base de datos que** contiene la fila. El primer archivo de datos tiene el número 1. La longitud de esta cadena depende de su sistema operativo.
* El **número de objeto de datos** , que es un número de identificación asignado a cada segmento de la base de datos. Puede recuperar el número de objetos de datos de las vistas del diccionario de datos USER\_OBJECTS, DBA\_OBJECTS y ALL\_OBJECTS. Los objetos que comparten el mismo segmento (tablas agrupadas en el mismo clúster, por ejemplo) tienen el mismo número de objeto.

Los filamentos se almacenan como valores de base 64 que pueden contener los caracteres AZ, az, 0-9 y el signo más (+) y la barra diagonal (/). Los filaidos no están disponibles directamente. Puede utilizar el paquete suministrado DBMS\_ROWID para interpretar los contenidos de rowid. Las funciones del paquete extraen y proporcionan información sobre los cuatro elementos de rowid enumerados anteriormente.

## Tipo de datos UROWID

Las filas de algunas tablas tienen direcciones que no son físicas o permanentes o que no fueron generadas por Oracle Database. Por ejemplo, las direcciones de fila de las tablas organizadas por índice se almacenan en hojas de índice, que pueden moverse. Los filaidos de tablas externas (como las tablas de DB2 a las que se accede a través de una puerta de enlace) no son filas de fila de Oracle estándar.

Oracle utiliza rowids universales ( **urowids** ) para almacenar las direcciones de tablas organizadas por índice y extranjeras. Las tablas organizadas por índice tienen urowids lógicos y las tablas extranjeras tienen urowids extranjeros. Ambos tipos de urowid se almacenan en la ROWID pseudocolumna (al igual que los filaidos físicos de las tablas organizadas en heap).

Oracle crea rowids lógicos basados ​​en la clave principal de la tabla. Los filaidos lógicos no cambian mientras la clave principal no cambie. La pseudocolumna ROWID de una tabla organizada por índice tiene un tipo de datos de UROWID. Puede acceder a esta pseudocolumna como lo haría con la pseudocolumna ROWID de una tabla organizada en heap (utilizando una declaración SELECT... ROWID). Si desea almacenar los rowids de una tabla organizada por índice, entonces puede definir una columna de tipo UROWID para la tabla y recuperar el valor de la pseudocolumna ROWID en esa columna.

## Tipos de datos REF

Un **identificador de objeto** (representado por la palabra clave OID) identifica de forma única un objeto y le permite hacer referencia al objeto desde otros objetos o desde tablas relacionales. Una categoría de tipo de datos llamada REF representa tales referencias. Un REF tipo de datos es un contenedor para un identificador de objeto. Los valores REF son punteros a objetos.

Cuando un valor REF apunta a un objeto inexistente, REF se dice que está "colgando". Un colgante REF es diferente de un nulo REF. Para determinar si a REF está colgando o no, use la condición IS[ NOT] DANGLING. Por ejemplo, dada la vista de objeto oc\_orders en el esquema de muestra oe, la columna customer\_ref es de tipo REF a tipo customer\_typ, que tiene un atributo cust\_email:

SELECT o.customer\_ref.cust\_email

FROM oc\_orders o

WHERE o.customer\_ref IS NOT DANGLING;

## Varrays

Una matriz es un conjunto ordenado de elementos de datos. Todos los elementos de una matriz dada son del mismo tipo de datos. Cada elemento tiene un **índice**, que es un número correspondiente a la posición del elemento en la matriz.

El número de elementos en una matriz es el tamaño de la matriz. Las matrices de Oracle son de tamaño variable, por lo que se denominan **varrays** . Debe especificar un tamaño máximo cuando declare la varray.

Cuando declara una **varray**, no asigna espacio. Define un tipo, que puedes usar como:

* El tipo de datos de una columna de una tabla relacional.
* Un atributo de tipo de objeto.
* Una variable PL/SQL, parámetro, o tipo de retorno de función.

*Oracle* normalmente almacena un objeto de matriz en línea (como parte de los datos de la fila) o fuera de línea (en un LOB), dependiendo de su tamaño. Sin embargo, si especifica características de almacenamiento separadas para una variedad, entonces Oracle las almacena fuera de línea, independientemente de su tamaño.

## Tablas anidadas

Un tipo de tabla anidada modela un conjunto desordenado de elementos. Los elementos pueden ser tipos incorporados (predefinidos) o tipos definidos por el usuario. Puede ver una tabla anidada como una tabla de una sola columna o, si la tabla anidada es un tipo de objeto, como una tabla de varias columnas, con una columna para cada atributo del tipo de objeto.

Una definición de tabla anidada no asigna espacio. Define un tipo, que puedes usar para declarar:

* El tipo de datos de una columna de una tabla relacional.
* Un atributo de tipo de objeto.
* Una variable PL / SQL, parámetro, o tipo de retorno de función.

Cuando una tabla anidada aparece como el tipo de una columna en una tabla relacional o como un atributo del tipo de objeto subyacente de una tabla de objetos, *Oracle* almacena todos los datos de la tabla anidada en una sola tabla, que asocia con el relacional adjunto o tabla de objetos

## ANYDATASET

Este tipo contiene una descripción de un tipo dado más un conjunto de instancias de datos de ese tipo. ANYDATASET se puede utilizar como un tipo de datos de parámetros de procedimiento donde se necesita tal flexibilidad. Los valores de las instancias de datos pueden ser de tipos integrados de SQL, así como tipos definidos por el usuario.

## Tipos de XML

Extensible Markup Language (XML) es un formato estándar desarrollado por el World Wide Web Consortium (W3C) para representar datos estructurados y no estructurados en la World Wide Web. Los identificadores de recursos universales (URI) identifican recursos como las páginas web en cualquier lugar de la web. Oracle proporciona tipos para manejar datos de URI y XML, así como una clase de URI llamados tipos DBURIRef para acceder a los datos almacenados dentro de la base de datos. También proporciona un conjunto de tipos para almacenar y acceder a los URI externos e internos desde la base de datos.

### XMLType

Este tipo suministrado por *Oracle* se puede usar para almacenar y consultar datos XML en la base de datos. XMLType tiene funciones miembro que puede usar para acceder, extraer y consultar los datos XML usando expresiones XPath. XPath es otro estándar desarrollado por el comité del W3C para atravesar documentos XML. Las funciones de Oracle XMLType soportan muchas expresiones W3C XPath. *Oracle* también proporciona un conjunto de funciones SQL y paquetes PL / SQL para crear valores XMLType a partir de datos relacionales existentes o datos relacionales de objetos.

XMLType es un tipo definido por el sistema, por lo que puede usarlo como un argumento de una función o como el tipo de datos de una tabla o columna de vista. También puede crear tablas y vistas de XMLType. Cuando crea una columna XMLType en una tabla, puede optar por almacenar los datos XML en una columna CLOB, como XML binario (almacenado internamente como a CLOB), u objeto de forma relacional.

También puede registrar el esquema (utilizando el paquete DBMS\_XMLSCHEMA) y crear una tabla o columna que se ajuste al esquema registrado. En este caso, Oracle almacena los datos XML en columnas subyacentes de objetos relacionales de forma predeterminada, pero puede especificar el almacenamiento en una columna CLOB XML o binaria incluso para datos basados ​​en esquemas.

Las consultas y el DML en las columna XMLType funcionan de la misma manera, independientemente del mecanismo de almacenamiento.

### Modelo de formato XML

La función SYS\_XMLGEN devuelve una instancia de tipo que XMLType contiene un documento XML. Oracle proporciona el objeto XMLFormat, que le permite formatear la salida de la función SYS\_XMLGEN.

[La Tabla 3-19](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements004.htm#g195161) enumera y describe los atributos del objeto XMLFormat. La función que implementa este tipo sigue la tabla.

***Tabla 3-19 Atributos del objeto XMLFormat***

| **Atributo** | **Tipo de datos** | **Propósito** |
| --- | --- | --- |
| enclTag | VARCHAR2(4000) | El nombre de la etiqueta adjunta para el resultado de la función  SYS\_XMLGEN. Si la entrada a la función es un nombre  de columna, entonces el valor predeterminado es el nombre  de la columna. De lo contrario, el valor predeterminado es ROW.   Cuando schemaType se establece en USE\_GIVEN\_SCHEMA,  este atributo también proporciona el nombre del elemento  XMLSchema. |
| schemaType | VARCHAR2(100) | El tipo de generación de esquema para el documento de salida.  Los valores válidos son NO\_SCHEMA" y USE\_GIVEN\_SCHEMA".  El valor predeterminado es ' NO\_SCHEMA'. |
| schemaName | VARCHAR2(4000) | El nombre del esquema de destino que utiliza Oracle si el valor  de schemaType es ' USE\_GIVEN\_SCHEMA'.  Si lo especifica schemaName, entonces Oracle utiliza la  etiqueta adjunta como nombre del elemento. |
| targetNameSpace | VARCHAR2(4000) | El espacio de nombres de destino si se especifica el  esquema (es decir, schemaType es GEN\_SCHEMA\_\* o  USE\_GIVEN\_SCHEMA) |
| dburlPrefix | VARCHAR2(4000) | La URL a la base de datos para usar si WITH\_SCHEMA  se especifica. Si no se especifica este atributo,  Oracle declara la URL a los tipos como una referencia  de URL relativa. |
| processingIns | VARCHAR2(4000) | Instrucciones de procesamiento proporcionadas  por el usuario, que se adjuntan a la parte superior de la salida  de la función antes del elemento. |

La función que implementa el objeto XMLFormat sigue:

STATIC FUNCTION createFormat(

enclTag IN varchar2 := 'ROWSET',

schemaType IN varchar2 := 'NO\_SCHEMA',

schemaName IN varchar2 := null,

targetNameSpace IN varchar2 := null,

dburlPrefix IN varchar2 := null,

processingIns IN varchar2 := null) RETURN XMLGenFormatType

deterministic parallel\_enable,

MEMBER PROCEDURE genSchema (spec IN varchar2),

MEMBER PROCEDURE setSchemaName(schemaName IN varchar2),

MEMBER PROCEDURE setTargetNameSpace(targetNameSpace IN varchar2),

MEMBER PROCEDURE setEnclosingElementName(enclTag IN varchar2),

MEMBER PROCEDURE setDbUrlPrefix(prefix IN varchar2),

MEMBER PROCEDURE setProcessingIns(pi IN varchar2),

CONSTRUCTOR FUNCTION XMLGenFormatType (

enclTag IN varchar2 := 'ROWSET',

schemaType IN varchar2 := 'NO\_SCHEMA',

schemaName IN varchar2 := null,

targetNameSpace IN varchar2 := null,

dbUrlPrefix IN varchar2 := null,

processingIns IN varchar2 := null) RETURN SELF AS RESULT

deterministic parallel\_enable,

STATIC function createFormat2(

enclTag in varchar2 := 'ROWSET',

flags in raw) return sys.xmlgenformattype

deterministic parallel\_enable

);

## Tipos de datos URI

Oracle proporciona una familia de tipos-URI URIType, DBURIType, XDBURIType, y HTTPURIType que están relacionados por una jerarquía de herencia. URIType es un tipo de objeto y los demás son subtipos de URIType. Dado que URIType es el supertipo, puede crear columnas de este tipo y almacenar DBURIType o HTTPURIType escribir instancias en esta columna.

**HTTPURIType:**Puede usarlo HTTPURIType para almacenar URL en páginas web externas o en archivos. *Oracle* accede a estos archivos mediante HTTP (Protocolo de transferencia de hipertexto).

**XDBURIType:**Puede usar XDBURIType para exponer documentos en la jerarquía de base de datos XML como URI que se pueden incrustar en cualquier columna URIType de una tabla. El XDBURIType consta de un URL, que comprende el nombre jerárquico del documento XML al que se refiere y un fragmento opcional que representa la sintaxis XPath. El fragmento está separado de la parte de la URL por un signo de número (#). Las siguientes líneas son ejemplos de XDBURIType:

/home/oe/doc1.xml

/home/oe/doc1.xml#/orders/order\_item

**DBURIType:**DBURIType se puede utilizar para almacenar valores DBURIRef, que hacen referencia a los datos dentro de la base de datos. El almacenamiento de valores DBURIRef le permite hacer referencia a los datos almacenados dentro o fuera de la base de datos y acceder a los datos de manera consistente.

los valores DBURIRef utilizan una representación similar a XPath para hacer referencia a los datos dentro de la base de datos. Si imagina la base de datos como un árbol XML, verá las tablas, filas y columnas como elementos en el documento XML. Por ejemplo, el usuario de recursos humanos de ejemplo hr verá el siguiente árbol XML:

<HR>

<EMPLOYEES>

<ROW>

<EMPLOYEE\_ID>205</EMPLOYEE\_ID>

<LAST\_NAME>Higgins</LAST\_NAME>

<SALARY>12008</SALARY>

.. <!-- other columns -->

</ROW>

... <!-- other rows -->

</EMPLOYEES>

<!-- other tables..-->

</HR>

<!-- other user schemas on which you have some privilege on..-->

El DBURIRef es una expresión XPath sobre este documento XML virtual. Para hacer referencia al valor SALARY en la tabla EMPLOYEES para el empleado con el número de empleado 205, puede escribir un DBURIRef como,

/HR/EMPLOYEES/ROW[EMPLOYEE\_ID=205]/SALARY

Usando este modelo, puede hacer referencia a los datos almacenados en columnas CLOB u otras columnas y exponerlos como URL al mundo externo.

### Paquete URIFactory

Oracle también proporciona el paquete URIFactory, que puede crear y devolver instancias de los diversos subtipos de URITypes. El paquete analiza la cadena de URL, identifica el tipo de URL (HTTP, DBUR etc.) y crea una instancia del subtipo. Para crear una DBURI instancia, la URL debe comenzar con el prefijo /oradb. Por ejemplo,  URIFactory.getURI('/oradb/HR/EMPLOYEES') crearía una instancia DBURIType y URIFactory.getUri('/sys/schema') crearía una instancia XDBURIType.

## Tipos espaciales

Oracle Spatial está diseñado para hacer que la administración de datos espaciales sea más fácil y más natural para los usuarios de las aplicaciones habilitadas para la ubicación, las aplicaciones del sistema de información geográfica (GIS) y las aplicaciones de geoimagen. Una vez que los datos espaciales se almacenan en una base de datos Oracle, puede manipularlos, recuperarlos y relacionarlos fácilmente con todos los demás datos almacenados en la base de datos. Los siguientes tipos de datos están disponibles solo si ha instalado Oracle Spatial.

### SDO\_GEOMETRY

La descripción geométrica de un objeto espacial se almacena en una sola fila, en una sola columna de tipo de objeto SDO\_GEOMETRY en una tabla definida por el usuario. Cualquier tabla que tenga una columna de tipo SDO\_GEOMETRY debe tener otra columna, o conjunto de columnas, que defina una clave primaria única para esa tabla. Las tablas de este tipo a veces se denominan tablas de geometría.

CREATE TYPE SDO\_GEOMETRY AS OBJECT

(sgo\_gtype NUMBER,

sdo\_srid NUMBER,

sdo\_point SDO\_POINT\_TYPE,

sdo\_elem\_info SDO\_ELEM\_INFO\_ARRAY,

sdo\_ordinates SDO\_ORDINATE\_ARRAY);

/

El tipo de objeto SDO\_GEOMETRY tiene la siguiente definición:

### SDO\_TOPO\_GEOMETRY

Este tipo describe una geometría de topología, que se almacena en una sola fila, en una sola columna de tipo de objeto SDO\_TOPO\_GEOMETRY en una tabla definida por el usuario. El tipo de objeto SDO\_TOPO\_GEOMETRY tiene la siguiente definición:

CREATE TYPE SDO\_TOPO\_GEOMETRY AS OBJECT

(tg\_type NUMBER,

tg\_id NUMBER,

tg\_layer\_id NUMBER,

topology\_id NUMBER);

/

/

### SDO\_GEORASTER

En el modelo relacional de objetos GeoRaster, una cuadrícula ráster o un objeto de imagen se almacena en una sola fila, en una sola columna de tipo de objeto SDO\_GEORASTER en una tabla definida por el usuario. Las tablas de este tipo se llaman tablas GeoRaster.

El tipo de objeto SDO\_GEORASTER tiene la siguiente definición:

CREATE TYPE SDO\_GEORASTER AS OBJECT

(rasterType NUMBER,

spatialExtent SDO\_GEOMETRY,

rasterDataTable VARCHAR2(32),

rasterID NUMBER,

metadata XMLType);

/

/

## Tipos de medios

Oracle Multimedia utiliza tipos de objetos, similares a las clases de Java o C ++, para describir datos multimedia. Una instancia de estos tipos de objetos consta de atributos, incluidos los metadatos y los datos de medios, y los métodos. Los tipos de datos multimedia se crean en el esquema ORDSYS. Existen sinónimos públicos para todos los tipos de datos, por lo que puede acceder a ellos sin especificar el nombre del esquema.

Oracle Multimedia proporciona los siguientes tipos de objetos:

* **ORDAudio**

Soporta el almacenamiento y gestión de datos de audio.

* **ORDDicom**

Admite el almacenamiento y la gestión de imágenes digitales y comunicaciones en medicina (DICOM), el formato universalmente reconocido como el estándar para imágenes médicas.

* **ORDDoc**

Admite el almacenamiento y la gestión de cualquier tipo de datos multimedia, incluidos datos de audio, imagen y video. Utilice este tipo cuando desee que todos los medios se almacenen en una sola columna.

* **ORDImage**

Soporta el almacenamiento y gestión de datos de imagen.

* **ORDVideo**

Soporta el almacenamiento y manejo de datos de video.

* **ORDImageSignature**

El tipo de objeto ORDImageSignature ha quedado en desuso y ya no debe introducirse en su código. Las ocurrencias existentes de este tipo de objeto continuarán funcionando como en el pasado.

Los siguientes tipos de datos cumplen con el estándar de imagen fija ISO-IEC 13249-5, comúnmente conocido como imagen fija de SQL / MM:

* **SI\_AverageColor**

Representa una característica que caracteriza una imagen por su color promedio.

* **SI\_Color**

Encapsula valores de color.

* **SI\_ColorHistogram**

Representa una característica que caracteriza una imagen por las frecuencias relativas de los colores exhibidos por muestras de la imagen en bruto.

* **SI\_FeatureList**

Una lista que contiene hasta cuatro de las características de la imagen representados por los tipos de objetos anteriores ( SI\_AverageColor, SI\_ColorHistogram, SI\_PositionalColor, y SI\_Texture), donde cada característica se asocia con un peso de entidad.

* **SI\_PositionalColor**

Dada una imagen dividida en n por m rectángulos, el tipo de objeto SI\_PositionalColor representa el rasgo que caracteriza a una imagen por la n de m la mayoría de los colores significativos de los rectángulos.

* **SI\_StillImage**

Representa imágenes digitales con características de imagen inherentes, como altura, ancho y formato.

* **SI\_Texture**

Representa una característica que caracteriza una imagen por el tamaño de los elementos que se repiten (grosor), las variaciones de brillo (contraste) y la dirección predominante (direccionalidad).

## Tipo de filtro de expresión

Oracle Expression Filter permite a los desarrolladores de aplicaciones administrar y evaluar expresiones condicionales que describen los intereses de los usuarios en los datos. El filtro de expresiones incluye el siguiente tipo de datos:

El Filtro de expresiones usa un tipo de datos virtual llamado Expression para administrar y evaluar expresiones condicionales como datos en tablas de bases de datos. El Filtro de Expresión crea una columna de tipo de datos Expression de una columna VARCHAR2 asignando un conjunto de atributos a la columna. Esta asignación habilita una restricción de datos que garantiza la validez de las expresiones almacenadas en la columna.

Puede definir condiciones utilizando el operador EVALUATE en un tipo de datos Expression para evaluar las expresiones almacenadas en una columna para algunos datos. Si está utilizando Enterprise Edition, también puede definir un índice de filtro de expresión en una columna de tipo de datos Expression para procesar consultas utilizando el operador EVALUATE.

# VARIABLES DE SUSTITUCIÓN

Hasta ahora las sentencias SQL se han ejecutado con columnas y condiciones predeterminadas y sus valores. Supongamos que desea realizar una consulta que muestre los empleados con diferentes cargos, excepto aquéllos cuyo job\_ID no sea SA\_REP. Puede editar la cláusula **WHERE** para proporcionar un valor diferente cada vez que ejecute el comando, pero existe también una forma más sencilla.

Si se utiliza una variable de sustitución en lugar de los valores exactos en la cláusula **WHERE**, puede ejecutar la misma consulta para diferentes valores.

Puede crear informes que soliciten a los usuarios que proporcionen sus propios valores para restringir el rango de datos devuelto utilizando las variables de sustitución. Puede embeber las *variables de sustitución* en un archivo de comandos o en una única sentencia SQL. Una variable se puede considerar como un contenedor en el que se almacenan los valores temporalmente. Cuando se ejecuta la sentencia, el valor se sustituye.



**Variables de Sustitución**

* Utilizar variables de sustitución para:
  + Almacenar valores temporalmente con una sustitución de un solo ampersand (&) y de dos ampersands (&&)
* Utilizar las variables de sustitución para complementar:
  + Condiciones WHERE
  + Cláusulas ORDER BY
  + Expresiones de columna
  + Nombres de tabla
  + Sentencias SELECT completas

Puede utilizar variables de sustitución de un solo ampersand (**&**) para almacenar valores temporalmente.

También puede predefinir variables mediante el comando **DEFINE**.

**DEFINE** crea y asigna un valor a una variable.

**Rangos de Datos Restringidos: Ejemplos**

* Informes de cifras sólo del trimestre actual o de un rango de fechas específico
* Informes sobre datos relevantes sólo del usuario que solicita el informe
* Visualización del personal de sólo un departamento determinado

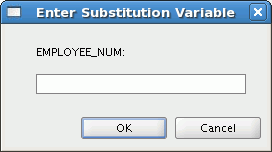
**Otros Efectos Interactivos**

Los efectos interactivos no están restringidos para dirigir la interacción del usuario a la cláusula **WHERE**. Los mismos principios se pueden utilizar también para conseguir otros objetivos, como:

* Obtener los valores de entrada de un archivo en lugar de obtenerlos de una persona
* Transferir valores de una sentencia SQL a otra

**Nota:** tanto SQL Developer como SQL\* Plus soportan las variables de sustitución y los comandos **DEFINE/UNDEFINE**. Ni SQL Developer ni iSQL\*Plus soportan las comprobaciones de validación (excepto para tipos de dato) en la entrada de usuario. Si se utilizan en scripts que se despliegan a usuarios, las variables de sustitución se pueden manipular para ataques de inyección de SQL.

### &



**Uso de la Variable de Sustitución de Un Solo Ampersand**

Utilizar una variable prefijada con un ampersand (**&**) para solicitar al usuario un valor:

SELECT employee\_id,

last\_name,

salary,

department\_id

FROM employees

WHERE employee\_id =&employee\_num

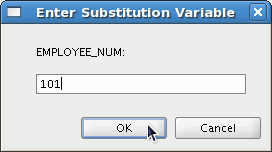
**Uso de la Variable de Sustitución de Un Solo Ampersand**

Al ejecutar un informe, los usuarios a menudo desean restringir los datos que se devuelve de forma dinámica. SQL\*Plus o SQL Developer proporciona esta flexibilidad con las variables de usuario. Utilice un ampersand (&) para identificar cada variable en la sentencia SQL. Sin embargo, no es necesario que defina el valor de cada variable.

|  |  |
| --- | --- |
| **Notación** | **Descripción** |
| *&user\_variable* | Indica una variable en una sentencia SQL; si la variable no existe, SQL\*Plus o SQL Developer solicita al Usuario un valor (la nueva variable se desecha después de utilizarla.) |

El ejemplo de la diapositiva crea una variable de sustitución de SQL Developer para un número de empleado. Al ejecutar la sentencia, SQL Developer solicita al usuario un número de empleado y, a continuación, muestra el número de empleado, apellido, salario y número de departamento de ese empleado.

Con un solo ampersand, si la variable no existe, se pregunta al usuario cada vez que se ejecuta el comando.



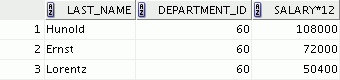
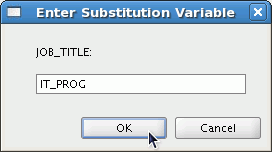
**Uso de la Variable de Sustitución de Un Solo Ampersand**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Cuando SQL Developer detecte que la sentencia SQL contiene un ampersand, se le solicitará que introduzca un valor para la variable de sustitución que se menciona en la sentencia SQL.

Después de introducir un valor y hacer clic en el botón OK, los resultados aparecen en el separador Results de la sesión de SQL Developer.



**Valores de Fecha y Carácter con Variables de Sustitución**

Utilizar las comillas simples para los valores de fecha y carácter:

SELECT last\_name,

department\_id,

salary\*12

FROM employees

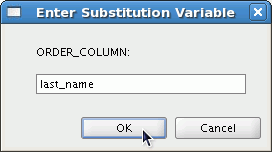
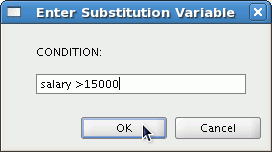
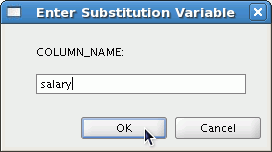
WHERE job\_id = '&job\_title';

**Valores de Fecha y Carácter con Variables de Sustitución**

En la cláusula **WHERE**, los valores de fecha y carácter se deben incluir entre comillas simples. La misma regla se aplica a las variables de sustitución.

Incluya la variable entre comillas simples en la sentencia SQL en sí.

La diapositiva muestra una consulta para recuperar los nombres de los empleados, números de departamento y salarios anuales de todos los empleados según el valor del puesto de la variable de sustitución SQL Developer.



**Especificación de Nombres de Columna, Expresiones y Texto**

SELECT employee\_id, last\_name,

job\_id,

&column\_name

FROM employees

WHERE condition

ORDER BY &order column;

**Especificación de Nombres de Columna, Expresiones y Texto**

No sólo puede utilizar las variables de sustitución en la cláusula **WHERE** de una sentencia SQL, sino también como sustitución de nombres de columna, expresiones o texto.

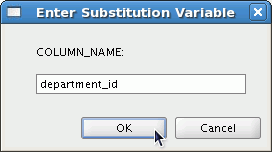
**Ejemplo:**

El ejemplo de la diapositiva muestra el número de empleado, apellido, puesto y cualquier otra columna especificada por el usuario en tiempo de ejecución de la tabla EMPLOYEES. Para cada variable de sustitución de la sentencia **SELECT**, se le solicitará que introduzca un valor y, a continuación, tendrá que hacer clic en el botón OK para continuar.

Si no introduce un valor para la variable de sustitución, obtendrá un error cuando ejecute la sentencia anterior.

**Nota:** una variable de sustitución se puede utilizar en cualquier lugar de la sentencia **SELECT**, excepto como la primera palabra introducida en el símbolo del sistema.

### &&



**Uso de Variables de Sustitución de Dos Ampersands**

Usar dos ampersands (&&) si se desea reutilizar el valor de la variable sin preguntar siempre al usuario:

SELECT employee\_id, last\_name,

job\_id,

&&column\_name

FROM employees

ORDER BY &column\_name;

**Uso de Variables de Sustitución de Dos Ampersands**

Puede utilizar la variable de sustitución de dos ampersands (&&) si desea reutilizar el valor de la variable sin preguntar siempre al usuario. El usuario visualiza la solicitud del valor sólo una vez. En el ejemplo de la diapositiva, se le solicita al usuario que proporcione el valor para la variable column\_name una vez. El valor proporcionado por el usuario (department\_id) se utiliza para la visualización y el orden de los datos. Si vuelve a ejecutar la consulta, no se le pedirá el valor de la variable.

SQL Developer almacena el valor proporcionado con el comando **DEFINE**; lo vuelve a utilizar cada vez que haga referencia al nombre de la variable. Después de que una variable de usuario esté colocada, necesitará utilizar el comando **UNDEFINE** para suprimirla de la siguiente forma:

**UNDEFINE** column\_name

### DEFINE



**Uso del Comando DEFINE**

* Usar el comando DEFINE para crear y asignar un valor a una variable.
* Usar el comando UNDEFINE de iSQL\*Plus para eliminar una variable.

DEFINE employee\_num = 200

SELECT employee\_id,

last\_name,

salary,

department\_id

FROM employees

WHERE employee\_id = &employee\_num

UNDEFINE employee\_num;

El ejemplo mostrado crea una variable de sustitución para un número de empleado utilizando el comando **DEFINE**. En tiempo de ejecución, muestra el número de empleado, nombre, salario y número de departamento de ese empleado.

Debido a que la variable se crea utilizando el comando **DEFINE** de SQL Developer, no se le solicita al usuario introducir un valor para el número de empleado. En su lugar, el valor de la variable definida se sustituye automáticamente en la sentencia **SELECT**. La variable de sustitución EMPLOYEE\_NUM está en la sesión hasta que el usuario anula su definición o si sale de la sesión de SQL Developer.

### VERIFY



**Uso del Comando VERIFY**

Usar el comando VERIFY para cambiar la visualización de la variable de sustitución, antes y después de que SQL Developer sustituya las variables de sustitución con los valores:

SET VERIFY ON

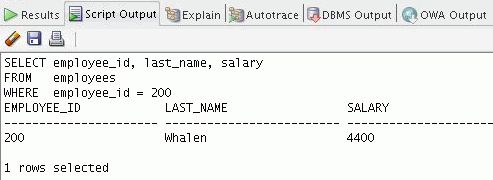
SELECT employee\_id,

last\_name,

salary

FROM employees

WHERE employee\_id = &employee\_num;



Para confirmar los cambios en la sentencia SQL, utilice el comando **VERIFY**. La definición de **SET VERIFY ON** fuerza a SQL Developer a mostrar el texto de un comando después de que sustituya las variables de sustitución por valores. Para ver la salida de **VERIFY**, debe utilizar el icono Run Script (F5) en la hoja de trabajo de SQL. SQL Developer muestra el texto de un comando después de que sustituya las variables de sustitución por valores**,** en el separador Script Output como se muestra en la diapositiva.

El ejemplo de la diapositiva muestra el nuevo valor de la columna EMPLOYEE\_ID en la sentencia SQL seguida de la salida.

**Variables del Sistema SQL\*Plus**

SQL\*Plus utiliza varias variables del sistema que controlan el entorno de trabajo. Una de esas variables es **VERIFY**. Para obtener una lista completa de todas las variables del sistema, puede emitir el comando **SHOW ALL** en el símbolo del sistema de SQL\*Plus.

# OPERADORES

## ARITMÉTICOS

|  |  |
| --- | --- |
| Operador | Descripción |
| + | Sumar |
| - | Restar |
| \* | Multiplicar |
| / | Dividir |

**Expresiones Aritméticas**

Puede que necesite modificar la forma en la que se muestran los datos, realizar cálculos o consultar casos de posibilidades. Todo esto es posible mediante las expresiones aritméticas. Una expresión aritmética puede contener nombres de columna, valores numéricos constantes y operadores aritméticos.

### Operadores Aritméticos

La diapositiva muestra los operadores aritméticos disponibles en SQL. Puede utilizar operadores aritméticos en cualquier cláusula de una sentencia SQL (excepto en la cláusula **FROM**).

**Nota:** con los tipos de dato **DATE** y **TIMESTAMP**, sólo puede utilizar los operadores de suma y resta.



SELECT last\_name,

salary,

salary + 300

FROM employees;



## Uso de Operadores Aritméticos

**…**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

El ejemplo de la diapositiva utiliza el operador de suma para calcular un aumento de salario de 300 dólares para todos los empleados. La diapositiva también muestra una columna SALARY+300 en la salida.

Tenga en cuenta que la columna calculada resultante, SALARY+300, no es una nueva columna en la tabla EMPLOYEES; es sólo para visualización. Por defecto, el nombre de la nueva columna procede del cálculo que lo genera, en este caso, salary+300.

**Nota:** el servidor de *Oracle* ignora los espacios en blanco antes y después del operador aritmético.

**Prioridad de Operadores**

Si una expresión aritmética contiene más de un operador, la multiplicación y división se evalúan primero. Si los operadores en una expresión tienen la misma prioridad, la evaluación se realiza de izquierda a derecha.

Puede utilizar los paréntesis para forzar la expresión que se incluye entre paréntesis para que se evalúe primero.

**Reglas de Prioridad**

* La multiplicación y división se producen antes de la suma y la resta.
* Los operadores de la misma prioridad se evalúan de izquierda a derecha.
* Los paréntesis se utilizan para sustituir la prioridad por defecto o para aclarar la sentencia.



SELECT last\_name,

salary,

12\*(salary+100)

FROM employees;

SELECT last\_name,

salary,

12\*salary+100

FROM employees;



## Prioridad de Operadores

El primer ejemplo de la diapositiva muestra el apellido, el salario y la compensación anual de los empleados. Calcula la compensación anual multiplicando el salario mensual por 12, más un incentivo de 100 dólares. Tenga en cuenta que la multiplicación se realiza antes de la suma.

**Nota:** utilice los paréntesis para reforzar el orden estándar de prioridad y mejorar la claridad. Por ejemplo, la expresión de la diapositiva se puede escribir como (12\*salary)+100 sin producir ningún cambio en el resultado.

**Uso de Paréntesis**

Puede sustituir las reglas de prioridad utilizando paréntesis para especificar el orden en el que desea que se ejecuten los operadores. El segundo ejemplo de la diapositiva muestra el apellido, el salario y la compensación anual de los empleados. Calcula la compensación anual de la siguiente forma: suma un incentivo mensual de 100 dólares al salario mensual y, a continuación, multiplica dicho subtotal por 12. Debido a los paréntesis, la suma tiene prioridad sobre la multiplicación.

SELECT last\_name,

job\_id,

salary,

commission\_pct

FROM employees;



## Definición de Valor Nulo

* Un valor nulo es un valor que no está disponible, sin asignar, desconocido o que no es aplicable.
* Un valor nulo no es lo mismo que un cero

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.



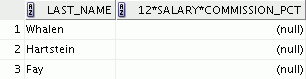
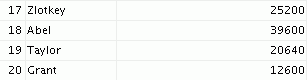
Si una fila carece de un valor de datos para una columna concreta, se dice que dicho valor es *nulo* o que contiene un valor nulo.

Un valor nulo es un valor que no está disponible, sin asignar, desconocido o que no es aplicable. Un valor nulo no es lo mismo que un cero o un espacio en blanco. El cero es un número y el espacio en blanco es un carácter.

Las columnas de cualquier tipo de dato pueden contener valores nulos. Sin embargo, algunas restricciones (**NOT NULL** y **PRIMARY KEY**) evitan que se utilicen valores nulos en la columna.

En la columna COMMISSION\_PCT de la tabla EMPLOYEES, observe que sólo pueden percibir una comisión un director de ventas o un vendedor. El resto de los empleados no tienen derecho a percibir comisiones. Un valor nulo representa este hecho.

**Nota:** por defecto, SQL Developer utiliza el literal, (**null**), para identificar los valores nulos. Sin embargo, se puede definir en un valor más relevante para el usuario. Para ello, seleccione Preferences en el menú Tools. En el cuadro de diálogo Preferences, amplíe el nodo Database. Haga clic en Advanced Parameters y en el panel derecho, para “Display Null value As”, introduzca un valor adecuado.



SELECT last\_name, 12\*salary\*commission\_pct

FROM employees;



**Valores Nulos en Expresiones Aritméticas**

Las expresiones aritméticas que contengan un valor nulo se evalúan como nulas.

**Valores Nulos en Expresiones Aritméticas**

Si cualquier valor de columna en una expresión aritmética es nulo, el resultado es nulo. Por ejemplo, si intenta realizar una división entre cero, recibirá un error. Sin embargo, si divide un número entre un valor nulo, el resultado será nulo o desconocido.

En el ejemplo de la diapositiva, el empleado Whalen no percibe ninguna comisión. Porque la columna COMMISSION\_PCT en la expresión aritmética es nula, por lo tanto, el resultado es nulo.

## DE CONCATENACIÓN, DE COMILLAS



SELECT last\_name||job\_id AS "Employees" FROM employees;



**Operador de Concatenación**

Un operador de concatenación:

* Enlaza columnas o cadenas de caracteres a otras columnas
* Se representa con dos barras verticales (||)
* Crea una columna resultante que es una expresión de carácter

**Operador de Concatenación**

Puede enlazar columnas a otras columnas, expresiones aritméticas o valores constantes para crear una expresión de carácter con el operador de concatenación (||). Las columnas a ambos lados del operador se combinan para crear una sola columna de salida.

En el ejemplo, LAST\_NAME y JOB\_ID están concatenados y se les ha otorgado el alias Employees. Observe que el apellido del empleado y código de trabajo se combinan para crear una única columna de salida.

La palabra clave **AS** antes del nombre de alias facilita la lectura de la cláusula **SELECT**.

**Valores Nulos con el Operador de Concatenación**

Si concatena un valor nulo con una cadena de caracteres, el resultado es una cadena de caracteres.

LAST\_NAME || NULL tiene como resultado LAST\_NAME.

**Nota:** también puede concatenar expresiones de fecha con otras expresiones o columnas.



**Cadenas de Caracteres Literales**

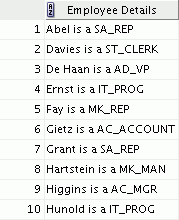
* Un literal es un carácter, un número o una fecha que se incluye en la sentencia SELECT.
* Los valores literales de caracteres y fecha se deben incluir entre comillas simples.
* Cada cadena de caracteres es la salida una vez para cada fila devuelta.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Cadenas de Caracteres Literales**

Un literal es un carácter, un número o una fecha que se incluye en la lista **SELECT**. No es un nombre de columna o un alias de columna. Se imprime para cada fila devuelta. Las cadenas literales de texto de formato libre se pueden incluir en el resultado de consultas y se tratan igual que la lista **SELECT**.

Los literales de caracteres y fecha se debendelimitar con comillas simples (' '); no es necesario delimitar los literales de número de forma similar.



SELECT last\_name ||' is a '||job\_id

AS "Employee Details"

FROM employees;



**Uso de Cadenas de Caracteres Literales**

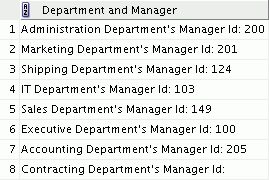
**Uso de Cadenas de Caracteres Literales**

El ejemplo de la diapositiva muestra los apellidos y códigos de trabajo de todos los empleados. La columna tiene la cabecera Employee Details. Observe los espacios incluidos entre comillas simples en la sentencia **SELECT**. Los espacios mejoran la legibilidad de la salida.

En el siguiente ejemplo, el apellido y el salario de cada empleado se concatenan con un literal para proporcionar a las filas devueltas más significado:

SELECT last\_name ||': 1 Month salary = '||salary Monthly

FROM employees;



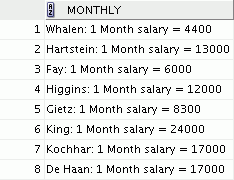
SELECT department\_name ||

q'[ Department's Manager Id: ]' ||

manager\_id

AS "Department and Manager"

FROM departments;



**Operador de Comillas (q) Alternativo**

* Especifique su propio delimitador de entrecomillado.
* Seleccione cualquier delimitador.
* Aumente la legibilidad y el uso.

**Operador de Comillas (q) Alternativo**

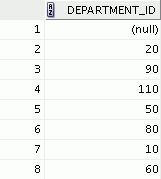
Muchas de las sentencias SQL utilizan literales de caracteres en expresiones o condiciones. Si el propio literal contiene una comilla simple, puede utilizar el operador de comillas (**q**) y seleccionar su propio delimitador de entrecomillado.

Puede seleccionar cualquier delimitador adecuado, ya sea de un solo byte o multibyte, o bien cualquiera de los siguientes pares de caracteres: **[ ]**, **{ }**, **( )** o **< >**.

En el ejemplo, la cadena contiene una comilla simple, que normalmente se interpreta como un delimitador de una cadena de caracteres. Sin embargo, al utilizar el operador q, los corchetes **[ ]** se utilizan como delimitador de entrecomillado. La cadena entre los delimitadores de corchetes se interpreta como una cadena de caracteres literales.

SELECT DISTINCT department\_id

FROM employees;



SELECT department\_id

FROM employees;

**Filas Duplicadas**

La visualización por defecto de las consultas incluye todas las filas, también las filas duplicadas.



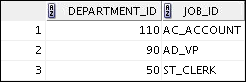
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Filas Duplicadas**

A menos que indique lo contrario, SQL muestra los resultados de una consulta sin eliminar las filas duplicadas. El primer ejemplo de la diapositiva muestra todos los números de departamento de la tabla EMPLOYEES. Observe que los números de departamento se repiten.

Para eliminar filas duplicadas en el resultado, incluya la palabra clave **DISTINCT** en la cláusula **SELECT** inmediatamente después de la palabra clave **SELECT**. En el segundo ejemplo de la diapositiva, la tabla EMPLOYEES en realidad contiene 20 filas, pero sólo hay siete números de departamento únicos en la tabla.

Puede especificar varias columnas después del cualificador **DISTINCT**. El cualificador **DISTINCT** afecta a todas las columnas seleccionadas y el resultado es cada combinación distinta de columnas.



SELECT DISTINCT department\_id,

job\_id

FROM employees;

**Nota:** también puede especificar la palabra clave **UNIQUE**, que es un sinónimo de la palabra clave **DISTINCT**.

## DE COMPARACIÓN

|  |  |
| --- | --- |
| Operador | Significado |
| = | Igual que |
| > | Mayor que |
| >= | Mayor o igual que |
| < | Menor que |
| <= | Menor o igual que |
| <> | Distinto de |
| BETWEEN  ...AND... | Entre dos valores (ambos incluidos) |
| IN(set) | Coincide con cualquiera de los |
| LIKE | Coincide con un patrón de |
| IS NULL | Es un valor nulo |

**Operadores de Comparación**

valores de una lista

caracteres



**Operadores de Comparación**

Los operadores de comparación se utilizan en condiciones que comparan una expresión con otra expresión o valor. Se utilizan en la cláusula **WHERE** en el siguiente formato:

**Sintaxis**

... **WHERE** *expr operator value*

**Ejemplo**

... **WHERE** hire\_date = '01-JAN-95'

... **WHERE** salary >= 6000

... **WHERE** last\_name = 'Smith'

Recuerde, un alias no se puede utilizar en la cláusula **WHERE**.

**Nota:** los símbolos **!=** y **^=** también pueden representar la condición ***not equal to***.

SELECT last\_name,

salary

FROM employees

WHERE salary <= 3000 ;



**Uso de Operadores de Comparación**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Uso de Operadores de Comparación**

En el ejemplo, la cláusula **SELECT** recupera el apellido y el salario de la tabla EMPLOYEES para cualquier empleado cuyo salario sea menor o igual que 3.000 dólares. Tenga en cuenta que existe un valor explícito proporcionado a la cláusula **WHERE**. El valor explícito de 3000 se compara con el valor de salario de la columna SALARY de la tabla EMPLOYEES.



**Condiciones de Rango mediante BETWEEN**

Utilizar el operador BETWEEN para mostrar las filas basadas en un rango de valores:

**Límite Límite**

**Inferior superior**

SELECT last\_name,

salary

FROM employees

WHERE salary

BETWEEN 2500 AND 3500;

**Uso de Condiciones de Rango mediante el Operador BETWEEN**

Puede utilizar las filas basadas en un rango de valores utilizando la condición de rango **BETWEEN**. El rango que especifique contiene un límite inferior y un límite superior.

La sentencia **SELECT**  de la diapositiva devuelve filas de la tabla EMPLOYEES para cualquier empleado cuyo salario esté entre 2.500 y 3.500 dólares.

También se incluyen los valores que se especifiquen con el operador **BETWEEN**. Debe especificar primero el límite inferior.

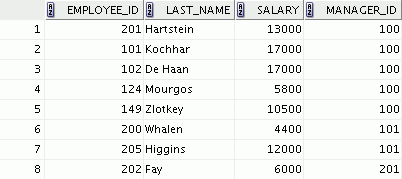
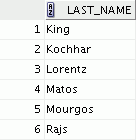
También puede utilizar la condición **BETWEEN**  en los valores de caracteres:

SELECT last\_name

FROM employees

WHERE last\_name

BETWEEN 'King' AND 'Smith';



SELECT employee\_id,

last\_name,

salary,

manager\_id

FROM employees

WHERE manager\_id

IN (100,101,102);



**Condición de Miembro mediante el Operador IN**

Utilizar el operador IN para probar los valores de una lista:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Condición de Miembro mediante el Operador IN**

Para probar valores de un juego especificado de valores, utilice el operador **IN**. La condición definida mediante el operador **IN** también se denomina *condición de miembro*.

El ejemplo de la diapositiva muestra los números de empleado, apellidos, salarios y números de empleado de los gestores de todos los empleados cuyo número de empleado del gestor sea 100, 101 o 201.

**Nota:** el juego de valores se puede especificar en cualquier orden aleatorio (por ejemplo, 201,100,101).

La condición **IN** se puede utilizar con cualquier tipo de dato. El siguiente ejemplo devuelve una fila de la tabla EMPLOYEES para cualquier empleado cuyo apellido esté incluido en la lista de nombres de la cláusula **WHERE**:

SELECT employee\_id,

manager\_id,

department\_id

FROM employees

WHERE last\_name IN ('Hartstein', 'Vargas');

Si los caracteres o fechas se utilizan en la lista, se deben incluir entre comillas simples (' ').

**Nota:** el servidor de *Oracle* evalúa el operador **IN** internamente como un juego de condiciones **OR**, como a=value1 o a=value2 o a=value3. Por lo tanto, el uso del operador **IN** no tiene ninguna ventaja de rendimiento y sólo se utiliza para la simplicidad lógica.

SELECT first\_name

FROM employees

WHERE first\_name LIKE 'S%'



**Coincidencia de Patrones mediante el Operador LIKE**

* Utilizar el operador LIKE para realizar búsquedas con comodines de valores de cadena de búsqueda válidos.
* Las condiciones de búsqueda pueden contener caracteres literales o números:
  + % indica cero o varios caracteres.
  + \_ indica un carácter.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Coincidencia de Patrones mediante el Operador LIKE**

Puede que no siempre conozca el valor exacto que debe buscar. Puede seleccionar filas que coincidan con un patrón de caracteres utilizando la condición **LIKE**. Se hace referencia a la operación de coincidencia de patrón de caracteres como búsqueda con *comodines*. Para crear la cadena de búsqueda se pueden utilizar dos símbolos.

|  |  |
| --- | --- |
| **Símbolo** | **Descripción** |
| % | Representa cualquier secuencia de cero o más caracteres |
| \_ | Representa un único carácter |

La sentencia **SELECT** de la diapositiva devuelve el nombre del empleado de la tabla EMPLOYEES de cualquier empleado cuyo nombre empiece por la letra “S”. Observe que se trata de la “S” mayúscula. No se devolverán los nombres que empiecen por “s” minúscula.

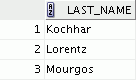
La condición **LIKE** se puede utilizar como un método abreviado para algunas comparaciones **BETWEEN**. El siguiente ejemplo muestra los apellidos y fechas de contratación de los empleados que comenzaron a trabajar entre enero y diciembre de 1995:

SELECT last\_name,

hire\_date

FROM employees

WHERE hire\_date LIKE '%95';



**Combinación de Caracteres Comodín**

* Puede combinar los dos caracteres comodín (%, \_) con caracteres literales para la coincidencia de patrones:
* Puede utilizar el identificador ESCAPE para buscar los símbolos % y \_ reales.

SELECT last\_name

FROM employees

WHERE last\_nameLIKE '\_o%';

**Combinación de Caracteres Comodín**

Los símbolos **%** y **\_** se pueden utilizar en cualquier combinación con los caracteres literales. El ejemplo de la diapositiva muestra los nombres de todos los empleados cuyos apellidos tengan la letra “o” como segundo carácter.

Identificador **ESCAPE**

Cuando necesite tener una coincidencia exacta para los caracteres **%**y ***\_***reales, utilice el identificador **ESCAPE**. Esta opción especifica lo que representa el carácter de escape. Si desea buscar cadenas que contengan SA\_, puede utilizar la siguiente sentencia SQL:

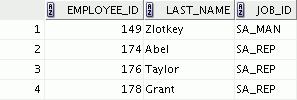
SELECT employee\_id,

last\_name,

job\_id

FROM employees

WHERE job\_id LIKE '%SA\\_%' ESCAPE '\';



El identificador **ESCAPE** identifica la barra invertida (**\**) como carácter de escape. En la sentencia SQL, el carácter de escape precede al guión bajo (**\_**). Esto hace que el servidor de *Oracle* interprete el carácter de subrayado literalmente.

## DE RANGO

## LÓGICOS



**Definición de Condiciones mediante los Operadores Lógicos**

|  |  |
| --- | --- |
| Operador | Significado |
| AND | Devuelve TRUE si *ambas* condiciones de componente son verdaderas |
| OR | Devuelve TRUE si *cualquier* condición de componente es verdadera |
| NOT | Devuelve TRUE si la condición es falsa |

**Definición de Condiciones mediante los Operadores Lógicos**

Una condición lógica combina el resultado de dos condiciones de componentes para producir un resultado único basado en dichas condiciones o invierte el resultado de una condición única. Se devuelve una fila sólo si el resultado global de la condición es verdadera.

En SQL, están disponibles tres operadores lógicos:

* **AND**
* **OR**
* **NOT**

Todos los ejemplos indicados hasta ahora han especificado sólo una condición en la cláusula **WHERE**. Puede utilizar varias condiciones en una única cláusula **WHERE** mediante los operadores **AND** y **OR**.



SELECT employee\_id,

last\_name,

job\_id, salary

FROM employees

WHERE salary >= 10000

AND job\_id LIKE '%MAN%' ;



**Uso del Operador AND**

AND necesita que ambas condiciones sean verdaderas:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Uso del Operador AND**

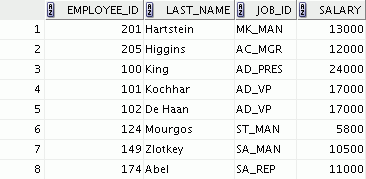
En el ejemplo, ambas condiciones deben ser verdaderas para poder seleccionar cualquier registro. Por lo tanto, sólo se seleccionan los empleados que tengan un puesto que contenga la cadena ‘MAN’ *y* que ganen 10.000 dólares o más.

Todas las búsquedas de caracteres son sensibles a mayúsculas/minúsculas, es decir, no se devuelve ninguna fila si ‘MAN’ no está en mayúsculas. Las cadenas de caracteres se deben incluir entre comillas simples.

**Tabla de TRUE AND**

La siguiente tabla muestra los resultados de combinar dos expresiones con **AND**:

|  |  |  |  |
| --- | --- | --- | --- |
| **AND** | **TRUE** | **FALSE** | **NULL** |
| **TRUE** | TRUE | FALSE | NULL |
| **FALSE** | FALSE | FALSE | FALSE |
| **NULL** | NULL | FALSE | NULL |



SELECT employee\_id,

last\_name,

job\_id, salary

FROM employees

WHERE salary >= 10000 OR job\_id LIKE '%MAN%' ;



**Uso del Operador OR**

OR necesita que cualquier condición sea verdadera:

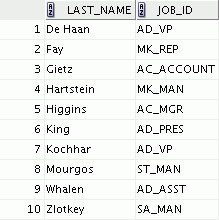
**Uso del Operador OR**

En el ejemplo, cualquier condición de componente debe ser verdadera para poder seleccionar cualquier registro. Por lo tanto, sólo se seleccionan los empleados que tengan un ID de trabajo que contenga la cadena ‘MAN’ *o* que ganen 10.000 dólares o más.

**Tabla de TRUE OR**

La siguiente tabla muestra los resultados de combinar dos expresiones con **OR**:

|  |  |  |  |
| --- | --- | --- | --- |
| **OR** | **TRUE** | **FALSE** | **NULL** |
| **TRUE** | TRUE | TRUE | TRUE |
| **FALSE** | TRUE | FALSE | NULL |
| **NULL** | TRUE | NULL | NULL |



**Uso del Operador NOT**

SELECT Last\_name,

job\_i d

FROM employees

WHERE job\_id NOT IN ('IT\_PROG', 'ST\_CLERK', 'SA\_REP');

**Uso del Operador NOT**

En el ejemplo de la diapositiva se muestra el apellido y el ID de trabajo de todos los empleados cuyo ID de trabajo *no sea* IT\_PROG, ST\_CLERK o SA\_REP.

**Tabla de NOT TRUE**

La siguiente tabla muestra el resultado de aplicar el operador NOT a una condición:

|  |  |  |  |
| --- | --- | --- | --- |
| **NOT** | **TRUE** | **FALSE** | **NULL** |
|  | FALSE | TRUE | NULL |

**Nota:** el operador **NOT** también se puede utilizar con otros operadores SQL, como **BETWEEN**, **LIKE** y **NULL**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ... | WHERE | job\_id | NOT | IN ('AC\_ACCOUNT', 'AD\_VP') |
| ... | WHERE | salary | NOT | BETWEEN 10000 AND 15000 |
| ... | WHERE | last\_name | NOT | LIKE '%A%' |

... WHERE commission\_pct IS NOT NULL

## REGLAS DE PRIORIDAD DE OPERADORES

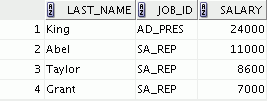
**Reglas de Prioridad**

**Puede utilizar los paréntesis para sustituir las reglas de prioridad.**

|  |  |
| --- | --- |
| Operador | Significado |
| 1 | Operadores aritméticos |
| 2 | Operador de concatenación |
| 3 | Condiciones de comparación |
| 4 | IS [NOT] NULL, LIKE, [NOT] IN |
| 5 | [NOT] BETWEEN |
| 6 | Distinto de |
| 7 | Condición lógica NOT |
| 8 | Condición lógica AND |
| 9 | Condición lógica OR |

**Reglas de Prioridad**

Las reglas de prioridad determinan el orden en el que se evalúan y calculan las expresiones. La tabla de la diapositiva muestra el orden de prioridad por defecto. Sin embargo, puede sustituir el orden por defecto utilizando paréntesis en las expresiones que desee calcular primero.



**1**

**employees**

**job\_id = 'SA\_REP' job\_id = 'AD\_PRES' salary > 15000;**

**FROM WHERE OR AND**

**SELECT last\_name, job\_id, salary**

**Reglas de Prioridad**

**2**

**employees**

**(job\_id = 'SA\_REP' job\_id = 'AD\_PRES') salary > 15000;**

**FROM WHERE OR AND**

**SELECT last\_name, job\_id, salary**

1. **Prioridad del Operador AND: Ejemplo**

En este ejemplo, hay dos condiciones:

- La primera condición es que el ID de cargo sea AD\_PRES y que el salario sea superior a 15.000 dólares.

- La segunda condición es que el ID de cargo sea SA\_REP. Por lo tanto, la sentencia **SELECT** sería la siguiente:

“Seleccione la fila si un empleado es presidente y gana más de $ 15,000, o si el empleado es un representante de ventas.”

**Uso de Paréntesis: Ejemplo**

En este ejemplo, hay dos condiciones:

* La primera condición es que el ID de trabajo sea AD\_PRES *o* SA\_REP.
* La segunda condición es que el salario sea superior a 15.000 dólares. Por lo tanto, la sentencia SELECT sería la siguiente:

"Seleccione la fila si un empleado es un presidente o un representante de ventas, y si el empleado gana más de $ 15,000".

# NULL



**Uso de las Condiciones NULL**

Probar condiciones nulas con el operador IS NULL.

SELECT last\_name,

manager\_id

FROM employees

WHERE manager\_id IS NULL

**Uso de las Condiciones NULL**

Las condiciones **NULL** incluyen las condiciones **IS NULL** e **IS NOT NULL**.

La condición **IS NULL** prueba las condiciones nulas. Un valor nulo significa que el valor no está disponible, no está asignado, se desconoce o no es aplicable. Por lo tanto, no puede probar con = porque un valor nulo no puede ser igual o desigual a cualquier valor. El ejemplo de la diapositiva recupera los apellidos y gestores de todos los empleados que no tienen un gestor.

Aquí se muestra otro ejemplo: Para mostrar un apellido, ID de cargo y comisión de todos los empleados que **no**tienen derecho a recibir una comisión, utilice la siguiente sentencia SQL:

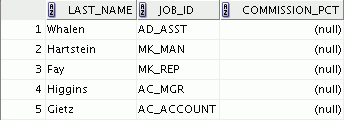
SELECT last\_name,

job\_id,

commission\_pct

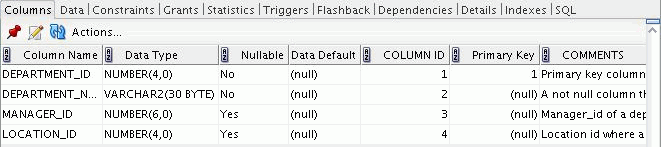
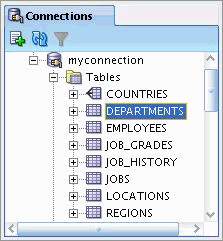
FROM employees

WHERE commission\_pct IS NULL;



# DESCRIBE

DESC[RIBE] *tablename*



**Visualización de la Estructura de la Tabla**

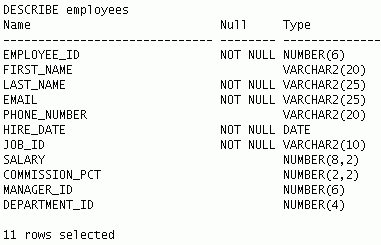
* Utilizar el comando DESCRIBE para mostrar la estructura de una tabla.
* O seleccionar la tabla en el árbol Connections y utilizar el separador Columns para ver la estructura de la tabla.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Visualización de la Estructura de la Tabla**

Puede mostrar la estructura de una tabla mediante el comando **DESCRIBE**. El comando muestra los nombres de columna y los tipos de dato e indica si una columna *debe* contener datos (es decir, si la columna tiene una restricción **NOT NULL**).

En la sintaxis, *table name*  es el nombre de cualquier tabla existente, vista o sinónimo accesible al usuario. Mediante la interfaz gráfica de usuario de SQL Developer, puede seleccionar el árbol Connections y utilizar el separador Columns para ver la estructura de la tabla. **Nota:** el comando **DESCRIBE** está soportado tanto en SQL\*Plus como en SQL Developer.



DESCRIBE employees



**Uso del Comando DESCRIBE**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Uso del Comando DESCRIBE**

En el ejemplo de la diapositiva se muestra la información sobre la estructura de la tabla EMPLOYEES mediante el comando **DESCRIBE**.

En la visualización resultante, ***Null***indica que los valores de esta columna pueden ser desconocidos. **NOT NULL** indica que una columna debe contener datos. **Type**muestra el tipo de dato de una columna.

Los tipos de dato se describen en la siguiente tabla:

|  |  |
| --- | --- |
| **Tipo de Dato** | **Descripción** |
| NUMBER(*p*,*s*) | Valor numérico que tiene un número máximo de dígitos *p*, con dígitos *s* a la derecha del punto decimal. |
| VARCHAR2(*s*) | Valor del carácter de longitud de variable del tamaño máximo *s*. |
| DATE | Valor de fecha y hora entre el 1 de enero de 4712 a.C. y el 31 de diciembre de 9999 d.C. |

# FUNCIONES

Las funciones son una característica muy potente de SQL. Se puede utilizar para realizar las siguientes acciones:

* Realizar cálculos en los datos
* Modificar elementos de datos individuales
* Manipular la salida para grupos de filas
* Formatear fechas y números para su visualización
* Convertir tipos de dato de columna

Algunas veces, las funciones SQL toman argumentos y siempre devuelven un valor.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Funciones** | | | |  | |
|  | | | |  | | | |
|  |  | | | | | |  |
| **Una sola fila funciones** | | |  | | **Varias filas funciones** | | |

Dos Tipos de Funciones SQL



**Dos Tipos de Funciones SQL**

**Devuelve un resultado por fila**

**Devuelve un resultado por juego de filas**

Hay dos tipos de grupos de funciones:

* Funciones de una sola fila.
* Funciones de varias filas.

**Funciones de Una Sola Fila**

Estas funciones funcionan sólo en filas únicas y devuelven un resultado por fila. Existen distintos tipos de funciones de una sola fila. En esta lección se abordan los siguientes temas:

* Carácter
* Número
* Fecha
* Conversión
* General

**Funciones de Varias Filas**

Las funciones pueden manipular grupos de filas para proporcionar un resultado por grupo de filas. Estas funciones también se conocen como *funciones de grupo* (se tratan en la lección titulada “Informes de Datos Agregados con Funciones de Grupo”).

## DE UNA SOLA FILA

function\_name [(arg1, arg2,...)]



**Funciones de Una Sola Fila**

Funciones de una sola fila:

* Manipular elementos de datos
* Aceptar argumentos y devolver un valor
* Actuar en cada fila devuelta
* Devolver un resultado por fila
* Posibilidad de modificar el tipo de dato
* Posibilidad de anidamiento
* Aceptar argumentos que pueden ser una columna o una expresión

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Las funciones de una sola fila se utilizan para manipular elementos de datos. Aceptan uno o varios argumentos y devuelven un valor para cada fila devuelta por la consulta. Un argumento puede ser uno de los siguientes elementos:

* Constante proporcionada por el usuario
* Valor de variable
* Nombre de columna
* Expresiones

Las características de las funciones de una sola fila son:

* + Actuar en cada fila devuelta en la consulta
  + Devolver un resultado por fila
  + Posibilidad de devolver un valor de datos de un tipo diferente al que se hace referencia
  + Posibilidad de esperar uno o más argumentos
  + Se pueden utilizar en cláusulas **SELECT**, **WHERE** y **ORDER BY**;
  + Posibilidad de anidamiento

En la sintaxis:

function\_name Es el nombre de la función.

*arg1, arg2* Es cualquier argumento que utilizará la función. Pueden estar

representados por un nombre de columna o expresión.



**Date**

**Conversión**

**Número**

**Una sola fila funciones**

**General**

**Carácter**

**Funciones de Una Sola Fila**

Funciones de Una Sola Fila (continuación)

Esta lección trata las siguientes funciones de una sola fila:

* **Funciones de carácter:** aceptan la entrada de caracteres y pueden devolver valores de número y de carácter.
* **Funciones numéricas:** aceptan valores de entrada y devuelven valores numéricos.
* **Funciones de fecha:** operan en valores del tipo de dato **DATE**. (Todas las funciones de fecha devuelven un valor de tipo de dato **DATE** excepto la función **MONTHS\_BETWEEN**, que devuelve un número.)
* **Funciones de conversión:** Convierten un valor de un tipo de dato a otro.

Funciones generales:

* + **NVL**

Nos permite obtener un valor concreto en vez de NULL como resultado

* + **NVL2**

NVL2 provee de funcionalidad similar a la [función NVL](https://www.administradortotal.org.es/mysql/la-funcion-nvl-de-oracle/) pero con un enfoque diferente.

Esta función evalúa una columna o expresión de manera que:

Si el resultado no es NULL, devuelve el segundo parámetro de NVL2.

Si el resultado es NULL, devuelve el tercer parámetro de NVL2

Como vemos, la principal diferencia con NVL es que en caso de no ser NULL, NVL2 no devuelve el dato original, sino que devuelve el segundo parámetro que le indicamos.

* + **NULLIF**

Esta función compara expr1 con expr2. Si son iguales devuelve NULL

* + **COALESCE**

Devuelve la primera expresión distinta de NULL entre sus argumentos.

* + **CASE**

Las expresiones CASE nos permiten realizar expresiones del tipo IF ... THEN ... ELSE en una instrucción sql  sin tener que llamar a procedimientos externos

* + **DECODE**

Traduce una expresión a un valor de retorno. Si expr es igual a value1, la función devuelve Return1. Si expr es igual a value2, la función devuelve Return2. Y asi sucesivamente. Si expr no es igual a ningun valor la funcion devuelve el valor por defecto.

- DE CARÁCTER

### DE CONVERSIÓN DE CARACTERES



**Funciones de Conversión de Caracteres**

Estas funciones convierten las mayúsculas/minúsculas para cadenas de caracteres:

|  |  |  |
| --- | --- | --- |
| **Función** | **Resultado** | |
| LOWER('SQL Course') | sql | course |
| UPPER('SQL Course') | SQL | COURSE |
| INITCAP('SQL Course') | Sql | Course |

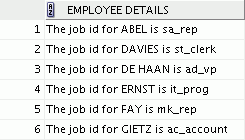
**LOWER**, **UPPER** y **INITCAP** son las tres funciones de conversión de caracteres.

* **LOWER**: convierte las cadenas de caracteres en mayúsculas o en mayúsculas/minúsculas a minúsculas.
* **UPPER**: convierte las cadenas de caracteres en minúscula o en mayúsculas/minúsculas a mayúsculas.
* **INITCAP**: convierte la primera letra de cada palabra a mayúsculas y el resto de las letras a minúsculas.

SELECT 'The job id for '||UPPER(last\_name)||

'is '||LOWER(job\_id) AS "EMPLOYEE DETAILS"

FROM employees;



SELECT employee\_id,

last\_name,

department\_id

FROM employees

WHERE LOWER(last\_name) = 'higgins';

SELECT employee\_id,

last\_name,

department\_id

FROM employees

WHERE last\_name = 'higgins';



**Uso de Funciones de Conversión de Caracteres**

Mostrar el número de empleado, nombre y número de departamento del empleado Higgins:



El ejemplo de la diapositiva muestra el número de empleado, nombre y número de departamento del empleado Higgins:

La cláusula **WHERE** de la primera sentencia SQL especifica el nombre del empleado como higgins. Debido a que todos los datos de la tabla EMPLOYEES están almacenados correctamente, el nombre Higgins no encuentra ninguna coincidencia en la tabla y no se selecciona ninguna fila.

La cláusula **WHERE** de la segunda sentencia SQL especifica que el nombre del empleado de la tabla EMPLOYEES se compara con higgins, convirtiendo la columna LAST\_NAME a minúsculas para poder compararla. Ya que ambos nombres no están en minúsculas, se ha encontrado una coincidencia y se ha seleccionado una fila. La cláusula **WHERE** se puede volver a escribir de la siguiente forma para que produzca el mismo resultado:

...WHERE last\_name = 'Higgins'

El nombre de la salida aparece tal y como se almacenó en la base de datos. Para mostrar el nombre en mayúsculas, utilice la función **UPPER** de la sentencia **SELECT**.

SELECT employee\_id,

UPPER(last\_name),

department\_id

FROM employees

WHERE INITCAP(last\_name) = 'Higgins';

### **DE MANIPULACIÓN DE CARACTERES**



**Funciones de Manipulación de Caracteres**

Estas funciones manipulan las cadenas de caracteres:

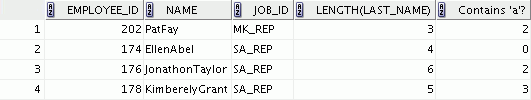
|  |  |
| --- | --- |
| **Función** | **Resultado** |
| CONCAT('Hello', 'World') | HelloWorld |
| SUBSTR('HelloWorld',1,5) | Hello |
| LENGTH('HelloWorld') | 10 |
| INSTR('HelloWorld', 'W') | 6 |
| LPAD(salary,10,'\*') | \*\*\*\*\*24000 |
| RPAD(salary, 10, '\*') | 24000\*\*\*\*\* |
| REPLACE  ('JACK and JUE','J','BL') | BLACK and BLUE |
| TRIM('H' FROM 'HelloWorld') | elloWorld |

Funciones de Manipulación de Caracteres

**CONCAT**, **SUBSTR**, **LENGTH**, **INSTR**, **LPAD**, **RPAD** y **TRIM** son funciones de manipulación de caracteres que se tratan en esta lección.

* **CONCAT**: une los valores. (Sólo se pueden utilizar dos parámetros con CONCAT).
* **SUBSTR**: extrae una cadena de una longitud determinada
* **LENGTH**: muestra la longitud de una cadena como un valor numérico
* **INSTR**: obtiene la posición numérica de un carácter denominado
* **LPAD**: Devuelve una expresión con relleno a la izquierda de caracteres *n* con una expresión de caracteres
* **RPAD**: devuelve una expresión con relleno a la derecha de caracteres *n* con una expresión de caracteres
* **TRIM**: recorta los caracteres finales o de encabezado (o ambos) de una cadena de caracteres (si *trim\_character* o *trim\_source* es un carácter literal, debe incluirlo entre comillas simples)

**Nota:** puede utilizar funciones como **UPPER** y **LOWER** con una sustitución con ampersand. Por ejemplo, utilice **UPPER**('&job\_title') para que el usuario no tenga que introducir el puesto en un formato específico.



**Uso de las Funciones de Manipulación de Caracteres**

SELECT employee\_id,

CONCAT(first\_name, last\_name) NAME, job\_id,

LENGTH (last\_name INSTR(last\_name, 'a')) "Contains 'a'?"

FROM employees

WHERE SUBSTR(job\_id, 4) = 'REP';

El ejemplo de la diapositiva muestra los nombres y apellidos de los empleados que se han unido, la longitud del apellido del empleado y la posición numérica de la letra “a” en el apellido del empleado de todos los empleados que tienen la cadena REP incluida en el ID de trabajo que empieza en la cuarta posición de dicho ID.

Ejemplo:

Modifique la sentencia SQL de la diapositiva para mostrar los datos de los empleados cuyos apellidos acaben con la letra “n”.

SELECT employee\_id,

CONCAT(first\_name, last\_name) NAME,

LENGTH (last\_name),

INSTR(last\_name, 'a') "Contains 'a'?"

FROM employees

WHERE SUBSTR(last\_name, -1, 1) = 'n';



#### - INITCAP

#### - SUBSTR

#### - LENGHT

#### - INSTR

#### - LPAD/RPAD

#### - TRIM

#### - REPLACE

### DE NÚMERO



**Funciones Numéricas**

* ROUND: redondea el valor a un decimal especificado
* TRUNC: trunca el valor a un decimal especificado
* MOD: devuelve el resto de la división

|  |  |
| --- | --- |
| **Función** | **Resultado** |
| ROUND(45.926, 2) | 45.93 |
| TRUNC(45.926, 2) | 45.92 |
| MOD(1600, 300) | 100 |

Las funciones numéricas aceptan entradas numéricas y devuelven valores numéricos. Esta sección describe algunas de las funciones numéricas.

|  |  |
| --- | --- |
| **Función** | **Objetivo** |
| **ROUND**(*column*|*expression*, *n*) | Redondear la columna, expresión o valor a n decimales o, si se omite *n*, a ningún decimal (Si *n* es negativo,  se redondearán los números a la izquierda del decimal.) |
| **TRUNC**(*column*|*expression*, *n*) | Truncar la columna, expresión o valor a *n* decimales o, si *n* se omite, el valor por defecto de *n* es cero |
| **MOD**(*m*,*n*) | Devolver el resto de *m* dividido entre *n* |

**Nota:** Esta lista contiene sólo algunas de las funciones numéricas disponibles.



**Uso de la Función ROUND**

**1**

**2**

**3**

DUAL es una tabla pública que puede utilizar para ver los resultados de funciones y cálculos.

SELECT ROUND(45.923,2),

ROUND(45.923,0),

ROUND(45.923,-1)

FROM dual;

La función **ROUND** redondea la columna, expresión o valor a *n* decimales. Si falta el segundo argumento o tiene un valor de 0, el valor se redondea a cero decimales. Si el segundo argumento tiene un valor de 2, el valor se redondea a dos decimales. Por el contrario, si el segundo argumento es –2, el valor se redondea a dos decimales a la izquierda (redondeados a la unidad más cercana a 100). La función **ROUND** también se puede utilizar con las funciones de fecha.

**Tabla DUAL:**

El usuario **SYS** es el propietario de la tabla **DUAL**, a la que pueden acceder todos los usuarios. Contiene una columna, DUMMY, y una fila con el valor X. La tabla DUAL es útil cuando sólo desea devolver un valor una vez (por ejemplo, el valor de una constante, pseudocolumna o expresión que no se deriva de una tabla con datos de usuario). La tabla DUAL se utiliza generalmente para obtener una visión más completa de la sintaxis de la cláusula **SELECT**, porque las cláusulas **SELECT** y **FROM** son obligatorias y muchos cálculos no tienen que realizar selecciones en las tablas reales.



**Uso de la Función TRUNC**

SELECT TRUNC(45.923,2),

TRUNC(45.923),

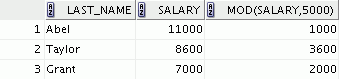
TRUNC(45.923,-1)

FROM dual;

La función **TRUNC** trunca la columna, expresión o valor a *n* decimales.

La función **TRUNC** funciona con argumentos similares a los de la función **ROUND**. Si falta el segundo argumento o tiene un valor de 0, el valor se trunca a cero decimales. Si el segundo argumento tiene un valor de 2, el valor se trunca a dos decimales. Por el contrario, si el segundo argumento tiene un valor de –2, el valor se trunca a dos decimales a la izquierda. Si el segundo argumento tiene un valor de –1, el valor se trunca a un decimal a la izquierda.

Al igual que la función **ROUND**, la función **TRUNC** se puede utilizar con funciones de fecha.



**Uso de la Función MOD**

Para todos los empleados con un puesto de vendedor, calcular el resto del salario después de dividirlo entre 5.000.

SELECT last\_name,

salary,

FROM employees

WHERE job\_id = 'SA\_REP';

La función **MOD** obtiene el resto del primer argumento dividido entre el segundo argumento. El ejemplo de la diapositiva calcula el resto del salario después de dividirlo entre 5.000 de todos los empleados cuyo ID de trabajo sea SA\_REP.

**Nota:** la función **MOD** se suele utilizar para determinar si un valor es par o impar. La función **MOD** es también la función hash de Oracle.

### DE FECHA



**Trabajo con Fechas**

Oracle Database almacena fechas en un formato numérico interno:

siglo, año, mes, día, horas, minutos y segundos.

* + El formato de visualización de la fecha por defecto es DD-MON-RR.

Permite almacenar fechas del siglo 21 en el siglo 20 especificando

sólo los dos últimos dígitos del año

* + - De la misma forma, permite almacenar fechas del siglo 20 en el siglo 21

SELECT last\_name,

hire\_date

FROM employees

WHERE hire\_date < '01-FEB-88';

# Trabajo con Fechas

*Oracle* *Database* almacena fechas en un formato numérico interno que representa el siglo, el año, el mes, las horas, los minutos y los segundos.

La visualización por defecto y el formato de entrada de cualquier fecha es DD-MON-RR. Las fechas de Oracle válidas son del 1 de enero de 4712 A.C. y el 31 de diciembre de 9999 D.C.

En el ejemplo de la diapositiva, la salida de la columna HIRE\_DATE aparece en el formato por defecto DD-MON-RR. Sin embargo, las fechas no se almacenan en la base de datos en este formato. Se almacenan todos los componentes de la fecha y la hora. Por lo tanto, aunque un valor HIRE\_DATE de 17-JUN-87 aparezca como el día, mes y año, también existe información de *hora* y *siglo* asociada a la fecha. Los datos completos podrían ser 17 de junio de 1987, 5:10:43 PM.



**Formato de Fecha RR**

|  |  |  |  |
| --- | --- | --- | --- |
| **Año Actual** | **Fecha Especificada** | **Formato RR** | **Formato YY** |
| **1995** | **27-OCT-95** | **1995** | **1995** |
| **1995** | **27-OCT-17** | **2017** | **1917** |
| **2001** | **27-OCT-17** | **2017** | **2017** |
| **2001** | **27-OCT-95** | **1995** | **2095** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Si el año de dos dígitos especificado es:** | |
| **0–49** | **50–99** |
| **Si dos de los dígitos del año actual son:** | **0–49** | **La fecha de devolución está en el siglo actual** | **La fecha de devolución está en el siglo anterior al actual** |
| **50–99** | **La fecha de devolución está en el siglo posterior al actual** | **La fecha de devolución está en el siglo actual** |

El formato de fecha RR es similar al elemento YY, pero puede utilizarlo para especificar siglos diferentes. Utilice el elemento de formato de fecha RR en lugar de YY para que el siglo del valor de retorno varíe según el año de dos dígitos especificado y los últimos dos dígitos del año actual. La tabla de la diapositiva resume el comportamiento del elemento RR.

|  |  |  |  |
| --- | --- | --- | --- |
| **Año Actual** | **Fecha Proporcionada** | **Interpretada (RR)** | **Interpretada (YY)** |
| 1994 | 27-OCT-95 | 1995 | 1995 |
| 1994 | 27-OCT-17 | 2017 | 1917 |
| 2001 | 27-OCT-17 | 2017 | 2017 |
| 2048 | 27-OCT-52 | 1952 | 2052 |
| 2051 | 27-OCT-47 | 2147 | 2047 |

Observe que los valores mostrados en las dos últimas filas de la tabla anterior. Conforme nos acercamos a la mitad del siglo, el comportamiento de RR puede que no sea lo que desea.

Estos datos se almacenan internamente de la siguiente forma:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CENTURY | YEAR | MONTH | DAY | HOUR MINUTE | SECOND |
| 19 | 87 | 06 | 17 | 17 10 | 43 |

**Siglos y el Año 2000**

Cuando se inserta un registro con una columna de fecha en una tabla, la información de siglose selecciona de la función **SYSDATE**. Sin embargo, cuando la columna de fecha se muestra en la pantalla, el componente de siglo no aparece (por defecto).

El tipo de dato **DATE** utiliza 2 bytes para la información de año, uno para el siglo y otro para el año. El valor de siglo siempre se incluye, independientemente de si se especifica o se muestra. En este caso, RR determina el valor por defecto para el siglo en **INSERT**.

SELECT sysdate

FROM dual;



**Uso de la Función SYSDATE**

SYSDATE es una función que devuelve:

* Fecha
* Hora



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de la Función SYSDATE

**SYSDATE** es una función de fecha que devuelve la fecha y hora actuales del servidor de base de datos. Puede utilizar **SYSDATE** como si utilizara cualquier otro nombre de columna. Por ejemplo, puede mostrar la fecha actual seleccionando **SYSDATE** de una tabla. Es muy común seleccionar **SYSDATE** de una tabla ficticia denominada **DUAL**.

**Nota: SYSDATE** devuelve la fecha y hora actuales definidas para el sistema operativo en el que reside la base de datos. Por lo tanto, si está en algún lugar de Australia y se conecta a una base de datos remota en una ubicación de Estados Unidos (EE. UU.), la función **SYSDATE** devolverá la fecha y hora de EE. UU. En ese caso, puede utilizar la función **CURRENT\_DATE** que devuelve la fecha actual en la zona horaria de la sesión.



**Operadores Aritméticos con Fechas**

* Sumar o restar un número de una fecha para obtener un valor de fecha resultante.
* Restar dos fechas para obtener el número de días entre esas fechas.
* Agregar horas a una fecha dividendo entre el número de horas entre 24.

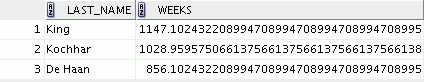
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Operadores Aritméticos con Fechas

Debido a que la base de datos almacena fechas como números, puede realizar cálculos utilizando operadores aritméticos como la suma y la resta. Puede agregar y restar constantes numéricas y fechas.

Puede realizar las siguientes operaciones:

|  |  |  |
| --- | --- | --- |
| **Operación** | **Resultado** | **Descripción** |
| fecha + número | Fecha | Suma un número de días a una fecha |
| fecha – número | Fecha | Resta un número de días de una fecha |
| fecha – fecha | Número de días | Resta una fecha de otra |
| fecha + número/24 | Fecha | Suma un número de horas a una fecha |



**Uso de Operadores Aritméticos con Fechas**

SELECT last\_name,

(SYSDATE-hire\_date)/7 AS WEEKS

FROM employees

WHERE department\_id = 90;

El ejemplo de la diapositiva muestra el apellido y el número de semanas durante las que han trabajado todos los empleados del departamento 90. Resta la fecha de contratación del empleado de la fecha actual (**SYSDATE**) y divide el resultado entre 7 para calcular el número de semanas durante las que ha trabajado el empleado.

**Nota: SYSDATE** es una función SQL que devuelve la fecha y hora actuales. Los resultados pueden ser diferentes según la fecha y hora definidas para el sistema operativo de la base de datos local al ejecutar la consulta SQL.

Si se resta una fecha más actual de una fecha más antigua, la diferencia es un número negativo.

**Agenda**

* Funciones SQL de una sola fila
* Funciones de carácter
* Funciones de número
* Trabajo con fechas
* Funciones de fecha



|  |  |
| --- | --- |
| **Función** | **Resultado** |
| MONTHS\_BETWEEN | Número de meses entre dos fechas |
| ADD\_MONTHS | Agregar meses de calendario a fecha |
| NEXT\_DAY | Siguiente día de la fecha especificada |
| LAST\_DAY | Último día del mes |
| ROUND | Redondear fecha |
| TRUNC | Truncar fecha |

### Funciones de Manipulación de Fecha

Las funciones de fecha funcionan en fechas de Oracle. Todas las funciones de fecha devuelven un valor del tipo de dato **DATE** excepto **MONTHS\_BETWEEN**, que devuelve un valor numérico.

* **MONTHS\_BETWEEN**(*date1, date2*): obtiene el número de meses entre *date1* y *date2*. El resultado puede ser positivo o negativo. Si *date1* es posterior a *date2*, el resultado es positivo; si *date1* es anterior a *date2*, el resultado es negativo. La parte del resultado que no sea un entero representa una parte del mes.
* **ADD\_MONTHS**(*date, n*): agrega el número *n* de los meses de calendario a *date*. El valor de *n* debe ser un entero y puede ser negativo.
* **NEXT\_DAY**(*date,* '*char*'): obtiene la fecha del siguiente día de la semana especificado ('*char*') que le sigue a *date*. El valor de *char* puede ser un número que representa un día o una cadena de caracteres.
* **LAST\_DAY**(*date*): obtiene la fecha del último día del mes que contiene *date.*

Esta lista es un subjuego de las funciones de fecha disponibles. Las funciones de número **ROUND** y **TRUNC** también se pueden utilizar para manipular los valores de fecha como se muestra a continuación:

* **ROUND**(*date*[,'*fmt*']): devuelve *date* redondeado a la unidad especificada por el modelo de formato *fmt.* Si se omite el modelo de formato *fmt* , *date* se redondea a la fecha más cercana.
* **TRUNC**(*date*[, '*fmt*']): devuelve *date* con la parte de la hora del día truncada a la unidad especificada por el modelo de formato *fmt*. Si se omite el modelo de formato *fmt*,*date* se redondea a la fecha más cercana.

Los siguientes modelos de formato se tratan en las lecciones tituladas “Uso de Funciones de Conversión y Expresiones Condicionales”:

|  |  |
| --- | --- |
| **Función** | **Resultado** |
| MONTHS\_BETWEEN  ('01-SEP-95','11-JAN-94') | 19.6774194 |
| ADD\_MONTHS (‘31-JAN-96',1) | '29-FEB-96' |
| NEXT\_DAY ('01-SEP-95','FRIDAY') | '08-SEP-95' |
| LAST\_DAY ('01-FEB-95') | '28-FEB-95' |

### Uso de las Funciones de Fecha

En el ejemplo de la diapositiva, la función **ADD\_MONTHS** agrega un mes al valor de fecha proporcionado “31-JAN-96” y devuelve “29-FEB-96”. La función reorganiza el año 1996 como año bisiesto y, por lo tanto, devuelve la última fecha del mes de febrero. Si cambia el valor de fecha de entrada a “31-JAN-95”, la función devuelve “28-FEB-95”.

Por ejemplo, muestre el número de empleado, fecha de contratación, número de meses durante los que ha trabajado, fecha de revisión de seis meses, primer viernes después de la fecha de contratación y último día del mes de contratación de todos los empleados que han trabajado menos de 150 meses.

SELECT employee\_id,

hire\_date,

MONTHS\_BETWEEN (SYSDATE, hire\_date) TENURE,

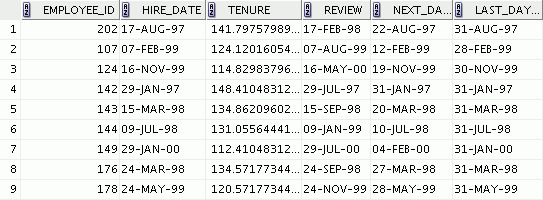
ADD\_MONTHS (hire\_date, 6) REVIEW,

NEXT\_DAY (hire\_date, 'FRIDAY'),

LAST\_DAY(hire\_date)

FROM employees

WHERE MONTHS\_BETWEEN (SYSDATE, hire\_date) < 150;



|  |  |
| --- | --- |
| **Función** | **Resultado** |
| ROUND(SYSDATE,'MONTH') | 01-AUG-03 |
| ROUND(SYSDATE ,'YEAR') | 01-JAN-04 |
| TRUNC(SYSDATE ,'MONTH') | 01-JUL-03 |
| TRUNC(SYSDATE ,'YEAR') | 01-JAN-03 |

### Uso de las Funciones ROUND y TRUNC con Fechas

Las funciones **ROUND** y **TRUNC** se pueden utilizar para valores de fecha y número. Al utilizarlas con fechas, estas funciones redondean o truncan el modelo de formato especificado. Por lo tanto, puede redondear las fechas al año o mes más cercano. Si el modelo de formato es mes, el resultado de las fechas 1-15 es el primer día del mes actual. El resultado de las fechas 16-31 es el primer día del siguiente mes. Si el modelo de formato es mes, el resultado de las fechas 1-6 es el 1 de enero del año actual. El resultado de los meses 7-12 es el 1 de enero del siguiente año.

**Ejemplo:**

Compare las fechas de contratación de todos los empleados que empezaron en 1997. Muestre el número de empleado, fecha de contratación y mes de inicio con las funciones **ROUND** y **TRUNC**.

SELECT employee\_id,

hire\_date,

ROUND (hire\_date, 'MONTH'),

TRUNC(hire\_date, 'MONTH')

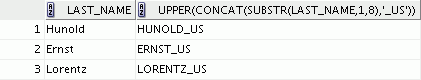
FROM employees

WHERE hire\_date LIKE '%97';



### Funciones de Anidación

Las funciones de una sola fila se pueden anidar en cualquier profundidad. Las funciones anidadas se evalúan desde el nivel más profundo hasta el nivel menos profundo. Los siguientes ejemplos muestran la flexibilidad de estas funciones.



SELECT last\_name, UPPER(CONCAT(SUBSTR (LAST\_NAME, 1, 8), '\_US'))

FROM employees

WHERE department\_id = 60;



**Funciones de Anidación: Ejemplo 1**

El ejemplo de la diapositiva muestra los apellidos de los empleados en el departamento 60. La evaluación de la sentencia SQL implica tres pasos:

1. La función interna recupera los primeros ocho caracteres del apellido.

Result1 = SUBSTR (LAST\_NAME, 1, 8)

1. La función externa concatena el resultado con:

\_US. Result2 = CONCAT(Result1, '\_US')

1. La función más externa convierte los resultados a mayúsculas.

La expresión completa se convierte en la cabecera de columna porque no se ha proporcionado ningún alias de columna.

**Ejemplo:**

Muestra la fecha del siguiente viernes seis meses después de la fecha de contratación. La fecha resultante debe aparecer como viernes 13 de agosto de 1999. Ordene los resultados por fecha de contratación.

SELECT TO\_CHAR(NEXT\_DAY(ADD\_MONTHS (

hire\_date, 6), 'FRIDAY'),

'fmDay, Month ddth, YYYY') "Next 6 Month Review"

FROM employees

ORDER BY hire\_date ;

SELECT TO\_CHAR(ROUND((salary/7), 2),'99G999D99', 'NLS\_NUMERIC\_CHARACTERS = '',.'' ') TO\_CHAR(ROUND((salary/7), 2),'99G999D99', 'NLS\_NUMERIC\_CHARACTERS = '',.'' ')AS "Formatted Salary"

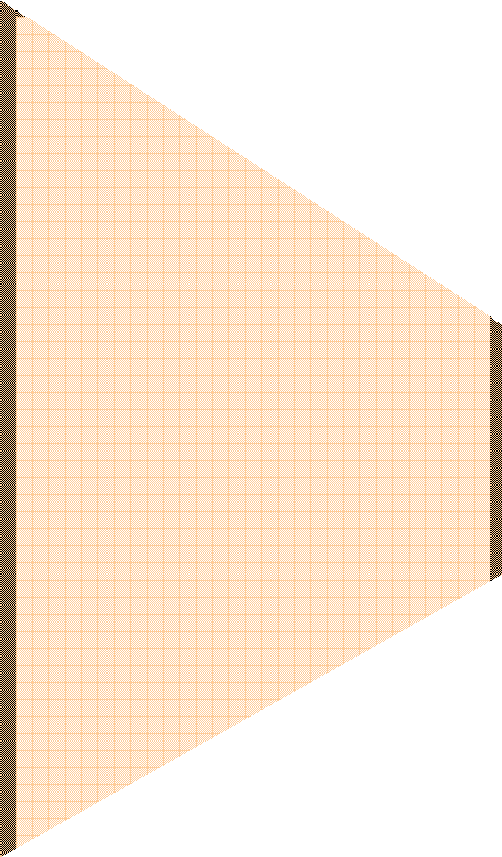
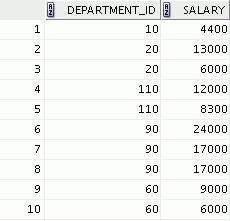
FROM employees;

El ejemplo de la diapositiva muestra los salarios de los empleados dividido entre7 y redondeado a dos decimales. El resultado se formatea para mostrar el salario con notación en danés. Es decir, la coma se utilizar como decimal y el punto para los miles.

En primer lugar, la función interna **ROUND** se ejecuta para redondear el valor del salario dividido entre 7 a dos decimales. La función **TO\_CHAR** se utiliza entonces para aplicar formato al resultado de la función **ROUND**.

**Nota:** los elementos D y G especificados en el parámetro de función **TO\_CHAR** son elementos de formato de número. D devuelve un carácter decimal en la posición especificada. G se utiliza como un separador de grupo.

## DE GRUPO



**¿Qué Son las Funciones de Grupo?**

Las funciones de grupo funcionan en juegos de filas para proporcionar un resultado por grupo.

**EMPLOYEES**

**Salario máximo en la tabla EMPLOYEES**

|  |  |
| --- | --- |
|  |  |
| **…** |  |
|  |  |



A diferencia de las funciones de una sola fila, las funciones de grupo funcionan en juegos de filas para proporcionar un resultado por grupo. Estos juegos pueden formar la tabla completa o la tabla dividida en grupos.



**Tipos de Funciones de Grupo**

* AVG
* COUNT
* MAX
* MIN
* STDDEV
* SUM
* VARIANCE

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Funciones de grupo**

**Tipos de Funciones de Grupo**

Cada una de las funciones acepta un argumento. La siguiente tabla identifica las opciones que se pueden utilizar en la sintaxis:

|  |  |
| --- | --- |
| **Función** | **Descripción** |
| **AVG**([DISTINCT|ALL]*n*) | Valor medio de *n*; ignora los valores nulos |
| **COUNT**({\*|[DISTINCT|ALL]*expr*  }) | Número de filas donde expr evalúa otros valores que no son nulos (tiene en cuenta todas  filas seleccionadas con \*, incluyendo duplicados y filas con valores nulos) |
| **MAX**([DISTINCT|ALL]*expr*) | Valor máximo de expr; ignora los valores nulos |
| **MIN**([DISTINCT|ALL]*expr*) | Valor mínimo de expr; ignora los valores nulos |
| **STDDEV**([DISTINCT|ALL]*n*) | Desviación estándar de *n*; ignora los valores nulos |
| **SUM**([DISTINCT|ALL]*n*) | Valores de suma de *n*; ignora los valores nulos |
| **VARIANCE**([DISTINCT|ALL]*n*) | Varianza de *n*; ignora los valores nulos |

SELECT group\_function(column), ...

FROM table

[WHERE condition]

[ORDER BY column];



**Funciones de Grupo: Sintaxis**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Funciones de Grupo: Sintaxis**

La función de grupo se coloca después de la palabra clave **SELECT**. Puede que tenga varias funciones de grupo separadas por comas.

Instrucciones para utilizar las funciones de grupo:

* **DISTINCT** hace que la función considere sólo los valores no duplicados; **ALL** hace que considere cada valor, incluyendo los duplicados. El valor por defecto es **ALL** y, por lo tanto, no es necesario especificarlo.
* Los tipos de dato para las funciones con el argumento expr pueden ser **CHAR**, **VARCHAR2**, **NUMBER** o **DATE**.
* Todas las funciones de grupo ignoran los valores nulos. Para sustituir un valor para valores nulos, utilice las funciones **NVL**, **NVL2**, **COALESCE**, **CASE** o **DECODE**.

SELECT AVG(salary),

MAX(salary),

MIN(salary),

SUM(salary)

FROM employees

WHERE job\_id LIKE '%REP%';



**Uso de las Funciones AVG y SUM**

Puede utilizar AVG y SUM para datos numéricos.



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de las Funciones AVG y SUM

Puede utilizar las funciones **AVG**, **SUM**, **MIN** y **MAX** en las columnas que pueden almacenar datos numéricos. El ejemplo de la diapositiva muestra la media, el valor más alto, más bajo y la suma de los salarios mensuales de todos los vendedores.



**Uso de Funciones MIN y MAX**

Puede utilizar MIN y MAX para tipos de dato numéricos, de caracteres y de fecha.

SELECT MIN(hire\_date), MAX(hire\_date)

FROM employees;

### Uso de Funciones MIN y MAX

Puede utilizar las funciones **MAX** y **MIN** para tipos de dato numéricos, de caracteres y de fecha. El ejemplo de la diapositiva muestra los empleados más y menos experimentados.

El siguiente ejemplo muestra el apellido del empleado que está en primer lugar y del que está en último lugar en una lista de todos los empleados ordenada alfabéticamente:

SELECT MIN(last\_name),

MAX(last\_name)

FROM employees;



**Nota:** las funciones **AVG**, **SUM**, **VARIANCE** y **STDDEV** se pueden utilizar sólo con los tipos de dato numéricos. **MAX** and **MIN** no se pueden utilizar con tipos de dato **LOB** o **LONG**.



**Uso de la Función COUNT**

COUNT(\*) devuelve el número de filas en una tabla:

**1**

COUNT(*expr*) devuelve el número de filas con valores no nulos para la expresión *expr*:

**2**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**SELECT COUNT(commission\_pct) FROM employees**

**WHERE department\_id = 80;**

**SELECT COUNT(\*)**

**FROM employees**

**WHERE department\_id = 50;**

### Uso de la Función COUNT

La función **COUNT** tiene tres formatos:

* **COUNT**(\*)
* **COUNT**(*expr*)
* **COUNT**(**DISTINCT** *expr*)

**COUNT(\*)** devuelve el número de filas en una tabla que cumplan con el criterio de la sentencia **SELECT**, incluyendo las filas duplicadas y las filas que contengan valores nulos en cualquiera de las columnas. Si se incluye una cláusula **WHERE** en la sentencia **SELECT**, **COUNT(\*)** devuelve el número de filas que cumpla con la condición de la cláusula **WHERE**.

Por el contrario, **COUNT(*expr)***devuelve el número de valores no nulos que están en la columna identificada con *expr*.

**COUNT(DISTINCT *expr*)** devuelve el número de valores únicos no nulos que están en la columna identificada con *expr*.

**Ejemplos:**

1. El ejemplo de la diapositiva muestra el número de empleados del departamento 50.
2. También muestra el número de empleados del departamento 80 que pueden percibir una comisión.

SELECT COUNT(DISTINCT department\_id)

FROM employees;



**Uso de la Palabra Clave DISTINCT**

* COUNT(DISTINCT expr) devuelve el número con valores distintos no nulos de *expr*.
* Para mostrar el número de valores distintos de departamento en la tabla EMPLOYEES:



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de la Palabra Clave DISTINCT

Utilice la palabra clave **DISTINCT** para suprimir el recuento de cualquier valor duplicado en una columna.

El ejemplo de la diapositiva muestra el número de valores de departamento distintos que están en la tabla EMPLOYEES.



**Funciones de Grupo y Valores Nulos**

Las funciones de grupo ignoran los valores nulos de la columna:

La función NVL fuerza las funciones de grupo para que incluyan valores nulos.



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**SELECT AVG(NVL(commission\_pct, 0)) FROM employees;**

**SELECT AVG(commission\_pct) FROM employees;**

**Funciones de Grupo y Valores Nulos**

Todas las funciones de grupo ignoran los valores nulos de la columna.

Sin embargo, **NVL** fuerza las funciones de grupo para que incluyan valores nulos.

**Ejemplos:**

1. La media se calcula *únicamente* en base a las filas de la tabla en las que se almacena un valor válido en la columna COMMISSION\_PCT. La media se calcula con la comisión total pagada a todos los empleados dividida entre el número de empleados que perciben una comisión (cuatro).
2. La media se calcula en base a *todas* las filas de la tabla, independientemente de si los valores nulos se almacenan en la columna COMMISSION\_PCT. La media se calcula con la comisión total pagada a todos los empleados dividida entre el número de empleados de la compañía (20).

### FUNCIONES DE GRUPO ANIDADAS

**F3(F2(F1(col,arg1),arg2),arg3)**



**Funciones de Anidación**

* Las funciones de una sola fila se pueden anidar en cualquier nivel.
* Las funciones anidadas se evalúan desde el nivel más profundo hasta el nivel menos profundo.

**Paso 1 = Resultado 1**

**Paso 2 = Resultado 2**

**Paso 3 = Resultado 3**

# funciones

Las funciones son similares a las de los operadores, ya que manipulan los elementos de datos y devuelven un resultado. Las funciones difieren de los operadores en el formato de sus argumentos. Este formato les permite operar en cero, uno, dos o más argumentos:

function ( argumento , argumento , ...)

Una función sin ningún argumento es similar a una pseudocolumna (consulte el [Capítulo 2, "Pseudocolumnas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/pseudocolumns.htm#g1020307) ). Sin embargo, una pseudocolumna generalmente devuelve un valor diferente para cada fila en el conjunto de resultados, mientras que una función sin ningún argumento generalmente devuelve el mismo valor para cada fila.

Este capítulo contiene estas secciones:

* [Acerca de las funciones SQL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions001.htm#CJAHCIID)
* [Funciones de una sola fila](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAJHBIA)
  + [Funciones Numéricas](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAIBHGG)
  + [Funciones de carácter que devuelven valores de carácter](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAEEJFC)
  + [Funciones de caracteres que devuelven valores numéricos](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAEHEJD)
  + [Funciones del conjunto de caracteres](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJADIIDD)
  + [Funciones de fecha y hora](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#i88891)
  + [Funciones de comparación general](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJADDIID)
  + [Funciones de conversión](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#i88892)
  + [Funciones de objetos grandes](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAHACCG)
  + [Funciones de colección](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAEEAJB)
  + [Funciones jerárquicas](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAECDEB)
  + [Funciones de minería de datos](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAIHADE)
  + [Funciones XML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CIHGFCGD)
  + [Funciones de codificación y decodificación](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAGDJCF)
  + [Funciones relacionadas con NULL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJAFHIFF)
  + [Entorno y funciones identificadoras](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#CJACFICF)
* [Funciones agregadas](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)
* [Funciones analiticas](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407)
* [Funciones de referencia de objetos](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions005.htm#CJABFGFJ)
* [Funciones del modelo](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions006.htm#CJADDCEB)
* [Funciones OLAP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions007.htm#CIHBFFEA)
* [Funciones del cartucho de datos](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions008.htm#CIHGCBEH)
* [Acerca de las funciones definidas por el usuario](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions256.htm#i1012049)

# Acerca de las funciones SQL

Las funciones de SQL están integradas en la base de datos Oracle y están disponibles para su uso en varias declaraciones SQL apropiadas. No confunda las funciones de SQL con las funciones definidas por el usuario escritas en PL / SQL.

Si llama a una función SQL con un argumento de un tipo de datos distinto al tipo de datos esperado por la función SQL, entonces Oracle intenta convertir el argumento al tipo de datos esperado antes de realizar la función SQL.

**Nulos en funciones SQL**

La mayoría de las funciones escalares devuelven nulo cuando se les da un argumento nulo. Puede utilizar la función NVL para devolver un valor cuando se produce un nulo. Por ejemplo, la expresión NVL(commission\_pct,0) devuelve 0 si commission\_pct es nulo o el valor de commission\_pct si no es nulo.

**Sintaxis para funciones SQL**En los diagramas de sintaxis para funciones SQL, los argumentos se indican por sus tipos de datos. Cuando el parámetro *function* aparece en la sintaxis SQL, reemplácelo con una de las funciones descritas en esta sección. Las funciones se agrupan por los tipos de datos de sus argumentos y sus valores de retorno.

**Nota:**

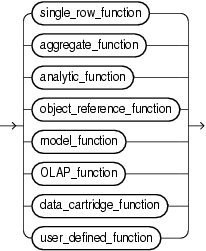
Cuando aplica funciones de SQL a las columnas de LOB, la base de datos Oracle crea LOB temporales durante el procesamiento de SQL y PL / SQL. Debe asegurarse de que la cuota de espacio de tablas temporal sea suficiente para almacenar estos LOB temporales para su aplicación.

**Nota:**

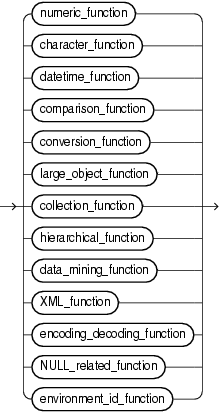
Los valores combinados de la configuración NLS\_COMP y NLS\_SORT determinan las reglas según las cuales se clasifican y comparan los caracteres. Si NLS\_COMP está configurado LINGUISTIC para su base de datos, todas las entidades en este capítulo se interpretarán de acuerdo con las reglas especificadas por el parámetro NLS\_SORT. Si NLS\_COMP no está configurado en LINGUISTIC, las funciones se interpretan sin importar la configuración NLS\_SORT. NLS\_SORT se puede establecer explícitamente. Si no se establece explícitamente, se deriva de NLS\_LANGUAGE.

La sintaxis que muestra las categorías de funciones es la siguiente:

***función* :: =**

  
[Descripción de la ilustración '' function.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/function.htm)

***single\_row\_function* :: =**

  
[Descripción de la ilustración '' single\_row\_function.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/single_row_function.htm)

Las secciones que siguen enumeran las funciones de SQL incorporadas en cada uno de los grupos ilustrados en los diagramas anteriores, excepto las funciones definidas por el usuario. Todas las funciones SQL incorporadas se describen en orden alfabético.

# Funciones de una sola fila

Las funciones de una sola fila devuelven una única fila de resultados para cada fila de una tabla o vista consultada. Estas funciones pueden aparecer en listas de selección, cláusulas  **WHERE START WITH**, **CONNECT BY**, y **HAVING**.

## Funciones Numéricas

Las funciones numéricas aceptan entradas numéricas y devuelven valores numéricos. La mayoría de las funciones numéricas devuelven valores NUMBER pue son exactos a 38 dígitos decimales.

Las funciones trascendentales COS, COSH, EXP, LN, LOG, SIN, SINH, SQRT, TAN, y TANH son exactos a 36 dígitos decimales.

Las funciones trascendentales ACOS, ASIN, ATAN, y ATAN2 tienen una precisión de 30 dígitos decimales. Las funciones numéricas son:

[ABS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions009.htm#i81802)  
[ACOS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions010.htm#i76693)  
[ASIN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions015.htm#i1056765)  
[ATAN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions016.htm#i76792)  
[ATAN2](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions017.htm#i76816)  
[BITAND](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions021.htm#i98000)  
[CEIL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions024.htm#i97801)  
[COS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions037.htm#i77138)  
[COSH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions038.htm#i82384)  
[EXP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions058.htm#i1017093)  
[FLOOR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions067.htm#i77449)  
[LN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions090.htm#i83932)  
[LOG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions093.htm#i84140)  
[MOD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions101.htm#i77996)  
[NANVL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions103.htm#i1273120)  
[POTENCIA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions129.htm#i78493)  
[RESTO](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions152.htm#i1300767)  
[ROUND (número)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions155.htm#i78633)  
[MUESTRA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions164.htm#i90056)  
[SIN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions165.htm#i78804)  
[SENOH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions166.htm#i78829)  
[SQRT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions168.htm#i1009289)  
[TAN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions193.htm#i79242)  
[TANH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions194.htm#i79267)  
[TRUNC (número)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions221.htm#i79729)  
[WIDTH\_BUCKET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions234.htm#i1001630)

## Funciones de carácter que devuelven valores de carácter

Las funciones de caracteres que devuelven valores de caracteres devuelven valores de los siguientes tipos de datos a menos que se documente lo contrario:

* Si el argumento de entrada es CHAR o VARCHAR2, entonces el valor devuelto es VARCHAR2.
* Si el argumento de entrada es NCHAR o NVARCHAR2, entonces el valor devuelto es NVARCHAR2.

La longitud del valor devuelto por la función está limitada por la longitud máxima del tipo de datos devuelto.

* Para las funciones que devuelven CHAR o VARCHAR2, si la longitud del valor devuelto supera el límite, la base de datos Oracle lo trunca y devuelve el resultado sin un mensaje de error.
* Para las funciones que devuelven valores CLOB, si la longitud de los valores devueltos excede el límite, Oracle genera un error y no devuelve datos.

Las funciones de caracteres que devuelven valores de caracteres son:

[CHR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions026.htm#i76965)  
[CONCAT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions033.htm#i77004)  
[INITCAP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions074.htm#i77574)  
[LOWER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions094.htm#i1043828)  
[LPAD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions095.htm#i1371196)  
[LTRIM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions096.htm#i77875)  
[NCHR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions104.htm#i1273067)  
[NLS\_INITCAP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions110.htm#i89841)  
[NLS\_LOWER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions111.htm#i78373)  
[NLS\_UPPER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions112.htm#i89889)  
[NLSSORT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions113.htm#i78399)  
[REGEXP\_REPLACE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions149.htm#i1305521)  
[REGEXP\_SUBSTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions150.htm#i1239858)  
[REPLACE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions153.htm#i78608)  
[RPAD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions159.htm#i78723)  
[RTRIM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions160.htm#i1018967)  
[SOUNDEX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions167.htm#i78853)  
[SUBSTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions181.htm#i87066)  
[TRANSLATE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions216.htm#i1501659)  
[TRANSLATE ... USANDO](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions217.htm#i79617)  
[TRIM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions219.htm#i79689)  
[UP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions226.htm#i90176)PPER

## Funciones de caracteres que devuelven valores numéricos

Las funciones de caracteres que devuelven valores numéricos pueden tomar como argumento cualquier tipo de datos de caracteres. Las funciones de caracteres que devuelven valores numéricos son:

[ASCII](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions013.htm#i1152151)  
[INSTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions080.htm#i77598)  
[LONGITUD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions088.htm#i77725)  
[REGEXP\_COUNT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions147.htm#CIHDAIHJ)  
[REGEXP\_INSTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions148.htm#i1239887)

## Funciones del conjunto de caracteres

Las funciones del conjunto de caracteres devuelven información sobre el conjunto de caracteres. Las funciones del conjunto de caracteres son:

[NLS\_CHARSET\_DECL\_LEN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions107.htm#i85356)  
[NLS\_CHARSET\_ID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions108.htm#i78230)  
[NLS\_CHARSET\_NAME](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions109.htm#i78300)

## Funciones de fecha y hora

Funciones de fecha y hora operan en la fecha ( DATE), marca de tiempo ( TIMESTAMP, TIMESTAMP WITH TIME ZONEy TIMESTAMP WITH LOCALTIME ZONE), y el intervalo ( INTERVAL DAY TO SECOND, INTERVAL YEAR TO MONTH) valores.

Algunas de las funciones de fecha y hora fueron diseñados para el tipo de datos DATE de Oracle ( ADD\_MONTHS, CURRENT\_DATE, LAST\_DAY, NEW\_TIME, y NEXT\_DAY). Si proporciona un valor de marca de tiempo como su argumento, la base de datos Oracle convierte internamente el tipo de entrada en un valor DATE y devuelve un valor DATE. Las excepciones son la función MONTHS\_BETWEEN, que devuelve un número, y las funciones ROUND y TRUNC, que no aceptan en absoluto la marca de tiempo o los valores de intervalo.

Las funciones de fecha y hora restantes se diseñaron para aceptar cualquiera de los tres tipos de datos (fecha, fecha y hora e intervalo) y para devolver un valor de uno de estos tipos.

Todas las funciones de fecha y hora que devuelven información de fecha y hora actual del sistema, tales como SYSDATE, SYSTIMESTAMP, CURRENT\_TIMESTAMP, y así sucesivamente, son evaluados una vez para cada instrucción SQL, sin tener en cuenta el número de veces que se hace referencia en esa declaración.

Las funciones de fecha y hora son:

[ADD\_MONTHS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions011.htm#i76717)  
[CURRENT\_DATE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions044.htm#i999792)  
[CURRENT\_TIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions045.htm#i999217)  
[DBTIMEZONE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions048.htm#i1327612)  
[EXTRACT (fecha y hora)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions059.htm#i1017161)  
[FROM\_TZ](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions068.htm#i999803)  
[LAST\_DAY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions084.htm#i83733)  
[LOCALTIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions092.htm#i999873)  
[MONTHS\_BETWEEN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions102.htm#i78039)  
[NEW\_TIME](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions105.htm#i1004085)  
[NEXT\_DAY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions106.htm#i78154)  
[numtodsinterval](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions117.htm#i88258)  
[NUMTOYMINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions118.htm#i89943)  
[ORA\_DST\_AFFECTED](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions121.htm#CJACAGAI)  
[ORA\_DST\_CONVERT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions122.htm#CJAGGAGA)  
[ORA\_DST\_ERROR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions123.htm#CJAGGCAB)  
[ROUND (fecha)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions154.htm#i78665)  
[SESSIONTIMEZONE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions162.htm#i999827)  
[SYS\_EXTRACT\_UTC](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions186.htm#i999831)  
[SYSDATE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions191.htm#i79216)  
[SYSTIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions192.htm#i999835)  
[TO\_CHAR (fecha y hora)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions200.htm#i1009324)  
[TO\_DSINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions204.htm#i1014645)  
[TO\_TIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions213.htm#i999843)  
[TO\_TIMESTAMP\_TZ](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions214.htm#i999847)  
[TO\_YMINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions215.htm#i999851)  
[TRUNC (fecha)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions220.htm#i79761)  
[TZ\_OFFSET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions222.htm#i1001830)

## Funciones de comparación general

Las funciones de comparación generales determinan el valor máximo o mínimo de un conjunto de valores. Las funciones de comparación generales son: ¿

## Funciones de conversión

Las funciones de conversión convierten un valor de un tipo de datos a otro. Generalmente, la forma de los nombres de funciones sigue la convención *datatype* TO *datatype*. El primer tipo de datos es el tipo de datos de entrada. El segundo tipo de datos es el tipo de datos de salida. Las funciones de conversión de SQL son:

[ASCIISTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions014.htm#i1000190)  
[BIN\_TO\_NUM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions020.htm#i1002025)  
[REPARTO](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions023.htm#i1269136)  
[CHARTOROWID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions025.htm#i76940)  
[COMPOSE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions032.htm#i1000195)  
[CONVERTIR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions034.htm#i77037)  
[DECOMPOSE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions050.htm#i1000217)  
[HEXTORAW](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions073.htm#i1023950)  
[numtodsinterval](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions117.htm#i88258)  
[NUMTOYMINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions118.htm#i89943)  
[RAWTOHEX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions143.htm#i78517)  
[RAWTONHEX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions144.htm#i1006713)  
[ROWIDTOCHAR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions157.htm#i78696)  
[ROWIDTONCHAR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions158.htm#i1293762)  
[SCN\_TO\_TIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions161.htm#BABEHBCB)  
[TIMESTAMP\_TO\_SCN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions195.htm#BABFJHAD)  
[TO\_BINARY\_DOUBLE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions196.htm#i1279235)  
[TO\_BINARY\_FLOAT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions197.htm#i1272891)  
[TO\_BLOB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions198.htm#CIHIGGFC)  
[TO\_CHAR (carácter)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions199.htm#i1006717)  
[TO\_CHAR (fecha y hora)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions200.htm#i1009324)  
[TO\_CHAR (número)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions201.htm#i79330)  
[TO\_CLOB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions202.htm#i1006718)  
[TO\_DATE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions203.htm#i1003589)  
[TO\_DSINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions204.htm#i1014645)  
[TO\_LOB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions205.htm#i79464)  
[TO\_MULTI\_BYTE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions206.htm#i79496)  
[TO\_NCHAR (carácter)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions207.htm#i1006719)  
[TO\_NCHAR (fecha y hora)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions208.htm#i1006720)  
[TO\_NCHAR (número)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions209.htm#i1006721)  
[TO\_NCLOB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions210.htm#i1006722)  
[TO\_NUMBER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions211.htm#i79512)  
[TO\_SINGLE\_BYTE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions212.htm#i79556)  
[TO\_TIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions213.htm#i999843)  
[TO\_TIMESTAMP\_TZ](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions214.htm#i999847)  
[TO\_YMINTERVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions215.htm#i999851)  
[TRATAR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions218.htm#i1018806)  
[UNISTR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions224.htm#i1000165)

## Funciones de objetos grandes

Las funciones de objetos grandes operan en LOBs. Las funciones de objeto grande son:

[BFILENAME](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions019.htm#i76871)  
[EMPTY\_BLOB, EMPTY\_CLOB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions056.htm#i77384)

## Funciones de colección

Las funciones de colección operan en tablas anidadas y varrays. Las funciones de recopilación de SQL son:

[CARDINALITY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions022.htm#i1269375)  
[COLLECT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions031.htm#i1271564)  
[POWERMULTISET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions130.htm#i1269496)  
[POWERMULTISET\_BY\_CARDINALITY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions131.htm#i1399971)  
[SET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions163.htm#i1269374)

## Funciones jerárquicas

Las funciones jerárquicas aplican información de ruta jerárquica a un conjunto de resultados. La función jerárquica es:

[SYS\_CONNECT\_BY\_PATH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions183.htm#i1038266)

## Funciones de minería de datos

Las funciones de minería de datos operan en modelos que se han construido utilizando el paquete DBMS\_DATA\_MINING o la API de Oracle Data Mining Java. Las funciones de minería de datos SQL son:

[CLUSTER\_ID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions027.htm#CJAJDDJB)  
[CLUSTER\_PROBABILITY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions028.htm#CJAIHHGE)  
[CLUSTER\_SET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions029.htm#CJACEBAB)  
[FEATURE\_ID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions062.htm#CJADFCIB)  
[FEATURE\_SET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions063.htm#CJAHHFJE)  
[FEATURE\_VALUE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions064.htm#CJAIFECA)  
[PREDICTION](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions132.htm#CJAFCHEG)  
[PREDICTION\_BOUNDS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions133.htm#CIHDDGEE)  
[PREDICTION\_COST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions134.htm#CJABAJED)  
[PREDICTION\_DETAILS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions135.htm#CJABHGHD)  
[PREDICTION\_PROBABILITY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions136.htm#CJAJDHHD)  
[PREDICTION\_SET](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions137.htm#CJAJGHAI)

## Funciones XML

Las funciones XML operan o devuelven documentos o fragmentos XML. Estas funciones utilizan argumentos que no están definidos como parte del Estándar ANSI / ISO / IEC SQL, pero están definidos como parte de los estándares del World Wide Web Consortium (W3C). El procesamiento y las operaciones que realizan las funciones están definidos por los estándares relevantes de W3C. La siguiente tabla proporciona un enlace a la sección correspondiente del estándar W3C para las reglas y pautas que se aplican a cada uno de estos argumentos relacionados con XML. Una declaración SQL que utiliza una de estas funciones XML, donde cualquiera de los argumentos no se ajusta a la sintaxis W3C relevante, generará un error. De especial interés es el hecho de que no todos los caracteres permitidos en el valor de una columna de base de datos se consideran legales en XML.

| **Elemento de sintaxis** | **URL estándar de W3C** |
| --- | --- |
| *value\_expr* | <http://www.w3.org/TR/2006/REC-xml-20060816> |
| *Xpath\_string* | <http://www.w3.org/TR/1999/REC-xpath-19991116> |
| *XQuery\_string* | <http://www.w3.org/TR/2007/REC-xquery-semantics-20070123/> |
| *namespace\_string* | <http://www.w3.org/TR/2006/REC-xml-names-20060816/> |
| *identifier* | <http://www.w3.org/TR/2006/REC-xml-20060816/#NT-Nmtoken> |

Las funciones SQL XML son:

[APPENDCHILDXML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions012.htm#CIHEGIFE)  
[DELETEXML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions051.htm#CIHEGJCB)  
[PROFUNDIDAD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions053.htm#i1150333)  
[EXISTSNODE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions057.htm#i1006711)  
[extracto (XML)](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions060.htm#i1006712)  
[EXTRACTVALUE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions061.htm#i1131042)  
[INSERTCHILDXML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions075.htm#CIHIJEBB)  
[INSERTCHILDXMLAFTER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions076.htm#CJABGABC)  
[INSERTCHILDXMLBEFORE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions077.htm#CJAECEHC)  
[INSERTXMLAFTER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions078.htm#CIHIGCDA)  
[INSERTXMLBEFORE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions079.htm#CIHICIAF)  
[PATH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions125.htm#i1150364)  
[SYS\_DBURIGEN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions185.htm#i1180070)  
[SYS\_XMLAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions189.htm#i1006716)  
[SYS\_XMLGEN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions190.htm#i1004367)  
[UPDATEXML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions225.htm#i1134878)  
[XMLAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions235.htm#i1286333)  
[XMLCAST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions236.htm#CIHHHACJ)  
[XMLCDATA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions237.htm#CIHFBEGB)  
[XMLCOLATTVAL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions238.htm#i1129374)  
[XMLCOMMENT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions239.htm#CIHJIBJA)  
[XMLCONCAT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions240.htm#i1166713)  
[XMLDiff](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions241.htm#CIHFDJAA)  
[XmlElement](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions242.htm#i1129193)  
[XMLEXISTS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions243.htm#CIHDEFCD)  
[XMLFOREST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions244.htm#i1172404)  
[XMLISVALID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions245.htm#BABJCJIG)  
[XMLPARSE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions246.htm#CIHCIIFA)  
[XMLPATCH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions247.htm#CIHDAEEC)  
[XMLPI](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions248.htm#CIHGJCBF)  
[XMLQUERY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions249.htm#CIHHFFGG)  
[XmlRoot](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions250.htm#CIHDAHBJ)  
[XMLSEQUENCE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions251.htm#i1129570)  
[XMLSERIALIZE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions252.htm#CIHDHGJD)  
[XMLTABLE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions253.htm#CIHGGHFB)  
[XmlTransform](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions254.htm#i1130976)

## Funciones de codificación y decodificación

Las funciones de codificación y decodificación le permiten inspeccionar y decodificar datos en la base de datos. Las funciones de codificación y decodificación son:

[DECODE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions049.htm#i1017437)  
[DUMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions055.htm#i77278)  
[ORA\_HASH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions124.htm#i1235081)  
[VSIZE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions233.htm#i80071)

## Funciones relacionadas con NULL

Las funciones NULL relacionadas facilitan el manejo nulo.

Las funciones NULL relacionadas son:

[COALESCE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions030.htm#i1001341)  
[LNNVL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions091.htm#i1479295)  
[NANVL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions103.htm#i1273120)  
[NULLIF](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions116.htm#i1001340)  
[NVL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions119.htm#i91798)  
[NVL2](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions120.htm#i91806)

## Entorno y funciones identificadoras

Las funciones de entorno e identificador proporcionan información sobre la instancia y la sesión. Las funciones de entorno e identificador son:

[SYS\_CONTEXT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions184.htm#i1038176)  
[SYS\_GUID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions187.htm#i79194)  
[SYS\_TYPEID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions188.htm#i1044156)  
[UID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions223.htm#i79792)  
[USER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions227.htm#i79833)  
[USERENV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions228.htm#i79862)

## Funciones agregadas

Las funciones agregadas devuelven una única fila de resultados basada en grupos de filas, en lugar de en filas individuales. Las funciones agregadas pueden aparecer en listas de selección y en cláusulas ORDER BY y HAVING. Se usan comúnmente con la cláusula GROUP BY en una declaración SELECT, donde la base de datos Oracle divide las filas de una tabla o vista consultada en grupos. En una consulta que contiene una cláusula GROUP BY, los elementos de la lista de selección pueden ser funciones agregadas, expresiones GROUP BY, constantes o expresiones que involucren una de estas. Oracle aplica las funciones agregadas a cada grupo de filas y devuelve una sola fila de resultados para cada grupo.

Si omite la cláusula GROUP BY, entonces Oracle aplica funciones agregadas en la lista de selección a todas las filas en la tabla o vista consultada. Utilice las funciones agregadas en la cláusula HAVING para eliminar grupos de la salida en función de los resultados de las funciones agregadas, en lugar de en los valores de las filas individuales de la tabla o vista consultada.

Muchas funciones agregadas (pero no todas) que toman un solo argumento aceptan estas cláusulas:

* DISTINCT y UNIQUE, que son sinónimos, hacen que una función agregada considere solo valores distintos de la expresión de argumento. Los diagramas de sintaxis para funciones agregadas en este capítulo usan la palabra clave DISTINCT por simplicidad.
* ALL hace que una función agregada considere todos los valores, incluidos todos los duplicados.

Por ejemplo, el promedio DISTINCT de 1, 1, 1 y 3 es 2. El promedio ALL es de 1.5. Si no especifica ninguno, entonces el valor predeterminado es ALL.

Algunas funciones agregadas permiten el *windowing\_clause*, que forma parte de la sintaxis de las funciones analíticas. Consulte *[windowing\_clause](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm" \l "i97640)* para obtener información sobre esta cláusula. En la lista de funciones agregadas al final de esta sección, las funciones que permiten que *windowing\_clause* sean seguidas por un asterisco (\*)

Todas las funciones agregadas, excepto COUNT(\*) GROUPING, e GROUPING\_ID ignoran los nulos. Puede utilizar la función NVL en el argumento de una función agregada para sustituir un valor por un valor nulo. COUNT y REGR\_COUNT nunca devuelve nulo, pero devuelve un número o cero. Para todas las funciones agregadas restantes, si el conjunto de datos no contiene filas, o solo contiene filas con nulos como argumentos a la función agregada, la función devuelve null.

Las funciones de agregado MIN, MAX, SUM, AVG, COUNT, VARIANCE, y STDDEV, cuando es seguida por la palabra clave KEEP, se pueden utilizar en conjunción con la función FIRST o LAST para operar en un conjunto de valores de un conjunto de filas que rango que el FIRST o  LAST con respecto a una especificación de clasificación dado .

Puedes anidar funciones agregadas. Por ejemplo, el siguiente ejemplo calcula el promedio de los salarios máximos de todos los departamentos en el esquema de muestra hr:

SELECT AVG(MAX(salary))

FROM employees

GROUP BY department\_id;

AVG(MAX(SALARY))

----------------

10926.3333

Este cálculo evalúa el agregado interno ( MAX( salary)) para cada grupo definido por la cláusula GROUP BY ( department\_id), y agrega los resultados nuevamente.

En la lista de funciones agregadas que sigue, las funciones seguidas de un asterisco (\*) permiten a *windowing\_clause*.

[AVG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions018.htm#i82074)  
[COLLECT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions031.htm#i1271564)  
[CORR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions035.htm#i82637)  
[CORR\_ \*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions036.htm#i1293691)  
[COUNT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions039.htm#i82697)  
[COVAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions040.htm#i1008854)  
[COVAR\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions041.htm#i82820)  
[CUME\_DIST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions043.htm#i82886)  
[DENSE\_RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions052.htm#i1064409)  
[PRIMER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions065.htm#i1000901)  
[group\_id](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions070.htm#i1002035)  
[AGRUPACIÓN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions071.htm#i77498)  
[GROUPING\_ID](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions072.htm#i1001964)  
[LAST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions083.htm#i1000905)  
[LISTAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions089.htm#CJABDFBD)  
[MAX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions098.htm#i89072)  
[MEDIANA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions099.htm#i1279886)  
[MIN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions100.htm#i1280029)  
[PERCENT\_RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions126.htm#i1043951)  
[PERCENTILE\_CONT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions127.htm#i1000909)  
[PERCENTILE\_DISC](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions128.htm#i1000913)  
[RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions141.htm#i1269223)  
[REGR\_ (regresión lineal) Funciones](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions151.htm#i85922)  
[STATS\_BINOMIAL\_TEST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions169.htm#i1279891)  
[STATS\_CROSSTAB](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions170.htm#i1333770)  
[STATS\_F\_TEST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions171.htm#i1279901)  
[STATS\_KS\_TEST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions172.htm#i1279906)  
[STATS\_MODE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions173.htm#i1279911)  
[STATS\_MW\_TEST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions174.htm#i1279921)  
[STATS\_ONE\_WAY\_ANOVA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions175.htm#i1335576)  
[STATS\_T\_TEST\_ \*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions176.htm#i1279931)  
[STATS\_WSR\_TEST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions177.htm#i1279951)  
[STDDEV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions178.htm#i89108)  
[STDDEV\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions179.htm#i86639)  
[STDDEV\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions180.htm#i86697)  
[SUM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions182.htm#i89126)  
[SYS\_XMLAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions189.htm#i1006716)  
[VAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions230.htm#i87119)  
[VAR\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions231.htm#i87169)  
[VARIACIÓN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions232.htm#i89144)  
[XMLAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions235.htm#i1286333)

## Funciones analiticas

Las funciones analíticas calculan un valor agregado basado en un grupo de filas. Se diferencian de las funciones agregadas en que devuelven varias filas para cada grupo. El grupo de filas se llama **ventana** y está definido por *analytic\_clause*. Para cada fila, se define una ventana deslizante de filas. La ventana determina el rango de filas utilizadas para realizar los cálculos de la fila actual. Los tamaños de las ventanas se pueden basar en un número físico de filas o en un intervalo lógico como el tiempo.

Las funciones analíticas son el último conjunto de operaciones realizadas en una consulta, excepto la cláusula final ORDER BY. Todo se une y todas las cláusulas WHERE, GROUP BY y HAVING se completan antes de que se procesan las funciones analíticas. Por lo tanto, las funciones analíticas solo pueden aparecer en la lista o cláusula de selección ORDER BY.

Las funciones analíticas se usan comúnmente para calcular agregados acumulativos, móviles, centrados y de informes.

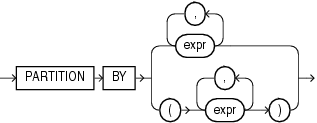
***analitic\_function* :: =**

La descripción de analytic_function.gif sigue.  
[Descripción de la ilustración '' analytic\_function.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/analytic_function.htm)

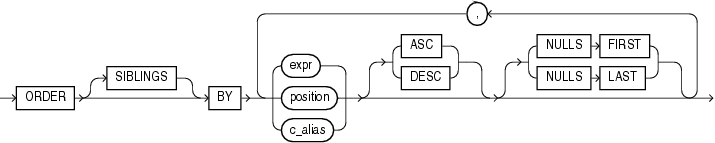
***analytic\_clause* :: =**

Descripción de analytic_clause.gif sigue  
[Descripción de la ilustración '' analytic\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/analytic_clause.htm)

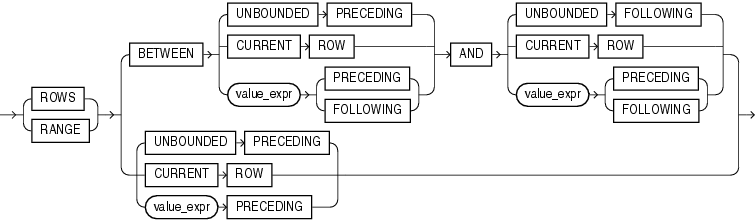
[***query\_partition\_clause***](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i1000371)**:: =**

  
[Descripción de la ilustración '' query\_partition\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/query_partition_clause.htm)

[***order\_by\_clause***](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i97469)**:: =**

  
[Descripción de la ilustración '' order\_by\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/order_by_clause.htm)

[***windowing\_clause***](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i97640)**:: =**

  
[Descripción de la ilustración '' windowing\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/windowing_clause.htm)

La semántica de esta sintaxis se discute en las secciones que siguen.

***funcion analítica***

Especifique el nombre de una función analítica (vea la lista de funciones analíticas después de esta discusión de la semántica).

***argumentos***

Las funciones analíticas toman de 0 a 3 argumentos. Los argumentos pueden ser cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que se pueden convertir implícitamente en un tipo de datos numéricos. Oracle determina el argumento con la precedencia numérica más alta y convierte implícitamente los argumentos restantes a ese tipo de datos. El tipo de retorno también es ese tipo de datos, a menos que se indique lo contrario para una función individual.

***analytic\_clause***

Se usa OVER *analytic\_clause* para indicar que la función opera en un conjunto de resultados de consulta. Esta cláusula se calcula después de las cláusulas FROM, WHERE, GROUP BY, y HAVING. Puede especificar funciones analíticas con esta cláusula en la lista de selección o cláusula ORDERBY. Para filtrar los resultados de una consulta basada en una función analítica, anide estas funciones dentro de la consulta principal y luego filtre los resultados de la subconsulta anidada.

**Notas sobre la *analytic\_clause* :**Las siguientes notas se aplican a *analytic\_clause*:

* No puede anidar funciones analíticas especificando ninguna función analítica en ninguna parte de *analytic\_clause*. Sin embargo, puede especificar una función analítica en una subconsulta y calcular otra función analítica sobre ella.
* Puede especificar OVER *analytic\_clause* con funciones analíticas definidas por el usuario, así como funciones analíticas integradas.

***query\_partition\_clause***

Utilice la PARTITION BY cláusula para dividir el conjunto de resultados de la consulta en grupos basados ​​en uno o más *value\_expr*. Si omite esta cláusula, la función trata todas las filas del conjunto de resultados de la consulta como un solo grupo.

Para usar la función analítica *query\_partition\_clause*, use la rama superior de la sintaxis (sin paréntesis). Para usar esta cláusula en una consulta modelo (en *model\_column\_clauses*) o en una combinación externa particionada (en *outer\_join\_clause*), use la rama inferior de la sintaxis (entre paréntesis).

Puede especificar múltiples funciones analíticas en la misma consulta, cada una con la misma clave  PARTITION BY o diferentes .

Si los objetos que se consultan tienen el atributo paralelo, y si especifica una función analítica con el *query\_partition\_clause*, los cálculos de la función también están paralelos.

Los valores válidos de *value\_expr* son constantes, columnas, funciones no analíticas, expresiones de función o expresiones que involucren cualquiera de estas.

***order\_by\_clause***

Use *order\_by\_clause* para especificar cómo se ordenan los datos dentro de una partición. Para todas las funciones analíticas, puede ordenar los valores en una partición en varias teclas, cada una definida por una *value\_expr* y cada una calificada por una secuencia de ordenación.

Dentro de cada función, puede especificar múltiples expresiones de orden. Hacerlo es especialmente útil cuando se usan funciones que clasifican valores, porque la segunda expresión puede resolver los vínculos entre valores idénticos para la primera expresión.

Siempre que los *order\_by\_clause* resultados en valores idénticos para varias filas, la función se comporta de la siguiente manera:

* CUME\_DIST, DENSE\_RANK, NTILE, PERCENT\_RANK, Y RANK devolver el mismo resultado para cada una de las filas.
* ROW\_NUMBER asigna a cada fila un valor distinto, incluso si hay un empate basado en el *order\_by\_clause*. El valor se basa en el orden en que se procesa la fila, que puede ser no determinista si ORDER BY no garantiza un orden total.
* Para todas las demás funciones analíticas, el resultado depende de la especificación de la ventana. Si especifica una ventana lógica con la palabra clave RANGE, la función devuelve el mismo resultado para cada una de las filas. Si especifica una ventana física con la palabra clave ROWS, el resultado no es determinista.

**Restricciones en la cláusula ORDER BY**Las siguientes restricciones se aplican a la cláusula ORDER BY:

* Cuando se utiliza en una función analítica, *order\_by\_clause* debe tomar una expresión ( *expr* ). La palabra clave SIBLINGS no es válida (solo es relevante en consultas jerárquicas). La posición ( *position* ) y los alias de columna (*c\_alias*) también son inválidos. De lo contrario, *order\_by\_clause* es el mismo que se usó para ordenar la consulta general o la subconsulta.
* Una función analítica que utiliza la palabra clave RANGE puede usar varias claves de clasificación en su cláusula ORDER BY si especifica alguna de las siguientes ventanas:
  + RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW. La forma corta de esto es RANGE UNBOUNDED PRECEDING.
  + RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
  + RANGE BETWEEN CURRENT ROW AND CURRENT ROW
  + RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING

Los límites de ventana distintos de estos cuatro pueden tener solo una clave de clasificación en la cláusula ORDER BY de la función analítica. Esta restricción no se aplica a los límites de ventana especificados por la palabra clave ROW.

**ASC | DESC**Especifique la secuencia de ordenación (ascendente o descendente). AS es el predeterminado

**NULLS FIRST | NULLS LAST**Especifique si las filas devueltas que contienen nulos deben aparecer primero o último en la secuencia de ordenación.

NULLS LAS es el valor predeterminado para orden ascendente, y NULLS FIRST es el valor predeterminado para orden descendente.

Las funciones analíticas siempre operan en filas en el orden especificado en la función *order\_by\_clause*. Sin embargo, el *order\_by\_clause* de la función no garantiza el orden del resultado. Utilice el botón *order\_by\_clause* de la consulta para garantizar el orden del resultado final.

***windowing\_clause***

Algunas funciones analíticas permiten la *windowing\_clause*. En la lista de funciones analíticas al final de esta sección, a las funciones que permiten *windowing\_clause* seguir un asterisco (\*).

**FILAS | RANGO**Estas palabras clave definen para cada fila una ventana (un conjunto de filas físico o lógico) que se utiliza para calcular el resultado de la función. La función se aplica a todas las filas de la ventana. La ventana se mueve a través del conjunto de resultados de la consulta o la partición de arriba a abajo.

* ROWS Especifica la ventana en unidades físicas (filas).
* RANGE Especifica la ventana como un desplazamiento lógico.

No puede especificar esta cláusula a menos que haya especificado el *order\_by\_clause*. Algunos límites de ventana definidos por la cláusula RANGE le permiten especificar solo una expresión en el *order\_by\_clause*.

El valor devuelto por una función analítica con un desplazamiento lógico es siempre determinista. Sin embargo, el valor devuelto por una función analítica con un desplazamiento físico puede producir resultados no deterministas a menos que la expresión de orden dé como resultado un ordenamiento único. Es posible que deba especificar varias columnas *order\_by\_clause* para lograr este orden único.

Use la cláusula BETWEEN... AND para especificar un punto de inicio y un punto final para la ventana. La primera expresión (antes AND) define el punto de inicio y la segunda expresión (después AND) define el punto final.

Si omite BETWEEN y especifica solo un punto final, Oracle lo considera el punto inicial y el punto final se establece de manera predeterminada en la fila actual.

Especifique UNBOUNDED PRECEDING para indicar que la ventana comienza en la primera fila de la partición. Esta es la especificación del punto de inicio y no se puede utilizar como una especificación de punto final.

**UNBOUNDED FOLLOWING**Especifique UNBOUNDED FOLLOWING para indicar que la ventana termina en la última fila de la partición. Esta es la especificación del punto final y no se puede utilizar como una especificación del punto de inicio.

**CURRENT ROW**Como punto de inicio, CURRENT ROW especifica que la ventana comienza en la fila o valor actual (dependiendo de si ha especificado ROW o RANGE, respectivamente). En este caso el punto final no puede ser *value\_expr* PRECEDING.

Como punto final, CURRENT ROW especifica que la ventana termina en la fila o valor actual (dependiendo de si ha especificado ROW o RANGE, respectivamente). En este caso el punto de inicio no puede ser *value\_expr* FOLLOWING.

***value\_expr* anterior o *value\_expr* SIGUIENTE**Para RANGE o ROW:

* Si *value\_expr* FOLLOWING es el punto de inicio, entonces el punto final debe ser *value\_expr* FOLLOWING.
* Si *value\_expr* PRECEDING es el punto final, entonces el punto inicial debe ser *value\_expr* PRECEDING.

Si está definiendo una ventana lógica definida por un intervalo de tiempo en formato numérico, es posible que deba usar funciones de conversión.

Si especificaste ROWS:

* *value\_expr* Es un desplazamiento físico. Debe ser una constante o expresión y debe evaluar un valor numérico positivo.
* Si *value\_expr* es parte del punto de inicio, debe evaluar una fila antes del punto final.

Si especificaste RANGE:

* *value\_expr* Es un desplazamiento lógico. Debe ser una constante o expresión que se evalúe como un valor numérico positivo o un literal de intervalo.
* Puede especificar solo una expresión en el *order\_by\_clause*.
* Si se *value\_expr* evalúa a un valor numérico, entonces ORDER BY *expr* debe ser un tipo numérico o de datos DATE.
* Si se *value\_expr* evalúa a un valor de intervalo, entonces ORDER BY *expr* debe ser un DATE tipo de datos.

Si omite el entero *windowing\_clause*, entonces el valor predeterminado es RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW.

Las funciones analíticas se utilizan comúnmente en entornos de almacenamiento de datos. En la lista de funciones analíticas que sigue, las funciones seguidas de un asterisco (\*) permiten la sintaxis completa, incluida la *windowing\_clause*.

[AVG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions018.htm#i82074) \*   
[CORR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions035.htm#i82637) \*   
[COUNT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions039.htm#i82697) \*   
[COVAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions040.htm#i1008854) \*   
[COVAR\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions041.htm#i82820) \*   
[CUME\_DIST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions043.htm#i82886)  
[DENSE\_RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions052.htm#i1064409)  
[PRIMER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions065.htm#i1000901)  
[FIRST\_VALUE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions066.htm#i83212) \*   
[LAG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions082.htm#i1327527)  
[LAST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions083.htm#i1000905)  
[LAST\_VALUE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions085.htm#i83648) \*   
[LEAD](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions086.htm#i83834)  
[LISTAGG](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions089.htm#CJABDFBD)  
[MAX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions098.htm#i89072) \*   
[MEDIANA](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions099.htm#i1279886)  
[MIN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions100.htm#i1280029) \*   
[NTH\_VALUE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions114.htm#CJAFEJBE) \*   
[NTILE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions115.htm#i85619)  
[PERCENT\_RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions126.htm#i1043951)  
[PERCENTILE\_CONT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions127.htm#i1000909)  
[PERCENTILE\_DISC](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions128.htm#i1000913)  
[RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions141.htm#i1269223)  
[RATIO\_TO\_REPORT](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions142.htm#i85800)  
[REGR\_ (regresión lineal) Funciones](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions151.htm#i85922) \*   
[ROW\_NUMBER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions156.htm#i86310)  
[STDDEV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions178.htm#i89108) \*   
[STDDEV\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions179.htm#i86639) \*   
[STDDEV\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions180.htm#i86697) \*   
[SUM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions182.htm#i89126)\*   
[VAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions230.htm#i87119) \*   
[VAR\_SAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions231.htm#i87169) \*   
[VARIACIÓN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions232.htm#i89144) \*

## Funciones de referencia de objetos

Las funciones de referencia de objetos manipulan valores REF, que son referencias a objetos de tipos de objetos específicos. Las funciones de referencia de objeto son:

[DEREF](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions054.htm#i1152051)  
[MAKE\_REF](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions097.htm#i77900)  
[REF](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions145.htm#i1009255)  
[REFTOHEX](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions146.htm#i1030787)  
[VALOR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions229.htm#i79988)

## Funciones del modelo

Las funciones del modelo solo se pueden utilizar en *model\_clause* la declaración SELECT. Las funciones del modelo son:

[CV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions046.htm#i1297925)  
[ITERATION\_NUMBER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions081.htm#i1327518)  
[PRESENTNNV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions138.htm#i1298026)  
[PRESENTV](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions139.htm#i1297947)  
[ANTERIOR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions140.htm#i1306001)

## Funciones OLAP

Las funciones OLAP devuelven datos de un objeto dimensional en formato relacional de dos dimensiones. La función OLAP es: ¿

## Funciones del cartucho de datos

Las funciones de Data Cartridge son útiles para los desarrolladores de Data Cartridge. La función del cartucho de datos es: ¿

# APENDICE DE FUNCIONES PREDEFINIDAS

## ABS

**Sintaxis**

Descripción de abs.gif sigue  
[Descripción de la ilustración '' abs.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/abs.htm)

**Propósito**

ABS devuelve el valor absoluto de *n*.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. La función devuelve el mismo tipo de datos que el tipo de datos numéricos del argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el valor absoluto de -15:

SELECT ABS(-15) "Absolute"

FROM DUAL;

Absolute

----------

15

## ACOS

**Sintaxis**

Descripción de acos.gif sigue  
[Descripción de la ilustración '' acos.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/acos.htm)

**Propósito**

ACOS Devuelve el arco coseno de *n*. El argumento *n*debe estar en el rango de -1 a 1, y la función devuelve un valor en el rango de 0 a *pi* , expresado en radianes.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el coseno del arco de .3:

SELECT ACOS(.3)"Arc\_Cosine"

FROM DUAL;

Arc\_Cosine

----------

1.26610367

## ADD\_MONTHS

**Sintaxis**

La descripción de add_months.gif sigue  
[Descripción de la ilustración '' add\_months.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/add_months.htm)

**Propósito**

ADD\_MONTHS Devuelve la fecha *date* más *integer* meses. Un mes está definido por el parámetro de sesión NLS\_CALENDAR. El argumento de fecha puede ser un valor de fecha y hora o cualquier valor al que se pueda convertir implícitamente DATE. El argumento *integer* puede ser un entero o cualquier valor que se pueda convertir implícitamente en un entero. El tipo de retorno es siempre DATE, independientemente del tipo de datos de *date*. Si *date* es el último día del mes o si el mes resultante tiene menos días que el componente del día *date*, entonces el resultado es el último día del mes resultante. De lo contrario, el resultado tiene el mismo componente día que *date*.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el mes siguiente al *hire\_date* de la tabla de muestra employees:

SELECT TO\_CHAR(ADD\_MONTHS(hire\_date, 1), 'DD-MON-YYYY') "Next month"

FROM employees

WHERE last\_name = 'Baer';

Next Month

-----------

07-JUL-2002

## APPENDCHILDXML

**Sintaxis**

Descripción de appendchildxml.gif sigue  
[Descripción de la ilustración '' appendchildxml.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/appendchildxml.htm)

**Propósito**

APPENDCHILDXML agrega un valor proporcionado por el usuario en el XML de destino como el elemento secundario del nodo indicado por una expresión XPath.

* *XMLType\_instance* Es una instancia de XMLType.
* *XPath\_string* es una expresión Xpath que indica uno o más nodos a los que se deben agregar uno o más nodos secundarios. Puede especificar un absoluto *XPath\_string* con una barra inicial o un pariente *XPath\_string* al omitir la barra inicial. Si omite la barra inicial, el contexto de la ruta relativa se establece de forma predeterminada en el nodo raíz.
* *value\_expr* Especifica uno o más nodos de XMLType. Se debe resolver a una cadena.
* El opcional *namespace\_string* proporciona información de espacio de nombres para el *XPath\_string*. Este parámetro debe ser de tipo VARCHAR2.

**Ver también:**

[*Guía del desarrollador de Oracle XML DB*](https://docs.oracle.com/cd/E11882_01/appdev.112/e23094/xdb04cre.htm#ADXDB0440) para obtener más información sobre esta función

**Ejemplos**

El siguiente ejemplo agrega un nodo /Owner al /Warehouse/Building nodo warehouse\_specen la tabla oe.warehouses si el valor del nodo /Building es "Alquilado":

UPDATE warehouses

SET warehouse\_spec = APPENDCHILDXML(warehouse\_spec, 'Warehouse/Building',

XMLType('<Owner>Grandco</Owner>'))

WHERE EXTRACTVALUE(warehouse\_spec, '/Warehouse/Building') = 'Rented';

SELECT warehouse\_id,

warehouse\_name,

EXTRACTVALUE(warehouse\_spec, '/Warehouse/Building/Owner') "Prop.Owner"

FROM warehouses

WHERE EXISTSNODE(warehouse\_spec, '/Warehouse/Building/Owner') = 1;

WAREHOUSE\_ID WAREHOUSE\_NAME Prop.Owner

------------ --------------- ----------

2 San Francisco Grandco

3 New Jersey Grandco

## ASCII

**Sintaxis**

La descripción de ascii.gif sigue.  
[Descripción de la ilustración '' ascii.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/ascii.htm)

**Propósito**

ASCII devuelve la representación decimal en el conjunto de caracteres de la base de datos del primer carácter de *char*.

*Char* puede ser de tipo de datos CHAR, VARCHAR2, NCHAR, o NVARCHAR2. El valor devuelto es de tipo de datos NUMBER. Si el conjunto de caracteres de su base de datos es ASCII de 7 bits, esta función devuelve un valor ASCII. Si el conjunto de caracteres de su base de datos es un código EBCDIC, esta función devuelve un valor EBCDIC. No hay una función de carácter EBCDIC correspondiente.

Esta función no soporta datos CLOB directamente. Sin embargo, CLOBs se puede pasar como argumentos a través de la conversión de datos implícita.

**Ver también:**

["Reglas de comparación de tipos de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i55214) para más información

**Ejemplos**

El siguiente ejemplo devuelve a los empleados cuyos apellidos comienzan con la letra L, cuyo equivalente ASCII es 76:

SELECT last\_name

FROM employees

WHERE ASCII(SUBSTR(last\_name, 1, 1)) = 76

ORDER BY last\_name;

LAST\_NAME

-------------------------

Ladwig

Landry

Lee

Livingston

Lorentz

## ASCIISTA

**Sintaxis**

A continuación se encuentra la descripción de asciistr.gif.  
[Descripción de la ilustración '' asciistr.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/asciistr.htm)

**Propósito**

ASCIISTR toma como argumento una cadena, o una expresión que se resuelve en una cadena, en cualquier conjunto de caracteres y devuelve una versión ASCII de la cadena en el conjunto de caracteres de la base de datos. Los caracteres no ASCII se convierten al formulario \xxxx, donde xxxx representa una unidad de código UTF-16.

**Ver también:**

[*Guía de soporte de globalización de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e10729/ch6unicode.htm#NLSPG006) para obtener información sobre conjuntos de caracteres Unicode y semántica de caracteres

**Ejemplos**

El siguiente ejemplo devuelve la cadena ASCII equivalente a la cadena de texto " ABÄCDE":

SELECT ASCIISTR('ABÄCDE')

FROM DUAL;

ASCIISTR('

----------

AB\00C4CDE

## ASIN

**Sintaxis**

Descripción de asin.gif sigue  
[Descripción de la ilustración '' asin.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/asin.htm)

**Propósito**

ASIN Devuelve el arco senoidal de *n*. El argumento *n*debe estar en el rango de -1 a 1, y la función devuelve un valor en el rango de - *pi* / 2 a *pi* / 2, expresado en radianes.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el arco seno de .3:

SELECT ASIN(.3) "Arc\_Sine"

FROM DUAL;

Arc\_Sine

----------

.304692654

## ATAN

**Sintaxis**

Descripción de atan.gif sigue  
[Descripción de la ilustración '' atan.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/atan.htm)

**Propósito**

ATAN Devuelve el arco tangente de *n*. El argumento *n* puede estar en un rango ilimitado y devuelve un valor en el rango de - *pi* / 2 a *pi* / 2, expresado en radianes.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[ATAN2](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions017.htm#i76816) para obtener información sobre la ATAN2función y la [Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el arco tangente de .3:

SELECT ATAN(.3) "Arc\_Tangent"

FROM DUAL;

Arc\_Tangent

----------

.291456794

## ATAN2

**Sintaxis**

Descripción de atan2.gif sigue  
[Descripción de la ilustración '' atan2.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/atan2.htm)

**Propósito**

ATAN2 Devuelve el arco tangente de *n1* y *n2*. El argumento *n1* puede estar en un rango ilimitado y devuelve un valor en el rango de - *pi* a *pi* , dependiendo de los signos de *n1* y *n2*, expresados ​​en radianes.

Esta función toma como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si algún argumento es BINARY\_FLOAT o BINARY\_DOUBLE, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función vuelve NUMBER.

**Ver también:**

[ATAN](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions016.htm#i76792) para obtener información sobre la ATANfunción y la [Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el arco tangente de .3 y .2:

SELECT ATAN2(.3, .2) "Arc\_Tangent2"

FROM DUAL;

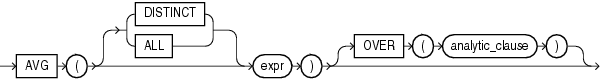
Arc\_Tangent2

------------

.982793723

## AVG

**Sintaxis**

  
[Descripción de la ilustración '' avg.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/avg.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

AVG devuelve el valor promedio de *expr*.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. La función devuelve el mismo tipo de datos que el tipo de datos numéricos del argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

Si lo especifica DISTINCT, puede especificar solo el *query\_partition\_clause* de *analytic\_clause*. Los *order\_by\_clause*y *windowing\_clause* no están permitidos.

**Ver también:**

["Acerca de las expresiones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions001.htm#i1002626) para obtener información sobre formas válidas de *expr*y ["Funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)

**Ejemplo Agregado**

El siguiente ejemplo calcula el salario promedio de todos los empleados en la tabla hr.employees:

SELECT AVG(salary) "Average"

FROM employees;

Average

--------------

6461.83178

**Ejemplo analítico**

El siguiente ejemplo calcula, para cada empleado en la tabla employees, el salario promedio de los empleados que reportan al mismo gerente que fueron contratados en el rango justo antes y justo después del empleado:

SELECT manager\_id, last\_name, hire\_date, salary,

AVG(salary) OVER (PARTITION BY manager\_id ORDER BY hire\_date

ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS c\_mavg

FROM employees

ORDER BY manager\_id, hire\_date, salary;

MANAGER\_ID LAST\_NAME HIRE\_DATE SALARY C\_MAVG

---------- ------------------------- --------- ---------- ----------

100 De Haan 13-JAN-01 17000 14000

100 Raphaely 07-DEC-02 11000 11966.6667

100 Kaufling 01-MAY-03 7900 10633.3333

100 Hartstein 17-FEB-04 13000 9633.33333

100 Weiss 18-JUL-04 8000 11666.6667

100 Russell 01-OCT-04 14000 11833.3333

100 Partners 05-JAN-05 13500 13166.6667

100 Errazuriz 10-MAR-05 12000 11233.3333

. . .

## BFILENAME

**Sintaxis**

Descripción de bfilename.gif sigue  
[Descripción de la ilustración '' bfilename.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/bfilename.htm)

**Propósito**

BFILENAME devuelve un localizador BFILE asociado con un archivo binario LOB físico en el sistema de archivos del servidor.

* ' *directory*' es un objeto de base de datos que sirve como alias para un nombre de ruta completo en el sistema de archivos del servidor donde se encuentran los archivos.
* ' *filename*' es el nombre del archivo en el sistema de archivos del servidor.

Debe crear el objeto de directorio y asociar un valor BFILE con un archivo físico antes de poder usarlos como argumentos BFILENAME en una instrucción SQL, PL / SQL, paquete DBMS\_LOB u operación OCI.

Puedes usar esta función de dos maneras:

* En una declaración DML para inicializar una columna BFILE
* En una interfaz programática para acceder a los datos BFILE asignando un valor al localizador BFILE

El argumento del directorio distingue entre mayúsculas y minúsculas. Debe asegurarse de que especifica el nombre del objeto de directorio exactamente como existe en el diccionario de datos. Por ejemplo, si un objeto de directorio "Admin" se creó utilizando un caso mixto y un identificador entre comillas en la declaración CREATE DIRECTORY, al usar la función BFILENAME debe referirse al objeto de directorio como 'Admin'. Debe especificar el argumento de nombre de archivo según el caso y las convenciones de puntuación para su sistema operativo.

**Ver también:**

* [*Guía de Oracle SecureFiles bases de datos y objetos de gran tamaño del desarrollador*](https://docs.oracle.com/cd/E11882_01/appdev.112/e18294/adlob_bfile_ops.htm#ADLOB012) y la[*Guía de Oracle Call Interface programador*](https://docs.oracle.com/cd/E11882_01/appdev.112/e10646/oci03typ.htm#LNOCI030) para obtener más información sobre los LOB y para ejemplos de recuperación de datos BFILE
* [CREAR DIRECTORIO](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_5007.htm#i2061958)

**Ejemplos**

El siguiente ejemplo inserta una fila en la tabla de muestra pm.print\_media. El ejemplo utiliza la función BFILENAME para identificar un archivo binario en el sistema de archivos del servidor en el directorio /demo/schema/product\_media. El ejemplo muestra cómo media\_dir se creó el objeto de base de datos de directorio en el pmesquema.

CREATE DIRECTORY media\_dir AS '/demo/schema/product\_media';

INSERT INTO print\_media (product\_id, ad\_id, ad\_graphic)

VALUES (3000, 31001, BFILENAME('MEDIA\_DIR', 'modem\_comp\_ad.gif'));

## BIN\_TO\_NUM

**Sintaxis**

Descripción de bin_to_num.gif sigue  
[Descripción de la ilustración '' bin\_to\_num.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/bin_to_num.htm)

**Propósito**

BIN\_TO\_NUM convierte un vector de bit a su número equivalente. Cada argumento de esta función representa un bit en el vector de bits. Esta función toma como argumentos cualquier tipo de datos numéricos, o cualquier tipo de datos no numéricos a los que se pueda convertir implícitamente NUMBER. Cada uno *expr* debe evaluar a 0 o 1. Esta función devuelve Oracle NUMBER.

BIN\_TO\_NUM es útil en aplicaciones de almacenamiento de datos para seleccionar grupos de interés de una vista materializada utilizando conjuntos de agrupación.

**Ver también:**

* [*group\_by\_clause*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_10002.htm#i2182483) para obtener información sobre la sintaxis GROUPING SETS
* [Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita
* [*Oracle Database Data Warehousing Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e25554/aggreg.htm#DWHSG020) para obtener información sobre la agregación de datos en general

**Ejemplos**

El siguiente ejemplo convierte un valor binario en un número:

SELECT BIN\_TO\_NUM(1,0,1,0)

FROM DUAL;

BIN\_TO\_NUM(1,0,1,0)

-------------------

10

El siguiente ejemplo convierte tres valores en un solo valor binario y utiliza BIN\_TO\_NUM para convertir ese binario en un número. El ejemplo utiliza una declaración PL / SQL para especificar los valores originales. Estos normalmente se derivarían de fuentes de datos reales.

SELECT order\_status

FROM orders

WHERE order\_id = 2441;

ORDER\_STATUS

------------

5

DECLARE

warehouse NUMBER := 1;

ground NUMBER := 1;

insured NUMBER := 1;

result NUMBER;

BEGIN

SELECT BIN\_TO\_NUM(warehouse, ground, insured) INTO result FROM DUAL;

UPDATE orders SET order\_status = result WHERE order\_id = 2441;

END;

/

PL/SQL procedure successfully completed.

SELECT order\_status

FROM orders

WHERE order\_id = 2441;

ORDER\_STATUS

------------

7

Consulte los ejemplos de [BITAND](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions021.htm#i98000) para obtener información sobre [cómo](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions021.htm#i98000) revertir este proceso extrayendo múltiples valores de un solo valor de columna.

## BITAND

**Sintaxis**

[Descripción de la ilustración '' bitand.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/bitand.htm)

**Propósito**

La función BITAND trata sus entradas y su salida como vectores de bits; La salida es el bit a bit AND de las entradas.

Los tipos de *expr1* y *expr2* son NUMBER, y el resultado es de tipo NUMBER. Si cualquiera de los dos argumentos a BITAND es NULL, el resultado es NULL.

Los argumentos deben estar en el rango - (2 (n-1) ) .. ((2 (n-1) ) -1). Si un argumento está fuera de este rango, el resultado no está definido.

El resultado se calcula en varios pasos. Primero, cada argumento A se reemplaza con el valor SIGN(A)\*FLOOR(ABS(A)). Esta conversión tiene el efecto de truncar cada argumento hacia cero. A continuación, cada argumento A (que ahora debe ser un valor entero) se convierte en un valor entero binario de complemento de dos bits de n bits. Los dos valores de bit se combinan utilizando una operación  AND bitwise . Finalmente, el valor complementario del n-bit dos resultante se convierte de nuevo a NUMBER.

**Notas sobre la función BITAND**

* La implementación actual de BITAND define *n*= 128.
* PL / SQL soporta una sobrecarga BITAND para la cual todos los tipos de entradas y del resultado son todos BINARY\_INTEGER y para los cuales *n*= 32.

**Ejemplos**

El siguiente ejemplo realiza una operación AND en los números 6 (binario 1,1,0) y 3 (binario 0,1,1):

SELECT BITAND(6,3)

FROM DUAL;

BITAND(6,3)

-----------

2

Este es el mismo que el siguiente ejemplo, que muestra los valores binarios de 6 y 3. La función BITAND opera solo en los dígitos significativos de los valores binarios:

SELECT BITAND(

BIN\_TO\_NUM(1,1,0),

BIN\_TO\_NUM(0,1,1)) "Binary"

FROM DUAL;

Binary

----------

2

Consulte el ejemplo de [BIN\_TO\_NUM](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions020.htm#i1002025) para obtener información sobre la codificación de varios valores en un solo valor de columna.

El siguiente ejemplo supone que la columna order\_status de la tabla de ejemplo oe.orders codifica varias opciones como bits individuales dentro de un único valor numérico. Por ejemplo, un pedido aún en el almacén está representado por un valor binario 001 (decimal 1). Una orden que se envía por transporte terrestre está representada por un valor binario 010 (decimal 2). Un paquete asegurado está representado por un valor binario 100 (decimal 4). El ejemplo utiliza la función DECODE para proporcionar dos valores para cada uno de los tres bits en el valor de order\_status, un valor si el bit está activado y otro si está desactivado.

SELECT order\_id, customer\_id, order\_status,

DECODE(BITAND(order\_status, 1), 1, 'Warehouse', 'PostOffice') "Location",

DECODE(BITAND(order\_status, 2), 2, 'Ground', 'Air') "Method",

DECODE(BITAND(order\_status, 4), 4, 'Insured', 'Certified') "Receipt"

FROM orders

WHERE sales\_rep\_id = 160

ORDER BY order\_id;

ORDER\_ID CUSTOMER\_ID ORDER\_STATUS Location Method Receipt

---------- ----------- ------------ ---------- ------ ---------

2416 104 6 PostOffice Ground Insured

2419 107 3 Warehouse Ground Certified

2420 108 2 PostOffice Ground Certified

2423 145 3 Warehouse Ground Certified

2441 106 5 Warehouse Air Insured

2455 145 7 Warehouse Ground Insured

Para la columna Location, BITAND primero se compara *order\_status* con 1 (binario 001). Solo se comparan los valores de bit significativos, por lo que cualquier valor binario con un 1 en su bit más a la derecha (cualquier número impar) se evaluará positivamente y devolverá 1. Los números pares devolverán 0. La función DECODE compara el valor devuelto por BITAND 1. Si ambos son 1, entonces la ubicación es "Almacén". Si son diferentes, entonces la ubicación es "Oficina de correos".

Las columnas Method y Receipt se calculan de manera similar. Para Method, BITAND realiza la operación AND en *order\_status* y 2 (010 binario). Para Receipt, BITAND realiza la operación AND en *order\_status* y 4 (binario 100).

## CARDINALIDAD

**Sintaxis**

Descripción de cardinality.gif sigue  
[Descripción de la ilustración '' cardinality.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cardinality.htm)

**Propósito**

CARDINALITY devuelve el número de elementos en una tabla anidada. El tipo de retorno es NUMBER. Si la tabla anidada está vacía, o es una colección nula, CARDINALITY devuelve NULL.

**Ejemplos**

El siguiente ejemplo muestra el número de elementos en la columna ad\_textdocs\_ntab de tabla anidada de la tabla de ejemplo pm.print\_media:

SELECT product\_id, CARDINALITY(ad\_textdocs\_ntab) cardinality

FROM print\_media

ORDER BY product\_id;

PRODUCT\_ID CARDINALITY

---------- -----------

2056 3

2268 3

3060 3

3106 3

## CAST

**Sintaxis**

Descripción de cast.gif sigue  
[Descripción de la ilustración '' cast.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cast.htm)

**Propósito**

CAST convierte un tipo de datos incorporado o un valor de tipo de colección en otro tipo de datos incorporado o un valor de tipo de colección.

CAST le permite convertir tipos de datos incorporados o valores de tipo de colección de un tipo en otro tipo de datos incorporado o tipo de colección. Puede convertir un operando sin nombre (como una fecha o el conjunto de resultados de una subconsulta) o una colección con nombre (como una varray o una tabla anidada) en un tipo de datos compatible con el tipo o colección con nombre. El *type\_name* debe ser el nombre de un tipo de datos o colección de tipo incorporado y el operando debe ser un tipo de datos incorporado o debe evaluarse como un valor de la colección.

Para el operando, *expr* puede ser un tipo de datos incorporado, un tipo de colección o una instancia de un tipo ANYDATA. Si *expr* es una instancia de un tipo ANYDATA, luego CAST intenta extraer el valor de la instancia ANYDATA y devolverlo si coincide con el tipo de destino de conversión, de lo contrario, se devolverá un valor nulo. MULTISET informa a la base de datos Oracle para tomar el conjunto de resultados de la subconsulta y devolver un valor de colección. [La Tabla 5-1](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions023.htm#g1514003) muestra qué tipos de datos incorporados se pueden convertir en qué otros tipos de datos incorporados. ( CAST no admite LONG, LONG RAW o los tipos proporcionados por Oracle.)

CAST no admite directamente ninguno de los tipos de datos LOB. Cuando se utiliza CAST para convertir un valor CLOB en un tipo de datos de carácter o un valor BLOB en el tipo de datos RAW, la base de datos convierte implícitamente el valor LOB en datos de caracteres o en bruto y luego convierte explícitamente el valor resultante en el tipo de datos de destino. Si el valor resultante es mayor que el tipo de destino, la base de datos devuelve un error.

Cuando usa CAST... MULTISET para obtener un valor de colección, cada elemento de la lista de selección en la consulta que se pasa a la función CAST se convierte al tipo de atributo correspondiente del tipo de elemento de la colección de destino.

***Tabla 5-1 Tipos de datos incorporados de conversión***

|  | **de BINARY\_FLOAT, BINARY\_DOUBLE** | **de CHAR, VARCHAR2** | **deNUMBER** | **de DATETIME / INTERVAL (Nota 1)** | **deRAW** | **de ROWID, UROWID (Nota 2)** | **de NCHAR, NVARCHAR2** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **to BINARY\_FLOAT, BINARY\_DOUBLE** | X | X | X | -- | -- | -- | X |
| **to CHAR, VARCHAR2** | X | X | X | X | X | X | -- |
| **to NUMBER** | X | X | X | -- | -- | -- | X |
| **to DATE, TIMESTAMP, INTERVAL** | -- | X | -- | X | -- | -- | -- |
| **to RAW** | -- | X | -- | -- | X | -- | -- |
| **to ROWID, UROWID** | -- | X | -- | -- | -- | X | -- |
| **to NCHAR, NVARCHAR2** | X | -- | X | X | X | X | X |

**Nota 1:** Fecha y hora / intervalo incluye DATE, TIMESTAMP, TIMESTAMP WITH TIMEZONE, INTERVAL DAY TO SECOND, y INTERVAL YEAR TO MONTH.

**Nota 2:** No puede convertir a UROWID a ROWID si UROWID contiene el valor de a ROWID de una tabla organizada por índice.

Si desea convertir un tipo de colección con nombre en otro tipo de colección con nombre, entonces los elementos de ambas colecciones deben ser del mismo tipo.

**Ver también:**

["Conversión de datos implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i163326) para obtener información sobre cómo Oracle Database convierte implícitamente los datos de tipo de recopilación en datos de caracteres y ["Consideraciones de seguridad para la conversión de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#CIHJCCEB)

Si el conjunto de resultados *subquery* puede evaluar varias filas, debe especificar la palabra clave MULTISET. Las filas resultantes de la subconsulta forman los elementos del valor de colección en el que se convierten. Sin la palabra clave MULTISET, la subconsulta se trata como una subconsulta escalar.

**Ejemplos de tipos de datos incorporados**

Los siguientes ejemplos utilizan la función CAST con tipos de datos escalares. El primer ejemplo convierte el texto en un valor de marca de tiempo aplicando el modelo de formato proporcionado en el parámetro de sesión NLS\_TIMESTAMP\_FORMAT. Si desea evitar la dependencia de este parámetro NLS, puede usar el TO\_DATE como se muestra en el segundo ejemplo.

SELECT CAST('22-OCT-1997'

AS TIMESTAMP WITH LOCAL TIME ZONE)

FROM DUAL;

SELECT CAST(TO\_DATE('22-Oct-1997', 'DD-Mon-YYYY')

AS TIMESTAMP WITH LOCAL TIME ZONE)

FROM DUAL;

En el ejemplo anterior, TO\_DATE convierte de texto a DATE, y CAST convierte de DATE a TIMESTAMP WITH LOCAL TIME ZONE, interpretando la fecha en la zona horaria de la sesión ( SESSIONTIMEZONE).

SELECT product\_id, CAST(ad\_sourcetext AS VARCHAR2(30)) text

FROM print\_media

ORDER BY product\_id;

**Ejemplos de colección**

Los ejemplos CAST que siguen a construir en la cust\_address\_typ que se encuentra en el esquema de la entrada de pedidos, muestra oe.

CREATE TYPE address\_book\_t AS TABLE OF cust\_address\_typ;

/

CREATE TYPE address\_array\_t AS VARRAY(3) OF cust\_address\_typ;

/

CREATE TABLE cust\_address (

custno NUMBER,

street\_address VARCHAR2(40),

postal\_code VARCHAR2(10),

city VARCHAR2(30),

state\_province VARCHAR2(10),

country\_id CHAR(2));

CREATE TABLE cust\_short (custno NUMBER, name VARCHAR2(31));

CREATE TABLE states (state\_id NUMBER, addresses address\_array\_t);

Este ejemplo arroja una subconsulta:

SELECT s.custno, s.name,

CAST(MULTISET(SELECT ca.street\_address,

ca.postal\_code,

ca.city,

ca.state\_province,

ca.country\_id

FROM cust\_address ca

WHERE s.custno = ca.custno)

AS address\_book\_t)

FROM cust\_short s

ORDER BY s.custno;

CAST convierte una columna de tipo varray en una tabla anidada:

SELECT CAST(s.addresses AS address\_book\_t)

FROM states s

WHERE s.state\_id = 111;

Los siguientes objetos crean la base del ejemplo que sigue:

CREATE TABLE projects

(employee\_id NUMBER, project\_name VARCHAR2(10));

CREATE TABLE emps\_short

(employee\_id NUMBER, last\_name VARCHAR2(10));

CREATE TYPE project\_table\_typ AS TABLE OF VARCHAR2(10);

/

El siguiente ejemplo de una expresión MULTISET usa estos objetos:

SELECT e.last\_name,

CAST(MULTISET(SELECT p.project\_name

FROM projects p

WHERE p.employee\_id = e.employee\_id

ORDER BY p.project\_name)

AS project\_table\_typ)

FROM emps\_short e

ORDER BY e.last\_name;

## CEIL

**Sintaxis**

Descripción de ceil.gif sigue  
[Descripción de la ilustración '' ceil.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/ceil.htm)

**Propósito**

CEIL devuelve el entero más pequeño que es mayor o igual que *n*. El número *n* siempre se puede escribir como la diferencia de un número entero *k* y una fracción positiva de *f* manera que 0 <= *f*<1 y *n*= *k*- *f*. El valor de CEIL es el entero *k*. Por lo tanto, el valor de sí CEIL es *n* si y solo si *n*es precisamente un número entero.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. La función devuelve el mismo tipo de datos que el tipo de datos numéricos del argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y [FLOOR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions067.htm#i77449)

**Ejemplos**

El siguiente ejemplo devuelve el entero más pequeño mayor o igual que el total del pedido de un orden especificado:

SELECT order\_total, CEIL(order\_total)

FROM orders

WHERE order\_id = 2434;

ORDER\_TOTAL CEIL(ORDER\_TOTAL)

----------- -----------------

268651.8 268652

## Chartorowid

**Sintaxis**

La descripción de chartorowid.gif sigue  
[Descripción de la ilustración '' chartorowid.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/chartorowid.htm)

**Propósito**

CHARTOROWID convierte un valor de CHAR, VARCHAR2, NCHAR, o NVARCHAR2 tipo de datos  ROWID.

Esta función no soporta datos CLOB directamente. Sin embargo, CLOBs se puede pasar como argumentos a través de la conversión de datos implícita.

**Ver también:**

["Reglas de comparación de tipos de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i55214) para más información.

**Ejemplos**

El siguiente ejemplo convierte una representación de rowid de carácter en un rowid. (El rowid real es diferente para cada instancia de base de datos).

SELECT last\_name

FROM employees

WHERE ROWID = CHARTOROWID('AAAFd1AAFAAAABSAA/');

LAST\_NAME

-------------------------

Greene

## CHR

**Sintaxis**

Descripción de chr.gif sigue  
[Descripción de la ilustración '' chr.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/chr.htm)

**Propósito**

CHR devuelve el carácter que tiene el equivalente binario *n* como un valor VARCHAR2 en el conjunto de caracteres de la base de datos o, si lo especifica USING NCHAR\_CS, el conjunto de caracteres nacional.

Para los conjuntos de caracteres de un solo byte, si n> 256, la base de datos Oracle devuelve el equivalente binario de n mod 256. Para conjuntos de caracteres multibyte, n debe resolverse en un punto de código completo. Los puntos de código no válidos no se validan, y el resultado de especificar puntos de código no válidos es indeterminado.

Esta función toma como argumento un valor NUMBER, o cualquier valor al que se pueda convertir implícitamente NUMBER, y devuelve un carácter.

**Nota:**

El uso de la función CHR (con o sin la cláusula opcional  USING NCHAR\_CS) da como resultado un código que no es portátil entre las arquitecturas de máquinas basadas en ASCII y EBCDIC.

**Ver también:**

[NCHR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions104.htm#i1273067) y la [Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo se ejecuta en una máquina basada en ASCII con el conjunto de caracteres de la base de datos definido como WE8ISO8859P1:

SELECT CHR(67)||CHR(65)||CHR(84) "Dog"

FROM DUAL;

Dog

---

CAT

Para producir los mismos resultados en una máquina basada en EBCDIC con el conjunto de caracteres WE8EBCDIC1047, el ejemplo anterior debería modificarse de la siguiente manera:

SELECT CHR(195)||CHR(193)||CHR(227) "Dog"

FROM DUAL;

Dog

---

CAT

Para conjuntos de caracteres multibyte, este tipo de concatenación da diferentes resultados. Por ejemplo, dado un carácter multibyte cuyo valor hexadecimal es a1a2(que a1r epresenta el primer byte y a2 el segundo byte), debe especificar nel equivalente decimal de ' a1a2', o 41378:

SELECT CHR(41378)

FROM DUAL;

No puede especificar el equivalente decimal de a1 concatenado con el equivalente decimal de a2, como en el siguiente ejemplo:

SELECT CHR(161)||CHR(162)

FROM DUAL;

Sin embargo, puede concatenar puntos de código multibyte completos, como en el siguiente ejemplo, que concatena los caracteres multibyte cuyos valores hexadecimales son a1a2 y a1a3:

SELECT CHR(41378)||CHR(41379)

FROM DUAL;

El siguiente ejemplo asume que el conjunto de caracteres nacional es UTF16:

SELECT CHR (196 USING NCHAR\_CS)

FROM DUAL;

CH

--

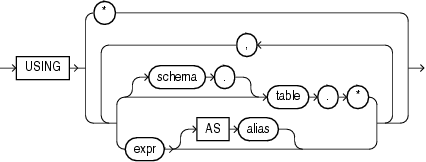
Ä

## CLUSTER\_ID

**Sintaxis**

La descripción de cluster_id.gif sigue.  
[Descripción de la ilustración '' cluster\_id.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cluster_id.htm)

***mining\_attribute\_clause* :: =**

  
[Descripción de la ilustración '' mining\_attribute\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/mining_attribute_clause.htm)

**Propósito**

Esta función es para uso con modelos de clustering creados por el paquete DBMS\_DATA\_MINING o con Oracle Data Miner. Devuelve el identificador de clúster del clúster predicho con la probabilidad más alta para el conjunto de predictores especificados en *mining\_attribute\_clause*. El valor devuelto es un Oracle NUMBER.

*mining\_attribute\_clause* se omporta como se describe para la función PREDICTION. Consulte *[mining\_attribute\_clause](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions132.htm" \l "CJAJACJD)* .

**Ver también:**

* [*Conceptos de minería de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16808/toc.htm) para obtener información detallada sobre la minería de datos de Oracle
* [*Guía del desarrollador de la aplicación Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e12218/models_deploying.htm#DMPRG004) para obtener información detallada sobre la puntuación con las funciones SQL de Data Mining

**Ejemplos**

El siguiente ejemplo enumera los grupos en los que se han agrupado los clientes de un conjunto de datos determinado.

Este ejemplo, y los requisitos previos de las operaciones de minería de datos, incluida la creación del modelo km\_sh\_clus\_sample y la vista mining\_data\_apply\_v, se pueden encontrar en el archivo de demostración $ORACLE\_HOME/rdbms/demo/dmkmdemo.sql. La información general sobre los archivos de demostración de minería de datos está disponible en [*la Guía del administrador de Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16807/sampleprogs.htm#DMADM009) . El ejemplo se presenta aquí para ilustrar el uso sintáctico de la función.

SELECT CLUSTER\_ID(km\_sh\_clus\_sample USING \*) AS clus, COUNT(\*) AS cnt

FROM mining\_data\_apply\_v

GROUP BY CLUSTER\_ID(km\_sh\_clus\_sample USING \*)

ORDER BY cnt DESC;

CLUS CNT

---------- ----------

2 580

10 216

6 186

8 115

19 110

12 101

18 81

16 39

17 38

14 34

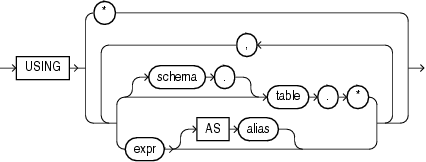
10 rows selected.

## CLUSTER\_PROBABILITY

**Sintaxis**

La descripción de cluster_probability.gif sigue  
[Descripción de la ilustración '' cluster\_probability.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cluster_probability.htm)

***mining\_attribute\_clause* :: =**

  
[Descripción de la ilustración '' mining\_attribute\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/mining_attribute_clause.htm)

**Propósito**

Esta función es para uso con modelos de clustering creados por el paquete DBMS\_DATA\_MINING o con Oracle Data Miner. Devuelve una medida del grado de confianza de la pertenencia a una fila de entrada en un clúster asociado con el modelo especificado.

* Para *cluster\_id*, especifique el identificador del cluster en el modelo. La función devuelve la probabilidad para el cluster especificado. Si omite esta cláusula, la función devuelve la probabilidad asociada con el mejor clúster predicho. Puede usar el formulario sin *cluster\_id* en combinación con la función CLUSTER\_ID para obtener el mejor par predicho de ID de clúster y probabilidad.
* *mining\_attribute\_clause* se omporta como se describe para la función PREDICTION. Consulte *[mining\_attribute\_clause](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions132.htm" \l "CJAJACJD)*

**Ver también:**

* [*Conceptos de minería de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16808/toc.htm) para obtener información detallada sobre la minería de datos de Oracle
* [*Guía del desarrollador de la aplicación Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e12218/models_deploying.htm#DMPRG004) para obtener información detallada sobre la puntuación con las funciones SQL de Data Mining

**Ejemplos**

El siguiente ejemplo determina los diez clientes más representativos, según la probabilidad, en el cluster 2.

Este ejemplo, y los requisitos previos de las operaciones de minería de datos, incluida la creación del modelo km\_sh\_clus\_sample y la vista mining\_data\_apply\_v, se pueden encontrar en el archivo de demostración $ORACLE\_HOME/rdbms/demo/dmkmdemo.sql. La información general sobre los archivos de demostración de minería de datos está disponible en [*la Guía del administrador de Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16807/sampleprogs.htm#DMADM009) . El ejemplo se presenta aquí para ilustrar el uso sintáctico de la función.

SELECT \*

FROM (SELECT cust\_id, CLUSTER\_PROBABILITY(km\_sh\_clus\_sample, 2 USING \*) prob

FROM mining\_data\_apply\_v

ORDER BY prob DESC)

WHERE ROWNUM < 11;

CUST\_ID PROB

---------- ----------

100256 .999387471

100988 .99936194

100889 .999335107

101086 .99928882

101215 .999266521

100390 .999264718

100985 .999251722

101026 .999247906

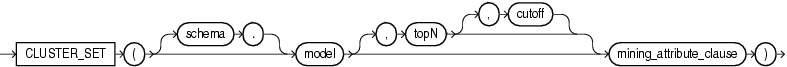
100601 .999242089

100672 .999235711

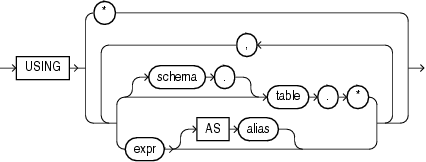
10 rows selected.

## CLUSTER\_SET

**Sintaxis**

  
[Descripción de la ilustración '' cluster\_set.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cluster_set.htm)

***mining\_attribute\_clause* :: =**

  
[Descripción de la ilustración '' mining\_attribute\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/mining_attribute_clause.htm)

**Propósito**

Esta función es para uso con modelos de clustering creados por el paquete DBMS\_DATA\_MINING o con Oracle Data Miner. Devuelve una variedad de objetos que contienen todos los grupos posibles a los que pertenece una fila determinada. Cada objeto en la varray es un par de valores escalares que contienen la ID del cluster y la probabilidad del cluster. Los campos de objeto se nombran CLUSTER\_ID y PROBABILITY, y ambos son Oracle NUMBER.

* Para el argumento opcional *topN* , especifique un entero positivo. Si lo hace, restringe el conjunto de clústeres predichos a aquellos que tienen uno de los Nvalores de probabilidad más altos. Si lo omite *topN*o lo configura NULL, todos los clústeres se devuelven en la colección. Si varios clústeres están vinculados por el valor Nth, la base de datos aún devuelve solo valores N.
* Para el argumento opcional  *cutoff*, especifique un entero positivo para restringir los grupos devueltos a aquellos con una probabilidad mayor o igual al límite especificado. Puede filtrar solo especificando  *cutoff* NULL para *topN* y el valor de corte deseado para *cutoff*.

Puede especificar *topN* y *cutoff* juntos restringir los clústeres devueltos a aquellos que están en la parte superior Ny tienen una probabilidad que supera el umbral.

*mining\_attribute\_clause* se comporta como se describe para la función PREDICTION. Consulte *[mining\_attribute\_clause](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions132.htm" \l "CJAJACJD)* .

**Ver también:**

* [*Conceptos de minería de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16808/toc.htm) para obtener información detallada sobre la minería de datos de Oracle
* [*Guía del desarrollador de la aplicación Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e12218/models_deploying.htm#DMPRG004) para obtener información detallada sobre la puntuación con las funciones SQL de Data Mining

**Ejemplos**

El siguiente ejemplo enumera los atributos más relevantes (con confianza> 55%) de cada grupo al que pertenece el cliente 101362 con> 20% de probabilidad.

Este ejemplo y el requisito previo de las operaciones de extracción de datos, incluida la creación del modelo km\_sh\_clus\_sample y las vistas y el tipo, se pueden encontrar en el archivo de demostración $ORACLE\_HOME/rdbms/demo/dmkmdemo.sql. La información general sobre los archivos de demostración de minería de datos está disponible en [*la Guía del administrador de Oracle Data Mining*](https://docs.oracle.com/cd/E11882_01/datamine.112/e16807/sampleprogs.htm#DMADM009) . El ejemplo se presenta aquí para ilustrar el uso sintáctico de la función.

WITH

clus\_tab AS (

SELECT id,

A.attribute\_name aname,

A.conditional\_operator op,

NVL(A.attribute\_str\_value,

ROUND(A.attribute\_num\_value),4)) val,

A.attribute\_support support,

A.attribute\_confidence confidence

FROM TABLE(DBMS\_DATA\_MINING.GET\_MODEL\_DETAILS\_KM('km\_sh\_clus\_sample')) T,

TABLE(T.rule.antecedent) A

WHERE A.attribute\_confidence > 0.55

),

clust AS (

SELECT id,

CAST(COLLECT(Cattr(aname, op, TO\_CHAR(val), support, confidence))

AS Cattrs) cl\_attrs

FROM clus\_tab

GROUP BY id

),

custclus AS (

SELECT T.cust\_id, S.cluster\_id, S.probability

FROM (SELECT cust\_id, CLUSTER\_SET(km\_sh\_clus\_sample, NULL, 0.2 USING \*) pset

FROM mining\_data\_apply\_v

WHERE cust\_id = 101362) T,

TABLE(T.pset) S

)

SELECT A.probability prob, A.cluster\_id cl\_id,

B.attr, B.op, B.val, B.supp, B.conf

FROM custclus A,

(SELECT T.id, C.\*

FROM clust T,

TABLE(T.cl\_attrs) C) B

WHERE A.cluster\_id = B.id

ORDER BY prob DESC, cl\_id ASC, conf DESC, attr ASC, val ASC;

PROB CL\_ID ATTR OP VAL SUPP CONF

------- ---------- -------------------------- -- ----------- -------- -------

.7745 8 HOUSEHOLD\_SIZE IN 9+ 124 .7500

.7745 8 CUST\_MARITAL\_STATUS IN Divorc. 116 .6000

.7745 8 CUST\_MARITAL\_STATUS IN NeverM 116 .6000

.7745 8 CUST\_MARITAL\_STATUS IN Separ. 116 .6000

.7745 8 CUST\_MARITAL\_STATUS IN Widowed 116 .6000

.2028 6 AGE >= 17 154 .6667

.2028 6 AGE <= 31.6 154 .6667

.2028 6 CUST\_MARITAL\_STATUS IN NeverM 172 .6667

8 rows selected.

## COALESCE

**Sintaxis**

Descripción de coalesce.gif sigue  
[Descripción de la ilustración '' coalesce.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/coalesce.htm)

**Propósito**

COALESCE devuelve el primer no-nulo *expr* en la lista de expresiones. Debe especificar al menos dos expresiones. Si todas las apariciones de *expr* evaluar a nulo, la función devuelve nulo.

La base de datos Oracle utiliza la **evaluación de cortocircuito** . La base de datos evalúa cada *expr*valor y determina si lo es NULL, en lugar de evaluar todos los valores *expr* antes de determinar si alguno de ellos es NULL.

Si todas las apariciones *expr* son de tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos, entonces la Base de datos Oracle determina el argumento con la mayor prioridad numérica, convierte implícitamente los argumentos restantes a ese tipo de datos y devuelve esos datos. tipo.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

Esta función es una generalización de la función NVL.

También se puede utilizar COALESCE como una variedad de la expresión CASE. Por ejemplo,

COALESCE (expr1, expr2)

es equivalente a:

CASE WHEN expr1 IS NOT NULL THEN expr1 ELSE expr2 END

Similar,

COALESCE (expr1, expr2, ..., expr *n* )

donde n> = 3, es equivalente a:

CASE WHEN expr1 IS NOT NULL THEN expr1

ELSE COALESCE (expr2, ..., expr*n*) END

**Ver también:**

[NVL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions119.htm#i91798) y ["Expresiones CASE"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions004.htm#i1033392)

**Ejemplos**

El siguiente ejemplo utiliza la tabla de muestra  oe.product\_information para organizar una venta de liquidación de productos. Da un descuento del 10% a todos los productos con un precio de lista. Si no hay un precio de lista, entonces el precio de venta es el precio mínimo. Si no hay un precio mínimo, entonces el precio de venta es "5":

SELECT product\_id, list\_price, min\_price,

COALESCE(0.9\*list\_price, min\_price, 5) "Sale"

FROM product\_information

WHERE supplier\_id = 102050

ORDER BY product\_id;

PRODUCT\_ID LIST\_PRICE MIN\_PRICE Sale

---------- ---------- ---------- ----------

1769 48 43.2

1770 73 73

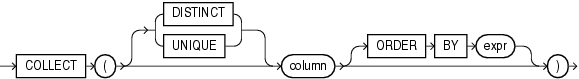
2378 305 247 274.5

2382 850 731 765

3355 5

## COLLECT

**Sintaxis**

  
[Descripción de la ilustración '' collect.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/collect.htm)

**Propósito**

COLLECT es una función agregada que toma como argumento una columna de cualquier tipo y crea una tabla anidada del tipo de entrada a partir de las filas seleccionadas. Para obtener resultados precisos de esta función, debe utilizarla dentro de una función CAST.

Si *columna* es una colección, la salida COLLECT es una tabla de colecciones anidadas. Si *columna* es de un tipo definido por el usuario, a continuación, *columna* debe tener una MAP o método definido ORDER en ella con el fin de que usted utilice los opcionales cláusulas DISTINCT, UNIQUEy ORDER BY.

**Ver también:**

[CAST](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions023.htm#i1269136) y ["funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)

**Ejemplos**

El siguiente ejemplo crea una tabla anidada de la columna varray de números de teléfono en la tabla de ejemplo oe.customers. La tabla anidada sólo incluye los números de teléfono de los clientes con un nivel de ingresos de L: 300,000 and above.

CREATE TYPE phone\_book\_t AS TABLE OF phone\_list\_typ;

/

SELECT CAST(COLLECT(phone\_numbers) AS phone\_book\_t) "Income Level L Phone Book"

FROM customers

WHERE income\_level = 'L: 300,000 and above';

Income Level L Phone Book

--------------------------------------------------------------------------------

PHONE\_BOOK\_T(PHONE\_LIST\_TYP('+1 414 123 4307'), PHONE\_LIST\_TYP('+1 608 123 4344'

), PHONE\_LIST\_TYP('+1 814 123 4696'), PHONE\_LIST\_TYP('+1 215 123 4721'), PHONE\_L

IST\_TYP('+1 814 123 4755'), PHONE\_LIST\_TYP('+91 11 012 4817', '+91 11 083 4817')

, PHONE\_LIST\_TYP('+91 172 012 4837'), PHONE\_LIST\_TYP('+41 31 012 3569', '+41 31

083 3569'))

El siguiente ejemplo crea una tabla anidada de la columna de nombres de almacén en la tabla de ejemplo oe. warehouses. Se utiliza ORDERBYpara ordenar los nombres de almacén.

CREATE TYPE warehouse\_name\_t AS TABLE OF VARCHAR2(35);

/

SELECT CAST(COLLECT(warehouse\_name ORDER BY warehouse\_name)

AS warehouse\_name\_t) "Warehouses"

FROM warehouses;

Warehouses

--------------------------------------------------------------------------------

WAREHOUSE\_NAME\_TYP('Beijing', 'Bombay', 'Mexico City', 'New Jersey', 'San Franci

sco', 'Seattle, Washington', 'Southlake, Texas', 'Sydney', 'Toronto')

## COMPOSE

**Sintaxis**

La descripción de compose.gif sigue  
[Descripción de la ilustración '' compose.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/compose.htm)

**Propósito**

COMPOSEtoma como argumento una cadena, o una expresión que se resuelve en una cadena, en cualquier tipo de datos, y devuelve una cadena Unicode en el mismo conjunto de caracteres que la entrada. *char*puede ser cualquiera de los tipos de datos CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, o NCLOB. Por ejemplo, un opunto de código calificado por un punto de código de diéresis se devolverá como el punto de código de o-umlaut.

COMPOSEdevuelve la cadena en forma normal NFC. Para una configuración más exclusiva, primero puede llamar DECOMPOSEcon la CANONICALconfiguración y luego COMPOSE. Esta combinación devuelve la cadena en forma normal NFKC.

CLOBy los NCLOBvalores son compatibles a través de la conversión implícita. Si *char*es un valor LOB de carácter, entonces se convierte en un VARCHARvalor antes de la COMPOSEoperación. La operación fallará si el tamaño del valor LOB supera la longitud admitida VARCHARen el entorno de desarrollo particular.

**Ver también:**

* [*Guía de soporte de globalización de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e10729/ch6unicode.htm#NLSPG006) para obtener información sobre conjuntos de caracteres Unicode y semántica de caracteres
* [DESCOMPONER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions050.htm#i1000217)

**Ejemplos**

El siguiente ejemplo devuelve el punto de código o-umlaut:

SELECT COMPOSE( 'o' || UNISTR('\0308') )

FROM DUAL;

CO

--

ö

## CONCAT

**Sintaxis**

Descripción de concat.gif sigue  
[Descripción de la ilustración '' concat.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/concat.htm)

**Propósito**

CONCATDevoluciones *char1*concatenadas con *char2*. Tanto *char1*y *char2*puede ser cualquiera de los tipos de datos CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, o NCLOB. La cadena devuelta está en el mismo conjunto de caracteres que *char1*. Su tipo de datos depende de los tipos de datos de los argumentos.

En las concatenaciones de dos tipos de datos diferentes, la base de datos Oracle devuelve el tipo de datos que resulta en una conversión sin pérdidas. Por lo tanto, si uno de los argumentos es un LOB, el valor devuelto es un LOB. Si uno de los argumentos es un tipo de datos nacional, entonces el valor devuelto es un tipo de datos nacional. Por ejemplo:

* CONCAT( CLOB, NCLOB) devuelveNCLOB
* CONCAT( NCLOB, NCHAR) devuelveNCLOB
* CONCAT( NCLOB, CHAR) devuelveNCLOB
* CONCAT( NCHAR, CLOB) devuelveNCLOB

Esta función es equivalente al operador de concatenación (||).

**Ver también:**

["Operador de concatenación"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/operators003.htm#i997789) para información sobre el CONCAToperador

**Ejemplos**

Este ejemplo utiliza el anidamiento para concatenar tres cadenas de caracteres:

SELECT CONCAT(CONCAT(last\_name, '''s job category is '), job\_id) "Job"

FROM employees

WHERE employee\_id = 152;

Job

------------------------------------------------------

Hall's job category is SA\_REP

## CONVERT

**Sintaxis**

Descripción de convert.gif sigue  
[Descripción de la ilustración '' convert.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/convert.htm)

**Propósito**

CONVERT convierte una cadena de caracteres de un conjunto de caracteres a otro.

* El *char*argumento es el valor a convertir. Puede ser cualquiera de los tipos de datos CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, o NCLOB.
* El *dest\_char\_set*argumento es el nombre del conjunto de caracteres al que *char*se convierte.
* El *source\_char\_set*argumento es el nombre del conjunto de caracteres en el que *char*se almacena en la base de datos. El valor predeterminado es el conjunto de caracteres de la base de datos.

El valor de retorno para CHARy VARCHAR2es VARCHAR2. Para NCHARy NVARCHAR2, lo es NVARCHAR2. Porque CLOBes CLOBy NCLOBes para NCLOB.

Tanto el destino como los argumentos del conjunto de caracteres de origen pueden ser literales o columnas que contienen el nombre del conjunto de caracteres.

Para una correspondencia completa en la conversión de caracteres, es esencial que el conjunto de caracteres de destino contenga una representación de todos los caracteres definidos en el conjunto de caracteres de origen. Cuando no existe un carácter en el conjunto de caracteres de destino, aparece un carácter de reemplazo. Los caracteres de reemplazo se pueden definir como parte de una definición de conjunto de caracteres.

**Nota:**

Oracle desalienta el uso de la CONVERTfunción en la versión actual de la base de datos Oracle. El valor de retorno de CONVERTtiene un tipo de datos de carácter, por lo que debe estar en el conjunto de caracteres de la base de datos o en el conjunto de caracteres nacional, según el tipo de datos. Cualquiera *dest\_char\_set*que no sea uno de estos dos conjuntos de caracteres no es compatible. El *char*argumento y el *source\_char\_set*tienen los mismos requisitos. Por lo tanto, el único uso práctico de la función es corregir datos que se han almacenado en un conjunto de caracteres incorrecto.

Los valores que no se encuentran en la base de datos ni en el conjunto de caracteres nacionales deben procesarse y almacenarse como RAWo BLOB. Los procedimientos en los paquetes PL / SQL UTL\_RAWy UTL\_I18N, por ejemplo, UTL\_RAW.CONVERTpermiten el procesamiento limitado de dichos valores. Procedimientos de aceptar un RAWargumento en los paquetes UTL\_FILE, UTL\_TCP, UTL\_HTTP, y UTL\_SMTPse pueden utilizar para dar salida a los datos procesados.

**Ejemplos**

El siguiente ejemplo ilustra la conversión de juegos de caracteres al convertir una cadena Latin-1 a ASCII. El resultado es el mismo que importar la misma cadena de una base de datos WE8ISO8859P1 a una base de datos US7ASCII.

SELECT CONVERT('Ä Ê Í Õ Ø A B C D E ', 'US7ASCII', 'WE8ISO8859P1')

FROM DUAL;

CONVERT('ÄÊÍÕØABCDE'

---------------------

A E I ? ? A B C D E ?

Los conjuntos de caracteres comunes incluyen:

* US7ASCII: juego de caracteres ASCII de 7 bits de EE. UU.
* WE8ISO8859P1: ISO 8859-1 Conjunto de caracteres de 8 bits de Europa occidental.
* EE8MSWIN1250: Código de Microsoft Windows Europa del Este página 1250
* WE8MSWIN1252: Código de Microsoft Windows Europa Occidental página 1252
* WE8EBCDIC1047: Código de IBM EBCDIC de Europa Occidental página 1047
* JA16SJISTILDE: Conjunto de caracteres japonés Shift-JIS, compatible con MS Code Página 932
* ZHT16MSWIN950: Página de códigos de chino tradicional de Microsoft Windows 950
* UTF8: Unicode 3.0 Forma de codificación de juego de caracteres universal CESU-8
* AL32UTF8: Forma de codificación UTF-8 del juego de caracteres universal de Unicode 5.0

Puede consultar la V$NLS\_VALID\_VALUESvista para obtener una lista de conjuntos de caracteres válidos, de la siguiente manera:

SELECT \* FROM V$NLS\_VALID\_VALUES WHERE parameter = 'CHARACTERSET';

## CORR

**Sintaxis**

Descripción de corr.gif sigue  
[Descripción de la ilustración '' corr.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/corr.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

CORRdevuelve el coeficiente de correlación de un conjunto de pares de números. Puedes usarlo como una función agregada o analítica.

Esta función toma como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Oracle determina el argumento con la precedencia numérica más alta, convierte implícitamente los argumentos restantes a ese tipo de datos y devuelve ese tipo de datos.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

La base de datos Oracle aplica la función al conjunto de ( *expr1*, *expr2*) después de eliminar los pares para los cuales *expr1*o *expr2*es nulo. Entonces Oracle realiza el siguiente cálculo:

COVAR\_POP(expr1, expr2) / (STDDEV\_POP(expr1) \* STDDEV\_POP(expr2))

La función devuelve un valor de tipo NUMBER. Si la función se aplica a un conjunto vacío, devuelve nulo.

**Nota:**

La CORRfunción calcula el coeficiente de correlación de Pearson, que requiere expresiones numéricas como argumentos. Oracle también proporciona las funciones ( CORR\_Scoeficiente rho de Spearman) y CORR\_K(coeficiente tau-b de Kendall) para admitir la correlación no paramétrica o de rango.

**Ver también:**

["Funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203) , ["Acerca de las expresiones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions001.htm#i1002626) para obtener información sobre formas válidas de *expr*, y [CORR\_ \*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions036.htm#i1293691) para obtener información sobre las funciones CORR\_SyCORR\_K

**Ejemplo Agregado**

El siguiente ejemplo calcula el coeficiente de correlación entre los precios de lista y los precios mínimos de los productos por clase de peso en la tabla de muestra oe.product\_information:

SELECT weight\_class, CORR(list\_price, min\_price) "Correlation"

FROM product\_information

GROUP BY weight\_class

ORDER BY weight\_class, "Correlation";

WEIGHT\_CLASS Correlation

------------ -----------

1 .999149795

2 .999022941

3 .998484472

4 .999359909

5 .999536087

**Ejemplo analítico**

El siguiente ejemplo muestra la correlación entre la duración en la empresa y el salario por la posición del empleado. El conjunto de resultados muestra la misma correlación para cada empleado en un trabajo determinado:

SELECT employee\_id, job\_id,

TO\_CHAR((SYSDATE - hire\_date) YEAR TO MONTH ) "Yrs-Mns", salary,

CORR(SYSDATE-hire\_date, salary)

OVER(PARTITION BY job\_id) AS "Correlation"

FROM employees

WHERE department\_id in (50, 80)

ORDER BY job\_id, employee\_id;

EMPLOYEE\_ID JOB\_ID Yrs-Mns SALARY Correlation

----------- ---------- ------- ---------- -----------

145 SA\_MAN +04-09 14000 .912385598

146 SA\_MAN +04-06 13500 .912385598

147 SA\_MAN +04-04 12000 .912385598

148 SA\_MAN +01-08 11000 .912385598

149 SA\_MAN +01-05 10500 .912385598

150 SA\_REP +04-05 10000 .80436755

151 SA\_REP +04-03 9500 .80436755

152 SA\_REP +03-10 9000 .80436755

153 SA\_REP +03-03 8000 .80436755

154 SA\_REP +02-07 7500 .80436755

155 SA\_REP +01-07 7000 .80436755

. . .

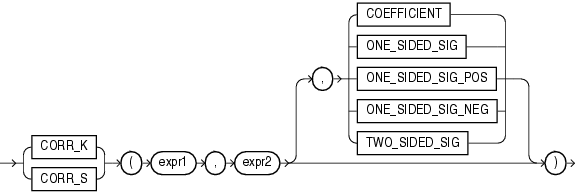
## CORR\_ \*

Las CORR\_\* funciones son:

* CORR\_S
* CORR\_K

**Sintaxis**

***correlación* :: =**

  
[Descripción de la ilustración '' correlation.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/correlation.htm)

**Propósito**

La CORRfunción (ver [CORR](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions035.htm#i82637) ) calcula el coeficiente de correlación de Pearson y requiere expresiones numéricas como entrada. Las CORR\_funciones \* admiten la correlación no paramétrica o de rango. Le permiten encontrar correlaciones entre las expresiones que tienen una escala ordinal (donde es posible clasificar los valores). Los coeficientes de correlación toman un valor que va de -1 a 1, donde 1 indica una relación perfecta, -1 una relación inversa perfecta (cuando una variable aumenta a medida que la otra disminuye), y un valor cercano a 0 significa que no hay relación.

Estas funciones toman como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. La base de datos de Oracle determina el argumento con la mayor prioridad numérica, convierte implícitamente los argumentos restantes a ese tipo de datos, realiza el cálculo y devuelve NUMBER.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

*expr1*y *expr2*son las dos variables que se están analizando. El tercer argumento es un valor de retorno de tipo VARCHAR2. Si omite el tercer argumento, entonces el valor predeterminado es COEFFICIENT. El significado de los valores de retorno se muestra en la tabla siguiente:

***Tabla 5-2 CORR\_ \* Valores de retorno***

| **Valor de retorno** | **Sentido** |
| --- | --- |
| COEFFICIENT | Coeficiente de correlación |
| ONE\_SIDED\_SIG | Significación positiva de una cola de la correlación. |
| ONE\_SIDED\_SIG\_POS | Igual que ONE\_SIDED\_SIG |
| ONE\_SIDED\_SIG\_NEG | Significación negativa de una cola de la correlación. |
| TWO\_SIDED\_SIG | Significado bilateral de la correlación. |

### CORR\_S

CORR\_Scalcula el coeficiente de correlación rho de Spearman. Las expresiones de entrada deben ser un conjunto de (x i , y i ) pares de observaciones. La función primero reemplaza cada valor con un rango. Cada valor de x i se reemplaza con su rango entre todos los otros x i s en la muestra, y cada valor de y i se reemplaza con su rango entre todos los demás y i s. Por lo tanto, cada x i y y i toman un valor de 1 a *n*, donde*n*es el número total de pares de valores. A los empates se les asigna el promedio de los rangos que habrían tenido si sus valores hubieran sido ligeramente diferentes. Luego la función calcula el coeficiente de correlación lineal de los rangos.

**Ejemplo CORR\_S**Utilizando el coeficiente de correlación rho de Spearman, el siguiente ejemplo deriva un coeficiente de correlación para cada una de dos comparaciones diferentes - salaryy commission\_pct, y salaryy y employee\_id:

SELECT COUNT(\*) count,

CORR\_S(salary, commission\_pct) commission,

CORR\_S(salary, employee\_id) empid

FROM employees;

COUNT COMMISSION EMPID

---------- ---------- ----------

107 .735837022 -.04473016

### CORR\_K

CORR\_Kcalcula el coeficiente de correlación tau-b de Kendall. En cuanto a CORR\_S, las expresiones de entrada son un conjunto de (x i , y i ) pares de observaciones. Para calcular el coeficiente, la función cuenta el número de pares concordantes y discordantes. Un par de observaciones es concordante si la observación con la x más grande también tiene un valor mayor de y. Un par de observaciones es discordante si la observación con la x más grande tiene una y más pequeña.

La importancia de tau-b es la probabilidad de que la correlación indicada por tau-b se deba al azar: un valor de 0 a 1. Un valor pequeño indica una correlación significativa para valores positivos de tau-b (o anticorrelación para valores negativos de tau-b).

**Ejemplo de CORR\_K**Utilizando el coeficiente de correlación tau-b de Kendall, el siguiente ejemplo determina si existe una correlación entre el salario de un empleado y el porcentaje de comisión:

SELECT CORR\_K(salary, commission\_pct, 'COEFFICIENT') coefficient,

CORR\_K(salary, commission\_pct, 'TWO\_SIDED\_SIG') two\_sided\_p\_value

FROM employees;

COEFFICIENT TWO\_SIDED\_P\_VALUE

----------- -----------------

.603079768 3.4702E-07

## COS

**Sintaxis**

Descripción de cos.gif sigue  
[Descripción de la ilustración '' cos.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cos.htm)

**Propósito**

COSdevuelve el coseno de *n*(un ángulo expresado en radianes).

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el coseno de 180 grados:

SELECT COS(180 \* 3.14159265359/180) "Cosine of 180 degrees"

FROM DUAL;

Cosine of 180 degrees

---------------------

-1

## COSH

**Sintaxis**

Descripción de cosh.gif sigue  
[Descripción de la ilustración '' cosh.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cosh.htm)

**Propósito**

COSHDevuelve el coseno hiperbólico *n*.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo devuelve el coseno hiperbólico de cero:

SELECT COSH(0) "Hyperbolic cosine of 0"

FROM DUAL;

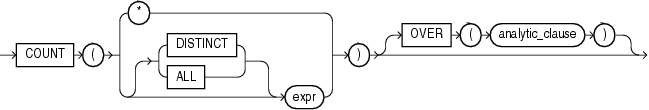
Hyperbolic cosine of 0

----------------------

1

## COUNT

**Sintaxis**

  
[Descripción de la ilustración '' count.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/count.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

COUNTdevuelve el número de filas devueltas por la consulta. Puedes usarlo como una función agregada o analítica.

Si lo especifica DISTINCT, puede especificar solo el *query\_partition\_clause*de *analytic\_clause*. Los *order\_by\_clause*y *windowing\_clause*no están permitidos.

Si lo especifica *expr*, COUNTdevuelve el número de filas donde *expr*no es nulo. Puede contar todas las filas o solo valores distintos de *expr*.

Si especifica el asterisco (\*), esta función devuelve todas las filas, incluidos los duplicados y los nulos. COUNTNunca devuelve nulo.

**Ver también:**

["Acerca de las expresiones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions001.htm#i1002626) para obtener información sobre formas válidas de *expr*y ["Funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)

**Ejemplos agregados**

Los siguientes ejemplos se usan COUNTcomo una función agregada:

SELECT COUNT(\*) "Total"

FROM employees;

Total

----------

107

SELECT COUNT(\*) "Allstars"

FROM employees

WHERE commission\_pct > 0;

Allstars

---------

35

SELECT COUNT(commission\_pct) "Count"

FROM employees;

Count

----------

35

SELECT COUNT(DISTINCT manager\_id) "Managers"

FROM employees;

Managers

----------

18

**Ejemplo analítico**

El siguiente ejemplo calcula, para cada empleado en la employeestabla, el recuento móvil de empleados que ganan salarios en el rango 50 menos que 150 más que el salario del empleado.

SELECT last\_name, salary,

COUNT(\*) OVER (ORDER BY salary RANGE BETWEEN 50 PRECEDING AND

150 FOLLOWING) AS mov\_count

FROM employees

ORDER BY salary, last\_name;

LAST\_NAME SALARY MOV\_COUNT

------------------------- ---------- ----------

Olson 2100 3

Markle 2200 2

Philtanker 2200 2

Gee 2400 8

Landry 2400 8

Colmenares 2500 10

Marlow 2500 10

Patel 2500 10

. . .

## COVAR\_POP

**Sintaxis**

La descripción de covar_pop.gif sigue  
[Descripción de la ilustración '' covar\_pop.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/covar_pop.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

COVAR\_POPdevuelve la covarianza de la población de un conjunto de pares de números. Puedes usarlo como una función agregada o analítica.

Esta función toma como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Oracle determina el argumento con la precedencia numérica más alta, convierte implícitamente los argumentos restantes a ese tipo de datos y devuelve ese tipo de datos.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

La base de datos de Oracle aplica la función al conjunto de pares ( *expr1*, *expr2*) después de eliminar todos los pares para los cuales *expr1*o *expr2*es nulo. Entonces Oracle realiza el siguiente cálculo:

(SUM(expr1 \* expr2) - SUM(expr2) \* SUM(expr1) / n) / n

donde *n*es el número de ( *expr1*, *expr2*) pares donde ni *expr1*ni *expr2*es nulo.

La función devuelve un valor de tipo NUMBER. Si la función se aplica a un conjunto vacío, devuelve nulo.

**Ver también :**

["Acerca de las expresiones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions001.htm#i1002626) para obtener información sobre formas válidas de *expr*y ["Funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)

**Ejemplo Agregado**

El siguiente ejemplo calcula la covarianza de la población y la covarianza de la muestra por el tiempo empleado ( SYSDATE- hire\_date) y el salario utilizando la tabla de muestra hr.employees:

SELECT job\_id,

COVAR\_POP(SYSDATE-hire\_date, salary) AS covar\_pop,

COVAR\_SAMP(SYSDATE-hire\_date, salary) AS covar\_samp

FROM employees

WHERE department\_id in (50, 80)

GROUP BY job\_id

ORDER BY job\_id, covar\_pop, covar\_samp;

JOB\_ID COVAR\_POP COVAR\_SAMP

---------- ----------- -----------

SA\_MAN 660700 825875

SA\_REP 579988.466 600702.34

SH\_CLERK 212432.5 223613.158

ST\_CLERK 176577.25 185870.789

ST\_MAN 436092 545115

**Ejemplo analítico**

El siguiente ejemplo calcula la covarianza de muestra acumulada del precio de lista y el precio mínimo de los productos en el esquema de muestra oe:

SELECT product\_id, supplier\_id,

COVAR\_POP(list\_price, min\_price)

OVER (ORDER BY product\_id, supplier\_id)

AS CUM\_COVP,

COVAR\_SAMP(list\_price, min\_price)

OVER (ORDER BY product\_id, supplier\_id)

AS CUM\_COVS

FROM product\_information p

WHERE category\_id = 29

ORDER BY product\_id, supplier\_id;

PRODUCT\_ID SUPPLIER\_ID CUM\_COVP CUM\_COVS

---------- ----------- ---------- ----------

1774 103088 0

1775 103087 1473.25 2946.5

1794 103096 1702.77778 2554.16667

1825 103093 1926.25 2568.33333

2004 103086 1591.4 1989.25

2005 103086 1512.5 1815

2416 103088 1475.97959 1721.97619

. . .

## COVAR\_SAMP

**Sintaxis**

La descripción de covar_samp.gif sigue  
[Descripción de la ilustración '' covar\_samp.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/covar_samp.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

COVAR\_SAMPdevuelve la muestra de covarianza de un conjunto de pares de números. Puedes usarlo como una función agregada o analítica.

Esta función toma como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Oracle determina el argumento con la precedencia numérica más alta, convierte implícitamente los argumentos restantes a ese tipo de datos y devuelve ese tipo de datos.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

La base de datos de Oracle aplica la función al conjunto de pares ( *expr1*, *expr2*) después de eliminar todos los pares para los cuales *expr1*o *expr2*es nulo. Entonces Oracle realiza el siguiente cálculo:

(SUM(expr1 \* expr2) - SUM(expr1) \* SUM(expr2) / n) / (n-1)

donde *n*es el número de ( *expr1*, *expr2*) pares donde ni *expr1*ni *expr2*es nulo.

La función devuelve un valor de tipo NUMBER. Si la función se aplica a un conjunto vacío, devuelve nulo.

**Ver también:**

["Acerca de las expresiones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions001.htm#i1002626) para obtener información sobre formas válidas de *expr*y ["Funciones agregadas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions003.htm#i89203)

**Ejemplo Agregado**

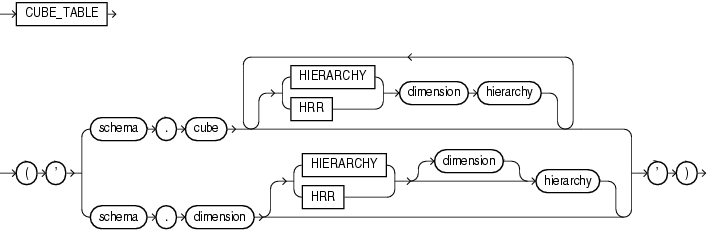
Consulte el ejemplo agregado para [COVAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions040.htm#i1008854) .

**Ejemplo analítico**

Consulte el ejemplo analítico para [COVAR\_POP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions040.htm#i1008854) .

## CUBE\_TABLE

**Sintaxis**

  
[Descripción de la ilustración '' cube\_table.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cube_table.htm)

**Propósito**

CUBE\_TABLE extrae datos de un cubo o dimensión y los devuelve en el formato bidimensional de una tabla relacional, que puede ser utilizada por aplicaciones basadas en SQL.

La función toma un solo VARCHAR2argumento. La cláusula de jerarquía opcional le permite especificar una jerarquía de dimensión. Un cubo puede tener múltiples cláusulas de jerarquía, una para cada dimensión.

Puedes generar estos diferentes tipos de tablas:

* Una tabla de cubos contiene una columna clave para cada dimensión y una columna para cada medida y medida calculada en el cubo. Para crear una tabla de cubos, puede especificar el cubo con o sin una cláusula de jerarquía de cubos. Para una dimensión con múltiples jerarquías, esta cláusula limita los valores de retorno a los miembros y niveles de la dimensión en la jerarquía especificada. Sin una cláusula de jerarquía, se incluyen todos los miembros de dimensión y todos los niveles.
* Una tabla de dimensiones contiene una columna clave y una columna para cada nivel y cada atributo. También contiene una MEMBER\_TYPEcolumna, que identifica a cada miembro con uno de los siguientes códigos:
  + L - Cargado desde una tabla, vista, o sinónimo
  + A - Miembro cargado y la raíz única de todas las jerarquías en la dimensión, es decir, el miembro agregado "todos"
  + C - Miembro calculado

Todos los miembros de dimensión y todos los niveles están incluidos en la tabla. Para crear una tabla de dimensiones, especifique la dimensión **sin** una cláusula de jerarquía de dimensiones.

* Una tabla de jerarquía contiene todas las columnas de una tabla de dimensiones más una columna para el miembro padre y una columna para cada nivel de origen. También contiene una MEMBER\_TYPEcolumna, como se describe para las tablas de dimensiones. Los miembros y niveles de dimensión que no forman parte de la jerarquía nombrada se excluyen de la tabla. Para crear una tabla de jerarquía, especifique la dimensión **con** una cláusula de jerarquía de dimensión.

CUBE\_TABLEes una función de tabla y siempre se utiliza en el contexto de una SELECTdeclaración con esta sintaxis:

SELECT ... FROM TABLE(CUBE\_TABLE('*arg*'));

**Ver también:**

[*La Guía del usuario de Oracle OLAP*](https://docs.oracle.com/cd/E11882_01/olap.112/e17123/query.htm#OLAUG400) para obtener información sobre objetos dimensionales y sobre las tablas generadas porCUBE\_TABLE.

**Ejemplos**

Los siguientes ejemplos requieren Oracle Database con la opción OLAP y el GLOBALesquema de muestra. Consulte [*la Guía del usuario de Oracle OLAP*](https://docs.oracle.com/cd/E11882_01/olap.112/e17123/start.htm#OLAUG9117) para obtener información sobre cómo descargar e instalar el GLOBALesquema de muestra.

La siguiente SELECTdeclaración genera una tabla de dimensiones CHANNELen el GLOBALesquema.

SELECT dim\_key, level\_name, long\_description, channel\_total\_id tot\_id,

channel\_channel\_id chan\_id, channel\_long\_description chan\_desc,

total\_long\_description tot\_desc

FROM TABLE(CUBE\_TABLE('global.channel'));

DIM\_KEY LEVEL\_NAME LONG\_DESCRIPTION TOT\_ID CHAN\_ID CHAN\_DESC TOT\_DESC

----------- ---------- ---------------- ------ ------- ------------ -------------

CHANNEL\_CAT CHANNEL Catalog TOTAL CAT Catalog Total Channel

CHANNEL\_DIR CHANNEL Direct Sales TOTAL DIR Direct Sales Total Channel

CHANNEL\_INT CHANNEL Internet TOTAL INT Internet Total Channel

TOTAL\_TOTAL TOTAL Total Channel TOTAL Total Channel

La siguiente sentencia genera una tabla de cubos de UNITS\_CUBE. Restringe la tabla a las jerarquías MARKETy CALENDAR.

SELECT sales, units, cost, time, customer, product, channel

FROM TABLE(CUBE\_TABLE('global.units\_cube HIERARCHY customer market HIERARCHY time calendar'))

WHERE rownum < 20;

SALES UNITS COST TIME CUSTOMER PRODUCT CHANNEL

---------- ---------- ---------- -------------------------- -------------- ----------- -----------

24538587.9 61109 22840853.7 CALENDAR\_QUARTER\_CY1998.Q1 TOTAL\_TOTAL TOTAL\_TOTAL TOTAL\_TOTAL

24993273.3 61320 23147171 CALENDAR\_QUARTER\_CY1998.Q2 TOTAL\_TOTAL TOTAL\_TOTAL TOTAL\_TOTAL

25080541.4 65265 23242535.4 CALENDAR\_QUARTER\_CY1998.Q3 TOTAL\_TOTAL TOTAL\_TOTAL TOTAL\_TOTAL

26258474 66122 24391020.6 CALENDAR\_QUARTER\_CY1998.Q4 TOTAL\_TOTAL TOTAL\_TOTAL TOTAL\_TOTAL

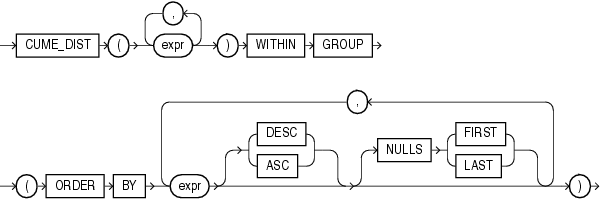
32785170 77589 30607218.1 CALENDAR\_QUARTER\_CY1999.Q1 TOTAL\_TOTAL TOTAL\_TOTAL TOTAL\_TOTAL

. . .

## CUME\_DIST

**Sintaxis agregada**

***cume\_dist\_aggregate* :: =**

  
[Descripción de la ilustración '' cume\_dist\_aggregate.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cume_dist_aggregate.htm)

**Sintaxis analítica**

***cume\_dist\_analytic* :: =**

La descripción de cume_dist_analytic.gif sigue  
[Descripción de la ilustración '' cume\_dist\_analytic.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cume_dist_analytic.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

CUME\_DISTcalcula la distribución acumulativa de un valor en un grupo de valores. El rango de valores devuelto por CUME\_DISTes> 0 a <= 1. Los valores de enlace siempre se evalúan al mismo valor de distribución acumulativa.

Esta función toma como argumentos cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. La base de datos de Oracle determina el argumento con la mayor prioridad numérica, convierte implícitamente los argumentos restantes a ese tipo de datos, realiza el cálculo y devuelve NUMBER.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita y ["Precedencia numérica"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i156865) para obtener información sobre la precedencia numérica

* Como una función agregada, CUME\_DISTcalcula, para una fila hipotética *r*identificada por los argumentos de la función y una especificación de clasificación correspondiente, la posición relativa de la fila *r*entre las filas en el grupo de agregación. Oracle realiza este cálculo como si la fila hipotética *r*se insertara en el grupo de filas que se agregarán. Los argumentos de la función identifican una única fila hipotética dentro de cada grupo agregado. Por lo tanto, todos deben evaluar las expresiones constantes dentro de cada grupo agregado. Las expresiones de argumento constante y las expresiones en la ORDER BYcláusula del agregado coinciden por posición. Por lo tanto, el número de argumentos debe ser el mismo y sus tipos deben ser compatibles.
* Como una función analítica, CUME\_DISTcalcula la posición relativa de un valor especificado en un grupo de valores. Para una fila *r*, asumiendo un orden ascendente, la CUME\_DISTde *r*es el número de filas con valores inferiores o iguales al valor de *r*, dividido por el número de filas que se evalúan (el conjunto de resultados de la consulta o una partición).

**Ejemplo Agregado**

El siguiente ejemplo calcula la distribución acumulativa de un empleado hipotético con un salario de $ 15,500 y una tasa de comisión del 5% entre los empleados en la tabla de muestra oe.employees:

SELECT CUME\_DIST(15500, .05) WITHIN GROUP

(ORDER BY salary, commission\_pct) "Cume-Dist of 15500"

FROM employees;

Cume-Dist of 15500

------------------

.972222222

**Ejemplo analítico**

El siguiente ejemplo calcula el percentil de salario para cada empleado en la división de compras. Por ejemplo, el 40% de los empleados tienen salarios menores o iguales a Himuro.

SELECT job\_id, last\_name, salary, CUME\_DIST()

OVER (PARTITION BY job\_id ORDER BY salary) AS cume\_dist

FROM employees

WHERE job\_id LIKE 'PU%'

ORDER BY job\_id, last\_name, salary, cume\_dist;

JOB\_ID LAST\_NAME SALARY CUME\_DIST

---------- ------------------------- ---------- ----------

PU\_CLERK Baida 2900 .8

PU\_CLERK Colmenares 2500 .2

PU\_CLERK Himuro 2600 .4

PU\_CLERK Khoo 3100 1

PU\_CLERK Tobias 2800 .6

PU\_MAN Raphaely 11000 1

## CURRENT\_DATE

**Sintaxis**

La descripción de current_date.gif sigue  
[Descripción de la ilustración '' current\_date.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/current_date.htm)

**Propósito**

CURRENT\_DATEdevuelve la fecha actual en la zona horaria de la sesión, en un valor en el calendario gregoriano de tipo de datos DATE.

**Ejemplos**

El siguiente ejemplo ilustra que CURRENT\_DATEes sensible a la zona horaria de la sesión:

ALTER SESSION SET TIME\_ZONE = '-5:0';

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY HH24:MI:SS';

SELECT SESSIONTIMEZONE, CURRENT\_DATE FROM DUAL;

SESSIONTIMEZONE CURRENT\_DATE

--------------- --------------------

-05:00 29-MAY-2000 13:14:03

ALTER SESSION SET TIME\_ZONE = '-8:0';

SELECT SESSIONTIMEZONE, CURRENT\_DATE FROM DUAL;

SESSIONTIMEZONE CURRENT\_DATE

--------------- --------------------

-08:00 29-MAY-2000 10:14:33

## CURRENT\_TIMESTAMP

**Sintaxis**

La descripción de current_timestamp.gif sigue  
[Descripción de la ilustración '' current\_timestamp.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/current_timestamp.htm)

**Propósito**

CURRENT\_TIMESTAMPdevuelve la fecha y hora actuales en la zona horaria de la sesión, en un valor de tipo de datos TIMESTAMP WITH TIMEZONE. El desplazamiento de la zona horaria refleja la hora local actual de la sesión de SQL. Si omite precisión, entonces el valor predeterminado es 6. La diferencia entre esta función y LOCALTIMESTAMPes que CURRENT\_TIMESTAMPdevuelve un TIMESTAMP WITH TIMEZONEvalor mientras que LOCALTIMESTAMPdevuelve un TIMESTAMPvalor.

En el argumento opcional, *precision*especifica la segunda precisión fraccional del valor de tiempo devuelto.

**Ver también:**

[LOCALTIMESTAMP](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions092.htm#i999873)

**Ejemplos**

El siguiente ejemplo ilustra que CURRENT\_TIMESTAMPes sensible a la zona horaria de la sesión:

ALTER SESSION SET TIME\_ZONE = '-5:0';

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY HH24:MI:SS';

SELECT SESSIONTIMEZONE, CURRENT\_TIMESTAMP FROM DUAL;

SESSIONTIMEZONE CURRENT\_TIMESTAMP

--------------- ---------------------------------------------------

-05:00 04-APR-00 01.17.56.917550 PM -05:00

ALTER SESSION SET TIME\_ZONE = '-8:0';

SELECT SESSIONTIMEZONE, CURRENT\_TIMESTAMP FROM DUAL;

SESSIONTIMEZONE CURRENT\_TIMESTAMP

--------------- ----------------------------------------------------

-08:00 04-APR-00 10.18.21.366065 AM -08:00

Cuando use el CURRENT\_TIMESTAMPcon una máscara de formato, tenga cuidado de que la máscara de formato coincida con el valor devuelto por la función. Por ejemplo, considere la siguiente tabla:

CREATE TABLE current\_test (col1 TIMESTAMP WITH TIME ZONE);

La siguiente declaración falla porque la máscara no incluye la TIME ZONEparte del tipo devuelto por la función:

INSERT INTO current\_test VALUES

(TO\_TIMESTAMP\_TZ(CURRENT\_TIMESTAMP, 'DD-MON-RR HH.MI.SSXFF PM'));

La siguiente declaración utiliza la máscara de formato correcta para coincidir con el tipo de retorno de CURRENT\_TIMESTAMP:

INSERT INTO current\_test VALUES

(TO\_TIMESTAMP\_TZ(CURRENT\_TIMESTAMP, 'DD-MON-RR HH.MI.SSXFF PM TZH:TZM'));

## CV

**Sintaxis**

Descripción de cv.gif sigue  
[Descripción de la ilustración '' cv.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cv.htm)

**Propósito**

La CVfunción se puede usar solo en el *model\_clause*de una SELECTdeclaración y luego solo en el lado derecho de una regla modelo. Devuelve el valor actual de una columna de dimensión o una columna de partición transportada desde el lado izquierdo al lado derecho de una regla. Esta función se utiliza *model\_clause*para proporcionar una indexación relativa con respecto a la columna de dimensión. El tipo de retorno es el del tipo de datos de la columna de dimensión. Si omite el argumento, entonces el valor predeterminado es la columna de dimensión asociada con la posición relativa de la función dentro de la referencia de celda.

La CVfunción se puede utilizar fuera de una referencia de celda. En este caso, *dimension\_column*es obligatorio.

**Ver también:**

[*model\_clause*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_10002.htm#i2172805) y["Model Expressions"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions010.htm#i1049969)para la sintaxis y la semántica

**Ejemplos**

El siguiente ejemplo asigna la suma de las ventas del producto representado por el valor actual de la columna de dimensiones (Mouse Pad o Standard Mouse) para los años 1999 y 2000 a las ventas de ese producto para el año 2001:

SELECT country, prod, year, s

FROM sales\_view\_ref

MODEL

PARTITION BY (country)

DIMENSION BY (prod, year)

MEASURES (sale s)

IGNORE NAV

UNIQUE DIMENSION

RULES UPSERT SEQUENTIAL ORDER

(

s[FOR prod IN ('Mouse Pad', 'Standard Mouse'), 2001] =

s[CV( ), 1999] + s[CV( ), 2000]

)

ORDER BY country, prod, year;

COUNTRY PROD YEAR S

---------- ----------------------------------- -------- ---------

France Mouse Pad 1998 2509.42

France Mouse Pad 1999 3678.69

France Mouse Pad 2000 3000.72

France Mouse Pad 2001 6679.41

France Standard Mouse 1998 2390.83

France Standard Mouse 1999 2280.45

France Standard Mouse 2000 1274.31

France Standard Mouse 2001 3554.76

Germany Mouse Pad 1998 5827.87

Germany Mouse Pad 1999 8346.44

Germany Mouse Pad 2000 7375.46

Germany Mouse Pad 2001 15721.9

Germany Standard Mouse 1998 7116.11

Germany Standard Mouse 1999 6263.14

Germany Standard Mouse 2000 2637.31

Germany Standard Mouse 2001 8900.45

16 rows selected.

El ejemplo anterior requiere la vista sales\_view\_ref. Consulte ["La cláusula MODELO: Ejemplos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_10002.htm#i2171160) para crear esta vista.

## DATAOBJ\_TO\_PARTITION

**Sintaxis**

La descripción de dataobj_to_partition.gif sigue  
[Descripción de la ilustración '' dataobj\_to\_partition.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dataobj_to_partition.htm)

**Propósito**

DATAOBJ\_TO\_PARTITIONsolo es útil para los desarrolladores de Data Cartridge que realizan operaciones de mantenimiento o consulta de datos en tablas particionadas del sistema que se utilizan para almacenar datos de índice de dominio. Las operaciones de consulta o DML se activan mediante las operaciones correspondientes en la tabla base del índice de dominio.

Esta función toma como argumentos el nombre de la tabla base y el ID de partición de la partición de tabla base, los cuales se pasan a la función por el método ODCIIndex apropiado. La función devuelve el ID de partición de la tabla particionada del sistema correspondiente, que se puede usar para realizar la operación (DML o consulta) en esa partición de la tabla particionada del sistema.

## DBTIMEZONE

**Sintaxis**

Descripción de dbtimezone.gif sigue  
[Descripción de la ilustración '' dbtimezone.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dbtimezone.htm)

**Propósito**

DBTIMEZONEdevuelve el valor de la zona horaria de la base de datos. El tipo de retorno es un desplazamiento de zona horaria (un tipo de carácter en el formato '[+|-]TZH:TZM') o un nombre de región de zona horaria, dependiendo de cómo el usuario especificó el valor de zona horaria de la base de datos en la declaración CREATE DATABASEo la más reciente ALTER DATABASE.

**Ejemplos**

El siguiente ejemplo asume que la zona horaria de la base de datos está configurada como zona horaria UTC:

SELECT DBTIMEZONE

FROM DUAL;

DBTIME

------

+00:00

## DECODE

**Sintaxis**

Descripción de decode.gif sigue  
[Descripción de la ilustración '' decode.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/decode.htm)

**Propósito**

DECODEse compara *expr*con cada *search*valor uno por uno. Si *expr*es igual a a *search*, la base de datos Oracle devuelve el correspondiente *result*. Si no se encuentra ninguna coincidencia, entonces vuelve Oracle *default*. Si *default*se omite, entonces Oracle devuelve nulo.

Los argumentos pueden ser cualquiera de los tipos numéricos ( NUMBER, BINARY\_FLOATo BINARY\_DOUBLE) o tipos de caracteres.

* Si *expr*y *search*son datos de caracteres, entonces Oracle los compara utilizando semánticas de comparación no rellenadas. *expr*, *search*Y *result*puede ser cualquiera de los tipos de datos CHAR, VARCHAR2, NCHAR, o NVARCHAR2. La cadena devuelta es de VARCHAR2tipo de datos y está en el mismo conjunto de caracteres que el primer *result*parámetro.
* Si el primer *search-result*par es numérico, entonces Oracle compara todas las *search-result*expresiones y el primero *expr*en determinar el argumento con la precedencia numérica más alta, convierte implícitamente los argumentos restantes a ese tipo de datos y devuelve ese tipo de datos.

El *search*, *result*y *default*los valores se pueden derivar de las expresiones. La base de datos Oracle utiliza la **evaluación de cortocircuito**. La base de datos evalúa cada *search*valor solo antes de compararlo con *expr*, en lugar de evaluar todos los *search*valores antes de comparar cualquiera de ellos con *expr*. En consecuencia, Oracle nunca evalúa a *search*si un anterior *search*es igual a *expr*.

Oracle convierte automáticamente *expr*y cada *search*valor al tipo de datos del primer *search*valor antes de comparar. Oracle convierte automáticamente el valor de retorno al mismo tipo de datos que el primero *result*. Si el primero *result*tiene el tipo de datos CHARo si el primero *result*es nulo, Oracle convierte el valor de retorno al tipo de datos VARCHAR2.

En una DECODEfunción, Oracle considera que dos nulos son equivalentes. Si *expr*es nulo, entonces Oracle devuelve *result*el primero *search*que también es nulo.

El número máximo de componentes de la DECODEfunción, incluyendo *expr*, *searches*, *results*, y *default*, es 255.

**Ver también:**

* ["Reglas de comparación de tipos de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i55214) para obtener información sobre la semántica de comparación
* ["Conversión de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i46862) para obtener información sobre la conversión de tipos de datos en general
* ["Números de punto flotante"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i140176) para obtener información sobre la semántica de comparación de punto flotante
* ["Conversión de datos implícita y explícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i53062) para obtener información sobre los inconvenientes de la conversión implícita
* ["COALESCE"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions030.htm#i1001341) y ["CASE Expressions"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/expressions004.htm#i1033392) , que proporcionan una funcionalidad similar a la deDECODE

**Ejemplos**

Este ejemplo decodifica el valor warehouse\_id. Si warehouse\_ides 1, entonces la función devuelve ' Southlake'; si warehouse\_ides 2, entonces devuelve ' San Francisco'; Etcétera. Si warehouse\_idno es 1, 2, 3 o 4, la función devuelve ' Non domestic'.

SELECT product\_id,

DECODE (warehouse\_id, 1, 'Southlake',

2, 'San Francisco',

3, 'New Jersey',

4, 'Seattle',

'Non domestic') "Location"

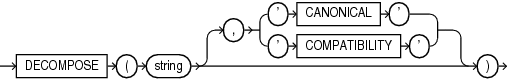
FROM inventories

WHERE product\_id < 1775

ORDER BY product\_id, "Location";

## DECOMPOSE

**Sintaxis**

  
[Descripción de la ilustración '' decompose.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/decompose.htm)

**Propósito**

DECOMPOSESólo es válido para caracteres Unicode. DECOMPOSEtoma como argumento una cadena en cualquier tipo de datos y devuelve una cadena Unicode después de la descomposición en el mismo conjunto de caracteres que la entrada. Por ejemplo, se devolverá un punto de código o-umlaut como el punto de código "o" seguido de un punto de código de diéresis.

* *string*puede ser cualquiera de los tipos de datos CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, o NCLOB.
* CANONICALprovoca la descomposición canónica, que permite la recomposición (por ejemplo, con la COMPOSEfunción) a la cadena original. Este es el valor predeterminado y devuelve la cadena en forma normal de NFD.
* COMPATIBILITYCausa la descomposición en modo compatibilidad. En este modo, la recomposición no es posible. Este modo es útil, por ejemplo, al descomponer caracteres *katakana de* ancho medio y ancho completo , donde la recomposición podría no ser deseable sin formato externo o información de estilo. Devuelve la cadena en forma normal NFKD.

CLOBy los NCLOBvalores son compatibles a través de la conversión implícita. Si *char*es un valor LOB de carácter, entonces se convierte en un VARCHARvalor antes de la COMPOSEoperación. La operación fallará si el tamaño del valor LOB supera la longitud admitida VARCHARen el entorno de desarrollo particular.

**Ver también:**

* [*Guía de soporte de globalización de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e10729/applocaledata.htm#NLSPG014) para obtener información sobre conjuntos de caracteres Unicode y semántica de caracteres
* [COMPONER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions032.htm#i1000195)

**Ejemplos**

El siguiente ejemplo descompone la cadena " Châteaux" en sus puntos de código de componente:

SELECT DECOMPOSE ('Châteaux')

FROM DUAL;

DECOMPOSE

---------

Cha^teaux

**Nota:**

Los resultados de este ejemplo pueden variar según el conjunto de caracteres de su sistema operativo.

## DELETEXML

**Sintaxis**

La descripción de deletexml.gif sigue  
[Descripción de la ilustración '' deletexml.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/deletexml.htm)

**Propósito**

DELETEXML elimina el nodo o los nodos que coinciden con la expresión XPath en el XML de destino.

* *XMLType\_instance*Es una instancia de XMLType.
* *XPath\_string*es una expresión Xpath que indica uno o más nodos que se eliminarán. Puede especificar un absoluto *XPath\_string*con una barra inicial o un pariente *XPath\_string*al omitir la barra inicial. Si omite la barra inicial, el contexto de la ruta relativa se establece de forma predeterminada en el nodo raíz. *XPath\_string*También se eliminan todos los nodos secundarios de los nodos especificados por .
* El opcional *namespace\_string*proporciona información de espacio de nombres para el *XPath\_string*. Este parámetro debe ser de tipo VARCHAR2.

**Ver también:**

[*Guía del desarrollador de Oracle XML DB*](https://docs.oracle.com/cd/E11882_01/appdev.112/e23094/xdb04cre.htm#ADXDB0440) para obtener más información sobre esta función

**Ejemplos**

El siguiente ejemplo elimina el /Ownernodo warehouse\_specde uno de los almacenes modificados en el ejemplo para [APPENDCHILDXML](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions012.htm#CIHEGIFE) :

UPDATE warehouses

SET warehouse\_spec = DELETEXML(warehouse\_spec, '/Warehouse/Building/Owner')

WHERE warehouse\_id = 2;

SELECT warehouse\_id, warehouse\_spec

FROM warehouses WHERE warehouse\_id in (2,3);

ID WAREHOUSE\_SPEC

---------- -----------------------------------

2 <?xml version="1.0"?>

<Warehouse>

<Building>Rented</Building>

<Area>50000</Area>

<Docks>1</Docks>

<DockType>Side load</DockType>

<WaterAccess>Y</WaterAccess>

<RailAccess>N</RailAccess>

<Parking>Lot</Parking>

<VClearance>12 ft</VClearance>

</Warehouse>

3 <?xml version="1.0"?>

<Warehouse>

<Building>Rented<Owner>Grandco</Owner>

</Building>

<Area>85700</Area>

<DockType/>

<WaterAccess>N</WaterAccess>

<RailAccess>N</RailAccess>

<Parking>Street</Parking>

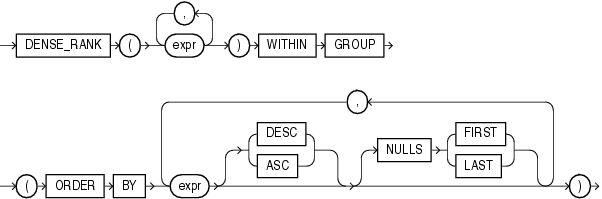
<VClearance>11.5 ft</VClearance>

</Warehouse>

## DENSE\_RANK

**Sintaxis agregada**

***dense\_rank\_aggregate* :: =**

  
[Descripción de la ilustración '' dense\_rank\_aggregate.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dense_rank_aggregate.htm)

**Sintaxis analítica**

***dense\_rank\_analytic* :: =**

A continuación se encuentra la descripción de dense_rank_analytic.gif.  
[Descripción de la ilustración '' dense\_rank\_analytic.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dense_rank_analytic.htm)

**Ver también:**

["Funciones analíticas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions004.htm#i81407) para obtener información sobre la sintaxis, la semántica y las restricciones

**Propósito**

DENSE\_RANKcalcula el rango de una fila en un grupo ordenado de filas y devuelve el rango como a NUMBER. Los rangos son enteros consecutivos que comienzan con 1. El valor de rango más grande es el número de valores únicos devueltos por la consulta. Los valores de rango no se saltan en caso de empate. Las filas con valores iguales para los criterios de clasificación reciben la misma clasificación. Esta función es útil para los informes de la parte superior y inferior de la N.

Esta función acepta como argumentos cualquier tipo de datos numéricos y devuelve NUMBER.

* Como una función agregada, DENSE\_RANKcalcula el rango denso de una fila hipotética identificada por los argumentos de la función con respecto a una especificación de clasificación dada. Los argumentos de la función deben evaluar todas las expresiones constantes dentro de cada grupo agregado, porque identifican una sola fila dentro de cada grupo. Las expresiones de los argumentos constantes y las expresiones en *order\_by\_clause*el agregado coinciden por posición. Por lo tanto, el número de argumentos debe ser el mismo y los tipos deben ser compatibles.
* Como una función analítica, DENSE\_RANKcalcula el rango de cada fila devuelta de una consulta con respecto a las otras filas, en función de los valores de *value\_exprs*en *order\_by\_clause*.

**Ejemplo Agregado**

El siguiente ejemplo calcula la clasificación de un empleado hipotético con el salario de $ 15,500 y una comisión del 5% en la tabla de muestra oe.employees:

SELECT DENSE\_RANK(15500, .05) WITHIN GROUP

(ORDER BY salary DESC, commission\_pct) "Dense Rank"

FROM employees;

Dense Rank

----------

3

**Ejemplo analítico**

La siguiente declaración clasifica a los empleados en el hresquema de muestra en el departamento 60 según sus salarios. Los valores salariales idénticos reciben el mismo rango. Sin embargo, no se saltan valores de rango. Compara este ejemplo con el ejemplo analítico de [RANK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions141.htm#i1269223) .

SELECT department\_id, last\_name, salary,

DENSE\_RANK() OVER (PARTITION BY department\_id ORDER BY salary) DENSE\_RANK

FROM employees WHERE department\_id = 60

ORDER BY DENSE\_RANK, last\_name;

DEPARTMENT\_ID LAST\_NAME SALARY DENSE\_RANK

------------- ------------------------- ---------- ----------

60 Lorentz 4200 1

60 Austin 4800 2

60 Pataballa 4800 2

60 Ernst 6000 3

60 Hunold 9000 4

## DEPTH

**Sintaxis**

A continuación se encuentra la descripción de depth.gif.  
[Descripción de la ilustración '' depth.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/depth.htm)

**Propósito**

DEPTHEs una función auxiliar utilizada solo con las condiciones UNDER\_PATHy EQUALS\_PATH. Devuelve el número de niveles en la ruta especificada por la UNDER\_PATHcondición con la misma variable de correlación.

El *correlation\_integer*puede ser cualquier NUMBERnúmero entero. Úselo para correlacionar esta función auxiliar con su condición primaria si la declaración contiene múltiples condiciones primarias. Los valores inferiores a 1 se tratan como 1.

**Ver también:**

[Condición EQUALS\_PATH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions009.htm#i1051094) , [condición](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions009.htm#i1051094)[UNDER\_PATH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions009.htm#i1041646) y la función relacionada [PATH](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions125.htm#i1150364)

**Ejemplos**

Las condiciones EQUALS\_PATHy UNDER\_PATHpueden tomar dos funciones auxiliares, DEPTHy PATH. El siguiente ejemplo muestra el uso de ambas funciones auxiliares. El ejemplo asume la existencia de XMLSchema warehouses.xsd(creado en ["Uso de XML en declaraciones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/ap_examples002.htm#i686084)).

SELECT PATH(1), DEPTH(2)

FROM RESOURCE\_VIEW

WHERE UNDER\_PATH(res, '/sys/schemas/OE', 1)=1

AND UNDER\_PATH(res, '/sys/schemas/OE', 2)=1;

PATH(1) DEPTH(2)

-------------------------------- --------

. . .

www.example.com 1

www.example.com/xwarehouses.xsd 2

. . .

## DEREF

**Sintaxis**

La descripción de deref.gif sigue.  
[Descripción de la ilustración '' deref.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/deref.htm)

**Propósito**

DEREFdevuelve la referencia de objeto del argumento *expr*, donde *expr*debe devolver una REFa un objeto. Si no utiliza esta función en una consulta, la base de datos Oracle devuelve el ID de objeto de la REFopción, como se muestra en el ejemplo que sigue.

**Ver también:**

[MAKE\_REF](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions097.htm#i77900)

**Ejemplos**

El esquema de muestra oecontiene un tipo de objeto cust\_address\_typ. Los ["Ejemplos de restricciones de REF"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/clauses002.htm#i1015744) crean un tipo similar cust\_address\_typ\_newy una tabla con una columna que es REFdel tipo. El siguiente ejemplo muestra cómo insertarse en dicha columna y cómo usarla DEREFpara extraer información de la columna:

INSERT INTO address\_table VALUES

('1 First', 'G45 EU8', 'Paris', 'CA', 'US');

INSERT INTO customer\_addresses

SELECT 999, REF(a) FROM address\_table a;

SELECT address

FROM customer\_addresses

ORDER BY address;

ADDRESS

--------------------------------------------------------------------------------

000022020876B2245DBE325C5FE03400400B40DCB176B2245DBE305C5FE03400400B40DCB1

SELECT DEREF(address)

FROM customer\_addresses;

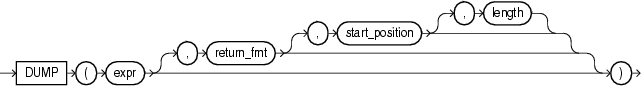
DEREF(ADDRESS)(STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID)

--------------------------------------------------------------------------------

CUST\_ADDRESS\_TYP\_NEW('1 First', 'G45 EU8', 'Paris', 'CA', 'US')

## DUMP

**Sintaxis**

  
[Descripción de la ilustración '' dump.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dump.htm)

**Propósito**

DUMPdevuelve un VARCHAR2valor que contiene el código del tipo de datos, la longitud en bytes y la representación interna de *expr*. El resultado devuelto está siempre en el conjunto de caracteres de la base de datos. Para el tipo de datos correspondiente a cada código, consulte la [Tabla 3-1, "Resumen de tipos de datos incorporados"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#BABCGCHG) .

El argumento *return\_fmt*especifica el formato del valor de retorno y puede tener cualquiera de los siguientes valores:

* 8 devuelve el resultado en notación octal.
* 10 devuelve el resultado en notación decimal.
* 16 devuelve el resultado en notación hexadecimal.
* 17 devuelve cada byte impreso como un carácter si y solo si puede interpretarse como un carácter imprimible en el conjunto de caracteres del compilador, generalmente ASCII o EBCDIC. Algunos caracteres de control ASCII también pueden imprimirse en la forma ^ X. De lo contrario, el carácter se imprime en notación hexadecimal. Todos los parámetros NLS son ignorados. No dependa de ningún formato de salida particular para DUMPcon *return\_fmt*17.

De forma predeterminada, el valor de retorno no contiene información de conjunto de caracteres. Para recuperar el nombre del conjunto de caracteres *expr*, agregue 1000 a cualquiera de los valores de formato anteriores. Por ejemplo, una *return\_fmt*de 1008 devuelve el resultado en octal y proporciona el nombre del conjunto de caracteres de *expr*.

Los argumentos *start\_position*y la *length*combinación para determinar qué parte de la representación interna devolver. El valor predeterminado es devolver toda la representación interna en notación decimal.

Si *expr*es nulo, entonces esta función retorna NULL.

Esta función no soporta CLOBdatos directamente. Sin embargo, CLOBs se puede pasar como argumentos a través de la conversión de datos implícita.

**Ver también:**

["Reglas de comparación de tipos de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#i55214) para más información

**Ejemplos**

Los siguientes ejemplos muestran cómo extraer información de volcado de una expresión de cadena y una columna:

SELECT DUMP('abc', 1016)

FROM DUAL;

DUMP('ABC',1016)

------------------------------------------

Typ=96 Len=3 CharacterSet=WE8DEC: 61,62,63

SELECT DUMP(last\_name, 8, 3, 2) "OCTAL"

FROM employees

WHERE last\_name = 'Hunold'

ORDER BY employee\_id;

OCTAL

-------------------------------------------------------------------

Typ=1 Len=6: 156,157

SELECT DUMP(last\_name, 10, 3, 2) "ASCII"

FROM employees

WHERE last\_name = 'Hunold'

ORDER BY employee\_id;

ASCII

--------------------------------------------------------------------

Typ=1 Len=6: 110,111

## EMPTY\_BLOB, EMPTY\_CLOB

**Sintaxis**

***empty\_LOB* :: =**

A continuación se encuentra la descripción de empty_lob.gif.  
[Descripción de la ilustración '' empty\_lob.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/empty_lob.htm)

**Propósito**

EMPTY\_BLOBy EMPTY\_CLOBdevuelve un localizador de LOB vacío que se puede usar para inicializar una variable de LOB o, en una declaración INSERTo UPDATE, para inicializar una columna o atributo de LOB a EMPTY. EMPTYsignifica que el LOB se inicializa, pero no se rellena con datos.

**Nota:**

Un LOB vacío no es lo mismo que un LOB nulo, y un vacío CLOBno es lo mismo que un LOB que contiene una cadena de 0 de longitud. Para obtener más información, consulte [*la Guía del desarrollador de Oracle Database SecureFiles y Large Objects*](https://docs.oracle.com/cd/E11882_01/appdev.112/e18294/adlob_working.htm#ADLOB002) .

**Restricción en los localizadores de LOB**No puede utilizar el localizador devuelto de esta función como un parámetro para el DBMS\_LOBpaquete o el OCI.

**Ejemplos**

El siguiente ejemplo inicializa la ad\_photocolumna de la pm.print\_mediatabla de ejemplo para EMPTY:

UPDATE print\_media

SET ad\_photo = EMPTY\_BLOB();

## EXISTSNODE

**Nota:**

La EXISTSNODEfunción está en desuso. Todavía es compatible con la compatibilidad con versiones anteriores. Sin embargo, Oracle recomienda que use la XMLEXISTSfunción en su lugar. Ver [XMLEXISTS](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions243.htm#CIHDEFCD) para más información.

**Sintaxis**

A continuación se encuentra la descripción de existingnode.gif  
[Descripción de la ilustración '' existirnode.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/existsnode.htm)

**Propósito**

EXISTSNODEdetermina si el recorrido de un documento XML mediante una ruta especificada da como resultado cualquier nodo. Toma como argumentos la XMLTypeinstancia que contiene un documento XML y una VARCHAR2cadena XPath que designa una ruta. La opción *namespace\_string*debe resolverse en un VARCHAR2valor que especifique una asignación predeterminada o una asignación de espacio de nombres para los prefijos, que Oracle Database utiliza cuando evalúa las expresiones XPath.

El *namespace\_string*argumento es por defecto el espacio de nombres del elemento raíz. Si se refiere a algún subelemento en *Xpath\_string*, entonces debe especificar *namespace\_string*, y debe especificar el prefijo "who" en ambos argumentos.

**Ver también:**

["Uso de XML en declaraciones SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/ap_examples002.htm#i686084) para ejemplos que especifican *namespace\_string*y usan el prefijo "who".

El valor de retorno es NUMBER:

* 0 si no quedan nodos después de aplicar el recorrido de XPath en el documento
* 1 si quedan nodos

**Ejemplos**

El siguiente ejemplo prueba la existencia del /Warehouse/Docknodo en la ruta XML de la warehouse\_speccolumna de la tabla de muestra oe.warehouses:

SELECT warehouse\_id, warehouse\_name

FROM warehouses

WHERE EXISTSNODE(warehouse\_spec, '/Warehouse/Docks') = 1

ORDER BY warehouse\_id;

WAREHOUSE\_ID WAREHOUSE\_NAME

------------ -----------------------------------

1 Southlake, Texas

2 San Francisco

4 Seattle, Washington

## EXP

**Sintaxis**

Descripción de exp.gif sigue  
[Descripción de la ilustración '' exp.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/exp.htm)

**Propósito**

EXPRetornos eelevados a la *n*potencia th, donde e= 2.71828183 .... La función devuelve un valor del mismo tipo que el argumento.

Esta función toma como argumento cualquier tipo de datos numéricos o cualquier tipo de datos no numéricos que puedan convertirse implícitamente en un tipo de datos numéricos. Si el argumento es BINARY\_FLOAT, entonces la función retorna BINARY\_DOUBLE. De lo contrario, la función devuelve el mismo tipo de datos numéricos que el argumento.

**Ver también:**

[Tabla 3-10, "Matriz de conversión de tipo implícita"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements002.htm#g195937) para obtener más información sobre la conversión implícita

**Ejemplos**

El siguiente ejemplo vuelve ea la 4ta potencia:

SELECT EXP(4) "e to the 4th power"

FROM DUAL;

e to the 4th power

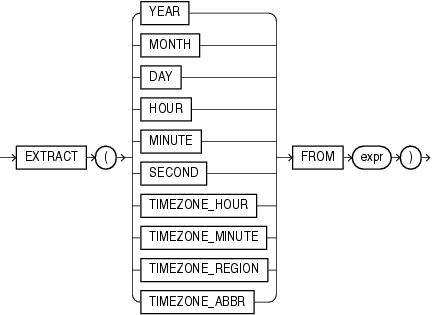
------------------

54.59815

## EXTRACT (datetime)

**Sintaxis**

***extract\_datetime* :: =**

  
[Descripción de la ilustración '' extract\_datetime.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/extract_datetime.htm)

**Propósito**

EXTRACTextrae y devuelve el valor de un campo de fecha y hora especificado desde una expresión de fecha y hora o intervalo. El *expr*puede ser cualquier expresión que se evalúe como una fecha y hora o el tipo de datos de intervalo compatible con el campo solicitado:

* Si YEARo MONTHse solicita, a continuación, *expr*se debe evaluar como una expresión de tipo de datos DATE, TIMESTAMP, TIMESTAMPWITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE, o INTERVAL YEAR TO MONTH.
* Si DAYse solicita, a continuación, *expr*se debe evaluar como una expresión de tipo de datos DATE, TIMESTAMP, TIMESTAMP WITH TIMEZONE, TIMESTAMP WITH LOCAL TIME ZONE, o INTERVAL DAY TO SECOND.
* Si HOUR, MINUTEo SECONDse solicita, a continuación, *expr*se debe evaluar como una expresión de tipo de datos TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE, o INTERVAL DAY TO SECOND. DATEno es válido aquí, porque Oracle Database lo trata como DATEun tipo de datos ANSI , que no tiene campos de tiempo.
* Si TIMEZONE\_HOUR, TIMEZONE\_MINUTE, TIMEZONE\_ABBR, TIMEZONE\_REGION, o TIMEZONE\_OFFSETse solicita, a continuación, *expr*se debe evaluar como una expresión de tipo de datos TIMESTAMP WITH TIME ZONEo TIMESTAMP WITH LOCAL TIME ZONE.

EXTRACTse interpreta *expr*como un tipo de datos de fecha y hora ANSI. Por ejemplo, se EXTRACTtrata DATEno como Oracle heredado DATEsino como ANSI DATE, sin elementos de tiempo. Por lo tanto, se puede extraer sólo YEAR, MONTHy DAYde un DATEvalor. Asimismo, puede extraer TIMEZONE\_HOURy TIMEZONE\_MINUTEsolo del TIMESTAMP WITH TIME ZONEtipo de datos.

Cuando especifica TIMEZONE\_REGIONo TIMEZONE\_ABBR(abreviatura), el valor devuelto es una VARCHAR2cadena que contiene el nombre o la abreviatura de la zona horaria correspondiente. Cuando especifica cualquiera de los otros campos de fecha y hora, el valor devuelto es un valor entero de NUMBERtipo de datos que representa el valor de fecha y hora en el calendario gregoriano. Cuando se extrae de una fecha y hora con un valor de zona horaria, el valor devuelto está en UTC. Para obtener una lista de los nombres de las regiones de zona horaria y sus abreviaturas correspondientes, consulte la V$TIMEZONE\_NAMESvista de rendimiento dinámico.

Esta función puede ser muy útil para manipular valores de campo de fecha y hora en tablas muy grandes, como se muestra en el primer ejemplo a continuación.

**Nota:**

La función de horario de verano necesita nombres de región de zona horaria. Estos nombres se almacenan en dos tipos de archivos de zona horaria: uno grande y otro pequeño. Uno de estos archivos es el archivo predeterminado, dependiendo de su entorno y del lanzamiento de la base de datos Oracle que esté utilizando. Para obtener más información sobre los archivos y nombres de zona horaria, consulte la [*Guía de soporte de globalización de bases de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e10729/applocaledata.htm#NLSPG014) .

Algunas combinaciones del campo de fecha y hora y la expresión de valor de fecha y tiempo dan como resultado ambigüedad. En estos casos, la base de datos Oracle devuelve UNKNOWN(consulte los ejemplos a continuación para obtener información adicional).

**Ver también:**

* [*Guía de soporte de globalización de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e10729/applocaledata.htm#NLSPG0141) para obtener una lista completa de los nombres de región de zona horaria en ambos archivos
* ["Fecha / Aritmética de intervalos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements001.htm#i48042) para una descripción de *datetime\_value\_expr*y*interval\_value\_expr*
* [*Referencia de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e40402/dynviews_part.htm#REFRN003) para obtener información sobre las vistas de rendimiento dinámico

**Ejemplos**

El siguiente ejemplo devuelve de la oe.orderstabla el número de pedidos realizados en cada mes:

SELECT EXTRACT(month FROM order\_date) "Month", COUNT(order\_date) "No. of Orders"

FROM orders

GROUP BY EXTRACT(month FROM order\_date)

ORDER BY "No. of Orders" DESC, "Month";

Month No. of Orders

---------- -------------

11 15

6 14

7 14

3 11

5 10

2 9

9 9

8 7

10 6

1 5

12 4

4 1

12 rows selected.

El siguiente ejemplo retorna el año 1998.

SELECT EXTRACT(YEAR FROM DATE '1998-03-07')

FROM DUAL;

EXTRACT(YEARFROMDATE'1998-03-07')

---------------------------------

1998

El siguiente ejemplo selecciona de la tabla de muestra a hr.employeestodos los empleados que fueron contratados después de 2007:

SELECT last\_name, employee\_id, hire\_date

FROM employees

WHERE EXTRACT(YEAR FROM TO\_DATE(hire\_date, 'DD-MON-RR')) > 2007

ORDER BY hire\_date;

LAST\_NAME EMPLOYEE\_ID HIRE\_DATE

------------------------- ----------- ---------

Johnson 179 04-JAN-08

Grant 199 13-JAN-08

Marvins 164 24-JAN-08

. . .

El siguiente ejemplo resulta en ambigüedad, por lo que Oracle devuelve UNKNOWN:

SELECT EXTRACT(TIMEZONE\_REGION FROM TIMESTAMP '1999-01-01 10:00:00 -08:00')

FROM DUAL;

EXTRACT(TIMEZONE\_REGIONFROMTIMESTAMP'1999-01-0110:00:00-08:00')

----------------------------------------------------------------

UNKNOWN

La ambigüedad surge debido a que el desplazamiento numérico de la zona horaria se proporciona en la expresión, y ese desplazamiento numérico se puede asignar a más de un nombre de región de zona horaria.

## EXTRACT (XML)

**Nota:**

La función EXTRACT(XML) está en desuso. Todavía es compatible con la compatibilidad con versiones anteriores. Sin embargo, Oracle recomienda que use la XMLQUERYfunción en su lugar. Ver [XMLQUERY](https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions249.htm#CIHHFFGG) para más información.

**Sintaxis**

***extract\_xml* :: =**

La descripción de extract_xml.gif sigue  
[Descripción de la ilustración '' extract\_xml.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/extract_xml.htm)

**Propósito**

EXTRACT(XML) es similar a la EXISTSNODEfunción. Aplica una VARCHAR2cadena XPath y devuelve una XMLTypeinstancia que contiene un fragmento XML. Puede especificar un absoluto *XPath\_string*con una barra inicial o un pariente *XPath\_string*al omitir la barra inicial. Si omite la barra inicial, el contexto de la ruta relativa se establece de forma predeterminada en el nodo raíz. La opción *namespace\_string*es obligatoria si el XML que está manejando usa un prefijo de espacio de nombres. Este argumento debe resolverse en un VARCHAR2valor que especifique una asignación predeterminada o una asignación de espacio de nombres para los prefijos, que Oracle Database utiliza cuando evalúa las expresiones de XPath.

**Ejemplos**

El siguiente ejemplo extrae el valor del /Warehouse/Docknodo de la ruta XML de la warehouse\_speccolumna en la tabla de muestra oe.warehouses:

SELECT warehouse\_name,

EXTRACT(warehouse\_spec, '/Warehouse/Docks') "Number of Docks"

FROM warehouses

WHERE warehouse\_spec IS NOT NULL

ORDER BY warehouse\_name;

WAREHOUSE\_NAME Number of Docks

------------------------- -------------------------

New Jersey

San Francisco <Docks>1</Docks>

Seattle, Washington <Docks>3</Docks>

Southlake, Texas <Docks>2</Docks>

# ONDICIONALES

Los dos métodos que se utilizan para implantar el procesamiento condicional (lógica **IF-THEN- ELSE**) en la sentencia SQL son la expresión **CASE** y la función **DECODE**.

**Nota:** la expresión **CASE** cumple con SQL ANSI. La función **DECODE** es específica de la sintaxis de *Oracle*.

## CASE

CASE expr WHEN comparison\_expr1 THEN return\_expr1

WHEN comparison\_expr2 THEN return\_expr2

WHEN comparison\_exprn THEN return\_exprn

ELSE else\_expr];



**Expresión CASE**

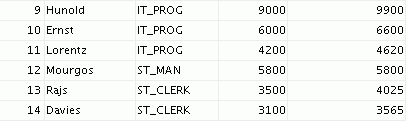
Facilita las consultas condicionales realizando el trabajo de una sentencia IF-THEN-ELSE:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Las expresiones CASE** permiten utilizar la lógica **IF-THEN-ELSE** en las sentencias SQL sin tener que llamar a los procedimientos.

En una expresión **CASE** simple, el servidor de Oracle busca el primer par **WHEN ... THEN** para el que expr es igual a comparison\_expr y devuelve return\_expr. Si ninguno de los pares **WHEN ... THEN** cumple con esta condición y existe una cláusula **ELSE**, el servidor de Oracle devuelve else\_expr. De lo contrario, el servidor de Oracle devuelve un valor nulo. No puede especificar el valor literal **NULL** para todas las expresiones return\_expr y else\_expr.

Las expresiones expr y comparison\_expr deben ser del mismo tipo de dato, que puede ser **CHAR**, **VARCHAR2**, **NCHAR** o **NVARCHAR2**. Todos los valores de retorno (return\_expr) deben ser del mismo tipo de dato.



SELECT last\_name,

job\_id, salary,

CASE job\_id WHEN 'IT\_PROG' THEN 1.10\*salary

WHEN 'ST\_CLERK' THEN 1.15\*salary

WHEN 'SA\_REP' THEN 1.20\*salary

ELSE salary "REVISED\_SALARY"

END

FROM employees

**Uso de la Expresión CASE**

Facilita las consultas condicionales realizando el trabajo de una sentencia IF-THEN-ELSE:

**Uso de la Expresión CASE**

En la sentencia SQL de la diapositiva, se descodifica el valor de JOB\_ID. Si JOB\_ID es IT\_PROG, el aumento de salario es del 10%; si JOB\_ID es ST\_CLERK, el aumento de salario es del 15%; si JOB\_ID es SA\_REP, el aumento del salario es del 20%. Para el resto de otros roles de cargo, no hay ningún aumento de salario.

La misma sentencia se puede escribir con la función **DECODE**.

El siguiente código es un ejemplo de la expresión buscada **CASE**. En una expresión **CASE** buscada, la búsqueda se realiza de izquierda a derecha hasta que se encuentra una coincidencia de la condición mostrada y, a continuación, se devuelve la expresión de retorno. Si no se encuentra ninguna condición verdadera y existe una cláusula **ELSE**, se devuelve la expresión de retorno en la cláusula **ELSE**; de lo contrario, se devuelve **NULL**.

SELECT last\_name,

salary,

(CASE WHEN salary<5000 THEN 'Low'

WHEN salary<10000 THEN 'Medium'

WHEN salary<20000 THEN 'Good' ELSE 'Excellent'

END) qualified\_salary

FROM employees;

## DECODE

DECODE(col|expression, search1, result1

[, search2, result2,...,] [, default])



**Función DECODE**

Facilita las consultas condicionales realizando el trabajo de una expresión CASE o una sentencia IF-THEN-ELSE:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

La función **DECODE** decodifica una expresión de una forma similar a la lógica **IF-THEN-ELSE** que se utiliza en varios idiomas. La función **DECODE** descodifica *expression*  después de compararla con cada valor *search*. Si la expresión es la misma que *search*, se devuelve *result*.

Si se omite el valor por defecto, se devuelve un valor nulo donde un valor de búsqueda no coincida con ninguno de los valores de resultado.



SELECT last\_name,

job\_id,

salary,

DECODE( job\_id, 'IT\_PROG', 1.10\*salary,'ST\_CLERK',

1.15\*salary, 'SA\_REP',

1.20\*salary, salary) REVISED\_SALARY



**Uso de la Función DECODE**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Uso de la Función DECODE**

En la sentencia SQL de la diapositiva, se prueba el valor de JOB\_ID. Si JOB\_ID es IT\_PROG, el aumento de salario es del 10%; si JOB\_ID es ST\_CLERK, el aumento de salario es del 15%; si JOB\_ID es SA\_REP, el aumento del salario es del 20%. Para el resto de otros roles de cargo, no hay ningún aumento de salario.

La misma sentencia se puede expresar en pseudocódigo como una sentencia **IF-THEN-ELSE**:

IF job\_id = 'IT\_PROG'

THEN salary = salary\*1.10

IF job\_id = 'ST\_CLERK'

THEN salary = salary\*1.15

IF job\_id = 'SA\_REP'

THEN salary = salary\*1.20

ELSE salary = salary

SELECT last\_name,

salary,

DECODE (TRUNC(salary/2000, 0),

0, 0.00,

1, 0.09,

2, 0.20,

3, 0.30,

4, 0.40,

5, 0.42,

6, 0.44,

0.45) TAX\_RATE

FROM employees

WHERE department\_id = 80;



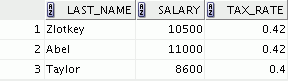
**Uso de la Función DECODE**

Mostrar el impuesto aplicable para cada empleado del departamento 80:

Esta diapositiva muestra otro ejemplo que utiliza la función **DECODE**. En este ejemplo, puede determinar los impuestos para cada empleado del departamento 80 según su salario mensual.

Los impuestos son los siguientes:

|  |  |
| --- | --- |
| *Rango de Salario Mensual* | *Impuesto* |
| $0,00–1.999,99 | 00% |
| $2.000,00–3.999,99 | 09% |
| $4.000,00–5.999,99 | 20% |
| $6.000,00–7.999,99 | 30% |
| $8.000,00–9.999,99 | 40% |
| $10.000,00–11.999,99 | 42% |
| $12.200,00–13.999,99 | 44% |
| 14.000,00 o superior | 45% |



# CONVERSIONES



**Tipo de dato explícito conversión**

**Tipo de dato implícito conversión**

**Tipo de Dato conversión**

**Funciones de Conversión**

# Funciones de Conversión

Además de los tipos de dato de Oracle, las columnas de las tablas de una BBDD Oracle se pueden definir utilizando los tipos de dato American National Standards Institute (ANSI), DB2 y SQL/DS. Sin embargo, el servidor de Oracle convierte internamente esos tipos de dato a tipos de dato de Oracle.

En algunos casos, el servidor de Oracle recibe datos de un tipo de dato cuando espera datos de un tipo de dato diferente. Cuando esto ocurre, el servidor de Oracle puede convertir automáticamente los datos al tipo de dato esperado. Esta conversión de tipo de dato puede realizarla el servidor de Oracle *implícitamente* o el usuario *explícitamente*.

Las conversiones de tipo de dato funcionan según las reglas explicadas en las siguientes diapositivas.

Las conversiones de tipo de dato explícitas se realizan utilizando las funciones de conversión. Las funciones de conversión sirven para convertir los valores de un tipo de dato a otro. Generalmente, el formato de los nombres de funciones sigue la convención *data type*  TO *data type*. El primer tipo de dato es el tipo de dato de entrada; el segundo tipo de dato es la salida.

**Nota:** aunque la conversión de tipo de dato implícita está disponible, se recomienda que realice una conversión de tipo de dato explícita para asegurar la fiabilidad de las sentencias SQL.

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | | | **De** |
| VARCHAR2 | o | CHAR | NUMBER |
| VARCHAR2 | o | CHAR | DATE |

## Conversión Implícita del Tipo de Dato

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **De** | | |
| NUMBER | VARCHAR2 | o | CHAR |
| DATE | VARCHAR2 | o | CHAR |

En general, el servidor de Oracle utiliza la regla de expresiones cuando se necesita una conversión de tipo de dato. Por ejemplo, la expresión grade = 2 tiene como resultado la conversión implícita del número 2 a la cadena “2” porque el grado es una columna CHAR(2)..

**Nota:** las conversiones de **CHAR** a **NUMBER** se realizan correctamente sólo si la cadena de caracteres representa un número válido.



**TO\_CHAR**

**TO\_CHAR**

**DATE**

**CARÁCTER**

**NUMBER**

**TO\_DATE**

**TO\_NUMBER**

**Conversión Explícita del Tipo de Dato**

## Conversión Explícita del Tipo de Dato

SQL proporciona tres funciones para convertir un valor de un tipo de dato a otro:

|  |  |
| --- | --- |
| **Función** | **Objetivo** |
| **TO\_CHAR**(*number*|*date*,[ *fmt*], *[nlsparams*]) | Convertir un valor de fecha o número a una cadena de caracteres VARCHAR2 con el modelo de formato *fmt*  **Conversión numérica:** el parámetro nlsparams especifica los siguientes caracteres, devueltos por los elementos de formato numérico:   * Carácter decimal * Separador de grupo * Símbolo de la divisa local * Símbolo de la divisa internacional Si se omite el parámetro nlsparams o   cualquier otro parámetro, esta función utiliza para  la sesión los valores de parámetro por defecto. |



**TO\_CHAR**

**TO\_CHAR**

**DATE**

**CARÁCTER**

**NUMBER**

**TO\_DATE**

**TO\_NUMBER**

**Conversión Explícita del Tipo de Dato**

|  |  |
| --- | --- |
| **Función** | **Objetivo** |
| **TO\_CHAR**(*number*|*date*,[  *fmt*], [*nlsparams*]) | **Conversión de fecha:** el parámetro nlsparams especifica el idioma en el que se devuelven las abreviaturas y los nombres del mes y del día. Si se omite este parámetro, esta función utiliza los idiomas de fecha por defecto para la sesión. |
| **TO\_NUMBER**(*char,[fmt], [nlsparams]*) | Convertir una cadena de caracteres que contiene dígitos a un número en el formato especificado por el modelo de formato opcional *fmt*.  El parámetro nlsparams tiene el mismo objetivo en esta función que en la función TO\_CHAR para la conversión numérica. |
| **TO\_DATE**(c*har*,[*fmt*],[*nlspa rams*]) | Convertir una cadena de caracteres que representa una fecha en un valor de fecha según el valor de *fmt* especificado. Si se omite *fmt*, el formato es DD-MON-YY.  El parámetro nlsparams tiene el mismo objetivo en esta función que en la función TO\_CHAR para la conversión de fecha. |

**Nota:** la lista de funciones mencionadas en esta lección incluye sólo algunas de las funciones de conversión disponibles.

TO\_CHAR(date, 'format\_model')



**Uso de la Función TO\_CHAR con Fechas**

El modelo de formato:

* Debe estar entre comillas simples
* Es sensible a mayúsculas/minúsculas
* Puede incluir cualquier elemento de formato de fecha válido
* Tiene un elemento fm para eliminar los espacios en blanco o suprimir ceros iniciales
* Está separado del valor de fecha por una coma

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de la Función TO\_CHAR con Fechas

**TO\_CHAR** convierte un tipo de dato de fecha y hora a un valor de tipo de dato **VARCHAR2** con el formato especificado por*format\_model*. Un modelo de formato es un carácter literal que describe el formato de fecha y hora almacenado en una cadena de caracteres. Por ejemplo, el modelo de formato de fecha y hora '11-Nov-1999' es 'DD-Mon-YYYY'. Puede utilizar la función **TO\_CHAR** para convertir una fecha de su formato por defecto a uno que especifique.

**Instrucciones**

* El modelo de formato debe estar entre comillas simples y es sensible a mayúsculas/minúsculas.
* El modelo de formato puede incluir cualquier elemento de formato de fecha válido. Asegúrese de separar el valor de fecha del modelo de formato con una coma.
* Los nombres de los días y meses en la salida se rellenan automáticamente con espacios en blanco.
* Para eliminar los espacios en blanco o suprimir los ceros iniciales, utilice el elemento *fm* de modo de rellen.

SELECT employee\_id,

TO\_CHAR(hire\_date, 'MM/YY') Month\_Hired

FROM employees

WHERE last\_name = 'Higgins';



**Elementos del Modelo de Formato de Fecha**

|  |  |
| --- | --- |
| **Elemento** | **Resultado** |
| YYYY | Año completo en números |
| YEAR | Año en letra (en inglés) |
| MM | Valor de dos dígitos del mes |
| MONTH | Nombre completo del mes |
| MON | Abreviatura de tres letras del mes |
| DY | Abreviatura de tres letras del día de la semana |
| DAY | Nombre completo del día de la semana |
| DD | Día numérico del mes |

**Elementos de Formato de Ejemplo de Formatos de Fecha Válidos**

|  |  |
| --- | --- |
| Elemento | Descripción |
| SCC o bien CC | Siglo; el servidor agrega como prefijo el signo - a los años A.C. |
| Años en fechas YYYY o bien SYYYY | Año; el servidor agrega como prefijo el signo - a los años A.C. |
| YYY o bien YY o bien Y | Indica el último, los tres últimos o los dos últimos dígitos de un año |
| Y,YYY | Año con una coma en esta posición |
| IYYY, IYY, IY, I | Año de cuatro, tres, dos o un dígito basado en el estándar ISO |
| SYEAR o bien YEAR | Año en letra; el servidor agrega como prefijo el signo - a los años A.C. |
| BC o bien AD | Indica un año A.C. o D.C. |
| B.C. o bien A.D. | Indica un año A.C. o D.C. que utiliza períodos |
| Q | Trimestre del año |
| MM | Mes: valor de dos dígitos |
| MONTH | Nombre del mes rellenado con espacios en blanco hasta una longitud de nueve caracteres |
| MON | Nombre del mes, abreviatura de tres letras |
| RM | Número romano del mes |
| WW o bien W | Semana del año o del mes |
| DDD o bien DD o bien D | Día del año, mes o semana |
| DAY | Nombre del día rellenado con espacios en blanco hasta una longitud de nueve caracteres |
| DY | Nombre del día, abreviatura de tres letras |
| J | Día juliano actual, número de días desde el 31 de diciembre de 4713 a.C. |
| IW | Semanas del año según el estándar (1 a 53) |



**Elementos del Modelo de Formato de Fecha**

* Los elementos de tiempo formatean la parte de la hora de la fecha:
* Agregan cadenas de caracteres entre comillas dobles:
* El número se agrega como sufijo de los números en letra:

**fourteenth**

**ddspth**

**15:45:32 PM**

**HH24:MI:SS AM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DD** | **"of"** | **MONTH** | **12** | **of** | **OCTOBER** |

Utilice los formatos de las siguientes tablas para mostrar la información de tiempo y los literales y para cambiar los numerales a números en letra.

|  |  |
| --- | --- |
| Elemento | Descripción |
| AM o PM | Indicador de meridiano |
| A.M. o P.M. | Indicador de meridiano con puntos |
| HH o HH12 o HH24 | Hora del día, hora (1–12) u hora (0–23) |
| MI | Minuto (0–59) |
| SS | Segundo (0–59) |
| SSSSS | Segundos después de la media noche (0–86399) |

Otros Formatos

|  |  |
| --- | --- |
| Elemento | Descripción |
| / . , | La puntuación se reproduce en el resultado. |
| “of the” | La cadena entre comillas se reproduce en el resultado. |

**Especificación de Sufijos para Influenciar la Visualización de Números**

|  |  |
| --- | --- |
| Elemento | Descripción |
| TH | Número ordinal (por ejemplo, DDTH para 4TH) |
| SP | Número en letra (por ejemplo, DDSP para FOUR) |
| SPTH or THSP | Números ordinales en letra (por ejemplo, DDSPTH para FOURTH) |

SELECT last\_name,

TO\_CHAR(hire\_date,

'fmDD Month YYYY') AS HIREDATE



**Uso de la Función TO\_CHAR con Fechas**

**…**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

La sentencia SQL de la diapositiva muestra los apellidos y fechas de contratación de todos los empleados. La fecha de contratación aparece como 17 de junio de 1987.

**Ejemplo:**

Modifique el ejemplo de la diapositiva para mostrar las fechas en un formato que aparezca como “Diecisiete de junio de 1987 12:00:00 AM.”

SELECT last\_name,

TO\_CHAR(hire\_date,

'fmDdspth "of" Month YYYY fmHH:MI:SS AM') HIREDATE

FROM employees;



…

Observe que el mes sigue el modelo de formato especificado; es decir, la primera letra está en mayúscula y el resto en minúsculas.



**Uso de la Función TO\_CHAR con Números**

Éstos son algunos de los elementos de formato que puede utilizar con la función TO\_CHAR para mostrar un valor de número como un carácter:

TO\_CHAR(number, 'format\_model')

|  |  |
| --- | --- |
| **Elemento** | **Resultado** |
| 9 | Representa un número |
| 0 | Fuerza para que aparezca un cero |
| $ | Coloca un signo de dólar flotante |
| L | Utiliza el símbolo de divisa local flotante |
| . | Imprime un punto decimal |
| , | Imprime una coma como indicador de miles |

### Uso de la Función TO\_CHAR con Números

Al trabajar con valores de número como cadenas de caracteres, debe convertir dichos números al tipo de dato de carácter utilizando la función **TO\_CHAR** que traduce un valor del tipo de dato **NUMBER** a un tipo de dato **VARCHAR2**. Esta técnica es muy útil con la concatenación.

Si está convirtiendo un número al tipo de dato de caracteres, puede utilizar los siguientes elementos de formato:

|  |  |  |  |
| --- | --- | --- | --- |
| **Elemento** | **Descripción** | **Ejemplo** | **Resultado** |
| 9 | Posición numérica (número de 9 que determinan el ancho de la visualización) | 999999 | 1234 |
| 0 | Muestra los ceros iniciales | 099999 | 001234 |
| $ | Signo de dólar flotante | $999999 | $1234 |
| L | Símbolo de divisa local flotante | L999999 | FF1234 |
| D | Devuelve el carácter decimal en la posición especificada. El valor por defecto es un punto (.) | 99D99 | 99.99 |
| . | Punto decimal en la posición especificada | 999999.99 | 1234.00 |
| G | Devuelve un separador de grupo en la posición especificada. Puede especificar varios separadores de grupo en un modelo de formato de número. | 9,999 | 9G999 |
| , | Coma en la posición especificada | 999,999 | 1,234 |
| MI | Signos menos a la derecha (valores negativos) | 999999MI | 1234- |
| PR | Introduce entre paréntesis los números negativos | 999999PR | <1234> |
| EEEE | Notación científica (el formato debe especificar cuatro E) | 99.999EEEE | 1.234E+03 |
| U | Devuelve la divisa dual "Euro" (u otra) en la posición especificada | U9999 | €1234 |
| V | Multiplica por 10 *n* veces (n = número de 9 después de V) | 9999V99 | 123400 |
| S | Devuelve un valor negativo o positivo | S9999 | -1234 or  +1234 |
| B | Muestra los valores cero como espacios en blanco, no como 0 | B9999.99 | 1234.00 |

SELECT TO\_CHAR(salary, '$99,999.00') SALARY

FROM employees

WHERE last\_name = 'Ernst';

* l servidor de Oracle muestra una cadena de signos de número (#) en lugar de un número completo cuyos dígitos exceden el número de dígitos proporcionado en el modelo de formato.
* El servidor de Oracle redondea el valor decimal almacenado al número de decimales proporcionado en el modelo de formato.

TO\_DATE(char[, 'format\_model'])

TO\_NUMBER(char[, 'format\_model'])



**Uso de Funciones TO\_NUMBER y TO\_DATE**

* Convertir una cadena de caracteres a un formato de número que utiliza la función TO\_NUMBER:
* Convertir una cadena de caracteres a un formato de fecha que utiliza la función TO\_DATE:
* Estas funciones tienen un modificador fx. Este modificador especifica la coincidencia exacta para el argumento de carácter y el modelo de formato de fecha de una función TO\_DATE.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Uso de Funciones TO\_NUMBER y TO\_DATE

Puede que desee convertir una cadena de caracteres a un número o a una fecha. Para realizar esta tarea, utilice las funcione **TO\_NUMBER** o **TO\_DATE**. El modelo de formato que seleccione está basado en los elementos de formato demostrados anteriormente.

El modificador fx especifica la coincidencia exacta para el argumento de carácter y el modelo de formato de fecha de una función **TO\_DATE**:

* La puntuación y el texto entre comillas del argumento de carácter debe coincidir exactamente (excepto en las mayúsculas/minúsculas) con las partes correspondientes del modelo de formato.
* El argumento de carácter no puede tener espacios en blanco . Sin fx, el servidor de Oracle ignora los espacios en blanco adicionales.
* Los datos numéricos del argumento de carácter deben tener el mismo número de dígitos que el elemento correspondiente en el modelo de formato. Sin fx, los números del argumento de carácter no pueden omitir los ceros iniciales.

**Ejemplo:**

Muestre el nombre y fecha de contratación de todos los empleados que empezaron a trabajar el 24 de mayo de 1999. En el siguiente ejemplo, hay dos espacios después del mes *May* y antes del número. Ya que se utiliza fx, se necesita una coincidencia exacta y los espacios después de la palabra *May* no se reconocerán:

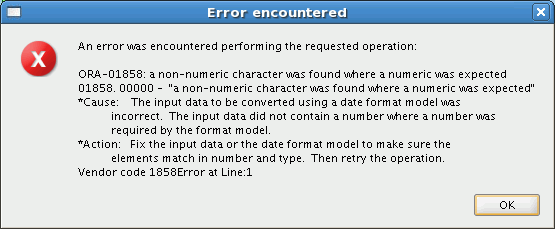
SELECT last\_name,

hire\_date

FROM employees

WHERE hire\_date = TO\_DATE('May 24, 1999', 'fxMonth DD, YYYY');

La salida del error resultante debe tener el siguiente aspecto:



Para ver la salida, corrija la consulta suprimiendo el espacio adicional entre "May" y "24".

SELECT last\_name,

hire\_date

FROM employees

WHERE hire\_date = TO\_DATE('May 24, 1999', 'fxMonth DD, YYYY');



SELECT last\_name,

TO\_CHAR(hire\_date, 'DD-Mon-YYYY')

FROM employees

WHERE hire\_date < TO\_DATE('01-Jan-90','DD-Mon-RR');



**Uso de las Funciones TO\_CHAR y TO\_DATE**

**con el Formato de Fecha RR**

Para buscar los empleados contratados antes de 1990, utilice el formato de fecha RR, que produce los mismos resultados si se ejecutara el comando en 1999 o en la actualidad:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Uso de las Funciones TO\_CHAR y TO\_DATE con el Formato de Fecha RR**

Para buscar los empleados contratados antes de 1990, se puede utilizar el formato RR. Debido a que el año actual es mayor que 1999, el formato RR interpreta la parte del año de la fecha de 1950 a 1999. Por otro lado, en el siguiente comando no se selecciona ninguna fila porque el formato YY interpreta la parte del año de la fecha en el siglo actual (2090).

SELECT last\_name,

TO\_CHAR(hire\_date, 'DD-Mon-yyyy')

FROM employees

WHERE TO\_DATE(hire\_date, 'DD-Mon-yy') < '01-Jan-1990';



## FUNCIONES GENERALES



**Funciones Generales**

Las siguientes funciones funcionan con cualquier tipo de dato y pertenecen al uso de valores nulos:

* NVL (expr1, expr2)
* NVL2 (expr1, expr2, expr3)
* NULLIF (expr1, expr2)
* COALESCE (expr1, expr2, ..., exprn)

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Funciones Generales

Estas funciones funcionan con cualquier tipo de dato y pertenecen al uso de valores nulos en la lista de expresiones.

|  |  |
| --- | --- |
| Función | Descripción |
| NVL | Convierte un valor nulo en un valor real |
| NVL2 | Si expr1 no es nulo, NVL2 devuelve expr2. Si expr1 es nulo, NVL2 devuelve expr3. El argumento expr1 puede tener cualquier tipo de dato. |
| NULLIF | Compara dos expresiones y devuelve un valor nulo si son iguales; si no son iguales, devuelve la primera expresión |
| COALESCE | Devuelve la primera expresión no nula en la lista de expresiones |



**Función NVL**

Convierte un valor nulo a un valor real:

* Los tipos de dato que se pueden utilizar son fecha, carácter y número.
* Los tipos de dato deben coincidir con:
  + NVL(commission\_pct,0)
  + NVL(hire\_date,'01-JAN-97')
  + NVL(job\_id,'No Job Yet')

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

#### Función NVL

Para convertir un valor nulo a un valor real, utilice la función **NVL**.

**Sintaxis**

NVL (*expr1*, *expr2*)

En la sintaxis:

* *expr1* es el valor de origen o expresión que puede contener un valor nulo
* *expr2* es el valor de destino para convertir el valor nulo

Puede utilizar la función NVL para convertir cualquier tipo de dato, pero el valor de retorno siempre es el mismo que el tipo de dato de *expr1*.

**Conversiones NVL para Varios Tipos de Dato**

|  |  |
| --- | --- |
| **Tipo de Dato** | **Ejemplo de Conversión** |
| **NUMBER** | NVL(*number\_column*,9) |
| **DATE** | NVL(*date\_column,* '01-JAN-95') |
| **CHAR** or **VARCHAR2** | NVL(*character\_column*, 'Unavailable') |

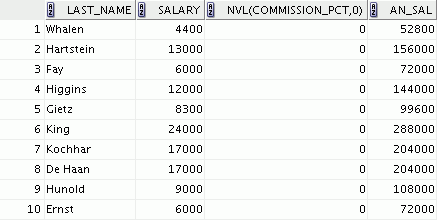
SELECT last\_name,

salary,

NVL(commission\_pct, 0),

(salary\*12)+(salary\*12\*NVL(commission\_pct, 0)) AN\_SAL

FROM employees;



**Uso de la Función NVL NVL**

Para calcular la compensación anual de los empleados, necesita multiplicar el salario mensual por 12 y, a continuación, agregue el porcentaje de comisión al resultado:

SELECT last\_name,

salary,

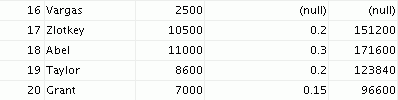
commission\_pct,

(salary\*12) + (salary\*12\*commission\_pct) AN\_SAL

FROM employees;



**…**



Tenga en cuenta que la compensación anual se calcula sólo para los empleados que obtienen una comisión. Si cualquier valor de columna de una expresión es nulo, el resultado es nulo. Para calcular los valores para todos los empleados, puede convertir los valores nulos a un número antes de aplicar el operador aritmético. En el ejemplo de la diapositiva, la función **NVL** se utiliza para convertir valores nulos a cero.

SELECT last\_name,

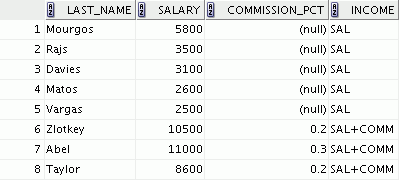
salary,

ommission\_pct,

NVL2(commission\_pct, 'SAL+COMM', 'SAL')income

FROM employees

WHERE department\_id IN (50, 80);



**Uso de la Función NVL2**

**Uso de la Función NVL2**

La función **NVL2** examina la primera expresión. Si la primera expresión no es nula, la función **NVL2** devuelve la segunda expresión. Si la primera expresión es nula, se devolverá la tercera expresión.

**Sintaxis**

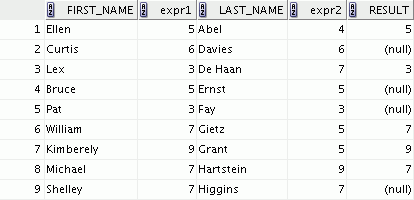
NVL2(*expr1*, *expr2, expr3*)

En la sintaxis:

* *expr1* es el valor de origen o expresión que puede contener un valor nulo
* *expr2* es el valor que se devuelve si *expr1* no es nulo
* *expr3* es el valor que se devuelve si *expr1* es nulo

En el ejemplo de la diapositiva se examina la columna COMMISSION\_PCT. Si se detecta un valor, se devolverá el valor literal de texto de SAL+COMM. Si la columna COMMISSION\_PCT contiene un valor nulo, se devolverá el valor literal de texto SAL.

**Nota:** el argumento *expr1* puede tener cualquier tipo de dato. Los argumentos *expr2* y *expr3* pueden tener cualquier tipo de dato excepto **LONG**.



**Uso de la Función NULLIF**

**…**

SELECT first\_name,

LENGTH(first\_name) "expr1",

last\_name,

LENGTH(last\_name) "expr2",

NULLIF(LENGTH(first\_name),

LENGTH(last\_name)) result

FROM employees;

#### Uso de la Función NULLIF

La función **NULLIF** compara dos expresiones.

**Sintaxis**

NULLIF (expr1, expr2)

En la sintaxis:

* **NULLIF** compara *expr1* y *expr2*. Si son iguales, la función devuelve un valor nulo. Si no, la función devuelve *expr1*. Sin embargo, no puede especificar el literal NULL para *expr1*.

En el ejemplo de la diapositiva, la longitud del nombre en la tabla EMPLOYEES se compara con la longitud del apellido en la tabla EMPLOYEES. Cuando las longitudes de ambos son iguales, se mostrará un valor nulo. Cuando las longitudes no son iguales, se mostrará la longitud del nombre.

**Nota:** la función **NULLIF** es el equivalente lógico de la siguiente expresión **CASE**.

CASE WHEN expr1 = expr2

THEN NULL

ELSE expr1

END



**Uso de la Función COALESCE**

* La ventaja de la función COALESCE con respecto a la función NVL es que la función COALESCE puede obtener múltiples valores alternativos.
* Si la primera expresión no es nula, la función COALESCE devuelve esa expresión; de lo contrario, aplica la función COALESCE de las expresiones restantes.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

#### Uso de la Función COALESCE

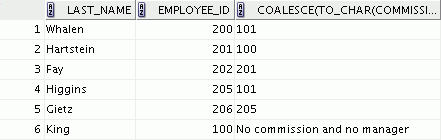
La función **COALESCE** devuelve la primera expresión no nula de la lista.

**Sintaxis**

COALESCE (expr1, expr2, ... exprn)

En la sintaxis:

* + *expr1* devuelve esta expresión si no es nula
  + *expr2* devuelve esta expresión si la primera expresión es nula y ésta no es nula
  + *exprn* devuelve esta expresión si las expresiones anteriores son nulas Tenga en cuenta que todas las expresiones deben ser del mismo tipo de dato.



SELECT last\_name,

employee\_id,

COALESCE(TO\_CHAR(commission\_pct),

TO\_CHAR(manager\_id),

'No commission and no manager')

FROM employees;



**Uso de la Función COALESCE**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

En el ejemplo de la diapositiva se muestra el valor manager\_id si éste no es nulo. Si el valor manager\_id es nulo, se mostrará commission\_pct. Si los valores manager\_id y commission\_pct son nulos, se muestra “No commission and no manager”. Tenga en cuenta que la función **TO\_CHAR** se aplica de modo que todas las expresiones sean del mismo tipo de dato.

**Ejemplo:**

Para los empleados que no perciben ninguna comisión, la organización desea proporcionar un aumento de salario de 2.000 dólares y para los empleados que perciben comisión, la consulta debe calcular el nuevo salario que es igual al salario existente sumado a la comisión.

SELECT last\_name,

salary,

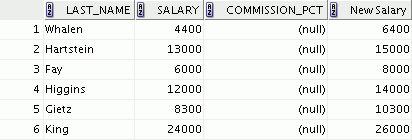
commission\_pct,

COALESCE((salary+(commission\_pct\*salary)),

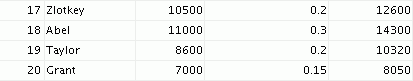
salary+2000, salary) "New Salary"

FROM employees;

**Nota:** examine la salida. Para los empleados que no perciben ninguna comisión, la columna de nuevo salario muestra el salario aumentado en 2.000 dólares y para los empleados que perciben comisión, la columna de nuevo salario muestra la comisión calculada sumada al salario.



**…**



# Consistencia de Lectura



**Consistencia de Lectura**

* La consistencia de lectura garantiza una vista consistente de los datos en todo momento.
* Los cambios realizados por un usuario no entran en conflicto con los cambios realizados por otro usuario.
* La consistencia de lectura garantiza que en los mismos datos:
  + Los lectores no esperen los escritores.
  + Los escritores no esperen a los lectores.
  + Los escritores esperen a los escritores.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Consistencia de Lectura

Los usuarios de la base de datos pueden acceder a la base de datos de dos formas:

* + Operaciones de lectura (sentencia **SELECT**)
  + Operaciones de escritura (sentencias **INSERT, UPDATE, DELETE**) La consistencia de lectura es necesaria para lo siguiente:
  + Que el lector y el escritor de la base de datos tengan garantizada una vista consistente de los datos.
  + Que los lectores no visualicen datos en proceso de cambio.
  + Que los escritores tengan garantizado que los cambios en la base de datos se realizarán de forma consistente.
  + Los cambios realizados por un escritor no entran en conflicto con los cambios realizados por otro escritor.

El objetivo de la consistencia es garantizar que todos los usuarios vean los datos como existían en el momento de la confirmación, antes de iniciar una operación DML.

**Nota:** el mismo usuario se puede conectar a diferentes sesiones. Cada sesión mantiene la consistencia de lectura tal y como se describe anteriormente, incluso si son los mismos usuarios.



**Antes del cambio (datos “antiguos”)**

**Usuario B**

**Datos cambiados**

**y sin modificar**

**Imagen de lectura**

**consistente**

**FROM userA.employees;**

**SELECT \***

**Segmentos de deshacer**

**Bloques de Datos**

**salary = 7000 last\_name = 'Grant';**

**SET WHERE**

**UPDATE employees**

**Usuario A**

**Implementación de Consistencia de Lectura**

Implementación de Consistencia de Lectura

La consistencia de lectura es una implantación automática. Mantiene una copia parcial de la base de datos en los segmentos de deshacer. La imagen de lectura consistente se crea a partir de los datos confirmados de la tabla y de los datos antiguos que se van a cambiar y que aún no se han confirmado del segmento de deshacer.

Al realizar una operación de inserción, actualización o supresión en la base de datos, el servidor de Oracle realiza una copia de los datos antes de cambiarlos y los escribe en un *segmento de deshacer*.

Todos los lectores, excepto el que haya emitido el cambio, seguirán viendo la base de datos como existía antes de que comenzaran los cambios; verán una “instantánea” de los datos del segmento de deshacer.

Antes de confirmar los cambios en la base de datos, sólo el usuario que modifica los datos ve la base de datos con modificaciones. Todos los demás verán la instantánea en el segmento de deshacer. Esto garantiza que los lectores de los datos lean datos consistentes en los que no se esté realizando actualmente ningún cambio.

Al confirmar una sentencia DML, el cambio realizado en la base de datos se hace visible para todos aquéllos que emiten una sentencia **SELECT** *después de* realizar la confirmación. El espacio ocupado por los datos *antiguos* en el archivo de segmentos de deshacer se libera para volver a utilizarlo.

Si se realiza un **ROLLBACK** de la transacción, los cambios se deshacen:

* + La versión original anterior de los datos del segmento de deshacer se vuelve a escribir en la tabla.
  + Todos los usuarios ven la base de datos como existía antes de comenzar la transacción.

# LENGUAJES DE SQL



**Sentencias SQL**

|  |
| --- |
| **SELECT INSERT**  **UPDATE** Lenguaje de Manipulación de Datos (DML)  **DELETE** **MERGE** |
| **CREATE ALTER**  **DROP** Lenguaje de Definición de Datos (DDL)  **RENAME**  **TRUNCATE COMMENT** |
| **GRANT** Lenguaje de Control de Datos (DCL)  **REVOKE** |
| **COMMIT**  **ROLLBACK** Control de Transacciones  **SAVEPOINT** |

**Sentencias SQL**

Sentencias SQL soportadas por el cumplimento de Oracle de los estándares de la industria Oracle Corporation asegura el cumplimiento futuro con la evolución de los estándares mediante la implicación de forma activa del personal clave en los comités de estándares SQL. Los comités aceptados por la industria son ANSI e ISO (International Standards Organization). Tanto ANSI como ISO han aceptado SQL como el lenguaje estándar de las bases de datos relacionales.

|  |  |
| --- | --- |
| **Sentencia** | **Descripción** |
| **SELECT INSERT UPDATE DELETE**  **MERGE** | Recupera datos de la base de datos, introduce nuevas filas, cambia las existentes y elimina las filas no deseadas de las tablas en la base de datos, respectivamente.  Conocidos colectivamente *lenguaje de manipulación de datos* (DML). |
| **CREATE ALTER DROP RENAME TRUNCATE**  **COMMENT** | Configura, cambia y elimina las estructuras de datos de las tablas. Conocidos colectivamente como *lenguaje de definición de datos* (DDL). |
| **GRANT REVOKE** | Proporciona o elimina los derechos de acceso a Oracle Database y a las. estructuras que contiene. |
| **COMMIT ROLLBACK**  **SAVEPOINT** | Gestiona los cambios realizados por las sentencias DML. Los cambios en los datos se pueden agrupar en transacciones lógicas. |

# DML

## SELECT

SELECT \*|{[DISTINCT] *column*|*expression* [*alias*],...}

FROM *table;*



**Sentencia SELECT Básica**

* SELECT identifica las columnas que se van a mostrar.
* FROM identifica la tabla que contiene estas columnas.

**Sentencia SELECT Básica**

En su formato más simple, una sentencia **SELECT** debe incluir lo siguiente:

* Una cláusula **SELECT**, que especifica las columnas que se van a mostrar.
* Una cláusula **FROM**, que identifica la tabla que contiene las columnas que se muestran en la cláusula **SELECT**.

En la sintaxis:

- **SELECT** es una lista de una o más columnas.

- **\*** selecciona todas las columnas.

- **DISTINCT** suprime los duplicados.

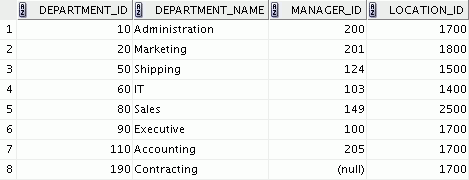
- **column|expression**selecciona la columna o expresión especificada.

*-* ***alias***proporciona diferentes cabeceras de las columnas seleccionadas.

- **FROM *table***especifica la tabla que contiene las columnas.

**Nota:** a lo largo de este curso, las palabras; ***palabra clave***, ***cláusula***y ***sentencia***se utilizan como se describe a continuación:

* Una *palabra clave* hace referencia a un elemento SQL individual (por ejemplo, **SELECT** y **FROM** son palabras clave).
* Una *cláusula* es parte de una sentencia SQL (por ejemplo, **SELECT** employee\_id, last\_name, etc.).
* Una *sentencia* es una combinación de dos o más cláusulas (por ejemplo, SELECT \* FROM employees).



SELECT \* FROM departments;



**Selección de Todas las Columnas**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Selección de Todas las Columnas**

Puede mostrar todas las columnas de datos en una tabla insertando la siguiente palabra clave **SELECT** con un asterisco (**\***). En el ejemplo de la diapositiva, la tabla DEPARTMENTS contiene cuatro columnas: DEPARTMENT\_ID, DEPARTMENT\_NAME, MANAGER\_ID y LOCATION\_ID. La tabla contiene ocho filas, una por cada departamento.

También puede mostrar todas las columnas en la tabla mostrando todas las columnas después de la palabra clave **SELECT**. Por ejemplo, la siguiente sentencia SQL (como el ejemplo de la diapositiva) muestra todas las columnas y filas de la tabla DEPARTMENTS:

SELECT department\_id,

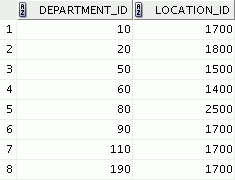
department\_name,

manager\_id,

location\_id

FROM departments;

**Nota:** en SQL Developer, puede introducir la sentencia SQL en una hoja de trabajo de SQL y hacer clic en el icono “Execute Statement” o pulsar [F9] para ejecutar la sentencia. La salida mostrada en la página con separadores Results aparece como se muestra en la diapositiva.



SELECT department\_id,

location\_id

FROM departments;



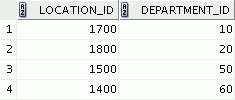
**Selección de Columnas Concretas**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Selección de Columnas Concretas**

También puede utilizar la sentencia **SELECT** para mostrar las columnas concretas de la tabla especificando los nombres de columna, separados por comas. El ejemplo de la diapositiva muestra todos los números de departamento y ubicación de la tabla DEPARTMENTS.

En la cláusula **SELECT**, especifique las columnas que desee, en el orden en el que desee que aparezcan en la salida. Por ejemplo, para mostrar la ubicación antes del número de departamento (de izquierda a derecha), utilice la siguiente sentencia:



SELECT location\_id,

department\_id

FROM departments;



**Escritura de Sentencias SQL**

Las sentencias SQL no son sensibles a mayúsculas/ minúsculas.

* + Las sentencias SQL se pueden introducir en una o más líneas.
  + Las palabras clave no se pueden abreviar o dividir entre líneas.
  + Las cláusulas se suelen colocar en líneas independientes.
  + El sangrado se utiliza para mejorar la legibilidad.
  + En SQL Developer, las sentencias SQL también pueden terminar con un punto y coma (;). Los puntos y comas son necesarios si ejecuta varias sentencias SQL.
  + En SQL\*Plus, debe finalizar cada sentencia SQL con un punto y coma (;).

**Escritura de Sentencias SQL**

Con estas sencillas reglas y directrices, puede construir sentencias válidas que son fáciles de leer y editar:

* Las sentencias SQL no son sensibles a mayúsculas/minúsculas (a menos que se indique).
* Las sentencias SQL se pueden introducir en una o más líneas.
* Las palabras clave no se pueden dividir entre líneas o abreviar.
* Las cláusulas se suelen colocar en líneas independientes para que resulte más fácil su lectura o edición.
* El sangrado se debe utilizar para que sea más fácil de leer el código.
* Normalmente, las palabras clave se introducen en mayúsculas; el resto de palabras, como nombres de tablas y columnas, se introducen en minúsculas.

**Ejecución de Sentencias SQL**

En SQL Developer, haga clic en el icono Run Script o pulse [F5] para ejecutar el comando o los comandos en la hoja de trabajo de SQL. También puede hacer clic en el icono Execute Statement o pulsar [F9] para ejecutar una sentencia SQL en la hoja de trabajo de SQL. El icono Execute Statement ejecuta la sentencia que indica el puntero del mouse en el cuadro Enter SQL Statement mientras que el icono Run Script ejecuta todas las sentencias del cuadro Enter SQL Statement. El icono Execute Statement muestra la salida de la consulta en la página con separadores Results, mientras que el icono Run Script emula la visualización SQL\*Plus y muestra la salida en la página con separadores Script Output.

Mediante SQL\*Plus, termine la sentencia SQL con un punto y coma y, a continuación, pulse la tecla [Intro] para ejecutar el comando.



**Valores por Defecto de Cabeceras de Columna**

* SQL Developer:
  + Alineación de cabeceras por defecto: alineación a la izquierda
  + Visualización de cabeceras por defecto: mayúsculas
* SQL\*Plus:
  + Las cabeceras de columna de carácter y de fecha se alinean a la izquierda
  + Las cabeceras de columna de número se alinean a la derecha
  + Visualización de cabeceras por defecto: mayúsculas

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Valores por Defecto de Cabeceras de Columna**

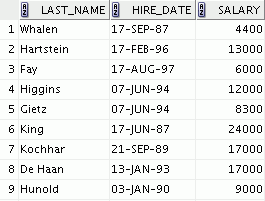
En SQL Developer, las cabeceras de columna se muestran en mayúsculas y alineadas a la izquierda.

SELECT last\_name,

hire\_date,

salary

FROM employees;



Puede sustituir la visualización de cabecera de columna por un alias.



### - DISTINC

Con la cláusula "distinct" se especifica que los registros con ciertos datos duplicados sean obviadas en el resultado. Por ejemplo, queremos conocer todos los autores de los cuales tenemos libros, si utilizamos esta sentencia:

select autor from libros;

Aparecen repetidos. Para obtener la lista de autores sin repetición usamos:

select distinct autor from libros;

También podemos tipear:

select autor from libros

group by autor;

Note que en los tres casos anteriores aparece "null" como un valor para "autor"· Si sólo queremos la lista de autores conocidos, es decir, no queremos incluir "null" en la lista, podemos utilizar la sentencia siguiente:

select distinct autor from libros

where autor is not null;

Para contar los distintos autores, sin considerar el valor "null" usamos:

select count(distinct autor)

from libros;

Note que si contamos los autores sin "distinct", no incluirá los valores "null" pero si los repetidos:

select count(autor)

from libros;

Esta sentencia cuenta los registros que tienen autor.

Podemos combinarla con "where". Por ejemplo, queremos conocer los distintos autores de la editorial "Planeta":

select distinct autor from libros

where editorial='Planeta';

También puede utilizarse con "group by" para contar los diferentes autores por editorial:

select editorial, count(distinct autor)

from libros

group by editorial;

La cláusula "distinct" afecta a todos los campos presentados. Para mostrar los títulos y editoriales de los libros sin repetir títulos ni editoriales, usamos:

select distinct titulo,editorial

from libros

order by titulo;

Note que los registros no están duplicados, aparecen títulos iguales pero con editorial diferente, cada registro es diferente.

Entonces, "**distinct**" elimina registros duplicados.

## - WHERE

**Limitación de Filas con una Selección**

En el ejemplo de la diapositiva, suponga que desea mostrar todos los empleados del departamento 90. Las filas con un valor de 90 de la columna DEPARTMENT\_ID son las únicas que se devuelve. Este método de restricción es la base de la cláusula **WHERE** en SQL.



**Limitación de las Filas Seleccionadas**

* Restringir las filas devueltas al utilizar la cláusula WHERE:
* La cláusula WHERE sigue a la cláusula FROM.

SELECT \*|{[DISTINCT] *column|expression* [*alias*],...}

FROM *table*

[WHERE *condition(s)*];

Puede restringir las filas que devuelve la consulta al utilizar la cláusula **WHERE**. Una cláusula **WHERE** contiene una condición que se debe. Si la condición es verdadera, se devolverá la fila que cumpla con la condición.

En la sintaxis:

**WHERE** restringe la consulta a filas que cumplan con una condición.

*condition* está compuesto por nombres de columna, expresiones,

constantes y un operador de comparación. Una condición especifica una combinación de una o más expresiones y operadores lógicos (booleanos) y devuelve un valor de TRUE, FALSE o UNKNOWN.

La cláusula **WHERE** puede comparar valores en columnas, literales, expresiones aritméticas o funciones. Consta de tres elementos:

* Nombre de la columna
* Condición de comparación
* Nombre de la columna, constante o lista de valores



**Uso de la Cláusula WHERE**

SELECT employee\_id,

last\_name,

job\_id,

department\_id

FROM employees

WHERE department\_id = 90;

En el ejemplo, la sentencia **SELECT** recupera el ID de empleado, apellido, ID de cargo y número de departamento de todos los empleados del departamento 90.

**Nota:** puede utilizar el alias de columna en la cláusula **WHERE**.

### - IS NULL

### - IS NOT NULL

## SELECT FOR UPDATE

SELECT employee\_id,

salary,

commission\_pct,

job\_id

FROM employees

WHERE job\_id = 'SA\_REP' FOR UPDATE

ORDER BY employee\_id;



**Cláusula FOR UPDATE en una Sentencia SELECT**

* Bloquea las filas de la tabla EMPLOYEES en las que

job\_id sea SA\_REP.

* El bloqueo sólo se libera si emite ROLLBACK o COMMIT.
* Si la sentencia SELECT intenta bloquear una fila bloqueada por otro usuario, la base de datos espera a que la fila esté disponible y, a continuación, devuelve los resultados de la sentencia SELECT.

# Cláusula FOR UPDATE en una Sentencia SELECT

Cuando se emite una sentencia **SELECT** en la base de datos para consultar algunos registros, no se coloca ningún bloqueo en las filas seleccionadas. En general, se debe a que el número de registros bloqueados en un momento determinado se mantiene (por defecto) en el mínimo absoluto: sólo se bloquean aquellos registros que se han cambiado, pero que aún no se han confirmado. Aun así, otros usuarios podrán leer dichos registros tal y como aparecían antes del cambio (la “imagen anterior” de los datos). Hay ocasiones, sin embargo, en las que puede que desee bloquear un juego de registros incluso antes de cambiarlos en el programa. Oracle ofrece la cláusula **FOR UPDATE** de la sentencia **SELECT** para realizar este bloqueo.

Cuando emite una sentencia **SELECT...FOR UPDATE**, el sistema de gestión de bases de datos relacionales (RDBMS) obtiene automáticamente los bloqueos a nivel de fila exclusivos de todas las filas identificadas por la sentencia **SELECT**, reteniendo así los registros “sólo para sus cambios”.

Nadie más podrá cambiar ninguno de estos registros hasta que realice una operación **ROLLBACK** o **COMMIT**.

Puede adjuntar la palabra clave opcional **NOWAIT** a la cláusula **FOR UPDATE** para indicar al servidor de Oracle que no espere si otro usuario ha bloqueado la tabla. En este caso, el control volverá inmediatamente al programa o al entorno de SQL Developer para que pueda realizar otro trabajo o simplemente esperar un período de tiempo antes de volver a intentarlo. Sin la cláusula **NOWAIT**, el proceso se bloqueará hasta que la tabla esté disponible, cuando otro usuario libere los bloqueos a través de la emisión un comando **COMMIT** o **ROLLBACK**.

SELECT e.employee\_id,

e.salary,

e.commission\_pct

FROM employees e

JOIN departments d

USING (department\_id)

WHERE job\_id = 'ST\_CLERK‘

AND location\_id = 1500 FOR UPDATE

ORDER BY e.employee\_id;



**Cláusula FOR UPDATE: Ejemplos**

* Puede utilizar la cláusula FOR UPDATE en una sentencia

SELECT en varias tablas.

* Las filas de las tablas EMPLOYEES y DEPARTMENTS están bloqueadas.
* Utilizar FOR UPDATE OF *column\_name* para cualificar la columna que desea cambiar, de modo que sólo se bloqueen las filas de esa tabla específica.

Ejemplos

En el ejemplo de la diapositiva, la sentencia bloquea las filas de la tabla EMPLOYEES con JOB\_ID definido en ST\_CLERK y LOCATION\_ID definido en 1500 y bloquea las filas de la tabla DEPARTMENTS con los departamentos de LOCATION\_ID definidos en 1500.

Puede utilizar **FOR UPDATE OF** *column\_name* para cualificar la columna que desea cambiar. La lista OF de la cláusula **FOR UPDATE** no limita el cambio sólo a aquellas columnas de las filas seleccionadas. Aún se mantienen los bloqueos en todas las filas; si simplemente declara **FOR UPDATE** en la consulta y no incluye una o más columnas después de la palabra clave **OF**, la base de datos bloqueará todas las filas identificadas en todas las tablas incluidas en la cláusula **FROM**.

La siguiente sentencia sólo bloquea las filas de la tabla EMPLOYEES con ST\_CLERK ubicado en LOCATION\_ID 1500. No se bloquea ninguna fila en la tabla DEPARTMENTS:

SELECT e.employee\_id,

e.salary,

e.commission\_pct

FROM employees e

JOIN departments d

USING (department\_id)

WHERE job\_id = 'ST\_CLERK'

AND location\_id = 1500 FOR UPDATE OF e.salary

ORDER BY e.employee\_id;

En el siguiente ejemplo**,** se indica a la base de datos que espere cinco segundos hasta que la fila esté disponible y, a continuación, se devuelve el control al usuario.

SELECT employee\_id,

salary,

commission\_pct,

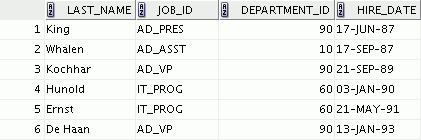
job\_id

FROM employees

WHERE job\_id = 'SA\_REP' FOR UPDATE WAIT 5

ORDER BY employee\_id;

## ORDER BY



**Uso de la Cláusula ORDER BY**

* Ordenar las filas recuperadas con la cláusula ORDER BY:
  + ASC: orden ascendente, valor por defecto
  + DESC: orden descendente
* La cláusula ORDER BY es la última en una sentencia

SELECT:

**…**

SELECT last\_name,

job\_id,

department\_id,

hire\_date

FROM employees

ORDER BY hire\_date;

**Uso de la Cláusula ORDER BY**

El orden de las filas devueltas en un resultado de consulta no está definido. La cláusula **ORDER BY** se puede utilizar para ordenar las filas. Sin embargo, si utiliza la cláusula **ORDER BY**, debe ser la última cláusula de la sentencia SQL. Además, puede especificar una expresión, un alias o una posición de columna como la condición de ordenación.

**Sintaxis**

SELECT *expr*

FROM *Tabla*

[WHERE *condition(s)*]

[ORDER BY {*column*, *expr, numeric\_position*} [ASC|DESC]];

En la sintaxis:

ORDER BY especifica el orden en el que aparecen las filas recuperadas.

ASC ordena las filas en orden ascendente (orden por defecto).

DESC ordena las filas en orden descendente.

Si la cláusula **ORDER** **BY** no se utiliza, el orden no está definido y puede que el servidor de Oracle no recupere dos veces las filas en el mismo orden para la misma consulta. Utilice la cláusula **ORDER BY** para mostrar las filas en un orden específico.

**Nota:** utilice las palabras clave **NULLS FIRST** o **NULLS LAST** para especificar si las filas devueltas que contengan valores nulos deben aparecer en primer o en último lugar en la secuencia de ordenación.

SELECT employee\_id,

last\_name,

salary\*12 annsal

FROM employees

ORDER BY annsal ;

* Ordenar por alias de columna:

SELECT last\_name,

job\_id,

department\_id,

hire\_date

FROM employees

ORDER BY hire\_date;



* Ordenar en orden descendente:

**Ordenación**

**Ordenación**

El orden por defecto es el orden ascendente:

* Los valores numéricos se muestran con los valores más bajos primero (por ejemplo, de 1 a 999).
* Los valores de fecha se muestran con el primer valor en primer lugar (por ejemplo, 01-ENE-92 antes de 01-ENE-95).
* Los valores de caracteres se muestran en orden alfabético (por ejemplo, primero la “A” y por último la “Z”).
* Los valores nulos se muestran al final para las secuencias ascendentes y al principio para las secuencias descendentes.
* Puede ordenar por una columna que no esté en la lista SELECT.

**Ejemplos**

1. Para invertir el orden en el que se muestran las filas, especifique la palabra clave **DESC** después del nombre de columna en la cláusula **ORDER BY**. El ejemplo de la diapositiva ordena el resultado por el empleado contratado más recientemente.
2. También puede utilizar un alias de columna en la cláusula **ORDER BY**. El ejemplo de la diapositiva ordena los datos por salario anual.

**Nota:** la palabra clave **DESC** que se utiliza aquí para la ordenación en orden descendente no se debe confundir con la palabra clave **DESC** utilizada para describir las estructuras de tabla.

* Ordenar por varias columnas:

SELECT last\_name,

department\_id,

salary

FROM employees

ORDER BY department\_id,

salary DESC;

Select last\_name,

job\_id,

department\_id,

hire\_date employees

FROM employees

ORDER BY 3;



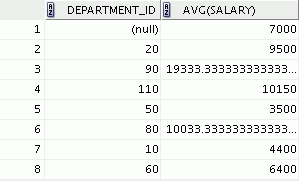
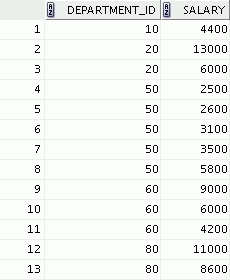
* Ordenar por posición numérica de la columna:

**Ordenación**

**Ejemplos**

1. Puede ordenar los resultados de la consulta especificando la posición numérica de la columna en la cláusula **SELECT**. El ejemplo de la diapositiva ordena el resultado por department\_id, ya que esta columna está en la tercera posición en la cláusula **SELECT**.
2. Puede ordenar los resultados de la consulta por más de una columna. El límite de ordenación es el número de columnas de la tabla determinada. En la cláusula **ORDER BY**, especifique las columnas y separe los nombres de columna con comas. Si desea invertir el orden de una columna, especifique **DESC** después del nombre. El resultado del ejemplo de consulta que se muestra en la diapositiva está ordenado por department\_id en orden ascendente y por salary en orden descendente.

## GROUP BY



**Creación de Grupos de Datos**

**EMPLOYEES**

**4400**

**9500**

**Salario medio en la tabla EMPLOYEES para cada departamento**

**3500**

**6400**

**10033**

**…**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Hasta este punto, todas las funciones de grupo han considerado la tabla como un grupo de información de gran tamaño. Sin embargo, en ocasiones es necesario dividir la tabla de información en grupos más pequeños. Para ello, hay que utilizar la cláusula **GROUP BY**.

SELECT column,

group\_function(column)

FROM table T

[WHERE condition]

[GROUP BY group\_by\_expression]

[ORDER BY column];



**Creación de Grupos de Datos: Sintaxis de la Cláusula GROUP BY**

Puede dividir las filas de una tabla en grupos más pequeños utilizando la cláusula GROUP BY.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

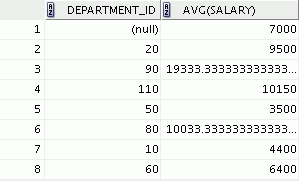
Puede utilizar la cláusula **GROUP BY** para dividir las filas de la tabla en grupos. A continuación puede utilizar las funciones de grupo para devolver información de resumen de cada grupo.

En la sintaxis:

*group\_by\_expression* Especifica columnas cuyos valores determinan la base para agrupar filas.

**Instrucciones**

* Si incluye una función de grupo en una cláusula **SELECT**, no puede seleccionar también resultados individuales *a menos que* la columna individual aparezca en la cláusula **GROUP BY**. Recibirá un mensaje de error si no puede incluir la lista de columnas en la cláusula **GROUP BY**.
* Al utilizar la cláusula **WHERE**, puede excluir las filas antes de dividirlas en grupos.
* Debe incluir las *columnas* en la cláusula **GROUP BY**.
* No puede utilizar un alias de columna en la cláusula **GROUP BY**.



SELECT department\_id,

AVG(salary)

FROM employees

GROUP BY department\_id ;



**Uso de la Cláusula GROUP BY**

Todas las columnas de la lista SELECT que no están incluidas en las funciones de grupo deben estar en la cláusula GROUP BY.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Al utilizar la cláusula **GROUP BY**, asegúrese de que todas las columnas de la lista **SELECT** que no están en las funciones de grupo están incluidas en la cláusula **GROUP BY**. El ejemplo de la diapositiva muestra el número de departamento y el salario medio de cada departamento. A continuación se muestra cómo se evalúa esta sentencia **SELECT** que contiene una cláusula **GROUP BY**:

* La cláusula **SELECT** especifica las columnas que se van a recuperar de la siguiente forma:
  + Columna del número de departamento de la tabla EMPLOYEES
  + Media de todos los salarios del grupo especificada en la cláusula **GROUP BY**
* La cláusula **FROM** especifica las dos tablas a las que la base de datos debe acceder: tabla EMPLOYEES.
* La cláusula **WHERE** especifica las filas que se van a recuperar. Ya que no existe ninguna cláusula **WHERE**, por defecto se recuperarán todas las filas.
* La cláusula **GROUP BY** especifica cómo se deben agrupar las filas. Las filas se agrupan por número de departamento, por lo tanto, la función **AVG** aplicada a la columna de salario calcula el salario medio de cada departamento.

**Nota:** para ordenar los resultados de las consultas en orden ascendente o descendente, incluya la cláusula **ORDER BY** en la consulta.

SELECT AVG(salary)

FROM employees

GROUP BY department\_id;

No es necesario que la columna **GROUP BY** esté en la cláusula **SELECT**. Por ejemplo, la sentencia **SELECT** de la diapositiva muestra los salarios medios de cada departamento sin mostrar los respectivos números de departamento. Sin embargo, sin los números de departamento, los resultados no parecen significativos.

También puede utilizar la función de grupo en la cláusula **ORDER BY**:

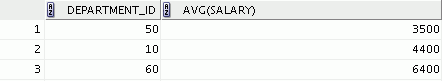
SELECT department\_id,

AVG(salary)

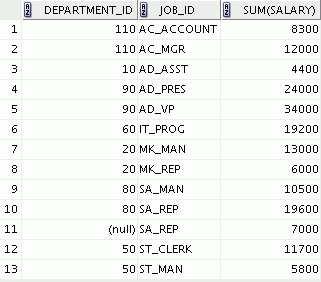
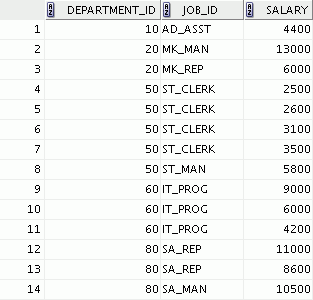
FROM employees

GROUP BY department\_id

ORDER BY AVG(salary);



**…**



**…**

**Agregar los salarios en la tabla EMPLOYEES para cada cargo, agrupado por departamento.**

**EMPLOYEES**

**Agrupación de Más de Una Columna**

### Agrupación de Más de Una Columna

En ocasiones necesitará ver los resultados de grupos dentro de grupos. La diapositiva muestra un informe que muestra el salario total pagado a cada puesto de cada departamento.

Según ese agrupamiento, la tabla EMPLOYEES se agrupa en primer lugar por número de departamento y, a continuación, por puesto. Por ejemplo, los cuatro oficinistas en el departamento de stock del departamento 50 se agrupan conjuntamente y se produce un resultado único (salario total) para todos los oficinistas en el departamento de stock del grupo.

La siguiente sentencia **SELECT** devuelve el resultado mostrado en la diapositiva:

SELECT department\_id,

job\_id,

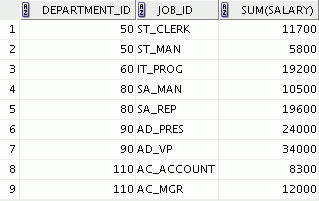
sum(salary)

FROM employees

GROUP BY department\_id,

job\_id

ORDER BY job\_id;



SELECT department\_id,

job\_id,

SUM(salary)

FROM employees

WHERE department\_id > 40

GROUP BY department\_id,

job\_id

ORDER BY department\_id;



**Uso de la Cláusula GROUP BY en Varias Columnas**

### Uso de la Cláusula Group By en Varias Columnas

Puede devolver resultados de resumen para grupos y subgrupos mostrando varias columnas **GROUP BY**. La cláusula **GROUP BY** agrupa filas pero no garantiza el orden del juego de resultados. Para ordenar los agrupamientos, utilice la cláusula **ORDER BY**.

En el ejemplo de la diapositiva, la sentencia **SELECT** que contiene una cláusula **GROUP BY** se evalúa de la siguiente forma:

* La cláusula **SELECT** especifica la columna que se van a recuperar:
  + ID de departamento en la tabla EMPLOYEES
  + ID de cargo de la tabla EMPLOYEES
  + Suma de todos los salarios del grupo especificada en la cláusula **GROUP BY**
* La cláusula **FROM** especifica las dos tablas a las que la base de datos debe acceder: tabla EMPLOYEES.
* La cláusula **WHERE** reduce el juego de resultados a aquellas filas en las que el ID de departamento es mayor de 40.
* La cláusula **GROUP BY** especifica cómo debe agrupar las filas resultantes:
  + En primer lugar, las filas se agrupan por ID de departamento.
  + En segundo lugar, las filas se agrupan por ID de cargo en los grupos de ID de departamento.
* La cláusula **ORDER BY** ordena los resultados por ID de departamento.

**Nota:** la función **SUM** se aplica a la columna de salario de todos los ID de cargo en el juego de resultados de cada grupo de ID de departamento. Además, tenga en cuenta que la fila SA\_REP no se devuelve. El ID de departamento para esta fila es **NULL** y, por lo tanto, no cumple la condición **WHERE**.

SELECT department\_id,

job\_id,

COUNT(last\_name)

FROM employees

GROUP BY department\_id ;

SELECT department\_id,

COUNT(last\_name)

FROM employees;



**Consultas No Válidas Realizadas con las Funciones de Grupo**

Cualquier columna o expresión de la lista SELECT que no sea una función de agregación debe estar en la cláusula GROUP BY:

**Una cláusula GROUP BY se debe agregar para el recuento de los apellidos para cada department\_id.**

**Agregar job\_id en GROUP BY o eliminar la columna job\_id de la lista SELECT.**



### Consultas No Válidas Realizadas con las Funciones de Grupo

Cuando utilice una mezcla de elementos individual (DEPARTMENT\_ID) y funciones de grupo (**COUNT**) en la misma sentencia **SELECT** debe incluir una cláusula **GROUP BY** que especifique los elementos individuales (en este caso, DEPARTMENT\_ID). Si falta la cláusula **GROUP BY**, aparecerá el mensaje de error “not a single-group group function” y la columna incorrecta estará indicada con un asterisco (**\***). Puede corregir el error del primer ejemplo de la diapositiva agregando la cláusula **GROUP BY**:

SELECT department\_id,

count(last\_name)

FROM employees

GROUP BY department\_id ;

Cualquier columna o expresión de la lista **SELECT** que no sea una función de agregación debe estar en la cláusula **GROUP BY**. En el segundo ejemplo de la diapositiva, job\_id no está en la cláusula **GROUP BY** ni la utiliza una función de grupo, por lo que se produce un error “not a **GROUP BY** expression”. Puede corregir el error del segundo ejemplo de la diapositiva agregando job\_id en la cláusula **GROUP BY**.

SELECT department\_id,

job\_id,

COUNT(last\_name)

FROM employees

GROUP BY department\_id,

job\_id;

SELECT department\_id,

AVG(salary)

FROM employees

WHERE AVG(salary) > 8000

GROUP BY department\_id;



**Consultas No Válidas Realizadas con las Funciones de Grupo**

* No puede utilizar la cláusula WHERE para restringir grupos.
* Debe utilizar la cláusula HAVING para restringir grupos.
* No puede utilizar las funciones de grupo de la cláusula WHERE.

**No puede utilizar la cláusula WHERE para restringir grupos.**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

No se puede utilizar la cláusula **WHERE** para restringir grupos. La sentencia **SELECT** del ejemplo de la diapositiva muestra un error porque se está utilizando la cláusula **WHERE** para restringir la visualización de los salarios medios de los departamentos que tienen un salario medio superior a 8000 dólares.

Puede corregir el error del ejemplo utilizando la cláusula **HAVING** para restringir grupos:

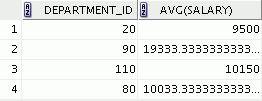
SELECT department\_id,

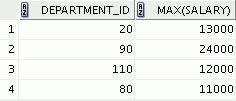
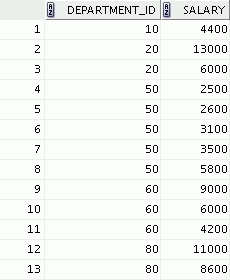
AVG(salary)

FROM employees

GROUP BY department\_id

HAVING AVG(salary) > 8000;





**Restricción de Resultados de Grupo**

**EMPLOYEES**

**Salario máximo por departamento cuando es superior a 10.000 euros.**

**…**

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |
|  | |
|  |  |
|  | |

### Restricción de Resultados de Grupo

Utilice la cláusula **HAVING** para restringir grupos de la misma forma que utiliza la cláusula **WHERE** para restringir las filas seleccionadas. Para buscar el salario máximo de cada uno de los departamentos que tienen un salario máximo superior a 10.000 euros, necesitará realizar las siguientes acciones:

* + 1. Buscar el salario medio de cada departamento realizando una agrupación por número de departamento.
    2. Restringir los grupos a los departamentos con un salario máximo superior a 10.000 euros.

SELECT column,

group\_function

FROM table

[WHERE condition]

[GROUP BY group\_by\_expression]

[HAVING group\_condition]

[ORDER BY column];



**Restricción de Resultados de Grupo con la Cláusula HAVING**

Al utilizar la cláusula HAVING, el servidor de Oracle restringe los grupos de la siguiente forma:

1. Agrupa las filas.
2. Aplica la función de grupo.
3. Muestra los grupos que coinciden con la cláusula HAVING.

### Restricción de Resultados de Grupo con la Cláusula HAVING

Puede utilizar la cláusula **HAVING** para especificar los grupos que se van a mostrar y, por lo tanto, restringir los grupos según la información de agregación.

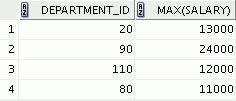
En la sintaxis, *group\_condition* restringe los grupos de filas de aquellos grupos para los que la condición especificada es verdadera.

El servidor de Oracle realiza los siguientes pasos al utilizar la cláusula **HAVING**:

1. Agrupa las filas.
2. Aplica la función de grupo al grupo.
3. Muestra los grupos que coinciden con los criterios de la cláusula **HAVING**.

La cláusula **HAVING** puede preceder a la cláusula **GROUP BY**, pero se recomienda que coloque la cláusula **GROUP BY** primero porque es más lógico. Los grupos están formados y las funciones de grupo se calculan antes de aplicar la cláusula **HAVING** a los grupos de la lista **SELECT**.

**Nota:** la cláusula **WHERE** restringe filas, mientras que la cláusula **HAVING** restringe grupos.



SELECT department\_id,

MAX(salary)

FROM employees

GROUP BY department\_id

HAVING MAX(salary)>10000 ;



**Uso de la Cláusula HAVING**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de la Cláusula HAVING

El ejemplo de la diapositiva muestra los números de departamento y los salarios máximos de los departamentos cuyo salario máximo sea superior a 10.000 dólares.

Puede utilizar la cláusula **GROUP BY** sin utilizar una función de grupo en la lista **SELECT**. Si restringe las filas según el resultado de una función de grupo, debe tener una cláusula **GROUP BY** y una cláusula **HAVING**.

El siguiente ejemplo muestra los números de departamento y los salarios medios de los departamentos cuyo salario máximo sea superior a 10.000 dólares:

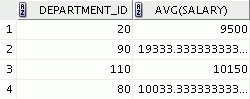
SELECT department\_id,

AVG(salary)

FROM employees

GROUP BY department\_id

HAVING max(salary)>10000;



SELECT job\_id,

SUM(salary) PAYROLL

FROM employees

WHERE job\_id NOT LIKE '%REP%'

GROUP BY job\_id

HAVING SUM(salary) > 13000;



**Uso de la Cláusula HAVING**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

El ejemplo de la diapositiva muestra el ID de cargo y el salario mensual total de cada cargo que tiene una nómina total excedente de 13.000 dólares. El ejemplo excluye a los vendedores y ordena la lista por el salario mensual total.

SELECT MAX(AVG(salary))

FROM employees

GROUP BY department\_id;



**Anidamiento de Funciones de Grupo**

Mostrar el salario máximo medio:



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Anidamiento de Funciones de Grupo**

Las funciones de grupo se pueden anidar en una profundidad de dos. El ejemplo de la diapositiva calcula el salario medio para cada department\_id y, a continuación, muestra el salario máximo medio.

Tenga en cuenta que la cláusula **GROUP BY** es obligatoria al anidar funciones de grupo.

## INSERT



**Lenguaje de Manipulación de Datos**

* Las sentencias DML se ejecutan al:
  + Agregar nuevas filas a una tabla
  + Modificar filas existentes en una tabla
  + Eliminar filas existentes de una tabla
* Una *transacción* consta de una recopilación de sentencias DML que forman una unidad lógica de trabajo.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

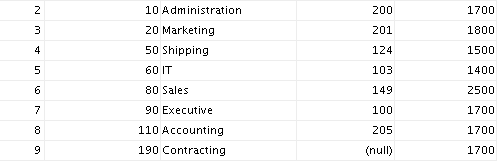
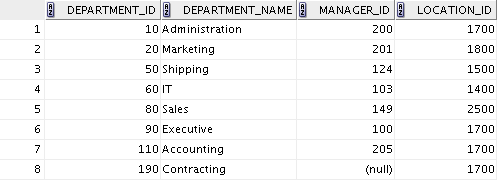
## Lenguaje de Manipulación de Datos

El lenguaje de manipulación de datos (DML) es una parte fundamentalmente de SQL. Para agregar, actualizar o suprimir los datos de la base de datos, ejecute una sentencia DML. La recopilación de sentencias DML que forman una unidad lógica de trabajo se denomina *transacción*.

Piense en una base de datos bancaria. Cuando un cliente del banco transfiere dinero de su cuenta de ahorro a una cuenta corriente, la transacción puede constar de las siguientes tres acciones diferentes: reducir la cuenta de ahorro, aumentar la cuenta corriente y registrar la transacción en el diario de transacciones. El servidor de Oracle debe garantizar que se ejecuten las tres sentencias SQL para mantener el balance correcto de las cuentas. Si algo impide que una de las sentencias de la transacción se ejecute, las demás sentencias de la transacción se deben deshacer.

Nota

* La mayoría de las sentencias DML de esta lección asumen que no se ha violado ninguna restricción de esta tabla. Las restricciones se tratarán más adelante en este curso.
* En SQL Developer, haga clic en el icono Run Script o pulse [F5] para ejecutar las sentencias DML. Los mensajes de comentarios se mostrarán en la página con separadores Script Output.



**Nuevo fila**

**Insertar una nueva fila en la tabla DEPARTMENTS.**

**DEPARTMENTS**

**Adición de una Nueva Fila a una Tabla**

## Adición de una Nueva Fila a una Tabla

El gráfico de la diapositiva ilustra un nuevo departamento a la tabla DEPARTMENTS.

INSERT INTO table [(column [, column...])]

VALUES (value [, value...]);



**Sintaxis de las Sentencias INSERT**

* Agregar nuevas filas a una tabla mediante la sentencia

INSERT:

* Con esta sintaxis, sólo se inserta una fila cada vez.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Sintaxis de las Sentencias INSERT

Puede agregar nuevas filas a una tabla emitiendo la sentencia INSERT.

En la sintaxis:

***table***es el nombre de la tabla.

***column***es el nombre de la columna de la tabla que se debe rellenar.

***value***es el valor correspondiente para la columna.

**Nota:** esta sentencia con la cláusula **VALUES** agrega sólo una fila cada vez a la tabla.

INSERT INTO departments(

department\_id,

department\_name,

manager\_id,

location\_id)

VALUES ( 70,

'Public Relations',

100,

1700);



**Inserción de Filas**

* Insertar una nueva fila que contenga los valores de cada columna.
* Mostrar valores en el orden por defecto de las columnas de la tabla.
* Opcionalmente, mostrar la lista de columnas en la cláusula

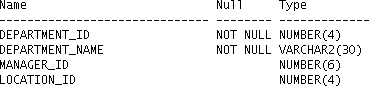
INSERT.

* Encerrar valores de caracteres y de fecha entre comillas simples.

### Inserción de Filas

Puesto que puede insertar una nueva fila que contenga los valores de cada columna, no es necesaria la lista de columnas en la cláusula **INSERT**. Sin embargo, si no utiliza la lista de columnas, los valores se deben mostrar según el orden por defecto de las columnas en la tabla y se debe proporcionar un valor para cada columna.

DESCRIBE departments



Para mayor claridad, utilice la lista de columnas en la cláusula **INSERT**.

Encierre los valores de caracteres y fecha entre comillas simples; sin embargo, no se recomienda encerrar valores numéricos entre comillas simples.

INSERT INTO departments

VALUES( 100,

'Finance',

NULL,

NULL);

INSERT INTO departments (department\_id,

department\_name)

VALUES (30,

'Purchasing');



**Inserción de Filas con Valores Nulos**

* Método implícito: omitir la columna de la lista de columnas.
* Método explícita: especificar la palabra clave NULL en la cláusula VALUES.

### Inserción de Filas con Valores Nulos

|  |  |
| --- | --- |
| **Metodo** | **Descripción** |
| Implícito | Omite la columna de la lista de columnas. |
| Explícito | Introduzca la palabra clave NULL en la lista VALUES; especifique la cadena vacía ('') en la lista VALUES de las cadenas de caracteres y fechas. |

Asegúrese de que puede utilizar valores nulos en la columna de destino mediante la verificación del estado **Null** con el comando **DESCRIBE**.

El servidor de Oracle aplica automáticamente todos los tipos de dato, rangos de datos y restricciones de integridad de los datos. Las columnas que no se muestran explícitamente obtienen un valor nulo en la nueva fila.

Los errores comunes que se producen durante la entrada del usuario se comprueban en el siguiente orden:

* Valor obligatorio que falta para una columna **NOT NULL**
* Valor duplicado que viola cualquier restricción de clave única o primaria
* Cualquier valor que viole una restricción **CHECK**
* Mantenimiento de la integridad referencial para la restricción de clave ajena
* No coincidencias de tipo de dato o valores demasiado anchos para la columna

**Nota:** se recomienda el uso de la lista de columnas porque hace la sentencia **INSERT** más legible y fiable o menos proclive a errores.

INSERT INTO employees (employee\_id,

first\_name,

last\_name,

email,

phone\_number,

hire\_date

)

VALUES ( 113,

'Louis',

'Popp',

'LPOPP',

'515.124.4567',

SYSDATE)



**Inserción de Valores Especiales**

La función SYSDATE registra la fecha y hora actuales.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Inserción de Valores Especiales

Puede utilizar funciones para introducir valores especiales en la tabla.

En el ejemplo de la diapositiva se registra información sobre el empleado Popp en la tabla EMPLOYEES. Proporciona la fecha y hora actuales en la columna HIRE\_DATE. Utiliza la función **SYSDATE** que devuelve la fecha y hora actuales del servidor de base de datos. También puede utilizar la función **CURRENT\_DATE** para obtener la fecha actual en la zona horaria de la sesión.

Además puede utilizar la función **USER** al insertar filas en la tabla. La función **USER** registra el nombre de usuario actual.

Confirmación de Adiciones a la Tabla

SELECT employee\_id,

last\_name,

job\_id,

hire\_date,

commission\_pct

FROM employees

WHERE employee\_id = 113;



**Inserción de Valores de Fecha y Hora Específicos**

* Agregar un nuevo empleado.
* Verificar la adición.

INSERT INTO employees

VALUES( 114,

'Den',

'Raphealy',

'DRAPHEAL',

'515.127.4561',

TO\_DATE('FEB 3, 1999', 'MON DD, YYYY'),

'SA\_REP',

11000,

0.2,

100,

60);



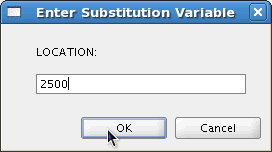
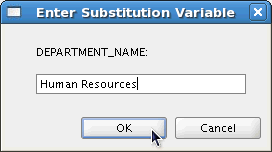
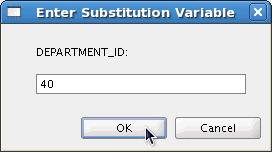
**Inserción de Valores de Fecha y Hora Específicos**

El formato DD-MON-RR se suele utilizar para insertar el valor de fecha. Con el formato RR, el sistema proporciona el siglo correcto automáticamente.

También puede especificar el valor de fecha en formato DD-MON-YYYY. Se recomienda hacerlo así porque especifica de forma clara el siglo y no depende de la lógica de formato RR interna de la especificación del siglo correcto.

Si necesita introducir una fecha en un formato diferente al formato por defecto (por ejemplo, con otro siglo o con una hora concreta), debe utilizar la función **TO\_DATE**.

En el ejemplo de la diapositiva se registra información del empleado Raphealy en la tabla EMPLOYEES. Define la columna HIRE\_DATE en el 3 de febrero de 1999.



INSERT INTO departments( department\_id,

department\_name,

location\_id)

VALUES( &department\_id,

'&department\_name',

&location);

**Creación de un Script**

* Utilizar el carácter de sustitución & en una sentencia SQL para solicitar valores.
* & es un marcador de posición del valor de la variable.

### Creación de un Script

Puede guardar comandos con variables de sustitución en un archivo y ejecutar los comandos en el archivo. En el ejemplo de la diapositiva se registra información sobre un departamento de la tabla DEPARTMENTS.

Ejecute el script y se le solicitará una entrada para cada una de las variables de sustitución (&). Después de introducir un valor para la variable de sustitución, haga clic en el botón OK. Los valores que introduce el usuario son los que se sustituyen en la sentencia. Esto permite ejecutar el mismo script una y otra vez, pero proporcionar un juego de valores diferente cada vez que lo ejecute.

INSERT INTO sales\_reps(id,

name,

salary,

commission\_pct)

SELECT employee\_id,

last\_name,

salary,

commission\_pct

FROM employees

WHERE job\_id LIKE '%REP%';



**Copia de Filas de Otra Tabla**

* Escribir la sentencia INSERT con una subconsulta:
* No utilizar la cláusula VALUES.
* Hacer coincidir el número de columnas de la cláusula

INSERT con el de la subconsulta.

* Inserta todas las filas devueltas por la subconsulta en la tabla, sales\_reps.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Copia de Filas de Otra Tabla

Puede utilizar la sentencia **INSERT** para agregar filas a la tabla en la que se derivan los valores de las tablas existentes. En el ejemplo de la diapositiva, para que la sentencia **INSERT INTO** funcione, debe haber creado antes la tabla sales\_reps mediante la sentencia **CREATE TABLE**. **CREATE TABLE** se trata en la lección titulada “Uso de Sentencias DDL para Crear y Gestionar Tablas”.

En lugar de la cláusula **VALUES**, utilice una subconsulta.

Sintaxis:

INSERT INTO table [ column (, column) ] subquery;

En la sintaxis:

*table* es el nombre de la tabla.

*column* es el nombre de la columna de la tabla que se debe rellenar.

*subquery* es la subconsulta que devuelve filas a la tabla.

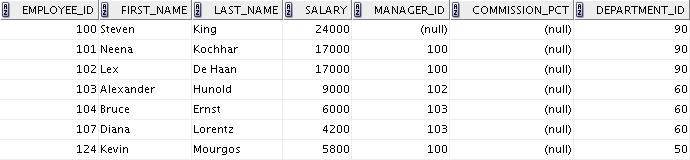
El número de columnas y sus tipos de dato de la lista de columnas de la cláusula **INSERT** deben coincidir con el número de valores y sus tipos de dato en la subconsulta. Se agregan cero o más filas en función del número de filas devueltas por la subconsulta. Para crear una copia de la filas de una tabla, utilice **SELECT** \* en la subconsulta:

INSERT INTO copy\_emp

SELECT \*

FROM employees;

## UPDATE



**Cambio de Datos en la Tabla**

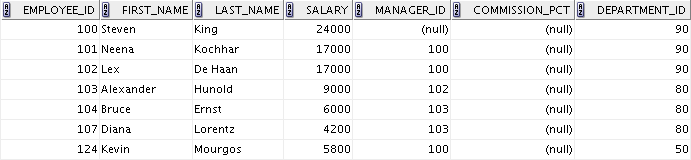
**EMPLOYEES**

**Actualizar filas en la tabla EMPLOYEES:**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Cambio de Datos en la Tabla



La diapositiva ilustra el cambio de número de departamento para los empleados del departamento 60 al departamento 80.

UPDATE Table

SET column = value [, column = [WHERE value, ...] condition];



**Sintaxis de Sentencias UPDATE**

* Modificar los valores existentes en una tabla con la sentencia UPDATE:
* Actualizar más de una fila cada vez (si es necesario).

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Sintaxis de Sentencias UPDATE

Puede modificar los valores existentes en una tabla mediante la sentencia **UPDATE**. En la sintaxis:

***table***es el nombre de la tabla.

***column***es el nombre de la columna de la tabla que se debe rellenar.

***value***es el valor o subconsulta correspondiente para la columna.

***condition***identifica las filas que se deben actualizar y se compone de nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

Para confirmar la operación de actualización, consulte la tabla para visualizar las filas actualizadas.

**Nota:** en general, la primera columna de clave primaria se utiliza en la cláusula **WHERE** para identificar una única fila para la actualización. El uso de otras columnas puede provocar una actualización inesperada de varias filas. Por ejemplo, identificar una fila de la tabla EMPLOYEES por nombre es peligroso, ya que puede que más de un empleado tengan el mismo nombre.

UPDATE employees

SET department\_id = 50

WHERE employee\_id = 113;



**Actualización de Filas en una Tabla**

* Si se especifica la cláusula WHERE, se modifican los valores de una fila o varias filas específicas:
* Si se omite la cláusula WHERE, se modifican los valores

de todas las filas de la tabla.

* Especificar SET *column\_name*= NULL para actualizar un valor de columna a NULL.

### Actualización de Filas en una Tabla

La sentencia **UPDATE** modifica los valores de una fila o varias filas específicas si se especifica la cláusula **WHERE**. El ejemplo de la diapositiva muestra la transferencia del empleado 113 (Popp) al departamento 50.Si omite la cláusula **WHERE**, se modifican todas las filas de la tabla. Examine las filas actualizadas en la tabla COPY\_EMP.

SELECT last\_name,

department\_id

FROM copy\_emp;



**…**

Por ejemplo, un empleado que era SA\_REP ahora ha cambiado su puesto a IT\_PROG. Por lo tanto, se debe actualizar su JOB\_ID y el campo de comisiones se debe definir en NULL.

UPDATE employees

SET job\_id = 'IT\_PROG',

commission\_pct = NULL

WHERE employee\_id = 114;

**Nota:** la tabla COPY\_EMP tiene los mismos datos que la tabla EMPLOYEES.

UPDATE employees

SET job\_id

UPDATE salary= (SELECT job\_id

FROM employees

WHERE employee\_id = 113;



**Actualización de Dos Columnas con una Subconsulta**

Actualizar el cargo y el salario del empleado 113 para que coincida con los del empleado 205. ¿

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Actualización de Dos Columnas con una Subconsulta

Puede actualizar varias columnas en la cláusula **SET** de una sentencia **UPDATE** mediante la escritura de varias subconsultas. La sintaxis es la siguiente:

UPDATE *table*

SET *column* = ( SELECT *column*

FROM *table*

WHERE *condition*)

[ ,

*Column* = (SELECT *column*

FROM *table*

WHERE *condition*)]

[WHERE *condition* ] ;

El ejemplo de la diapositiva también se puede escribir del siguiente modo:

UPDATE employees

SET ( job\_id,

Salary) = (SELECT job\_id,

salary

FROM employees

WHERE employee\_id = 205)

WHERE employee\_id = 113;

UPDATE copy\_emp

SET department\_id = ( SELECT department\_id

FROM employees

WHERE employee\_id = 100)

WHERE job\_id = (SELECT job\_id

FROM employees)



**Actualización de Filas Basada en Otra Tabla**

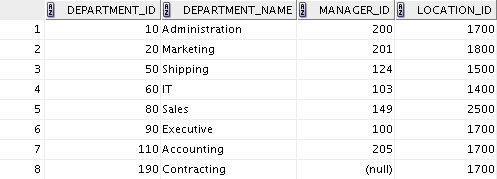
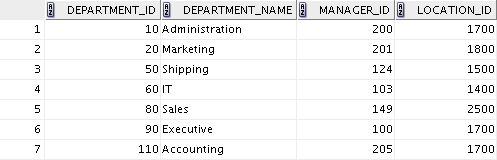
Utilizar subconsultas en las sentencias UPDATE para actualizar los valores de fila de una tabla según los valores de otra tabla:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Actualización de Filas Basada en Otra Tabla

Puede utilizar las subconsultas **UPDATE** para actualizar las filas de una tabla. El ejemplo de la diapositiva actualiza la tabla COPY\_EMP según los valores de la tabla EMPLOYEES. Cambia el número de departamento de todos los empleados con el ID de cargo del empleado 200 al número de departamento actual del empleado 100.

## DELETE



**Eliminación de Filas de Tablas**

**DEPARTMENTS**

**Suprimir una fila de la tabla DEPARTMENTS:**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Eliminación de Filas de Tablas

El departamento de contratos se ha eliminado de la tabla DEPARTMENTS (siempre que no haya violado ninguna restricción en la tabla DEPARTMENTS), como se muestra en el gráfico de la diapositiva.

DELETE [FROM] *table*

[WHERE *condition*];



**Sentencia DELETE**

Puede eliminar filas existentes de una tabla mediante la sentencia DELETE:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Sintaxis de Sentencias DELETE

Puede eliminar filas existentes de una tabla mediante la sentencia **DELETE**. En la sintaxis:

***table***es el nombre de la tabla.

***condition***identifica las filas que se deben suprimir y se compone de nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

**Nota:** si no se suprime ninguna fila, se devuelve el mensaje “0 rows deleted” (en el separador Script Output de SQL Developer)

DELETE FROM copy\_emp;

DELETE FROM departments

WHERE department\_name = 'Finance';



**Supresión de Filas de Tablas**

* Se suprimen filas concretas si se especifica la cláusula

WHERE:

* Se suprimen todas las filas de la tabla si omite la cláusula

WHERE:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Supresión de Filas de Tablas

Puede suprimir filas concretas mediante la especificación de la cláusula **WHERE** en la sentencia **DELETE**. En el primer ejemplo de la diapositiva se suprime el departamento Accounting de la tabla DEPARTMENTS. Para confirmar la operación de supresión, visualice las filas suprimidas con la sentencia **SELECT**.

SELECT \*

FROM departments

WHERE department\_name = 'Finance';



Si omite la cláusula **WHERE**, se suprimen todas las filas de la tabla. En el segundo ejemplo de la diapositiva se suprimen todas las filas de la tabla COPY\_EMP, porque no se ha especificado ninguna cláusula **WHERE**.

Ejemplo:

Se eliminan las filas identificadas en la cláusula **WHERE**.

DELETE FROM employees

WHERE employee\_id = 114;

****

DELETE FROM departments

WHERE department\_id IN (30, 40);



DELETE FROM employees

WHERE department\_id =(SELECT department\_id

FROM departments

WHERE department\_name LIKE '%Public%');



**Supresión de Filas Basada en Otra Tabla**

Utilizar subconsultas en las sentencias DELETE para eliminar filas de una tabla según los valores de otra tabla:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Supresión de Filas Basada en Otra Tabla

Puede utilizar subconsultas para suprimir filas de una tabla según los valores de otra tabla. En el ejemplo de la diapositiva se suprimen todos los empleados de un departamento, donde el nombre del departamento consta de la cadena Public.

La subconsulta busca en la tabla DEPARTMENTS el número de departamento según el nombre de departamento que contenga la cadena Public. A continuación, la subconsulta introduce el número de departamento en la consulta principal, con lo que se suprimen las filas de datos de la tabla EMPLOYEES en base a este número de departamento.

## TRUNCATE

TRUNCATE TABLE copy\_emp;

TRUNCATE TABLE *table\_name*;



**Sentencia TRUNCATE**

* Elimina todas las filas de una tabla, dejando la tabla vacía y la estructura de la misma intacta.
* Es una sentencia de lenguaje de definición de datos (DDL) en lugar de una sentencia DML; no se puede deshacer fácilmente.
* Sintaxis:
* Ejemplo:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Sentencia TRUNCATE

Una forma más eficaz de vaciar una tabla es con la sentencia **TRUNCATE**.

Puede utilizar la sentencia **TRUNCATE** para eliminar fácilmente todas las filas de una tabla o un cluster. La eliminación de filas con las sentencia **TRUNCATE** es más rápida que la eliminación con la sentencia **DELETE** por las siguientes razones:

* + La sentencia **TRUNCATE** es una sentencia de lenguaje de definición de datos (DDL) y no genera ninguna información de rollback. La información de rollback se trata más adelante en esta lección.
  + El truncamiento de una tabla no arranca los disparadores de supresión de la tabla.

Si la tabla es el principal de una restricción de integridad referencial, no puede truncarla. Debe desactivar la restricción antes de emitir la sentencia **TRUNCATE**. La desactivación de restricciones se trata en la lección titulada “Uso de Sentencias DDL para Crear y Gestionar Tablas”.

## MERGE

Oracle Server soporta la sentencia **MERGE** para operaciones **INSERT**, **UPDATE** y **DELETE**.

Mediante esta sentencia, puede actualizar, insertar o suprimir una fila condicionalmente en una tabla, con lo que se evitan varias sentencias DML. La decisión de actualizar, insertar o suprimir en la tabla destino se basa en una condición de la cláusula ON.

Hay que tener privilegios de objeto **INSERT** y **UPDATE** en la tabla destino y el privilegio de objeto **SELECT** en la tabla origen. Para especificar la cláusula **DELETE** de la cláusula merge\_update\_clause, también debe tener el privilegio de objeto **DELETE** en la tabla destino.

La sentencia **MERGE** es determinista. No se puede actualizar varias veces la misma fila de la tabla destino en la misma sentencia **MERGE**.

Un enfoque alternativo es utilizar bucles PL/SQL y varias sentencias DML. La sentencia **MERGE**, sin embargo, es fácil de utilizar y se expresa de forma más sencilla como una única sentencia SQL.

La sentencia **MERGE** es adecuada en diferentes aplicaciones de almacén de datos. Por ejemplo, en una aplicación de almacén de datos, es posible que necesite trabajar con datos procedentes de varios orígenes, algunos de los cuales pueden estar duplicados. Con la sentencia **MERGE**, puede agregar o modificar filas condicionalmente.

MERGE INTO table\_name table\_alias

USING (table|view|sub\_query) alias

ON (join condition)

WHEN MATCHED THEN

UPDATE SET

col1 = col\_val1,

col2 = col2\_val

WHEN NOT MATCHED THEN

INSERT (column\_list)

VALUES (column\_values);

**Fusión de Filas**

Puede actualizar filas existentes e insertar nuevas filas condicionalmente mediante la sentencia **MERGE**.

En la sintaxis:

Cláusula **INTO** especifica la tabla destino que se va a actualizar o en la que se va a insertar.

Cláusula **USING** identifica el origen de los datos que se van a actualizar o insertar; puede ser una tabla, una vista o una subconsulta.

Cláusula **ON** condición según la cual la operación **MERGE** actualiza o inserta

**WHEN MATCHED** | indica al servidor cómo responder a los resultados de la condición de unión **WHEN NOT MATCHED**

Para obtener más información, consulte Oracle Database 10g SQL Reference, “**MERGE**”.

Ejemplo:

Inserte o actualice las filas de la tabla EMPL3 para que se corresponda con la tabla SCOTT.EMP.

MERGE INTO empl3 c

USING scott.emp e

ON (c.empno = e.empno)

WHEN MATCHED THEN

UPDATE SET

c.first\_name = e.first\_name,

c.last\_name = e.last\_name,

c.email = e.email,

c.phone\_number = e.phone\_number,

c.hire\_date = e.hire\_date,

c.job = e.job,

c.sal = e.sal,

c.commission\_pct = e.commission\_pct,

c.mgr = e.mgr,

c.deptno = e.deptno

WHEN NOT MATCHED THEN

INSERT VALUES(e.empno, e.first\_name, e.last\_name,

e.email, e.phone\_number, e.hire\_date, e.job,

e.sal, e.commission\_pct, e.mgr,

e.deptno);

El ejemplo de la diapositiva hace corresponder EMPNO de la tabla EMPL3 con

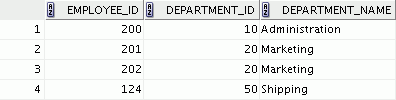
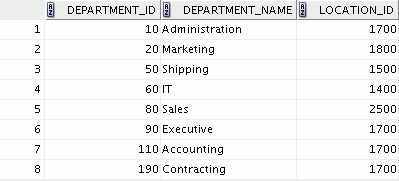
EMPNO de la tabla SCOTT.EMP. Si encuentra una correspondencia, la fila de la

tabla EMPL3 se actualiza para que se corresponda con la de la tabla SCOTT.EMP. Si no encuentra la fila, la inserta en la tabla EMPL3.

Se evalúa la condición c.empno = e.empno. Como la tabla EMPL3 está vacía, la condición devuelve false (no hay correspondencias). La lógica recae en la cláusula **WHEN NOT MATCHED** y el comando **MERGE** inserta las filas de la tabla SCOTT.EMP en la tabla EMPL3.

Si existieran las filas en la tabla EMPL3 y los identificadores de empleado se correspondieran en las dos tablas (en las tablas EMPL3 y SCOTT.EMP), las filas existentes en la tabla EMPL3 se actualizarían para corresponderse con la tabla SCOTT.EMP

# JOINS



**Obtención de Datos de Varias Tablas**

**EMPLOYEES**

**DEPARTMENTS**

**…**

**…**

|  |  |  |
| --- | --- | --- |
|  |  |  |

# Obtención de Datos de Varias Tablas

A veces necesita utilizar datos de más de una tabla. En el ejemplo de la diapositiva, el informe muestra datos de dos tablas independientes:

* + La tabla EMPLOYEES contiene los ID de empleado.
  + Las tablas EMPLOYEES y DEPARTMENTS contienen los ID de departamento.
  + La tabla DEPARTMENTS contiene los nombres de departamento.

Para producir el informe, necesita enlazar las tablas EMPLOYEES y DEPARTMENTS y acceder a los datos de ambas.



**Tipos de Uniones**

Las uniones compatibles con el estándar SQL:1999 incluyen los siguientes elementos:

* Uniones naturales:
  + Cláusula NATURAL JOIN
  + Cláusula USING
  + Cláusula ON
* Uniones OUTER:
  + LEFT OUTER JOIN
  + RIGHT OUTER JOIN
  + FULL OUTER JOIN
* Uniones cruzadas

**Tipos de Uniones**

Para unir tablas, puede utilizar la sintaxis de unión que cumpla con el estándar SQL:1999.

**Nota**

En versiones anteriores a Oracle9*i* la sintaxis de unión era diferente a los estándares American National Standards Institute (ANSI). La sintaxis de unión compatible con SQL:1999 no ofrece ninguna ventaja en el rendimiento respecto a la sintaxis de unión propietaria de Oracle que existía en las versiones anteriores. Para obtener más información sobre la sintaxis de unión propiedad de Oracle, consulte el apéndice F: Sintaxis de Unión en Oracle.

En la siguiente diapositiva se trata la sintaxis de unión compatible con SQL:1999.

SELECT table1.column,

table2.column

FROM table1

[NATURAL JOIN table2] |

[JOIN table2 USING (column\_name)] | [JOIN table2

ON (table1.column\_name = table2.column\_name)]| [LEFT|RIGHT|FULL OUTER JOIN table2

ON (table1.column\_name = table2.column\_name)]| [CROSS JOIN table2];



**Unión de Tablas mediante la Sintaxis SQL:1999**

Utilizar una unión para consultar datos de más de una tabla:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Unión de Tablas mediante la Sintaxis SQL:1999**

En la sintaxis:

* + table1.column indica la tabla y la columna desde las que se recuperan los datos
  + NATURAL JOIN une dos tablas basándose en el mismo nombre de la columna
  + JOIN table2 USING column\_name realiza una unión igualitaria basándose en el nombre de la columna
  + JOIN table2 ON table1.column\_name = table2.column\_name performs
  + realiza una unión igualitaria basándose en la condición de la cláusula ON
  + LEFT/RIGHT/FULL OUTER se utiliza para realizar uniones OUTER
  + CROSS JOIN devuelve un producto cartesiano de las dos tablas

**Cualificación de Nombres de Columna Ambiguos**



Utilizar prefijos de tabla para cualificar los nombres de columna

que están en varias tablas.

* + Utilizar prefijos de tabla para mejorar el rendimiento.
  + En lugar de prefijos de nombre de tabla completos, utilizar alias de tabla.
  + Los alias de tablas proporciona un nombre más corto de una tabla:

**– Mantiene el código SQL más pequeño, utiliza menos memoria**

* + Utilizar alias de columna para distinguir columnas que tienen nombres idénticos, pero que residen en diferentes tablas.

**Cualificación de Nombres de Columna Ambiguos**

Al unir dos o más tablas, debe cualificar los nombres de las columnas con el nombre de la tabla para evitar ambigüedad. Sin los prefijos de tabla, la columna DEPARTMENT\_ID de la lista **SELECT** puede provenir de la tabla DEPARTMENTS o de la tabla EMPLOYEES. Es necesario agregar el prefijo de tabla para ejecutar la consulta. Si no existen nombres de columna comunes entre las dos tablas, no es necesario cualificar las columnas. Sin embargo, el uso del prefijo de tabla mejora el rendimiento, ya que indica al servidor de *Oracle* dónde encontrar exactamente las columnas.

Sin embargo, la cualificación de nombres de columna con nombres de tabla puede llevar bastante tiempo, especialmente si los nombres de tabla son largos. En su lugar, puede utilizar *alias de tabla*. Igual que un alias de columna proporciona otro nombre a una columna, un alias de tabla proporciona otro nombre a una tabla. Los alias de tabla ayudan a mantener el código SQL más pequeño y, por lo tanto, menos uso de memoria

El nombre de tabla se especifica por completo, seguido de un espacio y del alias de tabla. Por ejemplo, a la tabla EMPLOYEES se le puede proporcionar el alias **e** y a la tabla DEPARTMENTS el alias **d**.

**Instrucciones**

* + Los alias de tabla pueden tener hasta 30 caracteres de longitud, pero los alias más cortos son mejores que los largos.
  + Si se utiliza un alias de tabla para un nombre de tabla determinado en la cláusula **FROM**, el alias de tabla se deberá sustituir por el nombre de tabla mediante la sentencia **SELECT**.
  + Los alias de tabla deben ser significativos.
  + El alias de tabla es válido sólo para la sentencia actual **SELECT**.

## NATURAL JOINTS



**Creación de Uniones Naturales**

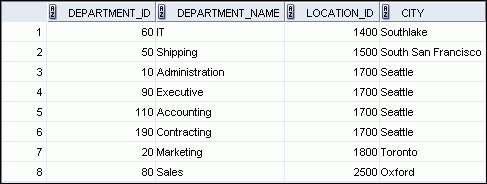
* La cláusula NATURAL JOIN está basada en todas las columnas de las dos tablas que tienen el mismo nombre.
* Selecciona filas de las dos tablas que tienen valores iguales en todas las columnas coincidentes.
* Si las columnas que tienen el mismo nombre tienen tipos de dato diferentes, se devolverá un error.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Creación de Uniones Naturales

Puede unir tablas automáticamente basándose en las columnas de las dos tablas que tienen el mismo nombre y los mismos tipos de dato. Puede realizar esta acción utilizando las palabras clave **NATURAL JOIN**.

**Nota:** la unión se puede producir únicamente en las columnas que tienen los mismos nombres y los mismos tipos de dato en ambas tablas. Si las columnas tienen el mismo nombre pero tipos de dato diferentes, la sintaxis **NATURAL JOIN** produce un error.



**Recuperación de Registros con Uniones Naturales**

SELECT department\_id,

department\_name,

location\_id,

city

FROM departments

NATURAL JOIN locations;

**Recuperación de Registros con Uniones Naturales**

En el ejemplo de la diapositiva, la tabla LOCATIONS está unida a la tabla DEPARTMENT mediante la columna LOCATION\_ID, que es la única con el mismo nombre en ambas. Si están presentes otras columnas comunes, la unión debe haberlas utilizado todas.

**Uniones Naturales con la Cláusula WHERE**

Las restricciones adicionales en una unión natural se implantan utilizando una cláusula **WHERE**. El siguiente ejemplo limita las filas de salida a las que tienen un ID de departamento igual a 20 o a 50:

SELECT department\_id,

department\_name,

location\_id,

city

FROM departments

NATURAL JOIN locations

WHERE department\_id IN (20, 50);



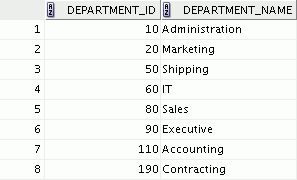
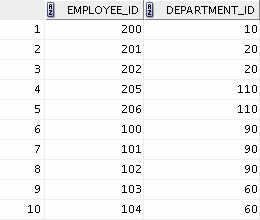
**Creación de Uniones con la Cláusula USING**

* Si varias columnas tienen el mismo nombre pero los tipos de dato no coinciden, utilizar la cláusula USING para especificar las columnas para la unión igualitaria.
* Utilizar USING para que sólo coincida una columna en caso de que coincida más de una.
* Las cláusulas NATURAL JOIN y USING se excluyen mutuamente.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Creación de Uniones con la Cláusula USING

Las uniones naturales utilizan todas las columnas con tipos de dato y nombres coincidentes para unir las tablas. La cláusula **USING** se puede utilizar para especificar sólo las columnas que se deben utilizar para una unión igualitaria.



**Clave ajena**

**Primary key**

**…**

**DEPARTMENTS**

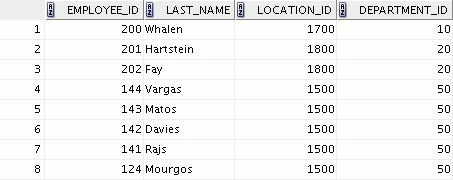
**EMPLOYEES**

**Unión de Nombres de Columna**

Unión de Nombres de Columna

Para determinar el nombre de departamento de un empleado, compare el valor de la columna DEPARTMENT\_ID en la tabla EMPLOYEES con los valores de DEPARTMENT\_ID de la tabla DEPARTMENTS. La relación entre las tablas EMPLOYEES y DEPARTMENTS es una *unión igualitaria*; es decir, los valores de la columna DEPARTMENT\_ID de ambas tablas deben ser iguales. Normalmente, este tipo de unión implica complementos de clave primaria y ajena.

**Nota:** las uniones igualitarias también se denominan *uniones simples* o *uniones internas*.



**Recuperación de Registros con la Cláusula USING**

**…**

SELECT employee\_id,

last\_name,

location\_id,

department\_id

FROM employees

JOIN departments

USING ( department\_id);

### Recuperación de Registros con la Cláusula USING

En el ejemplo de la diapositiva, las columnas DEPARTMENT\_ID de las tablas EMPLOYEES y DEPARTMENTS se unen y, por lo tanto, se muestra el LOCATION\_ID del departamento en el que trabaja el empleado.

SELECT l.city,

d.department\_name

FROM locations l

JOIN departments d

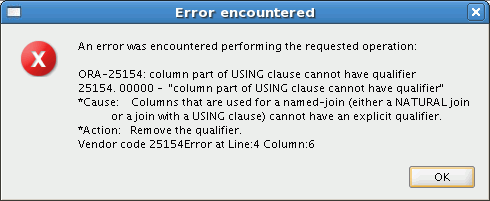
USING ( location\_id)

WHERE d.location\_id = 1400;



**Uso de Alias de Tabla con la Cláusula USING**

* No cualificar una columna que se utilice en la cláusula USING.
* Si la misma columna se utiliza en otro lugar de la sentencia SQL, no se le puede agregar un alias.



Uso de Alias de Tabla con la Cláusula USING

Al realizar una unión con la cláusula **USING**, no puede cualificar una columna que se utiliza en la cláusula **USING** en sí. Además, si esa columna se utiliza en cualquier ubicación de la sentencia SQL, no se le puede agregar ningún alias. Por ejemplo, en la consulta mencionada en la diapositiva, no debe agregar un alias a la columna location\_id en la cláusula **WHERE** porque la columna se utiliza en la cláusula **USING**.

Las columnas a las que se hace referencia en la cláusula **USING** no deben tener un cualificador (nombre o alias de la tabla) en cualquier ubicación de la sentencia SQL. Por ejemplo, la siguiente sentencia es válida:

SELECT l.city,

d.department\_name

FROM locations l

JOIN departments d

USING (location\_id)

WHERE location\_id = 1400;

Las columnas comunes en ambas tablas, pero que no se utilizan en la cláusula **USING**, se deben prefijar con un alias de tabla, de lo contrario aparecerá el error “column ambiguously defined”.

En la siguiente sentencia , manager\_id está presente tanto la tabla employees como en departments ; si no se prefija manager\_id con un alias de tabla, se produce el error “column ambiguously defined”.

El siguiente ejemplo es válido:

SELECT first\_name,

d.department\_name,

d.manager\_id

FROM employees e

JOIN departments d

USING (department\_id)

WHERE department\_id = 50;

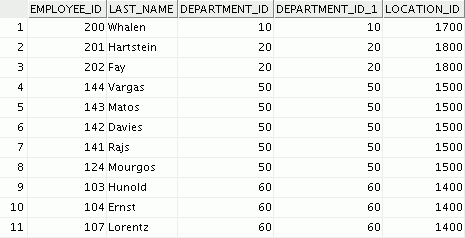
**Creación de Uniones con la Cláusula ON**

* La condición de unión de la unión natural es básicamente una unión igualitaria de todas las columnas con el mismo nombre.
* Utilizar la cláusula ON para especificar condiciones arbitrarias o columnas que se van a unir.
* La condición de unión está separada de otras condiciones de búsqueda.
* La clausula ON facilita la comprensión del código.



### Creación de Uniones con la Cláusula ON

Utilice la sentencia **ON** para especificar una condición de unión. Esto permite especificar condiciones de unión independientes de cualquier condición de filtro o de búsqueda en la cláusula **WHERE**.



**Recuperación de Registros con la Cláusula ON**

**…**

SELECT e.employee\_id,

e.last\_name,

e.department\_id,

d.department\_id,

d.location\_id

FROM employees e

JOIN departments d

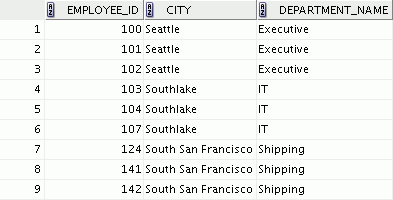
ON ( e.department\_id = d.department\_id)

Recuperación de Registros con la Cláusula ON

En este ejemplo, las columnas DEPARTMENT\_ID en la tabla EMPLOYEES y DEPARTMENTS se unen con la cláusula **ON**. Cuando un ID de departamento de la tabla EMPLOYEES sea igual al ID de departamento de la tabla DEPARTMENTS, se devolverá una fila. El alias de tabla es necesario para cualificar los column\_names coincidentes.

También puede utilizar la cláusula **ON** para unir columnas que tienen nombres diferentes. Los paréntesis de las columnas unidas, como se muestra en el ejemplo de la diapositiva, (e.department\_id = d.department\_id) son opcionales. Incluso, ON e.department\_id = d.department\_id funcionará.

**Nota:** al utilizar el icono Execute Statement para ejecutar la consulta, SQL Developer agrega el sufijo “\_1” para diferenciar entre los dos department\_ids.



**Creación de Uniones en 3 Direcciones con la Cláusula ON**

**…**

SELECT employee\_id,

city,

department\_name

FROM employees e

JOIN departments d

ON d.department\_id = e.department\_id

JOIN locations l

ON d.location\_id = l.location\_id;

### Creación de Uniones en 3 Direcciones con la Cláusula ON

Una unión en tres direcciones es una unión de tres tablas. En la sintaxis compatible con SQL:1999, las uniones se realizan de izquierda a derecha. Por lo tanto, la primera unión que se realiza es EMPLOYEES JOIN DEPARTMENTS. La primera condición de unión puede hacer referencia a las columnas de EMPLOYEES y DEPARTMENTS, pero no puede hacer referencia a las columnas de LOCATIONS. La segunda condición de unión puede hacer referencia a las columnas de las tres tablas.

**Nota:** el ejemplo de código de la diapositiva también se puede realizar con la cláusula USING:

SELECT e.employee\_id,

l.city,

d.department\_name

FROM employees e

JOIN departments d

USING (department\_id)

JOIN locations l

USING (location\_id);



**Aplicación de Condiciones Adicionales a una Unión**

Uso de la cláusula AND o la cláusula WHERE para aplicar condiciones adicionales:

SELECT e.employee\_id,

e.last\_name,

e.department\_id,

d.department\_id,

d.location\_id

FROM employees e

JOIN departments d

ON ( e.department\_id = d.department\_id)

AND e.manager\_id = 149;

**O bien**

SELECT e.employee\_id,

e.last\_name,

e.department\_id,

d.department\_id,

d.location\_id

FROM employees e

JOIN departments d

ON (e.department\_id = d.department\_id)

WHERE e.manager\_id = 149;

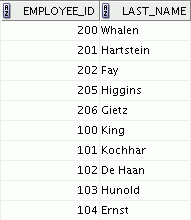
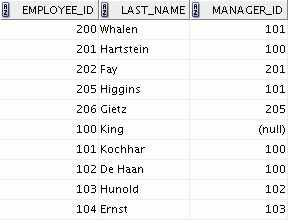
### Aplicación de Condiciones Adicionales a una Unión

Puede aplicar condiciones adicionales a la unión.

El ejemplo realiza una unión en las tablas EMPLOYEES y DEPARTMENTS y, además, muestra sólo los empleados que tienen un ID de gestor de 149. Para agregar condiciones adicionales a la cláusula **ON**, puede agregar cláusulas **AND**. Por otro lado, puede utilizar una cláusula **WHERE** para aplicar condiciones adicionales.



## AUTOUNION



**MANAGER\_ID en la tabla WORKER es igual a**

**EMPLOYEE\_ID en la tabla MANAGER.**

**…**

**…**

**EMPLOYEES (MANAGER)**

**EMPLOYEES (WORKER)**

**Unión de una Tabla consigo Misma**

### Unión de una Tabla consigo Misma

Puede que a veces necesite unir una tabla consigo misma. Para buscar el nombre de cada gestor del empleado, necesita unir la tabla EMPLOYEES consigo misma o realizar una autounión. Por ejemplo, para buscar el nombre del gestor de Lorentz, necesita:

Buscar a Lorentz en la tabla EMPLOYEES buscando en la columna LAST\_NAME.

Busque el número de gestor de Lorentz consultando la columna MANAGER\_ID. El número de gestor de Lorentz es 103.

Busque el nombre del gestor con un EMPLOYEE\_ID de 103 consultando la columna LAST\_NAME. El número de empleado de Hunold es 103, por lo que Hunold en el gestor de Lorentz.

Durante este proceso, busca dos veces en la tabla. La primera vez, cuando consulta la tabla para buscar a Lorentz en la columna LAST\_NAME y el valor de MANAGER\_ID de 103. La segunda vez, cuando busca en la columna EMPLOYEE\_ID para buscar 103 y en la columna LAST\_NAME para buscar a Hunold.



SELECT worker.last\_name emp,

manager.last\_name mgr

FROM employees worker

JOIN employees manager

ON ( worker.manager\_id = manager.employee\_id);



**Autouniones que Utilizan la Cláusula ON**

### Autouniones que Utilizan la Cláusula ON

La cláusula **ON** también se puede utilizar para unir columnas con nombres diferentes, ya sea en la misma tabla o en otra diferente.

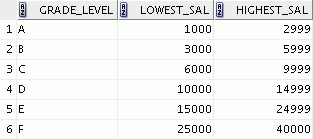
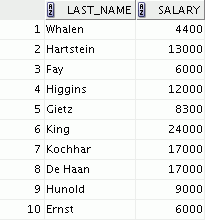
En el ejemplo se muestra una autounión de la tabla EMPLOYEES basada en las columnas

EMPLOYEE\_ID y MANAGER\_ID.

**Nota:** los paréntesis de las columnas unidas, como se muestra en la diapositiva (e.manager\_id = m.employee\_id) son **opcionales**. Incluso:

**ON** e.manager\_id = m.employee\_id funcionará.

## UNIONES NO IGUALITARIAS



**JOB\_GRADES define el rango de valores de LOWEST\_SAL y HIGHEST\_SAL de cada GRADE\_LEVEL.**

**Por lo tanto, la columna GRADE\_LEVEL se puede utilizar para asignar grados a cada empleado.**

**…**

**JOB\_GRADES**

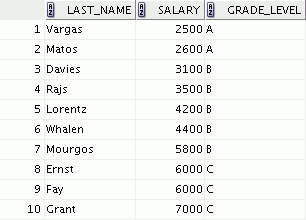
**EMPLOYEES**

**Uniones no igualitarias**

## Uniones no igualitarias

Una unión no igualitaria es una condición de unión que contiene algún operador diferente del operador de igualdad.

La relación entre la tabla EMPLOYEES y JOB\_GRADES es un ejemplo de unión no igualitaria. Los rangos de la columna SALARY en la tabla EMPLOYEES entre los valores en las columnas LOWEST\_SAL y HIGHEST\_SAL de la tabla JOB\_GRADES. Por lo tanto, se pueden agregar grados a cada empleado según el salario. La relación se obtiene mediante un operador distinto del de igualdad (=).



SELECT e.last\_name,

e.salary,

j.grade\_level

FROM employees e

JOIN job\_grades j

ON e.salary

BETWEEN j.lowest\_sal

AND j.highest\_sal;



**Recuperación de Registros con Uniones no Igualitarias**

**…**

### Recuperación de Registros con Uniones no Igualitarias

La diapositiva del ejemplo crea una unión no igualitaria para evaluar el grado de salario de un empleado. El salario debe estar *entre* cualquier par de los rangos de salario bajos y altos.

Es importante tener en cuenta que todos los empleados aparecerán sólo una vez al ejecutar esta consulta. No se repite ningún empleado en la lista. Existen dos motivos por los que se produce este hecho:

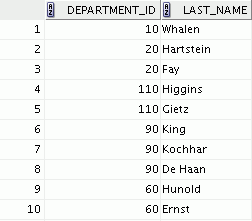
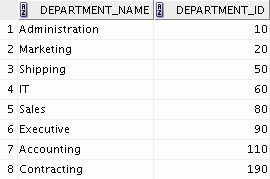
Ninguna de las filas de la tabla JOB\_GRADES contiene grados que se solapen. Es decir, el valor de salario de un empleado sólo puede oscilar entre los valores de salario bajo y alto de una de las filas de la tabla de grados de salario.

Todos los salarios de los empleados oscilan entre los límites proporcionados por la tabla de grados de cargo. Es decir, ningún empleado gana menos que el valor más bajo de la columna LOWEST\_SAL o más que el valor más alto de la columna HIGHEST\_SAL.

**Nota:** se pueden utilizar otras condiciones (como <= y >=), pero **BETWEEN** es la más simple. Recuerde especificar primero el valor bajo y, a continuación, el alto al utilizar **BETWEEN**. El servidor Oracle convierte la condición **BETWEEN** en un par de condiciones **AND**. Por lo tanto, el uso de **BETWEEN** no proporciona ninguna ventaja, se recomienda sólo para la simplicidad lógica.

En el ejemplo de la diapositiva se han especificado los alias de tabla por motivos de rendimiento, no por una posible ambigüedad.

## OUTER JOINTS



**Al empleado “Grant” no se le ha asignado un ID departamento.**

**…**

**No hay ningún empleado en el departamento 190.**

**Uniones Igualitarias con EMPLOYEES**

**DEPARTMENTS**

**Devolución de Registros sin Coincidencia Directa con las uniones OUTER**

### Devolución de Registros sin Coincidencia Directa con Uniones OUTER

Si una fila no cumple una condición de unión, la fila no aparece en el resultado de consultas. En el ejemplo de la diapositiva, se utiliza una condición de unión igualitaria en las tablas EMPLOYEES y DEPARTMENTS para devolver el resultado a la derecha. El juego de resultados no contiene lo siguiente:

* + El ID de departamento 190, porque no hay ningún empleado con dicho ID registrado en la tabla EMPLOYEES.
  + El empleado con apellido Grant, porque a este empleado no se ha asignado un ID departamento.

Para volver al registro de departamento que no tiene ningún empleado o empleados que no tienen un departamento asignado, puede utilizar la unión **OUTER**.



**Uniones INNER frente a Uniones OUTER**

* En SQL:1999, la unión de dos tablas que devuelven sólo filas coincidentes se denomina unión INNER.
* Una unión entre dos tablas que devuelve los resultados de la unión INNER y las filas no coincidentes de las tablas izquierda (o derecha) se denomina una unión OUTER.
* Una unión entre dos tablas que devuelve los resultados de una unión INNER y los resultados de una unión izquierda y derecha da como resultado una unión OUTER completa.

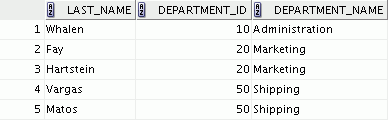
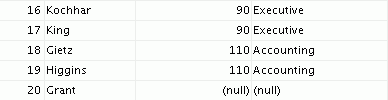
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uniones INNER frente a Uniones OUTER

La unión de las tablas con cláusulas **NATURAL JOIN, USING** u **ON** da como resultado una unión **INNER**. Cualquier fila no coincidente no aparece en la salida. Para devolver las filas no coincidentes, puede utilizar una unión **OUTER**. Una unión **OUTER** devuelve todas las filas que cumplen la condición de unión y también algunas o todas las filas de una tabla para la que ninguna fila de la otra tabla cumple la condición de unión.

Hay tres tipos de uniones **OUTER**:

* + LEFT OUTER
  + RIGHT OUTER
  + FULL OUTER



SELECT e.last\_name,

e.department\_id,

d.department\_name

FROM employees e

LEFT OUTER JOIN departments d

ON (e.department\_id = d.department\_id);

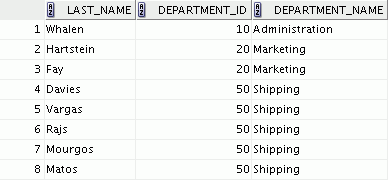


**LEFT OUTER JOIN**

**…**

### LEFT OUTER JOIN

Esta consulta recupera todas las filas de la tabla EMPLOYEES que es la tabla de la izquierda, incluso si no hay ninguna coincidencia en la tabla DEPARTMENTS.



SELECT e.last\_name,

d.department\_id,

d.department\_name

FROM employees e

RIGHT OUTER JOIN departments d

ON (e.department\_id = d.department\_id);

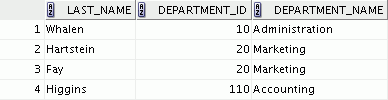


**RIGHT OUTER JOIN**

**…**

### RIGHT OUTER JOIN

Esta consulta recupera todas las filas de la tabla DEPARTMENTS que es la tabla de la derecha, incluso si no hay ninguna coincidencia en la tabla EMPLOYEES.



SELECT e.last\_name,

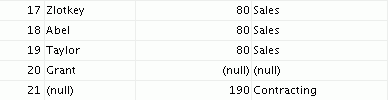
d.department\_id,

d.department\_name

FROM employees e

FULL OUTER JOIN departments d

ON (e.department\_id = d.department\_id);



**FULL OUTER JOIN**

### FULL OUTER JOIN

Esta consulta recupera todas las filas de la tabla EMPLOYEES, incluso si no hay ninguna coincidencia en la tabla DEPARTMENTS. También recupera todas las filas de la tabla DEPARTMENTS, incluso si no hay ninguna coincidencia en la tabla EMPLOYEES.

## SINTAXIS ANTIGUA. (+)

Esta es la sintaxis antigua de Oracle de LEFT OUTER JOIN antes que se estandarizase.

A partir de Oracle9i, la sintaxis de combinación externa confusa mediante la notación "(+)" ha sido reemplazada por la sintaxis de combinación externa ISO 99. Como sabemos, hay tres tipos de uniones externas, izquierda, derecha y combinación externa completa. El propósito de una combinación externa es incluir filas no coincidentes y la combinación externa devuelve estas columnas que faltan como valores NULL.

La equivalente con la sintaxis actual sería:

SELECT A.IDTABLAA,

B.NOMBRETABLAB

FROM TABLA A LEFT OUTER JOIN TABLA B

ON A.IDTABLAA = B.IDTABLAB;

La diferencia es que con el LEFT OUTER JOIN ( o (+) al lado derecho del = en el WHERE ) se incluyen en el resultado todas las filas de la tabla A a pesar que no tengan correspondencia en la tabla B colocando un NULL como valor en ese caso.

El operador (+) indica que se hará un OUTER JOIN, dependiendo de la posición del operador. Un OUTER JOIN regresa todas las filas que cumplen con la condición especificada, y además 0 o más filas de una tabla que no tengan coincidencia en la otra tabla.

Para hacer un LEFT OUTER JOIN, el operador debe aparecer en la tabla del lado derecho:

SELECT \*

FROM TABLA\_A JOIN TABLA\_B(\*);

En el caso específico de tus tablas se está realizando un LEFT OUTER JOIN, por lo que el resultado serán todas las filas que cumplan con la condición de JOIN y además las filas de A.IDTABLAA que no tengan coincidencia.

Para hacer un RIGHT OUTER JOIN, el operador debe aparecer en la tabla del lado izquierdo:

SELECT \*

FROM TABLA\_A(\*) JOIN TABLA\_B;

También hay que considerar las siguientes restricciones:

* Sólo se puede usar el operador (\*) dentro de la cláusula WHERE
* Para que los resultados sean correctos en una consulta con más de un JOIN, se debe usar el operador en todos los JOIN.
* No se puede usar para realizar un self-join
* Si en la cláusula WHERE aparece (\*), no se puede combinar con otras condiciones con los operadores AND o OR.  
    
  Este es un operador exclusivo de Oracle, por lo que si quisieras migrar este código a algún otro RDBMS es recomendable usar la sintaxis estándar de OUTER JOINS.

Revisemos las diferencias de sintaxis entre estas variaciones en la sintaxis de combinación:

Left outer join:  Oracle8i

SELECT   last\_name,

   department\_name

FROM employees e,

   departments d

WHERE e.department\_id = d.department\_id(+);

Left outer join:  Oracle9i and beyond

SELECT   last\_name,

   department\_name

FROM    employees e

LEFT OUTER JOINT   departments d

ON   e.department\_id = d.department\_id;

## PRODUCTO CARTESIANO, UNION CRUZADA.



**Productos Cartesianos**

* Un producto cartesiano se forma cuando:
  + Se omite una condición de unión
  + Una condición de unión no es válida
  + Todas las filas de la primera tabla se unen a todas las filas de la segunda tabla
* Se incluye siempre una condición de unión válida si desea evitar un producto cartesiano.

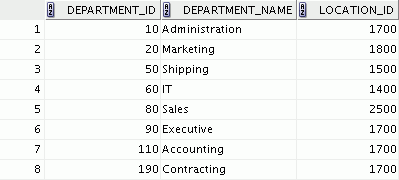
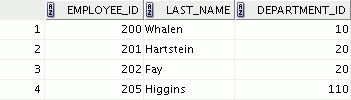
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Productos Cartesianos

Cuando una condición de unión no es válida o se omite completamente, el resultado es un *producto cartesiano*, en el que se muestran todas las combinaciones de filas. Todas las filas de la primera tabla se unen a todas las filas de la segunda tabla.

Un producto cartesiano tiende a generar un gran número de filas y el resultado es poco útil. Debe incluir siempre una condición de unión válida a menos que tenga necesidades específicas de combinar todas las filas de todas las tablas.

Los productos cartesianos son útiles para algunas pruebas para las que necesite generar un gran número de filas para simular una cantidad razonable de datos.



**…**

**…**

**Producto cartesiano:**

**20 x 8 = 160 filas**

**…**

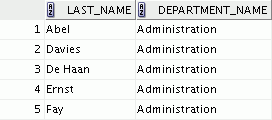
**DEPARTMENTS (8 filas)**

**EMPLOYEES (20 filas)**

**Generación de un Producto Cartesiano**

### Generación de un Producto Cartesiano

Se genera un producto cartesiano si se omite una condición de unión. En el ejemplo de la diapositiva se muestra el apellido del empleado y el nombre del departamento de las tablas EMPLOYEES y DEPARTMENTS. Ya que no se ha especificado ninguna condición de unión, todas las filas (20 filas) de la tabla EMPLOYEES se unen con todas las filas (8 filas) de la tabla DEPARTMENTS por lo que se generan 160 filas en la salida.



**Creación de Uniones Cruzadas**

* La cláusula CROSS JOIN produce el producto combinado de dos tablas.
* Esto también se denomina un producto cartesiano entre dos tablas.

**…**

SELECT last\_name,

department\_name

FROM employees

CROSS JOIN departments;

### Creación de Uniones Cruzadas

El ejemplo de la diapositiva produce un producto cartesiano de las tablas EMPLOYEES y DEPARTMENTS.

La técnica CROSS JOIN se puede aplicar a muchas situaciones de forma útil. Por ejemplo, para devolver el costo de mano de obra total por oficina por mes, incluso si el mes X no tiene costo de mano de obra, puede realizar un unión cruzada de oficinas con la tabla de todos los meses.

Es una práctica aconsejable declarar de forma explícita CROSS JOIN en SELECT si desea crear un producto cartesiano. Por lo tanto, queda muy claro que desea que esto ocurra y que no se trata del resultado de las uniones que faltan

# SUBCONSULTAS



**Uso de una Subconsulta para Solucionar Problemas**

¿Quién tiene un salario mayor que el de Abel?

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Consulta principal:**

**¿Qué empleados tienen un salario mayor que el de Abel?**

**Subconsulta:**

**¿Cuál es el salario de Abel?**

### Uso de una Subconsulta para Solucionar Problemas

Supongamos que desea escribir una consulta para saber quién gana un salario mayor que el de Abel.

Para solucionar este problema, necesita *dos* consultas: Una para saber cuánto gana Abel y otra para saber quién gana más que esa cantidad.

También puede solucionar este problema combinando las dos consultas y colocando una *dentro* de la otra.

La consulta interna (o *subconsulta*) devuelve un valor que utiliza la consulta externa (o *consulta principal*). El uso de una subconsulta es equivalente a la realización de dos consultas secuenciales y al uso del resultado de la primera consulta como el valor de búsqueda en la segunda consulta.



**Sintaxis de la Subconsulta**

* La subconsulta (consulta interna) se ejecuta una vez *antes*

de la consulta principal (consulta externa).

* La consulta principal utiliza el resultado de la subconsulta.

SELECT select\_list

FROM Tabla

WHERE expr operator(SELECT select\_list

FROM table);

## Sintaxis de la Subconsulta

Una subconsulta es una sentencia **SELECT** que está embebida en la cláusula de otra sentencia **SELECT**. Puede crear sentencias potentes a partir de sentencias simples utilizando las subconsultas. Pueden ser muy útiles cuando necesite seleccionar filas de una tabla con una condición de que dependa de los datos de la propia tabla.

Puede colocar la subconsulta en diferentes cláusulas SQL, entre las que se incluyen las siguientes:

* + Cláusula **WHERE**
  + Cláusula **HAVING**
  + Cláusula **FROM**

En la sintaxis:

***operator*** incluye una condición de comparación como **>**, **=** o **IN**

**Nota:** las condiciones de comparación pueden ser de dos clases: operadores de una sola fila (**>**, **=,** **>=, <, <>, <=**) y operadores de varias filas (**IN**, **ANY**, **ALL**, **EXISTS**).

Se suele hacer referencia a la subconsulta como una sentencia **SELECT** anidada, una subconsulta **SELECT** o una sentencia **SELECT** interna. Por lo general, la subconsulta se ejecuta en primer lugar y su resultado se utiliza para completar la condición de consulta para la consulta principal (o externa).



SELECT last\_name,

salary

FROM employees

WHERE salary > (SELECT salary

FROM employees

WHERE last\_name = 'Abel');



**Uso de Subconsultas**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de Subconsultas

En la diapositiva, la consulta interna determina el salario del empleado Abel. La consulta externa obtiene el resultado de la consulta interna y lo utiliza para mostrar a todos los empleados que ganan más que el empleado Abel.



**Instrucciones para el Uso de Subconsultas**

* Incluir subconsultas entre paréntesis.
* Coloque las subconsultas a la derecha de la condición de comparación para facilitar la legibilidad. (Sin embargo, la subconsulta puede aparecer a cualquier lado del operador de comparación.)
* Utilizar operadores de una sola fila con subconsultas de una fila y utilizar los operadores de varias filas con subconsultas de varias filas.

Instrucciones para el Uso de Subconsultas

* Una subconsulta debe estar incluida entre paréntesis.
* Coloque las subconsultas a la derecha de la condición de comparación para facilitar la legibilidad. Sin embargo, la subconsulta puede aparecer a cualquier lado del operador de comparación.
* En las subconsultas, se utilizan dos clases de condiciones de comparación: operadores de una sola fila y operadores de varias filas.



**Tipos de Subconsultas**

* Subconsulta de una sola fila

**devuelve**

**ST\_CLERK**

* Subconsulta de varias filas

**devuelve**

**ST\_CLERK SA\_MAN**

**Consulta principal**

**Subconsulta**

|  |  |  |
| --- | --- | --- |
| **Consulta principal** | | |
|  | **Subconsulta** |  |

## Tipos de Subconsultas

* **Subconsultas de una sola fila:** consultas que devuelven sólo una fila de la sentencia **SELECT**

interna.

* **Subconsultas de varias filas:** consultas que devuelven más de una fila de la sentencia **SELECT**

interna.

**Nota:** también existen varias subconsultas de varias columnas, que son consultas que devuelven más de una columna de la sentencia **SELECT** interna.

## DE UNA FILA

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| = | Igual que |
| > | Mayor que |
| >= | Mayor o igual que |
| < | Menor que |
| <= | Menor o igual que |
| <> | Distinto de |

### Subconsultas de Una Sola Fila



**Subconsultas de Una Sola Fila**

* Devuelven una sola fila
* Utilizan operadores de comparación de una sola fila

Una subconsulta de una sola fila devuelve una fila de la sentencia **SELECT** interna. Este tipo de subconsulta utiliza un operador de una sola fila. La diapositiva proporciona una lista de los operadores de una sola fila.

Ejemplo:

Muestre los empleados cuyo ID de cargo sea el mismo que el del empleado 141:

SELECT last\_name,

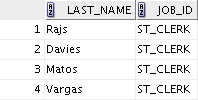
job\_id

FROM employees

WHERE job\_id = (SELECT job\_id

FROM employees

WHERE employee\_id = 141);



SELECT last\_name,

job\_id,

salary

FROM employees SA\_REP

WHERE job\_id =(SELECT job\_id

FROM employees

WHERE last\_name = 'Taylor')

AND salary > 8600

(SELECT salary

FROM employees

WHERE last\_name = 'Taylor');



**Ejecución de Subconsultas de una Sola Fila**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Ejecución de Subconsultas de una Sola Fila

Una sentencia **SELECT** se puede considerar como un bloque de consulta. El ejemplo de la diapositiva muestra los empleados que realizan el mismo trabajo que “Taylor”, pero que tienen un salario mayor que él.

El ejemplo tiene tres bloques de consulta: la consulta externa y las dos consultas internas. Los bloques de consulta internos se ejecutan primero, produciendo los resultados de consulta SA\_REP y 8600, respectivamente. A continuación, se procesa el bloque de consulta externo y utiliza los valores devueltos por las consultas internas para completar sus condiciones de búsqueda.

Tanto las consultas internas como externas devuelven valores únicos (SA\_REP y 8600, respectivamente), por lo que a esta sentencia SQL se la denomina subconsulta de una sola fila.

**Nota:** las consultas externas e internas pueden obtener datos de diferentes tablas.

SELECT last\_name,

job\_id,

salary employees

FROM employees 250

WHERE salary=(SELECT MIN(salary)

FROM employees);



**Uso de Funciones de Grupo en una Subconsulta**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de Funciones de Grupo en una Subconsulta

Puede mostrar los datos de una consulta principal utilizando una función de grupo en una subconsulta para que devuelva una sola fila. La subconsulta está entre paréntesis y se coloca al final de la condición de comparación.

El ejemplo de la diapositiva muestra el apellido del empleado, el ID de cargo y el salario de todos los empleados cuyo salario sea igual al salario mínimo. La función de grupo **MIN** devuelve un valor único (2500) a la consulta externa.

SELECT department\_id,

MIN(salary)

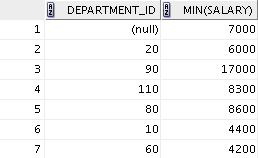
FROM employees

GROUP BY department\_id 2500

HAVING MIN(salary) >( SELECT MIN(salary)

FROM employees

WHERE department\_id = 50);



**Cláusula HAVING con Subconsultas**

* El servidor de Oracle ejecuta primero las subconsultas.
* El servidor de Oracle devuelve los resultados a la cláusula

HAVING de la consulta principal.

### Cláusula HAVING con Subconsultas

Puede utilizar las subconsultas en la cláusula **WHERE** y en la cláusula **HAVING**. El servidor de *Oracle* ejecuta la subconsulta y los resultados se devuelven en la cláusula **HAVING** de la consulta principal.

La sentencia SQL de la diapositiva muestra todos los departamentos que tienen un salario mínimo mayor que el del departamento 50.

Ejemplo:

Busque el puesto con el salario medio más bajo.

SELECT job\_id,

AVG(salary)

FROM employees

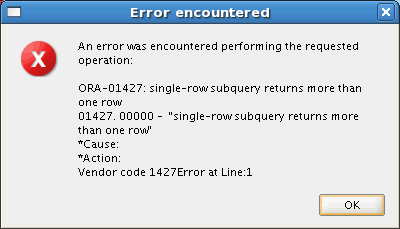
GROUP BY job\_id

HAVING AVG(salary) = ( SELECT MIN(AVG(salary))

FROM employees

GROUP BY job\_id);





SELECT employee\_id,

last\_name

FROM employees

WHERE salary =(SELECT MIN(salary)

FROM employees

GROUP BY department\_id);



**¿Qué Parte de esta Sentencia Es Incorrecta?**

**Operador de una sola fila con una subconsulta de varias filas**

¿Qué Parte de esta Sentencia Es Incorrecta?

Un error muy común en las subconsultas se produce cuando se devuelve más de una fila para una subconsulta de una sola fila.

En la sentencia SQL de la diapositiva, la subconsulta contiene una cláusula **GROUP BY**, que implica que la subconsulta devolverá varias filas, una por cada grupo que encuentre. En este caso, los resultados de la subconsulta son 4400, 6000, 2500, 4200, 7000, 17000 y 8300.

La consulta externa toma esos resultados y los utiliza en su cláusula **WHERE**. La cláusula **WHERE** contiene un operador igual (=), un operador de comparación de una sola fila que espera un único valor. El operador = no puede aceptar más de un valor de la subconsulta y, por lo tanto, genera el error.

Para corregir este error, cambie el operador = a **IN**.

SELECT last\_name,

job\_id

FROM employees

WHERE job\_id =(SELECT job\_id

FROM employees

WHERE last\_name = 'Haas');



**La Consulta Interna No Devuelve Ningún Resultado**

**La subconsulta no devuelve ninguna fila porque no hay ningún empleado con el nombre “Haas”.**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

La Consulta Interna No Devuelve Ningún Resultado

Otro problema común de las subconsultas se produce cuando la consulta interna no devuelve ninguna fila.

En la sentencia SQL de la diapositiva, la subconsulta contiene una cláusula **WHERE**. Supuestamente, la intención es encontrar el empleado cuyo nombre es Haas. La sentencia es correcta, pero no selecciona ninguna fila al ejecutarse porque no hay ningún empleado con el nombre Haas. Por lo tanto, la subconsulta no devuelve ninguna fila.

La consulta externa toma los resultados de la subconsulta (nula) y los utiliza en la cláusula **WHERE**. La consulta externa no encuentra ningún empleado con un ID de cargo igual que un valor nulo, por lo tanto, no devuelve ninguna fila. Si existía un cargo con un valor nulo, la fila no se devolverá porque la comparación de dos valores nulos tiene como resultado un valor nulo; por lo tanto, la condición **WHERE** no es verdadera.

## DE VARIAS FILAS

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| IN | Igual a cualquier miembro de la lista |
| ANY | Debe estar precedido por =, !=, >, <, <=, >=. Compara un valor con cada valor de la lista o devuelto por una consulta. Se evalúa en FALSE si la  consulta no devuelve ninguna fila. |
| ALL | Debe estar precedido por =, !=, >, <, <=, >=. Compara un valor con cada valor de la lista o devuelto por una consulta. Se evalúa en TRUE si la  consulta no devuelve ninguna fila. |

### Subconsultas de Varias Filas



**Subconsultas de Varias Filas**

* Devuelven más de una fila
* Utilizan operadores de comparación de varias filas

Las subconsultas que devuelven más de una fila se denominan subconsultas de varias filas.

Con una subconsulta de varias filas, puede utilizar un operador de varias filas en lugar de utilizar un operador de una sola fila. El operador de varias filas espera uno o más valores: ¿

SELECT last\_name,

salary,

department\_id

FROM employees

WHERE salary IN (SELECT MIN(salary)

FROM employees

GROUP BY department\_id);

Ejemplo:

Busque los empleados que ganan el mismo salario que el salario mínimo de cada departamento.

Primero se ejecuta la consulta interna, produciendo el resultado de consulta. A continuación, se procesa el bloque de consulta principal y se utilizan los valores devueltos por la consulta interna para completar la condición de búsqueda. De hecho, la consulta principal aparece en el servidor de Oracle de la siguiente forma:

SELECT last\_name,

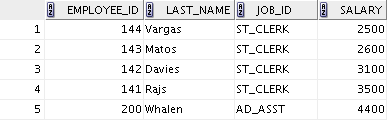
salary,

department\_id

FROM employees

WHERE salary IN (2500, 4200, 4400, 6000,

7000, 8300, 8600, 17000);



SELECT employee\_id,

last\_name,

job\_id, salary

FROM employees

WHERE salary < ANY 9000, 6000, 4200

(SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG') AND job\_id <> 'IT\_PROG';



**Uso del Operador ANY**

**en Subconsultas de Varias Filas**

**…**



### Uso del Operador ANY en Subconsultas de Varias Filas

El operador **ANY** (y su sinónimo, el operador SOME) compara un valor con *cada* valor devuelto por una subconsulta. El ejemplo de la diapositiva muestra los empleados que no son programadores de TI y cuyo salario es menor al de cualquier programador de TI. El salario máximo de un programador es de 9.000 dólares.

**<ANY** significa menos que el máximo.

**>ANY** significa más que el mínimo.

**=ANY** es igual que **IN**.



SELECT employee\_id,

last\_name,

job\_id,

salary

FROM employees

WHERE salary < ALL 9000, 6000, 4200

(SELECT salary

FROM employees

WHERE job\_id = 'IT\_PROG')

AND job\_id <> 'IT\_PROG';

**Uso del Operador ALL**

**en Subconsultas de Varias Filas**

### Uso del Operador ALL en Subconsultas de Varias Filas

El operador **ALL** compara un valor con *cada* valor devuelto por una subconsulta. El ejemplo de la diapositiva muestra los empleados cuyo salario es inferior al salario de todos los empleados con un ID de cargo IT\_PROG y cuyo trabajo no es IT\_PROG.

**>ALL** significa más que el máximo y **<ALL** significa menos que el mínimo. El operador **NOT** se puede utilizar con los operadores **IN**, **ANY** y **ALL**.

SELECT \*

FROM departments

WHERE NOT EXISTS(

SELECT \*

FROM employees

WHERE employees.department\_id =

departments.department\_id);



**Uso del Operador EXISTS**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso del Operador EXISTS

El operador **EXISTS** se utiliza en consultas en las que el resultado depende de si existen determinadas filas en la tabla o no. Se evalúa en **TRUE** si la subconsulta devuelve al menos una fila.

El ejemplo de la diapositiva muestra los departamentos que no tienen empleados. Para cada fila de la tabla DEPARTMENTS, la condición comprueba si existe o no una fila en la tabla EMPLOYEES con el mismo ID de departamento. En caso de que no exista dicha fila, la condición se cumple para la fila en cuestión y se selecciona. Si existe una fila correspondiente en la tabla EMPLOYEES, la fila no se selecciona.

SELECT emp.last\_name

FROM employees emp

WHERE emp.employee\_id NOT IN(

SELECT mgr.manager\_id

FROM employees mgr);



**Valores Nulos en una Subconsulta**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Valores Nulos en una Subconsulta

La sentencia SQL de la diapositiva intenta mostrar todos los empleados que no tienen ningún subordinado. Lógicamente, esta sentencia SQL debe haber devuelto 12 filas. Sin embargo, la sentencia SQL no devuelve ninguna fila. Uno de los valores devueltos por la consulta interna es un valor nulo y, por lo tanto, la consulta completa no devuelve ninguna fila.

El motivo es que todas las condiciones que comparan un valor nulo tienen un resultado nulo. Por lo tanto, cuando los valores nulos probablemente formen parte del juego de resultados de una subconsulta, no utilice el operador **NOT IN**. El operador **NOT IN** es equivalente a **<>** **ALL**.

Tenga en cuenta que el valor nulo que forma parte del juego de resultados de una subconsulta no representa ningún problema si se utiliza el operador **IN.** El operador **IN** es equivalente a **=ANY**.

Por ejemplo, para mostrar los empleados que tienen subordinados, utilice la siguiente sentencia SQL:

SELECT emp.last\_name

FROM employees emp

WHERE emp.employee\_id IN(

SELECT mgr.manager\_id

FROM employees mgr);

Por otro lado, una cláusula **WHERE** se puede incluir en la subconsulta para mostrar todos los empleados que no tienen ningún subordinado:

SELECT last\_name

FROM employees

WHERE employee\_id NOT IN(

SELECT manager\_id

FROM employees

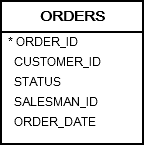
WHERE manager\_id IS NOT NULL);

## - IS NULL

En el mundo de las bases de datos, NULLes especial. Es un marcador de información faltante o la información no es aplicable.

NULLes especial en el sentido de que no es un valor como un número, una cadena de caracteres o una fecha y hora, por lo tanto, no puede compararlo con ningún otro valor como cero (0) o una cadena vacía (”). En términos generales, NULLni siquiera es igual a NULL.

Veamos la orderstabla de la [base de datos de muestra](https://www.oracletutorial.com/getting-started/oracle-sample-database/) .



La salesman\_idcolumna almacena la identificación de vendedor del vendedor que está a cargo de la orden de venta.

La siguiente [SELECT](https://www.oracletutorial.com/oracle-basics/oracle-select/)declaración intenta devolver todos los pedidos de venta que no tienen un vendedor responsable:

**SELECT** \* **FROM** orders

**WHERE** salesman\_id = NULL

**ORDER** **BY** order\_date **DESC**;

Devuelve una fila vacía.

La consulta usa el operador de comparación ( =) para comparar los valores de la salesman\_idcolumna con NULL, lo cual no es correcto.

Para verificar si un valor es NULLo no, debe usar el IS NULLoperador de la siguiente manera:

expression | column IS NULL

El IS NULLoperador devuelve verdadero si la expresión o columna es NULL. De lo contrario, devuelve falso.

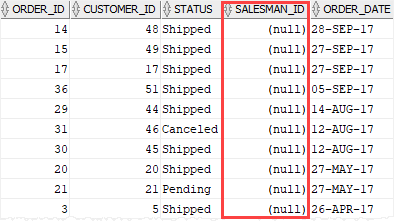
La siguiente consulta devuelve todos los pedidos de venta que no tienen un vendedor responsable:

**SELECT** \* **FROM** orders

**WHERE** salesman\_id **IS** NULL

**ORDER** **BY** order\_date **DESC**;

Aquí está el resultado parcial de la consulta:



## IS NOT NULLOperador de Oracle

Para negar el IS NULLoperador, utilice el IS NOT NULLoperador de la siguiente manera:

expression | column IS NOT NULL

El operador IS NOT NULLdevuelve verdadero si la expresión o el valor de la columna no es nulo. De lo contrario, devuelve falso.

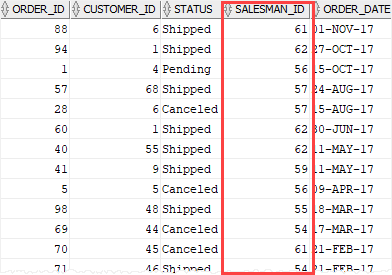
Por ejemplo, el siguiente ejemplo devuelve todos los pedidos de venta que tienen un vendedor responsable:

**SELECT** \* **FROM** orders

**WHERE** salesman\_id **IS** **NOT** NULL

**ORDER** **BY** order\_date **DESC**;

Esta imagen ilustra la salida parcial:



# DDL

## - CREATE

Para crear una nueva tabla en Oracle Database, use la CREATE TABLEdeclaración. A continuación se ilustra la sintaxis básica de la CREATE TABLEdeclaración:

**CREATE** **TABLE** schema\_name.table\_name (

column\_1 data\_type column\_constraint,

column\_2 data\_type column\_constraint,

...

table\_constraint

);

En esta sintaxis:

* Primero, especifique el nombre de la tabla y el nombre del esquema al que pertenece la nueva tabla en la CREATE TABLEcláusula.
* En segundo lugar, enumere todas las columnas de la tabla entre paréntesis. En caso de que una tabla tenga varias columnas, debe separarlas con comas (,). Una definición de columna incluye el nombre de la columna seguido de su [tipo de datos](https://www.oracletutorial.com/oracle-basics/oracle-data-types/) , por ejemplo, [NUMBER](https://www.oracletutorial.com/oracle-basics/oracle-number-data-type/), [VARCHAR2](https://www.oracletutorial.com/oracle-basics/oracle-varchar2/), y una restricción de columna tal como [NOT NULL](https://www.oracletutorial.com/oracle-basics/oracle-not-null/), [clave primaria](https://www.oracletutorial.com/oracle-basics/oracle-primary-key/) , [cheque](https://www.oracletutorial.com/oracle-basics/oracle-check-constraint/) .
* En tercer lugar, agregue restricciones de tabla si corresponde, por ejemplo, [clave principal](https://www.oracletutorial.com/oracle-basics/oracle-primary-key/) , [clave externa](https://www.oracletutorial.com/oracle-basics/oracle-foreign-key/) , [verificación](https://www.oracletutorial.com/oracle-basics/oracle-check-constraint/) .

Tenga en cuenta que debe tener CREATE TABLEprivilegios de sistema para crear una nueva tabla en su esquema y CREATE ANY TABLEprivilegios de sistema para crear una nueva tabla en el esquema de otro usuario. Además de esto, el propietario de la nueva tabla debe tener la cuota para el [espacio de tabla](https://www.oracletutorial.com/oracle-administration/oracle-tablespace/) que contiene la nueva tabla o UNLIMITED TABLESPACEprivilegio del sistema.

**CREATE TABLE**Ejemplo de declaración de Oracle

El siguiente ejemplo muestra cómo crear una nueva tabla nombrada personsen el otesquema:

**CREATE** **TABLE** ot.persons(

person\_id NUMBER **GENERATED** **BY** **DEFAULT** **AS** **IDENTITY**,

first\_name VARCHAR2(50) **NOT** NULL,

last\_name VARCHAR2(50) **NOT** NULL,

PRIMARY **KEY**(person\_id)

);

En este ejemplo, la   personstabla tiene tres columnas: person\_id, first\_name, y last\_name.

El person\_ides la [columna de identidad](https://www.oracletutorial.com/oracle-basics/oracle-identity-column/) que identifica filas únicas en la tabla. El tipo de datos de la person\_idcolumna es NUMBER. La cláusula GENERATED BY DEFAULT AS IDENTITYindica a Oracle que genere un nuevo entero para la columna siempre que se [inserte](https://www.oracletutorial.com/oracle-basics/oracle-insert/) una nueva fila en la tabla.

La first\_namecolumna tiene un tipo de datos VARCHAR2con una longitud máxima de 50. Significa que no puede insertar un nombre cuya longitud sea superior a 50 en la first\_namecolumna. Además, la [NOT NULL](https://www.oracletutorial.com/oracle-basics/oracle-not-null/)restricción de columna evita que la first\_namecolumna tenga valores NULL.

La last\_namecolumna tiene las mismas características que la first\_namecolumna.

La [PRIMARY KEY](https://www.oracletutorial.com/oracle-basics/oracle-primary-key/)cláusula especifica la person\_idcolumna como la columna de clave principal que se utiliza para identificar la fila única en la   personstabla.

### - ALIAS



**Definición de Alias de Columna**

Un alias de columna:

* Cambia el nombre de una cabecera de columna
* Es útil para realizar cálculos
* Sigue inmediatamente al nombre de columna (también puede ser la palabra clave opcional AS entre el nombre de columna y el alias)
* Necesita comillas dobles si contiene espacios o caracteres especiales o si es sensible a mayúsculas/minúsculas

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Definición de Alias de Columna

Al mostrar el resultado de una consulta, SQL Developer normalmente utiliza el nombre de la columna seleccionada como la cabecera de columna. Esta cabecera puede que no sea descriptiva y, por lo tanto, puede resultar difícil de entender. Puede cambiar una cabecera de columna mediante un alias de columna.

Especifique el alias después de la columna en la lista **SELECT** con un espacio como separador. Por defecto, las cabeceras de alias aparecen en mayúscula. Si el alias contiene espacios o caracteres especiales (como # o $) o si es sensible a mayúsculas/minúsculas, incluya el alias entre comillas dobles ("").

SELECT last\_name "Name",

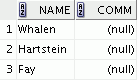
salary\*12 "Annual Salary"

FROM employees;

SELECT last\_name AS name,

commission\_pct comm

FROM employees;

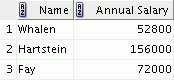


**Uso de Alias de Columna**

**…**



**Uso de Alias de Columna**



El primer ejemplo muestra los nombres y los porcentajes de comisión de todos los empleados. Observe que la palabra clave opcional AS se ha utilizado antes que el nombre de alias de columna. El resultado de la consulta es el mismo si se utiliza o no la palabra clave **AS**. Observe también que la sentencia SQL tiene los alias de columna name y comm en minúscula, mientras que el resultado de la consulta muestra las cabeceras de columna en mayúscula. Como se indica en la diapositiva anterior, las cabeceras de columna aparecen en mayúscula por defecto.

El segundo ejemplo muestra los apellidos y los salarios anuales de todos los empleados. Debido a que Annual Salary contiene un espacio, está incluido entre comillas dobles. Tenga en cuenta que la cabecera de columna de la salida es exactamente la misma que en el alias de columna.

## ALTER



**Sentencia ALTER TABLE**

Utilizar la sentencia ALTER TABLE para:

* Agregar una nueva columna
* Modificar una definición de columna existente
* Definir un valor por defecto para la nueva columna
* Borrar una columna
* Cambiar el nombre de una columna
* Cambiar la tabla al estado de sólo lectura

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Sentencia ALTER TABLE

Después de crear una tabla, puede que necesite cambiar la estructura de la tabla por cualquiera de las siguientes razones:

* + Ha omitido una columna.
  + Debe cambiar la definición de columna o su nombre.
  + Debe eliminar columnas.
  + Desea definir la tabla en modode sólo lectura. Para ello, puede utilizar la sentencia **ALTER TABLE**.

ALTER TABLE employees READ ONLY;

-- perform table maintenance and then

-- return table back to read/write mode

ALTER TABLE employees READ WRITE;



**Tablas de Sólo Lectura**

Puede utilizar la sentencia ALTER TABLE para:

* Definir una tabla en modo de sólo lectura, que evita que DDL o DML cambie durante el mantenimiento de la tabla
* Volver a definir la tabla en modo de lectura/escritura

Tablas de Sólo Lectura

Con *Oracle Database 11g*, puede especificar **READ ONLY** para definir una tabla en modo de sólo lectura. Cuando la tabla esté en modo **READ-ONLY**, no puede ejecutar ninguna sentencia DML que afecte a la tabla o a cualquier sentencia **SELECT** ... **FOR** **UPDATE**. Puede ejecutar las sentencias DDL siempre y cuando no modifique los datos de la tabla. Se permiten operaciones sobre índices asociados a la tabla cuando la tabla esté en modo **READ ONLY**.

Especifique **READ/WRITE** para volver a definir una tabla de sólo lectura en modo de lectura/escritura.

**Nota:** si es necesario puede borrar una tabla con modo **READ ONLY**. El comando **DROP** se ejecuta sólo en el diccionario de datos, por lo que no es necesario el acceso al contenido de la tabla. El espacio utilizado por la tabla no se reclamará hasta que el tablespace se vuelva a definir en lectura/escritura y, a continuación, los cambios necesarios se podrán realizar en las cabeceras de segmento de bloque, etc.

## DROP

DROP TABLE dept80;



**Borrado de una Tabla**

* Mueve una tabla a la papelera de reciclaje
* Elimina la tabla y todos sus datos completamente si se especifica la cláusula PURGE
* Invalida objetos dependientes y elimina privilegios de objeto en la tabla

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Borrado de una Tabla

La sentencia **DROP TABLE** mueve una tabla a la papelera de reciclaje o elimina la tabla y todos sus datos de la base de datos completamente. A menos que especifique la cláusula **PURGE**, la sentencia **DROP TABLE** no vuelve a liberar espacio en los tablespaces para que lo utilicen otros objetos y el espacio sigue contando en la cuota de espacio del usuario. El borrado de una tabla invalida objetos dependientes y elimina privilegios de objeto en la tabla.

Al borrar una tabla, la base de datos pierde todos los datos de la tabla y los índices asociados a los mismos.

Sintaxis:

DROP TABLE table [PURGE];

En la sintaxis, *table* es el nombre de la tabla.

Instrucciones

* Se suprimen todos los datos de la tabla.
* Se mantienen las vistas y los sinónimos, pero no son válidos.
* Se confirman las transacciones pendientes.
* Sólo el creador de la tabla o un usuario con el privilegio **DROP ANY TABLE** puede eliminar una tabla.

**Nota:** utilice la sentencia **FLASHBACK TABLE** para restaurar una tabla borrada de la papelera de reciclaje.

## - RENAME

## - TRUNCATE

### - COMMENT

# DCL

## - GRANT

## REVOKE

# CONTROL DE TRANSACCIONES



**Transacciones de Base de Datos**

Una transacción de base de datos consiste en una de las siguientes opciones:

* Sentencias DML que constituyen un cambio consistente de los datos
* Una sentencia DDL
* Una sentencia de lenguaje de control de datos (DCL)

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Transacciones de Base de Datos

El servidor de Oracle garantiza la consistencia de los datos basada en transacciones. Las transacciones proporcionan más flexibilidad y control al cambiar los datos y garantizan la consistencia de los datos en caso de fallo de proceso de usuario o del sistema.

Las transacciones constan de sentencias DML que suponen un cambio en los datos. Por ejemplo, una transferencia de fondos entre dos cuentas debe incluir un débito en una cuenta y un crédito en otra por la misma cantidad. Ambas acciones deben ser correctas o incorrectas por igual; el crédito no se debe confirmar sin el débito.

Tipos de Transacción

|  |  |
| --- | --- |
| **Tipo** | **Descripción** |
| Lenguaje de manipulación de datos (DML) | Consta de un número de sentencias DML que el servidor de Oracle trata como una entidad única o unidad de trabajo lógica. |
| Lenguaje de definición de datos (DDL) | Consta de una única sentencia DDL. |
| Lenguaje de control de datos (DCL) | Consta de una única sentencia DCL. |



**Transacciones de la Base de Datos: Inicio y Fin**

* Empezar cuando se ejecute la primera sentencia SQL de DML.
* Terminar con uno de los siguientes eventos:
  + Se emite una sentencia COMMIT o ROLLBACK.
  + Se ejecuta una sentencia DDL o DCL (confirmación automática).
  + El usuario sale de SQL Developer o SQL\*Plus.
  + El sistema falla.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

¿Cuándo se inicia y termina una transacción de la base de datos?

Una transacción empieza cuando se encuentra la primera sentencia DML y termina cuando se produce una de las siguientes condiciones:

* Se emite una sentencia COMMIT o ROLLBACK.
* Se emite una sentencia DDL, como CREATE.
* Se emite una sentencia DCL.
* El usuario sale de SQL Developer o SQL\*Plus.
* Falla una máquina o el sistema.

Tras la finalización de una transacción, la siguiente sentencia SQL ejecutable inicia automáticamente la próxima transacción.

Se confirma automáticamente una nueva sentencia DDL o DCL y, por lo tanto, finaliza de forma implícita una transacción.

**Ventajas de las Sentencias COMMIT**

**y ROLLBACK**

Con las sentencias COMMIT y ROLLBACK, puede:

* Garantizar la consistencia
* Visualizar una presentación preliminar de los cambios de los datos antes de hacerlos permanentes
* Agrupar componentes relacionados de forma lógica



### Ventajas de las Sentencias COMMIT y ROLLBACK

Con las sentencias **COMMIT** y **ROLLBACK**, puede controlar cuándo hacer permanentes los cambios realizados a los datos.



**ROLLBACK**

**ROLLBACK**

**to SAVEPOINT A**

**ROLLBACK**

**to SAVEPOINT B**

**SAVEPOINT B**

**SAVEPOINT A**

**Sentencias de Control de Transacciones Explícitas**

***Hora COMMIT***

**Transacción**

**INSERT**

**UPDATE**

**INSERT**

**DELETE**

Sentencias de Control de Transacciones Explícitas

Puede controlar la lógica de las transacciones mediante las sentencias **COMMIT**, **SAVEPOINT** y **ROLLBACK**.

|  |  |
| --- | --- |
| **Sentencia** | **Descripción** |
| COMMIT | COMMIT finaliza la transacción actual convirtiendo todos los cambios de datos pendientes en permanentes. |
| SAVEPOINT *name* | SAVEPOINT marca un punto de grabación en la transacción actual. |
| ROLLBACK | ROLLBACK finaliza la transacción actual descartando todos los cambios de datos pendientes. |
| ROLLBACK TO  *SAVEPOINT name* | ROLLBACK TO SAVEPOINT realiza un rollback de la transacción actual en el punto de grabación especificado, descartando de esta forma la posibilidad de realizar cambios y puntos de grabación creados después del punto de grabación en el que está realizando el rollback. Si omite la cláusula TO SAVEPOINT, la sentencia ROLLBACK realiza un rollback  de toda la transacción. Puesto que los puntos de grabación son lógicos, no hay forma de mostrar lo que ha creado. |

**Nota:** no puede aplicar la sentencia **COMMIT** en **SAVEPOINT**.

**SAVEPOINT** no es SQL estándar de ANSI.

UPDATE...

SAVEPOINT update\_done;



INSERT...





**Rollback de los Cambios de un Marcador**

* Crear un marcador en una transacción actual mediante la sentencia SAVEPOINT.
* Realizar rollback en dicho marcador mediante la sentencia

ROLLBACK TO SAVEPOINT.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Rollback de los Cambios de un Marcador

Puede crear un marcador en la transacción actual mediante la sentencia **SAVEPOINT**, que divide la transacción en secciones más pequeñas. Puede descartar los cambios pendientes hasta dicho marcador con la sentencia **ROLLBACK TO SAVEPOINT**.

Si crea un segundo punto de grabación con el mismo nombre que un punto de grabación anterior, éste se suprime.



**Procesamiento de Transacciones Implícitas**

* Una transacción automática se produce en las siguientes circunstancias:
  + Se emite una sentencia DDL.
  + Se emite una sentencia DCL.
  + Salida normal de SQL Developer o SQL\*Plus, sin emitir explícitamente las sentencias COMMIT o ROLLBACK.
* Un rollback automático se produce con la terminación anormal de SQL Developer o SQL\*Plus o por un fallo del sistema.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Procesamiento de Transacciones Implícitas

|  |  |
| --- | --- |
| **Estado** | **Circumstances** |
| Confirmación automática | Se emite una sentencia DDL o DCL, se sale normalmente de  SQL\*Plus sin emitir explícitamente los comandos COMMIT o ROLLBACK |
| Rollback automático | Terminación anormal de SQL Developer o SQL\*Plus o fallo del sistema |

**Nota:** en SQL\*Plus, el comando **AUTOCOMMIT** se puede activar o desactivar. Si se define en **ON**, cada sentencia DML individual se confirma en cuanto se ejecuta. No puede realizar rollback de los cambios. Si se define en **OFF**, la sentencia **COMMIT** aún se puede emitir explícitamente. Además, la sentencia **COMMIT** se emite al emitir una sentencia DDL o al salir de SQL\*Plus. El comando **SET AUTOCOMMIT ON/OFF** se salta en SQL Developer. DML se confirma en una salida normal de SQL Developer sólo si tiene activada la preferencia Autocommit. Para activar Autocommit, realice los siguientes pasos:

* En el menú Tools, seleccione Preferences. En el cuadro de diálogo Preferences, amplíe Database y seleccione Worksheet Parameters.
* En el panel de la derecha, seleccione la opción “Autocommit in SQL Worksheet”. Haga clic en OK.

**Fallos del Sistema**

Cuando se interrumpe una transacción debido a un fallo del sistema, se realiza un rollback automático de toda la transacción. Esto impide que el error provoque cambios no deseados en los datos y que devuelva las tablas a su estado en el momento de la última confirmación. De esta forma, el servidor de Oracle protege la integridad de las tablas.

En SQL Developer, la salida normal de la sesión se realiza seleccionando **Exit** en el menú File. Con SQL\*Plus, la salida normal se realiza introduciendo el comando **EXIT** en el símbolo del sistema. Si se cierra la ventana, se interpreta como una salida anormal.



**Estado de los Datos antes de COMMIT o ROLLBACK**

* Se puede recuperar el estado anterior de los datos.
* El usuario actual puede revisar los resultados de las operaciones DML mediante la sentencia SELECT.
* Los demás usuarios *no pueden* ver los resultados de las sentencias DML del usuario actual.
* Las filas afectadas están *bloqueadas*; otros usuarios no pueden cambiar los datos de las filas afectadas.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Estado de los Datos antes de COMMIT o ROLLBACK

Todos los cambios de datos realizados durante la transacción son temporales hasta que se confirma la transacción.

El estado de los datos antes de emitir las sentencias **COMMIT** o **ROLLBACK** se describe a continuación:

* Las operaciones de manipulación de datos afectan principalmente al buffer de la base de datos y, por lo tanto, se puede recuperar el estado anterior de los datos.
* El usuario actual puede revisar los resultados de las operaciones de manipulación de datos mediante la consulta de las tablas.
* Los demás usuarios no pueden ver los resultados de las operaciones de manipulación realizadas por el usuario actual. El servidor de Oracle establece la consistencia de lectura para garantizar que todos los usuarios ven los datos tal y como estaban en el momento de la última confirmación.
* Las filas afectadas están bloqueadas; otros usuarios no pueden cambiar los datos de las filas afectadas.



**Estado de los Datos después de COMMIT**

* Los cambios de datos se guardan en la base de datos.
* Se sobrescribe el estado anterior de los datos.
* Todos los usuarios pueden ver los resultados.
* Los bloqueos de las filas afectadas se liberan y dichas filas quedan disponibles para que las manipulen otros usuarios.
* Se borran todos los puntos de grabación.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Estado de los Datos después de COMMIT

Para convertir todos los cambios pendientes en permanentes, utilice la sentencia **COMMIT**. A continuación, se muestra lo que ocurre después de ejecutar una sentencia **COMMIT**:

* Los cambios de datos se escriben en la base de datos.
* El estado anterior de los datos ya no está disponible con las consultas SQL normales.
* Todos los usuarios pueden ver los resultados de la transacción.
* Los bloqueos de las filas afectadas se liberan y las filas quedan ahora disponibles para que otros usuarios realicen nuevos cambios en los datos.
* Se borran todos los puntos de grabación.

COMIT;



DELETE FROM employees

WHERE employee\_id = 99999;



INSERT INTO departments;



VALUES (290, 'Corporate Tax', NULL, 1700);



**Confirmación de Datos**

* Realice estos cambios:
* Confirme los cambios:

Confirmación de Datos

En el ejemplo de la diapositiva, se suprime una fila de la tabla EMPLOYEES y se inserta una nueva en la tabla DEPARTMENTS. Los cambios se guardan emitiendo la sentencia **COMMIT**.

Ejemplo:

Elimine los departamentos 290 y 300 de la tabla DEPARTMENTS y actualice una fila en la tabla EMPLOYEES. Guarde el cambio de datos.

DELETE FROM departments

WHERE department\_id IN (290, 300);

UPDATE employees

SET department\_id = 80

WHERE employee\_id = 206;

COMMIT;

DELETE FROM copy\_emp;

ROLLBACK ;



**Estado de los Datos después de ROLLBACK**

Desechar todos los cambios pendientes mediante la sentencia

ROLLBACK:

* Se deshacen los cambios de datos.
* Se restaura el estado anterior de los datos.
* Se liberan los bloqueos de las filas afectadas.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Estado de los Datos después de ROLLBACK

Para descartar todos los cambios pendientes, utilice la sentencia **ROLLBACK** obteniendo los siguientes resultados:

* Se deshacen los cambios de datos.
* Se restaura el estado anterior de los datos.
* Se liberan los bloqueos de las filas afectadas.

DELETE FROM test;

25,000 rows deleted.

ROLLBACK ;

Rollback complete.

SELECT \*

FROM test

WHERE id = 100;

No rows selected.

COMMIT;



**Estado de los Datos después de ROLLBACK: Ejemplo**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Estado de los Datos después de ROLLBACK: Ejemplo

Al intentar eliminar un registro de la tabla TEST, puede borrar accidentalmente la tabla. Puede corregir el error, volver a emitir la sentencia correcta y hacer permanentes los cambios de datos.



**Rollback a Nivel de Sentencias**

* Si falla una sentencia DML durante la ejecución, sólo se realiza un rollback de dicha sentencia.
* El servidor de Oracle implanta un punto de grabación implícito.
* Los demás cambios se retienen.
* El usuario debe terminar las transacciones explícitamente con la ejecución de una sentencia COMMIT o ROLLBACK.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Rollback a Nivel de Sentencias

Se puede desechar parte de una transacción mediante un rollback implícito si se detecta un error de ejecución de sentencia. Si falla una única sentencia DML durante la ejecución de una transacción, su efecto se deshace mediante un rollback a nivel de sentencia, pero los cambios realizados por las sentencias DML anteriores en la transacción no se desechan. El usuario los puede omitir o realizar un rollback explícito.

El servidor de Oracle emite una confirmación implícita antes y después de cualquier sentencia DDL. Por lo tanto, si la sentencia DDL no se ejecuta correctamente, no podrá realizar rollback de la sentencia anterior porque el servidor ha emitido una confirmación.

El usuario debe terminar las transacciones explícitamente con la ejecución de una sentencia **COMMIT** o **ROLLBACK**.

# OPERADORES DE DEFINICION



**MINUS**

**B**

**A**

**INTERSECT**

**B**

**A**

**UNION/UNION ALL**

**B**

**A**

**B**

**A**

**Operadores de Definición**

Operadores de Definición

Los operadores de definición combinan el resultado de dos o más consultas de componente en un resultado. Las consultas que contienen operadores de definición se denominan *consultas compuestas*.

|  |  |
| --- | --- |
| .  **Operador** | **Devuelve** |
| UNION | Resultados de ambas consultas después de eliminar la duplicación |
| UNION ALL | Resultados de ambas consultas, incluidas todas las duplicaciones |
| INTERSECT | Resultados comunes a ambas consultas |
| MINUS | Resultados de la primera consulta no presentes en la segunda |

Todos los operadores de definición tienen la misma prioridad. Si una sentencia SQL contiene varios operadores de definición, el servidor de Oracle los evalúa de izquierda (parte superior) a derecha (parte inferior), si no hay ningún paréntesis que especifique explícitamente otro orden. Debe utilizar los paréntesis para especificar el orden de evaluación explícitamente en las consultas que utilizan el operador **INTERSECT** en otros operadores de definición.



**Instrucciones de los Operadores de Definición**

* La expresiones de las listas SELECT debe coincidir en número.
* Los tipos de dato para cada columna de la segunda consulta deben coincidir con los tipos de dato de su columna correspondiente en la primera consulta.
* Los paréntesis se pueden utilizar para modificar la secuencia de ejecución.
* La sentencia ORDER BY puede aparecer sólo una vez al final de la sentencia.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Instrucciones de los Operadores de Definición

* La expresiones de las listas **SELECT** deben coincidir en número y tipo de dato. Las consultas que utilizan los operadores **UNION**, **UNION** **ALL**, **INTERSECT** y **MINUS** en su cláusula **WHERE** deben tener el mismo número y tipo de dato de columnas en su lista **SELECT**. El tipo de dato de las columnas de la lista **SELECT** de las consultas de la consulta compuesta puede no ser exactamente el mismo. La columna de la segunda consulta debe estar en el mismo grupo de tipo de dato (por ejemplo, numérico o de caracteres) que la columna correspondiente de la primera consulta.
* Los operadores de definición se pueden utilizar en subconsultas.
* Debe utilizar los paréntesis para especificar el orden de evaluación en las consultas que utilizan el operador **INTERSECT** en otros operadores de definición. Esto garantiza el cumplimiento de los emergentes estándares SQL que otorgarán al operador **INTERSECT** mayor prioridad que los operadores de definición.



**Servidor de Oracle y Operadores de Definición**

* Las filas duplicadas se eliminan automáticamente excepto en UNION ALL.
* Los nombres de columna de la primera consulta aparecen en el resultado.
* Por defecto, la salida se ordena en orden ascendente, excepto en UNION ALL.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Servidor de Oracle y Operadores de Definición

Cuando una consulta utiliza operadores de definición, el servidor de *Oracle* elimina automáticamente las filas duplicadas, excepto en el caso del operador **UNION ALL**. Los nombres de columna de la salida están determinados por la lista de columnas de la primera sentencia **SELECT**. Por defecto, la salida se ordena en orden ascendente según la primera columna de la cláusula **SELECT**.

Las expresiones correspondientes en las listas **SELECT** de las consultas de componente de una consulta compuesta deben coincidir en número y tipo de dato. Si las consultas de componente seleccionan datos de caracteres, el tipo de dato del valor de retorno se determina de la siguiente forma:

* Si ambas consultas seleccionan valores del tipo de dato **CHAR**, de la misma longitud, los valores devueltos tienen el mismo tipo de dato **CHAR** con esa longitud. Si las consultas seleccionan valores del tipo **CHAR** con diferentes longitudes, el valor devuelto es **VARCHAR2** con la longitud del valor **CHAR** más largo.
* Si una o ambas consultas seleccionan valores del tipo de dato **VARCHAR2**, los valores devueltos tienen un tipo de dato **VARCHAR2**.

Si las consultas de componente seleccionan datos numéricos, el tipo de dato del valor de retorno se determina según la prioridad numérica. Si todas las consultas seleccionan valores del tipo **NUMBER**, los valores devueltos tienen el tipo de dato **NUMBER**. En las consultas que utilizan operadores de definición, el servidor de Oracle no realiza una conversión explícita a través de los grupos de tipos de dato. Por lo tanto, si las expresiones correspondientes de las consultas de componente se resuelven tanto en datos numéricos como de caracteres, el servidor de Oracle devuelve un error.

## UNION/UNION ALL



**El operador UNION devuelve los resultados de ambas consultas después de eliminar la duplicación.**

**B**

**A**

**Operador UNION**

## Operador UNION

El operador **UNION** devuelve todas las filas seleccionadas en cualquier consulta. Utilice el operador **UNION** para devolver todas las filas de varias tablas y eliminar las filas duplicadas.

Instrucciones:

* El número de columnas seleccionadas debe ser el mismo.
* Los tipos de dato de las columnas seleccionadas deben pertenecer al mismo grupo de tipo de dato (por ejemplo, numérico o de caracteres).
* No es necesario que los nombres de las columnas sean idénticos.
* El operador **UNION** funciona en todas las columnas seleccionadas.
* Los valores **NULL** no se ignoran durante la comprobación de duplicados.
* Por defecto, la salida se ordena en orden ascendente según de las columnas de la cláusula

SELECT.



SELECT employee\_id, job\_id

FROM employees

UNION SELECT employee\_id, job\_id

FROM job\_history;



**Uso del Operador UNION**

Mostrar los detalles actuales y anteriores del puesto de todos los empleados. Mostrar cada empleado sólo una vez.

**…**

**…**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Uso del Operador UNION

El operador **UNION** elimina cualquier registro duplicado. Si los registros de las tablas EMPLOYEES y JOB\_HISTORY son idénticos, éstos sólo aparecerán una vez. Observe en la salida de la diapositiva que el registro del empleado con un EMPLOYEE\_ID de 200 aparece dos veces porque JOB\_ID es diferente en cada fila.

Considere el siguiente ejemplo:

SELECT employee\_id,

job\_id,

department\_id

FROM employees

UNION

SELECT employee\_id,

job\_id,

department\_id

FROM job\_history;





En la salida anterior, el empleado 200 aparece tres veces. ¿Por qué? Observe los valores de DEPARTMENT\_ID del empleado 200. Una fila tiene un DEPARTMENT\_ID de 90, otra 10 y la tercera, 90. Debido a estas combinaciones únicas de ID de cargos y de departamentos, cada fila del empleado 200 es única y, por lo tanto, no se considera un duplicado. Observe que la salida está ordenada en orden ascendente según la primera columna de la cláusula **SELECT** (en este caso, EMPLOYEE\_ID).



**El operador UNION ALL devuelve los resultados de ambas consultas, incluidas todas las duplicaciones.**

**B**

**A**

**Operador UNION ALL**

## Operador UNION ALL

Utilice el operador **UNION ALL** para devolver todas las filas de varias consultas.

Instrucciones:

Las instrucciones para **UNION** y **UNION ALL** son las mismas, excepto en los dos siguientes casos que pertenecen a **UNION ALL**: a diferencia de **UNION**, las filas duplicadas no se eliminan y la salida no se ordena por defecto.



**Uso del Operador UNION ALL**

Mostrar los departamentos actuales y anteriores de todos los empleados.

**…**

SELECT employee\_id,

job\_id,

department\_id

FROM employees

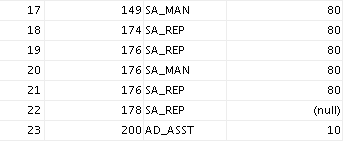
UNION ALL

SELECT employee\_id,

job\_id,

department\_id

FROM job\_history;



**…**

Uso del Operador UNION ALL

En el ejemplo se seleccionan 30 filas. La combinación de las dos tablas tiene un total de 30 filas. El operador **UNION ALL** no elimina las filas duplicadas. **UNION** devuelve todas las filas distintas seleccionadas por cualquier consulta. **UNION ALL** devuelve todas las filas seleccionadas por cualquier consulta, incluyendo las duplicadas. Considere la consulta de la diapositiva escrita ahora con la cláusula **UNION**:

SELECT employee\_id,

job\_id,

department\_id

FROM employees

UNION

SELECT employee\_id,

job\_id,

department\_id

FROM job\_history

ORDER BY employee\_id;

La consulta anterior devuelve 29 filas. Esto se debe a que elimina la siguiente fila (porque está duplicada):



## INTERSECT



**El operador INTERSECT devuelve filas comunes a ambas consultas.**

**B**

**A**

**Operador INTERSECT**

## Operador INTERSECT

Utilice el operador **INTERSECT** para devolver todas las filas comunes a varias consultas.

Instrucciones

* El número de columnas y los tipos de dato de las columnas seleccionadas por las sentencias **SELECT** en las consultas deben ser idénticos en todas las sentencias **SELECT** utilizadas en la consulta. No es necesario, sin embargo, que los nombres de las columnas sean idénticos.
* Si se invierte el orden de las tablas intersectadas no se alterará el resultado.
* **INTERSECT** no ignora los valores **NULL**.

SELECT employee\_id,

job\_id FROM employees

INTERSECT

SELECT employee\_id,

job\_id

FROM job\_history;



**Uso del Operador INTERSECT**

Mostrar los ID de empleado y de cargo de los empleados que actualmente tienen el mismo puesto que anteriormente (es decir, han cambiado de cargo pero ahora han vuelto a realizar el mismo trabajo que realizaban anteriormente).



Uso del Operador **INTERSECT**

En el ejemplo de la diapositiva, la consulta devuelve sólo los registros que tienen los mismos valores en las columnas seleccionadas en ambas tablas.

¿Cuál será el resultado si agrega la columna DEPARTMENT\_ID a la sentencia SELECT de la tabla EMPLOYEES y la columna DEPARTMENT\_ID a la sentencia SELECT de la tabla JOB\_HISTORY y ejecuta esta consulta? El resultado puede variar debido a la introducción de otra columna cuyos valores pueden o no estar duplicados.

Ejemplo:

SELECT employee\_id,

job\_id,

department\_id

FROM employees

INTERSECT

SELECT employee\_id,

job\_id,

department\_id

FROM job\_history;



El empleado 200 ya no forma parte de los resultados porque el valor EMPLOYEES.DEPARTMENT\_ID es diferente del valor JOB\_HISTORY.DEPARTMENT\_ID.

## MINUS



**El operador MINUS devuelve todas las filas distintas seleccionadas por la primera consulta, pero que no están presentes en el juego de resultados de la segunda consulta.**

**B**

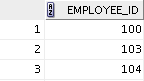
**A**

**Operador MINUS**

## Operador MINUS

Utilice el operador **MINUS** para devolver todas las filas distintas seleccionadas por la primero consulta, pero que no están presentes en el juego de resultados de la segunda consulta (la primera sentencia SELECT **MINUS** la segunda sentencia **SELECT**).

**Nota:** el número de columnas y los tipos de dato de las columnas seleccionadas por las sentencias **SELECT** de las consultas deben pertenecer al mismo grupo de tipo de dato en todas las sentencias **SELECT** utilizadas en la consulta. No es necesario, sin embargo, que los nombres de las columnas sean idénticos.



SELECT employee\_id

FROM employees

MINUS

SELECT employee\_id

FROM job\_history;



**Uso del Operador MINUS**

Mostrar los identificadores de empleado cuyos empleados no han cambiado sus puestos ni una vez.

**…**



Uso del Operador **MINUS**

En el ejemplo de la diapositiva, los ID de empleado de la tabla JOB\_HISTORY se restan de los de la tabla EMPLOYEES. El juego de resultados muestra los empleados resultantes después de la resta; están representados por filas que existen en la tabla EMPLOYEES pero que no existen en la tabla JOB\_HISTORY. Éstos son los registros de los empleados que no han cambiado sus puestos ni una vez.

# TABLAS ANIDADAS

CREATE TABLE *table*

[(*column*, *column*...)]

AS *subquery;*



**Creación de una Tabla mediante una Subconsulta**

* Crear una tabla e insertar filas combinando la sentencia

CREATE TABLE y la opción AS *subquery*.

* Hacer coincidir el número de columnas especificadas con el número de columnas de la subconsulta.
* Definir columnas con nombres de columna y valores por defecto.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Creación de una Tabla mediante una Subconsulta

Un segundo método para crear tablas es aplicar la cláusula AS *subquery*, que crea la tabla e inserta las filas desvueltas de la subconsulta.

En la sintaxis:

***table***es el nombre de la tabla

***column***es el nombre de la columna, el valor por defecto y la restricción de integridad

***subquery***es la sentencia **SELECT** que define el juego de filas que se va a insertar en la nueva tabla

Instrucciones:

* + La tabla se crea con los nombres de columna especificados y las filas recuperadas por la sentencia **SELECT** se insertan en la tabla.
  + La definición de columna sólo puede contener el nombre de la columna y el valor por defecto.
  + Si se proporcionan especificaciones de columna, el número de columnas debe ser igual al número de columnas de la lista **SELECT** de la subconsulta.
  + Si no se proporcionan especificaciones de columna, los nombres de columna de la tabla son los mismos que los de la subconsulta.
  + Las definiciones de tipo de dato de columna y de la restricción **NOT** **NULL** se transfieren a la nueva tabla. Tenga en cuenta que sólo la restricción explícita **NOT NULL** se heredará. La columna **PRIMARY** **KEY** no transferirá la función **NOT NULL** a la nueva columna. Las demás reglas de restricción no se transfieren a la nueva tabla. Sin embargo, puede agregar restricciones en la definición de columna.



DESCRIBE dept80;

CREATE TABLE dept80

AS

SELECT employee\_id,

last\_name,

salary\*12 ANNSAL,

hire\_date

FROM employees

WHERE department\_id = 80;



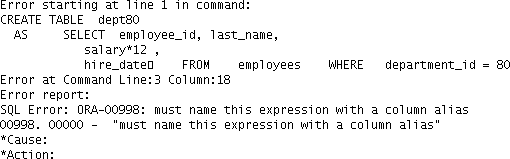
**Creación de una Tabla mediante una Subconsulta**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

En el ejemplo de la diapositiva, se crea una tabla denominada DEPT80, que contiene los detalles de todos los empleados que trabajan en el departamento 80. Tenga en cuenta que los datos de la tabla DEPT80 proceden de la tabla EMPLOYEES.

Puede verificar la existencia de una tabla de base de datos y comprobar las definiciones de columna mediante el comando **DESCRIBE**.

Sin embargo, asegúrese de proporcionar un alias de columna al seleccionar una expresión. La expresión SALARY\*12 recibe el alias ANNSAL. Sin el alias, se genera el siguiente error:



# Operadores Multiset

Los operadores de conjuntos múltiples combinan los resultados de dos tablas anidadas en una sola tabla anidada.

Los ejemplos relacionados con los operadores de conjuntos múltiples requieren que se creen dos tablas anidadas y se carguen con los datos de la siguiente manera:

Primero, haga una copia de la tabla oe.customers llamada customers\_demo:

CREATE TABLE customers\_demo

AS

SELECT \*

FROM customers;

A continuación, cree un tipo de tabla llamada cust\_address\_tab\_typ. Este tipo se utilizará al crear las columnas de la tabla anidada.

CREATE TYPE cust\_address\_tab\_typ

AS

TABLE OF cust\_address\_typ;

/

Ahora, crea dos columnas de tabla anidadas en la tabla customers\_demo:?

ALTER TABLE customers\_demo

ADD ( cust\_address\_ntab,

cust\_address\_tab\_typ,

cust\_address2\_ntab,

cust\_address\_tab\_typ)

NESTED TABLE cust\_address\_ntab STORE AS cust\_address\_ntab\_store

NESTED TABLE cust\_address2\_ntab STORE AS cust\_address2\_ntab\_store;

Finalmente, cargue los datos en las dos nuevas columnas de la tabla anidada usando los datos de la columna cust\_address de la tabla oe.customers:

UPDATE customers\_demo cd

SET cust\_address\_ntab =

CAST(MULTISET(SELECT cust\_address

FROM customers c

WHERE c.customer\_id =

cd.customer\_id) as cust\_address\_tab\_typ);

UPDATE customers\_demo cd

SET cust\_address2\_ntab =

CAST(MULTISET(SELECT cust\_address

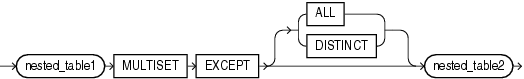
FROM customers c

WHERE c.customer\_id =

cd.customer\_id) as cust\_address\_tab\_typ);

## MULTISET EXCEPTO

MULTISET EXCEPT toma como argumentos dos tablas anidadas y devuelve una tabla anidada cuyos elementos están en la primera tabla anidada pero no en la segunda tabla anidada. Las dos tablas anidadas de entrada deben ser del mismo tipo, y la tabla anidada devuelta también es del mismo tipo.

  
[Descripción de la ilustración '' multiset\_except.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/multiset_except.htm)

* La palabra clave ALL le indica a Oracle que devuelva todos los elementos *nested\_table1* que no están en *nested\_table2*. Por ejemplo, si un elemento en particular aparece varias *m* v eces *nested\_table1* y luego varias *n* veces *nested\_table2*, el resultado tendrá ocurrencias *(m-n)* del elemento if *m >n* y 0 ocurrencias si *m<=n*. ALL es el predeterminado
* La palabra clave DISTINCT le indica a Oracle que elimine cualquier elemento en el *nested\_table1* que también se encuentre *nested\_table2*, independientemente del número de ocurrencias.
* Los tipos de elementos de las tablas anidadas deben ser comparables. Consulte ["Condiciones de comparación"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions002.htm#i1033286) para obtener información sobre la comparabilidad de los tipos no escalares.

**Ejemplo**

El siguiente ejemplo compara dos tablas anidadas y devuelve una tabla anidada de aquellos elementos que se encuentran en la primera tabla anidada pero no en la segunda tabla anidada:

SELECT customer\_id, cust\_address\_ntab

MULTISET EXCEPT DISTINCT cust\_address2\_ntab multiset\_except

FROM customers\_demo

ORDER BY customer\_id;

CUSTOMER\_ID MULTISET\_EXCEPT(STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID)

----------- --------------------------------------------------------------------------------

101 CUST\_ADDRESS\_TAB\_TYP()

102 CUST\_ADDRESS\_TAB\_TYP()

103 CUST\_ADDRESS\_TAB\_TYP()

104 CUST\_ADDRESS\_TAB\_TYP()

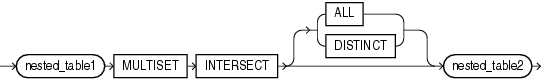
105 CUST\_ADDRESS\_TAB\_TYP()

. . .

El ejemplo anterior requiere la tabla customers\_demo y dos columnas de tabla anidadas que contienen datos. Consulte ["Operadores de conjuntos múltiples"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/operators006.htm#i1035617) para crear esta tabla y columnas de tablas anidadas.

## MULTISET INTERSECT

MULTISET INTERSECT toma como argumentos dos tablas anidadas y devuelve una tabla anidada cuyos valores son comunes en las dos tablas anidadas de entrada. Las dos tablas anidadas de entrada deben ser del mismo tipo, y la tabla anidada devuelta también es del mismo tipo.

  
[Descripción de la ilustración '' multiset\_intersect.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/multiset_intersect.htm)

* La palabra clave ALL le indica a Oracle que devuelva todas las apariciones comunes de elementos que se encuentran en las dos tablas anidadas de entrada, incluidos los valores comunes duplicados y las apariciones  NULL comunes duplicadas . Por ejemplo, si un valor en particular ocurre m veces en *nested\_table1* y n tiempos en *nested\_table2*, entonces el resultado contendrá los min(m,n) tiempos del elemento . ALL es el predeterminado
* La palabra clave DISTINCT le indica a Oracle que elimine los duplicados de la tabla anidada devuelta, incluidos los duplicados de NULL, si existen.
* Los tipos de elementos de las tablas anidadas deben ser comparables. Consulte ["Condiciones de comparación"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions002.htm#i1033286) para obtener información sobre la comparabilidad de los tipos no escalares.

**Ejemplo**

El siguiente ejemplo compara dos tablas anidadas y devuelve una tabla anidada de los elementos encontrados en ambas tablas anidadas de entrada:

SELECT customer\_id, cust\_address\_ntab

MULTISET INTERSECT DISTINCT cust\_address2\_ntab multiset\_intersect

FROM customers\_demo

ORDER BY customer\_id;

CUSTOMER\_ID MULTISET\_INTERSECT(STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID

----------- -----------------------------------------------------------------------------------

101 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('514 W Superior St', '46901', 'Kokomo', 'IN', 'US'))

102 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('2515 Bloyd Ave', '46218', 'Indianapolis', 'IN', 'US'))

103 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('8768 N State Rd 37', '47404', 'Bloomington', 'IN', 'US'))

104 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('6445 Bay Harbor Ln', '46254', 'Indianapolis', 'IN', 'US'))

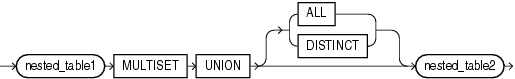
105 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('4019 W 3Rd St', '47404', 'Bloomington', 'IN', 'US'))

. . .

El ejemplo anterior requiere la tabla customers\_demo y dos columnas de tabla anidadas que contienen datos. Consulte ["Operadores de conjuntos múltiples"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/operators006.htm#i1035617) para crear esta tabla y columnas de tablas anidadas.

## Unión múltiple

MULTISET UNION toma como argumentos dos tablas anidadas y devuelve una tabla anidada cuyos valores son los de las dos tablas anidadas de entrada. Las dos tablas anidadas de entrada deben ser del mismo tipo, y la tabla anidada devuelta también es del mismo tipo.

  
[Descripción de la ilustración '' multiset\_union.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/multiset_union.htm)

* La palabra clave ALL le indica a Oracle que devuelva todos los elementos que se encuentran en las dos tablas anidadas de entrada, incluidos los valores duplicados y las apariciones NULL duplicadas . Este es el valor predeterminado.
* La palabra clave DISTINCT le indica a Oracle que elimine los duplicados de la tabla anidada devuelta, incluidos los duplicados de NULL, si existen.
* Los tipos de elementos de las tablas anidadas deben ser comparables. Consulte ["Condiciones de comparación"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/conditions002.htm#i1033286) para obtener información sobre la comparabilidad de los tipos no escalares.

**Ejemplo**

El siguiente ejemplo compara dos tablas anidadas y devuelve una tabla anidada de elementos de ambas tablas anidadas de entrada:

SELECT customer\_id, cust\_address\_ntab

MULTISET UNION cust\_address2\_ntab multiset\_union

FROM customers\_demo

ORDER BY customer\_id;

CUSTOMER\_ID MULTISET\_UNION(STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID)

----------- -------------------------------------------------------------------------------

101 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('514 W Superior St', '46901', 'Kokomo', 'IN', 'US'),

CUST\_ADDRESS\_TYP('514 W Superior St', '46901', 'Kokomo', 'IN', 'US'))

102 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('2515 Bloyd Ave', '46218', 'Indianapolis', 'IN', 'US'),

CUST\_ADDRESS\_TYP('2515 Bloyd Ave', '46218', 'Indianapolis', 'IN','US'))

103 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('8768 N State Rd 37', '47404', 'Bloomington', 'IN', 'US'),

CUST\_ADDRESS\_TYP('8768 N State Rd 37', '47404', 'Bloomington', 'IN', 'US'))

104 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('6445 Bay Harbor Ln', '46254', 'Indianapolis', 'IN', 'US'),

CUST\_ADDRESS\_TYP('6445 Bay Harbor Ln', '46254', 'Indianapolis', 'IN', 'US'))

105 CUST\_ADDRESS\_TAB\_TYP(CUST\_ADDRESS\_TYP('4019 W 3Rd St', '47404', 'Bloomington', 'IN', 'US'),

CUST\_ADDRESS\_TYP('4019 W 3Rd St', '47404', 'Bloomington', 'IN', 'US'))

. . .

El ejemplo anterior requiere la tabla customers\_demo y dos columnas de tabla anidadas que contienen datos. Consulte ["Operadores de conjuntos múltiples"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/operators006.htm#i1035617) para crear esta tabla y columnas de tablas anidadas.

# Operadores definidos por el usuario

Al igual que los operadores integrados, los operadores definidos por el usuario toman un conjunto de operandos como entrada y devuelven un resultado. Sin embargo, los creas con la declaración CREATE OPERATOR y se identifican con nombres definidos por el usuario. Residen en el mismo espacio de nombres que las tablas, vistas, tipos y funciones independientes.

Después de haber definido un nuevo operador, puede usarlo en sentencias de SQL como cualquier otro operador integrado. Por ejemplo, puede usar operadores definidos por el usuario en la lista de selección de una declaración SELECT, la condición de una cláusulas WHERE o en cláusulas ORDER BY y cláusulas GROUP BY. Sin embargo, debe tener privilegios EXECUTE sobre el operador para hacerlo, porque es un objeto definido por el usuario.

SELECT location\_id,

department\_name "Department",

TO\_CHAR(NULL) "Warehouse location"

FROM departments

UNION

SELECT location\_id,

TO\_CHAR(NULL) "Department",

state\_province

FROM locations;



**Coincidencia de las Sentencias SELECT**

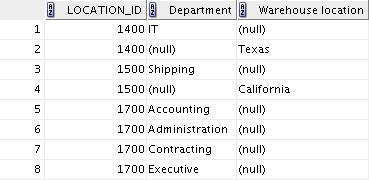
* Con el operador UNION, se muestra el ID de ubicación, el nombre de departamento y el estado en el que está ubicado.
* Debe hacer que coincida el tipo de dato (mediante la función TO\_CHAR o cualquier otra función de conversión) cuando las columnas no existan en una tabla o en la otra.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

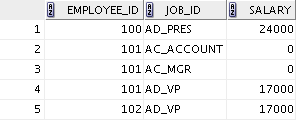
## Coincidencia de las Sentencias SELECT

Debido a que las expresiones de las listas **SELECT** de las consultas deben coincidir en número, puede utilizar columnas ficticias y funciones de conversión de tipos de dato para cumplir con esta regla. En la diapositiva, se asigna el nombre Warehouse location como la cabecera de columna ficticia. La función **TO\_CHAR** se utiliza en la primera consulta para que coincida el tipo de dato **VARCHAR2** de la columna state\_province que recupera la segunda consulta. Igualmente, la función **TO\_CHAR** se utiliza en la segunda consulta para que coincida el tipo de dato **VARCHAR2** de la columna department\_name que recupera la primera consulta.

La salida de la consulta se muestra a continuación:



**…**



SELECT employee\_id,

job\_id,

salary

FROM employees

UNION

SELECT employee\_id,

job\_id,0

FROM job\_history;



**Coincidencia de las Sentencias SELECT: Ejemplo**

Utilizar el operador UNION, mostrar el ID de empleado, ID de cargo y salario de todos los empleados.

**…**



Coincidencia de las Sentencias SELECT:

Ejemplo:

Las tablas EMPLOYEES y JOB\_HISTORY tienen varias columnas en común (por ejemplo, EMPLOYEE\_ID, JOB\_ID y DEPARTMENT\_ID). Pero, ¿y si lo que desea es que la consulta muestre el ID de empleado, ID de cargo y salario con el operador UNION sabiendo que el salario sólo existe en la tabla EMPLOYEES?

El código de ejemplo de la diapositiva coinciden las columnas EMPLOYEE\_ID y JOB\_ID de las tablas EMPLOYEES y JOB\_HISTORY. Se agrega el valor literal 0 a la sentencia JOB\_HISTORY SELECT para que coincida con columna numérica SALARY de la sentencia EMPLOYEES SELECT.

En los resultados mostrados en la diapositiva, cada fila de la salida que corresponde a un registro de la tabla JOB\_HISTORY contiene un 0 en la columna SALARY.



**Uso de la Cláusula ORDER BY en Operaciones de Definición**

* La cláusula ORDER BY sólo puede aparecer una vez al final de la consulta compuesta.
* Las consultas de componente no pueden tener cláusulas

ORDER BY individuales.

* La cláusula ORDER BY reconoce sólo las columnas de la primera consulta SELECT.
* Por defecto, la primera columna de la primera consulta SELECT se utiliza para ordenar la salida en orden ascendente.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Uso de la Cláusula ORDER BY en Operaciones de Definición

La cláusula **ORDER BY** sólo se puede utilizar una vez en una consulta compuesta. Si se utiliza, la cláusula **ORDER BY** se debe colocar al final de la consulta. La cláusula **ORDER BY** acepta el nombre de columna o alias. Por defecto, la salida se ordena en orden ascendente según la primera columna de la cláusula **SELECT**.

**Nota:** la cláusula **ORDER BY** no reconoce los nombres de columna de la segunda consulta **SELECT**. Para evitar la confusión con los nombres de columna, es una práctica habitual utilizar cláusula **ORDER BY** según las posiciones de las columnas.

Por ejemplo, en la siguiente sentencia, la salida se mostrará en orden ascendente según job\_id.

SELECT employee\_id,

job\_id,salary

FROM employees

UNION

SELECT employee\_id,

job\_id,0

FROM job\_history

ORDER BY 2;

Si omite **ORDER BY**, por defecto, la salida se ordenará en orden ascendente según employee\_id. No puede utilizar las columnas de la segunda consulta para ordenar la salida.

# OBJETOS DE LA BBDD



**Objetos de Base de Datos**

|  |  |
| --- | --- |
| **Objeto** | **Descripción** |
| Tabla | Unidad básica de almacenamiento; compuesta por filas. |
| Vista | Representa de forma lógica subjuegos de datos de una o más tablas. |
| Secuencia | Genera valores numéricos. |
| Índice | Mejora el rendimiento de las consultas de recuperación de datos. |
| Sinónimo | Ofrece nombres alternativos para los objetos. |

## Objetos de Base de Datos

Existen otros objetos en la base de datos además de las tablas. Con las vistas, puede presentar y ocultar datos de las tablas.

Muchas aplicaciones exigen el uso de números únicos como valores de clave primaria. Puede crear código en la aplicación para cumplir este requisito o utilizar una secuencia para generar números únicos.

Si desea mejorar el rendimiento de las consultas de recuperación de datos, debe crear un índice. También puede utilizar índices para reforzar la unicidad en una columna o recopilación de columnas.

Puede proporcionar nombres alternativos para los objetos mediante sinónimos.



**Objetos de Base de Datos**

|  |  |
| --- | --- |
| **Objeto** | **Descripción** |
| Tabla | Unidad básica de almacenamiento; compuesta por filas. |
| Vista | Representa de forma lógica subjuegos de datos de una o más tablas. |
| Secuencia | Genera valores numéricos. |
| Índice | Mejora el rendimiento de algunas consultas. |
| Sinónimo | Ofrece nombres alternativos para un objeto |

# Objetos de Base de Datos

Oracle Database puede contener varias estructuras de datos. Cada estructura se debe describir en el diseño de la base de datos para que se pueda crear durante la fase de creación del desarrollo de la base de datos.

* + **Tabla:** almacena datos.
  + **Vista:** subjuego de datos de una o más tablas.
  + **Secuencia:** genera valores numéricos.
  + **Índice:** mejora el rendimiento de algunas consultas.
  + **Sinónimo:** ofrece nombres alternativos para un objeto.

Estructuras de Tabla de Oracle

* + Las tablas se pueden crear en cualquier momento, incluso mientras los usuarios utilizan la base de datos.
  + No necesita especificar el tamaño de la tabla. El tamaño se define en última instancia según la cantidad de espacio asignado a la base de datos completa. No obstante, es importante calcular la cantidad de espacio que utilizará una tabla a lo largo del tiempo.
  + La estructura de la tabla se puede modificar en línea.

**Nota:** hay más objetos de base de datos disponibles, pero no se tratan en este curso.

## Objetos de esquema

Un **esquema** es una colección de estructuras lógicas de datos u objetos de esquema. Un esquema es propiedad de un usuario de la base de datos y tiene el mismo nombre que ese usuario. Cada usuario posee un solo esquema. Los objetos de esquema se pueden crear y manipular con SQL e incluyen los siguientes tipos de objetos:

Clusters   
Restricciones   
de base de datos vincula   
la base de datos activa   
Dimensiones   
bibliotecas procedimiento externo   
mesas organizadas Índice   
Índices   
Indextypes   
clases de Java, los recursos de Java, Java fuentes de   
vistas materializadas   
vista materializada registros de   
modelos de minería   
de objeto Tablas de   
tipos de objeto   
Objeto vistas   
Operadores   
Paquetes   
Secuencias   
funciones almacenadas, procedimientos almacenados   
sinónimos   
Tablas   
Vistas

# Nombres de objetos de base de datos y calificadores

Algunos objetos de la base de datos están formados por partes que puede o debe nombrar, como las columnas en una tabla o vista, particiones y subparticiones de índice y tabla, restricciones de integridad en una tabla y objetos que se almacenan dentro de un paquete, incluidos los procedimientos y Funciones almacenadas. Esta sección proporciona:

* Reglas para nombrar objetos de base de datos y calificadores de ubicación de objetos de base de datos
* Pautas para nombrar objetos de base de datos y calificadores

**Nota :**

Oracle utiliza los nombres generados por el sistema que comienzan con " SYS\_" para los objetos y subobjetos generados de forma implícita en la base de datos, y los nombres que comienzan con " ORA\_" para algunos objetos suministrados por Oracle. Oracle no le permite utilizar estos prefijos en los nombres que proporciona explícitamente a los objetos y subobjetos de su base de datos para evitar posibles conflictos en la resolución de nombres.

## Reglas de nombres de objetos de base de datos

Cada objeto de base de datos tiene un nombre. En una declaración SQL, usted representa el nombre de un objeto con un **identificador entre comillas** o un **identificador sin comillas** .

* Un identificador entre comillas comienza y termina con comillas dobles ("). Si nombra un objeto de esquema con un identificador entre comillas, debe usar comillas dobles cada vez que haga referencia a ese objeto.
* Un identificador sin comillas no está rodeado por ninguna puntuación.

Puede usar los identificadores entre comillas o no para nombrar cualquier objeto de base de datos. Sin embargo, los nombres de bases de datos, los nombres de bases de datos globales y los nombres de enlaces de bases de datos siempre distinguen entre mayúsculas y minúsculas y se almacenan en mayúsculas. Si especifica nombres como identificadores entre comillas, las comillas se ignorarán silenciosamente.

**Nota:**

Oracle no recomienda el uso de identificadores entre comillas para los nombres de objetos de la base de datos. SQL \* Plus acepta estos identificadores citados, pero pueden no ser válidos cuando se usan otras herramientas que administran objetos de base de datos.

La siguiente lista de reglas se aplica a los identificadores entre comillas y no entre comillas, a menos que se indique lo contrario:

1. Los nombres deben tener una longitud de 1 a 30 bytes con estas excepciones:
   * Los nombres de las bases de datos están limitados a 8 bytes.
   * Los nombres de los enlaces de la base de datos pueden tener hasta 128 bytes.

Si un identificador incluye varias partes separadas por puntos, cada atributo puede tener una longitud de hasta 30 bytes. Cada separador de período, así como las comillas dobles que lo rodean, cuentan como un byte. Por ejemplo, supongamos que identifica una columna como esta:

" *esquema* ". " *tabla* ". " *columna* "

El nombre del esquema puede ser de 30 bytes, el nombre de la tabla puede ser de 30 bytes y el nombre de la columna puede ser de 30 bytes. Cada una de las comillas y los puntos es un carácter de un solo byte, por lo que la longitud total del identificador en este ejemplo puede ser de hasta 98 ​​bytes.

1. Los identificadores sin comillas no pueden ser palabras reservadas de Oracle SQL. Los identificadores entre comillas pueden ser palabras reservadas, aunque esto no se recomienda.

Dependiendo del producto de Oracle que planee usar para acceder a un objeto de base de datos, los nombres pueden ser restringidos aún más por otras palabras reservadas específicas del producto.

**Nota:**

La palabra reservada ROWID es una excepción a esta regla. No puede usar la palabra en mayúscula ROWID, ya sea entre comillas o sin comillas, como nombre de columna. Sin embargo, puede usar la palabra en mayúscula como un identificador entre comillas que no es un nombre de columna, y puede usar la palabra con una o más letras minúsculas (por ejemplo, " Rowid" o " rowid") como cualquier identificador entre comillas, incluido el nombre de una columna .

**Ver también:**

* + ["Palabras reservadas de Oracle SQL"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/ap_keywd001.htm#BABCJAEB) para una lista de todas las palabras reservadas de Oracle SQL
  + El manual para un producto específico, como [*Oracle Database PL / SQL Language Reference*](https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/reservewords.htm#LNPLS019) , para una lista de las palabras reservadas de ese producto

1. El lenguaje SQL de Oracle contiene otras palabras que tienen significados especiales. Estas palabras incluyen tipos de datos, nombres de esquema, nombres de funciones, la tabla de sistema simulado DUAL y palabras clave (las palabras en mayúsculas en las instrucciones SQL, tales como DIMENSION, SEGMENT, ALLOCATE, DISABLE, etc.). Estas palabras no están reservadas. Sin embargo, Oracle los utiliza internamente de maneras específicas. Por lo tanto, si usa estas palabras como nombres para objetos y partes de objetos, entonces sus declaraciones SQL pueden ser más difíciles de leer y pueden llevar a resultados impredecibles.

En particular, no use palabras que comiencen con SYS o ORA\_ como nombres de objetos de esquema, y ​​no use los nombres de las funciones integradas de SQL para los nombres de objetos de esquema o funciones definidas por el usuario.

1. Debe usar caracteres ASCII en nombres de bases de datos, nombres de bases de datos globales y nombres de enlaces de bases de datos, ya que los caracteres ASCII proporcionan una compatibilidad óptima en diferentes plataformas y sistemas operativos.
2. Puede incluir caracteres multibyte en contraseñas.
3. Los identificadores sin comillas deben comenzar con un carácter alfabético de su conjunto de caracteres de la base de datos. Los identificadores entre comillas pueden comenzar con cualquier carácter.
4. Los identificadores sin comillas pueden contener solo caracteres alfanuméricos de su conjunto de caracteres de la base de datos y el subrayado (\_), el signo de dólar ($) y el signo de libra (#). Los enlaces de la base de datos también pueden contener puntos (.) Y signos "at" (@). Oracle lo desalienta enérgicamente de usar $ y # en identificadores sin comillas.

Los identificadores entre comillas pueden contener caracteres y signos de puntuación, así como espacios. Sin embargo, ni los identificadores entre comillas ni los que no están entre comillas pueden contener comillas dobles o el carácter nulo ( \0).

1. Dentro de un espacio de nombres, no hay dos objetos que puedan tener el mismo nombre.

Los siguientes objetos de esquema comparten un espacio de nombres:

* + Paquetes
  + Sinónimos privados
  + Secuencias
  + Procedimientos independientes
  + Funciones almacenadas autónomas
  + Operadores definidos por el usuario
  + Tipos definidos por el usuario
  + Mesas
  + Puntos de vista

Cada uno de los siguientes objetos de esquema tiene su propio espacio de nombres:

* + Agrupaciones
  + Restricciones
  + Activadores de base de datos
  + Dimensiones
  + Índices
  + Vistas materializadas (Cuando crea una vista materializada, la base de datos crea una tabla interna con el mismo nombre. Esta tabla tiene el mismo espacio de nombres que las otras tablas del esquema. Por lo tanto, un esquema no puede contener una tabla y una vista materializada de la misma nombre.)
  + Enlaces de bases de datos privadas

Debido a que las tablas y las secuencias están en el mismo espacio de nombres, una tabla y una secuencia en el mismo esquema no pueden tener el mismo nombre. Sin embargo, las tablas y los índices están en diferentes espacios de nombres. Por lo tanto, una tabla y un índice en el mismo esquema pueden tener el mismo nombre.

Cada esquema en la base de datos tiene sus propios espacios de nombres para los objetos que contiene. Esto significa, por ejemplo, que dos tablas en diferentes esquemas están en diferentes espacios de nombres y pueden tener el mismo nombre.

Cada uno de los siguientes objetos que no son de esquema también tiene su propio espacio de nombres:

* + Ediciones
  + Los archivos de parámetro ( PFILEs) y archivos de parámetros del servidor ( SPFILEs)
  + Perfiles
  + Enlaces de bases de datos públicas
  + Sinónimos públicos
  + Espacios de tabla
  + Roles del usuario

Debido a que los objetos en estos espacios de nombres no están contenidos en esquemas, estos espacios de nombres abarcan toda la base de datos.

1. Los identificadores sin comillas no distinguen entre mayúsculas y minúsculas. Oracle los interpreta como mayúsculas. Los identificadores entre comillas distinguen entre mayúsculas y minúsculas.

Al incluir nombres entre comillas dobles, puede asignar los siguientes nombres a diferentes objetos en el mismo espacio de nombres:

"empleados"

"Empleados"

"EMPLEADOS"

Tenga en cuenta que Oracle interpreta los siguientes nombres de la misma manera, por lo que no se pueden usar para diferentes objetos en el mismo espacio de nombres:

empleados

EMPLEADOS

"EMPLEADOS"

1. Cuando Oracle almacena o compara los identificadores en mayúsculas, la forma en mayúsculas de cada carácter en los identificadores se determina aplicando las reglas de mayúsculas y minúsculas del conjunto de caracteres de la base de datos. Las reglas específicas de idioma determinadas por la configuración de la sesión NLS\_SORT no se consideran. Este comportamiento corresponde a aplicar la función SQL UPPER al identificador en lugar de a la función NLS\_UPPER.

El conjunto de caracteres de la base de datos de reglas de uppercasing puede producir resultados que son incorrectos cuando se ven en un lenguaje natural determinado. Por ejemplo, la minúscula sharp s ("ß"), utilizada en alemán, no tiene una forma en mayúsculas de acuerdo con las reglas del conjunto de caracteres de la base de datos. No se modifica cuando un identificador se convierte en mayúsculas, mientras que la forma en mayúsculas esperada en alemán es la secuencia de dos caracteres en mayúscula S ("SS"). De manera similar, la letra mayúscula de la letra minúscula i, según las reglas del conjunto de caracteres de la base de datos, es la letra mayúscula I. Sin embargo, la forma mayúscula esperada en turco y azerbaiyano es la letra mayúscula I con el punto arriba.

Las reglas de conjunto de caracteres de la base de datos uppercasing aseguran que los identificadores se interpreten de la misma manera en cualquier configuración lingüística de una sesión. Si desea que un identificador se vea correctamente en un cierto lenguaje natural, entonces puede citarlo para conservar la forma en minúsculas o puede usar la forma en mayúsculas corregida lingüísticamente siempre que use ese identificador.

1. Las columnas en la misma tabla o vista no pueden tener el mismo nombre. Sin embargo, las columnas en diferentes tablas o vistas pueden tener el mismo nombre.
2. Los procedimientos o funciones contenidos en el mismo paquete pueden tener el mismo nombre, si sus argumentos no son del mismo número y tipos de datos. La creación de múltiples procedimientos o funciones con el mismo nombre en el mismo paquete con diferentes argumentos se denomina **sobrecarga** del procedimiento o función.

## Ejemplos de nombres de objetos de esquema

Los siguientes ejemplos son nombres de objetos de esquema válidos:

last\_name

horse

hr.hire\_date

"EVEN THIS & THAT!"

a\_very\_long\_and\_valid\_name

Todos estos ejemplos se adhieren a las reglas enumeradas en ["Reglas de nombres de objetos de base de datos"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements008.htm#i27570) . El siguiente ejemplo no es válido porque supera los 30 caracteres:

a\_very\_very\_long\_and\_not\_valid\_name

Aunque los alias de columnas, los alias de tablas, los nombres de usuario y las contraseñas no son objetos o partes de objetos, también deben seguir estas reglas de denominación, a menos que se especifique lo contrario en las propias reglas.

## Pautas para nombrar objetos de esquema

Aquí hay varias pautas útiles para nombrar objetos y sus partes:

* Utilice nombres completos, descriptivos y pronunciables (o abreviaturas conocidas).
* Use reglas de nombres consistentes.
* Utilice el mismo nombre para describir la misma entidad o atributo en las tablas.

Al nombrar objetos, equilibre el objetivo de mantener los nombres cortos y fáciles de usar con el objetivo de hacer que los nombres sean lo más descriptivos posible. En caso de duda, elija el nombre más descriptivo, ya que los objetos en la base de datos pueden ser utilizados por muchas personas durante un período de tiempo. Su contraparte dentro de diez años puede tener dificultades para entender una columna de tabla con un nombre como pmdd en  lugar de payment\_due\_date.

El uso de reglas de nomenclatura coherentes ayuda a los usuarios a comprender la parte que desempeña cada tabla en su aplicación. Una de estas reglas podría ser comenzar con los nombres de todas las tablas que pertenecen a la aplicación FINANCE fin\_.

Use los mismos nombres para describir las mismas cosas en las tablas. Por ejemplo, las columnas de número de departamento de la muestra employees y las tablas departments se denominan department\_id.

# Sintaxis para objetos de esquema y partes en declaraciones SQL

Esta sección le indica cómo hacer referencia a los objetos de esquema y sus partes en el contexto de una declaración SQL. Esta sección te muestra:

* La sintaxis general para referirse a un objeto.
* Cómo Oracle resuelve una referencia a un objeto
* Cómo referirse a objetos en esquemas diferentes al suyo
* Cómo referirse a objetos en bases de datos remotas
* Cómo referirse a particiones y subparticiones de tablas e índices

El siguiente diagrama muestra la sintaxis general para referirse a un objeto o una parte:

***database\_object\_or\_part* :: =**

La descripción de database_object_or_part.gif sigue  
[Descripción de la ilustración '' database\_object\_or\_part.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/database_object_or_part.htm)

( *[dblink](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements009.htm" \l "BABGJDDB)*[:: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements009.htm" \l "BABGJDDB) )

dónde:

* *object* es el nombre del objeto
* *schema* es el esquema que contiene el objeto. El calificador de esquema le permite referirse a un objeto en un esquema diferente al suyo. Se le deben otorgar privilegios para referirse a objetos en otros esquemas. Si omite *schema*, Oracle asume que se está refiriendo a un objeto en su propio esquema.

Solo los objetos de esquema pueden ser calificados con *schema*. Los objetos de esquema se muestran con el elemento de lista [8](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements008.htm#i78631) . Los objetos que no son de esquema , que también se muestran con el elemento de lista [8](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements008.htm#i78631) , no se pueden calificar *schema* porque no son objetos de esquema. Una excepción son los sinónimos públicos, que pueden calificarse opcionalmente con " PUBLIC". Las comillas son obligatorias.

* *parte* Es una parte del objeto. Este identificador le permite referirse a una parte de un objeto de esquema, como una columna o una partición de una tabla. No todos los tipos de objetos tienen partes.
* *Dblink* Solo se aplica cuando se utiliza la funcionalidad distribuida de la base de datos Oracle. Este es el nombre de la base de datos que contiene el objeto. El calificador *dblink* le permite referirse a un objeto en una base de datos que no sea su base de datos local. Si omite *dblink*, Oracle asume que se está refiriendo a un objeto en su base de datos local. No todas las sentencias de SQL le permiten acceder a objetos en bases de datos remotas.

Puede incluir espacios alrededor de los períodos que separan los componentes de la referencia al objeto, pero es convencional omitirlos.

## Cómo Oracle Database resuelve referencias de objetos de esquema

Cuando hace referencia a un objeto en una declaración SQL, Oracle considera el contexto de la declaración SQL y ubica el objeto en el espacio de nombres apropiado. Después de ubicar el objeto, Oracle realiza la operación especificada por la declaración en el objeto. Si el objeto nombrado no se puede encontrar en el espacio de nombres apropiado, entonces Oracle devuelve un error.

El siguiente ejemplo ilustra cómo Oracle resuelve las referencias a los objetos dentro de las declaraciones SQL. Considere esta declaración que agrega una fila de datos a una tabla identificada por el nombre departments:

INSERT INTO departments

VALUES (280, 'ENTERTAINMENT\_CLERK', 206, 1700);

En función del contexto de la declaración, Oracle determina que departments puede ser:

* Una tabla en tu propio esquema.
* Una vista en tu propio esquema.
* Un sinónimo privado de una mesa o vista.
* Un sinonimo publico

Oracle siempre intenta resolver una referencia de objeto dentro de los espacios de nombres en su propio esquema antes de considerar los espacios de nombres fuera de su esquema. En este ejemplo, *Oracle* intenta resolver el nombre de departments de la siguiente manera:

1. Primero, Oracle intenta localizar el objeto en el espacio de nombres en su propio esquema que contiene tablas, vistas y sinónimos privados. Si el objeto es un sinónimo privado, entonces Oracle localiza el objeto que representa el sinónimo. Este objeto podría estar en su propio esquema, otro esquema o en otra base de datos. El objeto también podría ser otro sinónimo, en cuyo caso Oracle localiza el objeto que representa este sinónimo.
2. Si el objeto está en el espacio de nombres, entonces Oracle intenta realizar la declaración en el objeto. En este ejemplo, Oracle intenta agregar la fila de datos a departments. Si el objeto no es del tipo correcto para la declaración, entonces Oracle devuelve un error. En este ejemplo, departments debe ser una tabla, una vista o un sinónimo privado que se resuelva en una tabla o vista. Si departments es una secuencia, entonces Oracle devuelve un error.
3. Si el objeto no se encuentra en ningún espacio de nombres buscado hasta ahora, Oracle busca el espacio de nombres que contiene sinónimos públicos. Si el objeto está en ese espacio de nombres, entonces Oracle intenta realizar la declaración en él. Si el objeto no es del tipo correcto para la declaración, entonces Oracle devuelve un error. En este ejemplo, si departments es un sinónimo público de una secuencia, Oracle devuelve un error.

Si un sinónimo público tiene tablas dependientes o tipos definidos por el usuario, no puede crear un objeto con el mismo nombre que el sinónimo en el mismo esquema que los objetos dependientes.

Si un sinónimo no tiene tablas dependientes ni tipos definidos por el usuario, puede crear un objeto con el mismo nombre en el mismo esquema que los objetos dependientes. Oracle invalida los objetos dependientes e intenta volver a validarlos cuando se accede a ellos nuevamente.



**Reglas de Nomenclatura**

Los nombres de tabla y de columna deben:

* Empezar por una letra
* Tener entre 1 y 30 caracteres
* Contener sólo A–Z, a–z, 0–9, \_, $ y #
* No deben ser un duplicado de otro nombre de objeto propiedad del mismo usuario
* No debe ser una palabra reservada del servidor de Oracle

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Reglas de Nomenclatura

Las tablas y columnas de la base de datos se deben nombrar según las reglas estándar de nomenclatura de objetos de Oracle Database:

Los nombres de tabla y de columna deben comenzar por una letra y tener entre 1 y 30 caracteres.

Los nombres sólo deben contener los caracteres A–Z, a–z, 0–9, \_ (subrayado), $ y # (caracteres válidos, pero cuyo uso no es aconsejable).

Los nombres no deben ser un duplicado de otro nombre de objeto propiedad del mismo usuario del servidor de Oracle.

Los nombres no deben ser una palabra reservada del servidor de *Oracle*.

- Puede utilizar identificadores entre comillas para representar el nombre de un objeto. Un identificador entre comillas empieza y finaliza con comillas dobles (""). Si asigna un nombre a un objeto de esquema mediante un identificador entre comillas, debe utilizar las comillas dobles cuando haga referencia a dicho objeto. Los identificadores entre comillas pueden ser palabras reservadas, aunque no se recomienda.

### Directrices de Nomenclatura

Utilice nombres descriptivos para las tablas y otros objetos de la base de datos.

**Nota:** los nombres no son sensibles a mayúsculas/minúsculas. Por ejemplo, EMPLOYEES se considera el mismo nombre que eMPloyees o eMpLOYEES. Sin embargo, los identificadores entre comillas son sensibles a mayúsculas/minúsculas.

## TABLA

CREATE TABLE [*schema*.]*table*

(*column datatype* [DEFAULT *expr*][, ...]);



**Sentencia CREATE TABLE**

* Debe tener:
  + El privilegio CREATE TABLE
  + Un área de almacenamiento
* Debe especificar:
  + El nombre de tabla
  + El nombre de columna, tipo de dato de columna y tamaño de columna



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Sentencia CREATE TABLE

Puede crear tablas para almacenar datos ejecutando la sentencia SQL **CREATE TABLE**. Esta sentencia es una de las sentencias DDL, que son un subjuego de sentencias SQL que se utilizan para crear, modificar o eliminar estructuras de *Oracle* Database. Estas sentencias tienen un efecto inmediato en la base de datos y registran información en el diccionario de datos.

Para crear una tabla, un usuario debe tener el privilegio **CREATE TABLE** y un área de almacenamiento en la que crear los objetos. El administrador de la base de datos (DBA) utiliza sentencias de lenguaje de control de datos (DCL) para otorgar privilegios a los usuarios.

En la sintaxis:

*schema* es el mismo nombre que el del propietario

*table* es el nombre de la tabla

**DEFAULT  *expr***especifica un valor por defecto si se omite un valor en la sentencia

INSERT.

**column**es el nombre de la columna.

***datatype***es el tipo de dato y la longitud de la columna.



**USERB**

**SELECT \***

**FROM userA.employees;**

**USERA**

**SELECT \***

**FROM userB.employees;**

* Las tablas que pertenecen a otros usuarios no se incluyen en el esquema del usuario.
* Debe utilizar el nombre del propietario como prefijo de dichas tablas.

**Referencia a Tablas de Otro Usuario**

Referencia a Tablas de Otro Usuario

Un esquema es una recopilación de estructuras lógicas de datos u *objetos de esquema*. Un esquema es propiedad de un usuario de base de datos y tiene el mismo nombre que el usuario. Cada usuario es propietario de un único esquema.

Los objetos de esquema se pueden crear y manipular con SQL e incluyen tablas, vistas, sinónimos, secuencias, procedimientos almacenados, índices, clusters y enlaces de base de datos.

Si una tabla no pertenece al usuario, el nombre del propietario se debe anteponer a la tabla. Por ejemplo, dados los esquemas USERA y USERB y ambos contienen la tabla EMPLOYEES, si USERA desea acceder a la tabla EMPLOYEES que pertenece a USERB, USERA debe anteponer el nombre del esquema al nombre de la tabla:

SELECT \*

FROM userb.employees;

Si USERB desea acceder a la tabla EMPLOYEES propiedad de USERA, USERB debe anteponer el nombre del esquema al nombre de la tabla:

SELECT \*

FROM usera.employees;



CREATE TABLE hire\_dates(

id NUMBER(8),

hire\_date DATE DEFAULT SYSDATE);

... hire\_date DATE DEFAULT SYSDATE, ...



**Opción DEFAULT**

* Especificar un valor por defecto para una columna durante una inserción.
* Los valores literales, expresiones o funciones SQL son valores válidos.
* El nombre de otra columna o una pseudocolumna son valores no válidos.
* El tipo de dato por defecto debe coincidir con el tipo de dato de la columna.

### Opción DEFAULT

Al definir una tabla, puede especificar que se proporcione un valor por defecto a una columna mediante la opción **DEFAULT**. Esta opción evita que se introduzcan valores nulos en las columnas si se inserta una fila sin un valor para la columna. El valor por defecto puede ser un literal, una expresión o una función SQL (como **SYSDATE** o **USER**), pero el valor no puede ser el nombre de otra columna o una pseudocolumna (como **NEXTVAL** o **CURRVAL**). La expresión por defecto debe coincidir con el tipo de dato de la columna.

Considere los siguientes ejemplos:

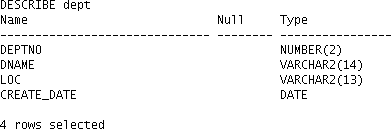
**INSERT INTO** hire\_dates values(45, NULL);

La sentencia anterior insertará el valor nulo en lugar del valor por defecto.

**INSERT INTO** hire\_dates(id) values(35);

La sentencia anterior insertará **SYSDATE** para la columna HIRE\_DATE.

**Nota:** En SQL Developer, haga clic en el icono Run Script o pulse [F5] para ejecutar las sentencias DDL. Los comentarios se mostrarán en la página con separadores Script Output.



DESCRIBE dept;

CREATE TABLE dept(

deptno NUMBER(2),

dname VARCHAR2(14),

loc VARCHAR2(13),

create\_date DATE DEFAULT SYSDATE);



**Creación de Tablas**

* Crear una tabla:
* Confirmar la creación de la tabla:

Creación de Tablas

El ejemplo de la diapositiva crea la tabla DEPT, con cuatro columnas: DEPTNO, DNAME, LOC y CREATE\_DATE. La columna CREATE\_DATE tiene un valor por defecto. Si no se proporciona ningún valor para una sentencia **INSERT**, se inserta automáticamente la fecha del sistema.

Para confirmar que se ha creado la tabla, ejecute el comando **DESCRIBE**.

Puesto que la creación de una tabla constituye una sentencia DDL, se realiza una confirmación automática al ejecutar esta sentencia.

**Nota:** puede ver la lista de las tablas que posee consultando el diccionario de datos. Por ejemplo:

**SELECT** table\_name

**FROM** user\_tables

Mediante las vista de diccionario de datos, puede obtener información sobre otros objetos de base de datos como vistas, índices, etc.

## VISTA

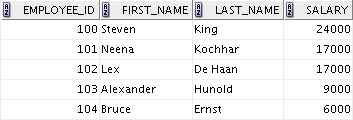
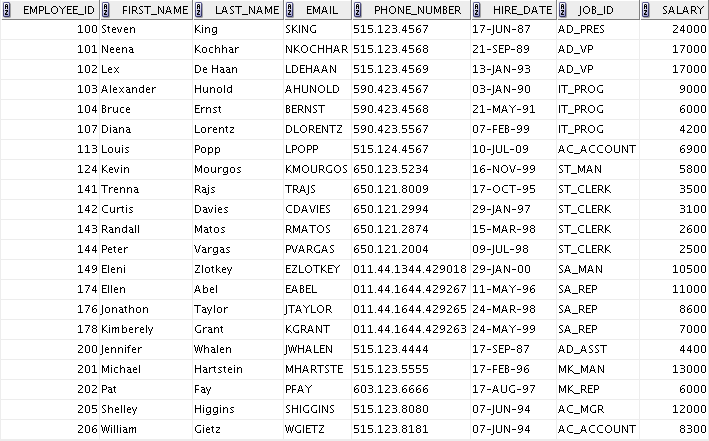


**¿Qué es una Vista?**

Tabla **EMPLOYEES**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

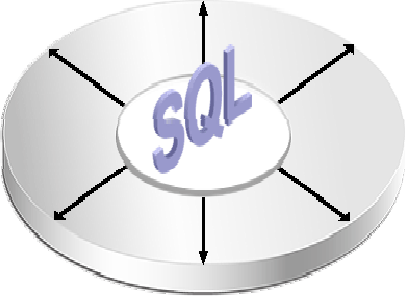
### ¿Qué es una Vista?



Puede representar combinaciones de datos o subjuegos lógicos mediante la creación de vistas de tablas. Una vista es una tabla lógica basada en una tabla o en otra vista. Una vista no contiene ningún dato propio, sino que se trata de una ventana a través de la que se pueden visualizar o cambiar los datos de las tablas. Las tablas en las que se basa la vista se denominan *tablas base*. Las vista se almacena como una sentencia **SELECT** en el diccionario de datos.



**Ventajas de las Vistas**



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Para mejorar la independencia de los datos**

**Para presentar diferentes vistas de los mismos datos**

**Para facilitar consultas complejas**

**Para restringir el acceso a los datos**

Ventajas de las Vistas

* Las vistas restringen el acceso a los datos porque muestra las columnas seleccionadas de la tabla.
* Las vistas se pueden utilizar para crear consultas simples para recuperar los resultados de consultas complicadas. Por ejemplo, las vistas se pueden utilizar para consultar información de varias tablas sin que el usuario sepa cómo escribir una sentencia de unión.
* Las vistas proporcionan independencia de los datos para usuarios ad-hoc y programas de aplicación. Las vistas se pueden utilizar para recuperar datos de varias tablas.
* Las vistas proporcionan a los grupos de usuarios acceso a los datos según sus criterios concretos.



**Vistas Simples y Complejas**

|  |  |  |
| --- | --- | --- |
| **Función** | **Vistas Simples** | **Vistas Complejas** |
| Número de tablas | Una | Una o más |
| Contienen funciones | No | Sí |
| Contienen grupos de datos | No | Sí |
| Operaciones DML a través de una vista | Sí | No siempre |

### Vistas Simples y Complejas

Existen dos clasificaciones para las vistas: simples y complejas.. La principal diferencia está relacionada con las operaciones (**INSERT**, **UPDATE** y **DELETE**).

* Las vistas simples:
  + Derivan datos de una sola tabla.
  + No contienen funciones ni grupos de datos.
  + Pueden realizar operaciones DML a través de la vista.
* Las vistas complejas:
  + Derivan datos de varias tablas.
  + Contienen funciones o grupos de datos.
  + No siempre permiten operaciones DML a través de la vista.

CREATE [OR REPLACE] [FORCE|NOFORCE]

VIEW *view* [(*alias*[, *alias*]...)]

AS *subquery*

[WITH CHECK OPTION [CONSTRAINT *constraint*]]

[WITH READ ONLY [CONSTRAINT *constraint*]];



**Creación de Vistas**

* Embeber una subconsulta en la sentencia CREATE VIEW:
* La subconsulta puede contener la sintaxis compleja

SELECT.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Vistas

Para crear una vista, puede embeber una subconsulta en la sentencia **CREATE VIEW**. En la sintaxis:

**OR**  **REPLACE** vuelve a crear la vista si ya existe.

**FORCE** crea la vista independientemente de si existen o no las tablas base. **NO FORCE** crea la vista sólo si existen las tablas base (opción por defecto).

***view***es el nombre de la vista.

***alias***especifica los nombre de las expresiones seleccionadas por la consulta de la vista. (El numero de alias debe coincidir con el número de expresiones seleccionadas por la vista).

***subquery***Es una sentencia **SELECT** completa (Puede utilizar alias para las columnas de la lista **SELECT**).

**WITH CHECK OPTION** especifica que sólo las filas accesibles para la vista se puedan insertar o actualizar.

***constraint***es el nombre asignado a la restricción **CHECK OPTION**.

**WITH READ ONLY** garantiza que no se pueda realizar ninguna operación DML en esta vista.

**Nota:** en SQL Developer, haga clic en el icono Run Script o pulse [F5] para ejecutar las sentencias de lenguaje de definición de datos (DLL). Los comentarios se mostrarán en la página con separadores Script Output.



DESCRIBE empvu80;

CREATE VIEW empvu80

AS SELECT employee\_id,

last\_name,

salary

FROM employees

WHERE department\_id = 80;



**Creación de Vistas**

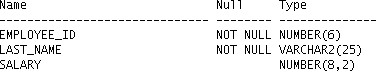
* Crear la vista EMPVU80, que contiene los detalles de los empleados del departamento 80:
* Describir la estructura de la vista mediante el comando

DESCRIBE de SQL\*Plus:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

En el ejemplo de la diapositiva se crea una vista que contiene el número de empleado, el apellido y el salario de cada empleado del departamento 80.

Puede visualizar la estructura de la vista mediante el comando **DESCRIBE**.



Instrucciones

* La subconsulta que define una vista puede contener la sintaxis compleja **SELECT**, incluidas uniones, grupos y subconsultas.
* Si no especifica un nombre de restricción para la vista creada con **WITH CHECK OPTION**, el sistema asigna un nombre por defecto con el formato SYS\_C*n*.
* Puede utilizar la opción **OR REPLACE** para cambiar la definición de la vista sin borrarla y volver a crearla o sin necesidad de volver a otorgarle los privilegios de objeto otorgados previamente.



CREATE VIEW salvu50

AS SELECT employee\_id ID\_NUMBER,

last\_name NAME,

salary\*12 ANN\_SALARY

FROM employees

WHERE department\_id = 50;



**Creación de Vistas**

* Crear una vista mediante alias de columna en la subconsulta:
* Seleccionar las columnas de esta vista según los nombres de alias proporcionados.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Puede controlar los nombres de columna incluyendo alias de columna en la subconsulta.El ejemplo de la diapositiva crea una vista que contiene el número de empleado (EMPLOYEE\_ID) con el alias ID\_NUMBER, el nombre (LAST\_NAME) con el alias NAME y el salario anual (SALARY) con el alias ANN\_SALARY para cada empleado del departamento 50.

Como alternativa, puede utilizar un alias después de la sentencia **CREATE** y antes de la subconsulta **SELECT**. El numero de alias mostrado debe coincidir con el número de expresiones seleccionadas en la vista).



CREATE OR REPLACE VIEW salvu50 ( ID\_NUMBER,

NAME,

ANN\_SALARY)

AS

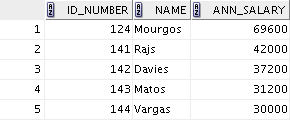
SELECT employee\_id,

last\_name,

salary\*12

FROM employees

WHERE department\_id = 50;



**Recuperación de Datos de una Vista**

SELECT \*

FROM salvu50;

Recuperación de Datos de una Vista

Puede recuperar datos de una vista al igual que de una tabla. Puede visualizar el contenido de toda la vista o sólo de filas y columnas específicas.



CREATE OR REPLACE VIEW empvu80

(id\_number, name, sal, department\_id)

AS

SELECT employee\_id, first\_name || ' '

|| last\_name, salary, department\_id

FROM employees

WHERE department\_id = 80;



**Modificación de Vistas**

* Modificar la vista EMPVU80 mediante una cláusula CREATE OR REPLACE VIEW. Agregar alias para cada nombre de columna:
* Los alias de columna de la cláusula CREATE OR REPLACE VIEW se muestran en el mismo orden que las columnas de la subconsulta.

Modificación de Vistas

Con la opción **OR REPLACE**, se puede crear una vista incluso si ya existe otra con este nombre, sustituyendo de esta forma la antigua versión de la vista para su propietario. Esto significa que la vista se puede modificar sin borrar, volver a crear y volver a otorgar los privilegios de objeto.

**Nota:** al asignar alias de columna en la cláusula **CREATE OR REPLACE VIEW**, recuerde que los alias se muestran en el mismo orden que las columnas de la subconsulta.



CREATE OR REPLACE VIEW dept\_sum\_vu (name,

minsal,

maxsal,

avgsal)

AS

SELECT d.department\_name,

MIN(e.salary),

MAX(e.salary),

AVG(e.salary)

FROM employees e

JOIN departments d

ON (e.department\_id = d.department\_id)

GROUP BY d.department\_name;



**Creación de Vistas Complejas**

Crear una vista compleja que contenga funciones de grupo para visualizar valores de dos tablas:

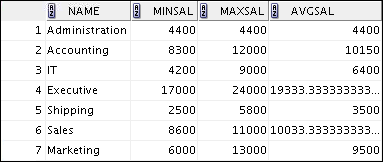
Creación de Vistas Complejas

En el ejemplo de la diapositiva se crea una vista compleja de nombres de departamento, salarios mínimos y salarios medios por departamento. Tenga en cuenta que se han especificado nombres alternativos para la vista. Éste es un requisito necesario si alguna de las columnas de la vista se deriva de una función o expresión.

Puede visualizar la estructura de la vista mediante el comando **DESCRIBE**. Para visualizar el contenido de la vista, emita una sentencia **SELECT**.

SELECT \*

FROM dept\_sum\_vu;





* No puede eliminar una fila si la vista contiene lo siguiente:
  + Funciones de grupo
  + Una cláusula GROUP BY
  + La palabra clave DISTINCT
  + La palabra clave ROWNUM de pseudocolumna

en

**Reglas para Realizar Operaciones DML en una Vista**

Normalmente, puede realizar operaciones DML las vistas simples.

•

Reglas para Realizar Operaciones DML en una Vista

* Puede realizar operaciones DML en los datos a través de una vista si dichas operaciones siguen ciertas reglas.
* Puede eliminar una fila de una vista a menos que contenga alguna de las siguientes opciones:
  + Funciones de grupo
  + Una cláusula **GROUP BY**
  + La palabra clave **DISTINCT**
  + La palabra clave **ROWNUM** de pseudocolumna



**Reglas para Realizar Operaciones DML en una Vista**

No puede modificar datos de una vista si contiene:

* Funciones de grupo
* Una cláusula GROUP BY
* La palabra clave DISTINCT
* La palabra clave ROWNUM de pseudocolumna
* Columnas definidas por expresiones

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Puede modificar datos a través de una vista a menos que contenga cualquiera de las condiciones mencionadas en la diapositiva anterior o columnas definidas por expresiones (por ejemplo, SALARY \* 12).



**Reglas para Realizar Operaciones DML en una Vista**

No puede agregar datos a través de una vista si ésta incluye:

* Funciones de grupo
* Una cláusula GROUP BY
* La palabra clave DISTINCT
* La palabra clave ROWNUM de pseudocolumna
* Columnas definidas por expresiones
* Columnas NOT NULL de las tablas base no seleccionadas por la vista

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Puede agregar datos a través de una vista a menos que contenga cualquiera de los elementos mostrados en la diapositiva. No puede agregar datos a una vista si ésta contiene columnas **NOT** **NULL** sin valores por defecto en la tabla base. Todos los valores necesarios deben estar presentes en la vista. Recuerde que va a agregar valores directamente a la tabla subyacente *a través* de la vista.



CREATE OR REPLACE VIEW empvu20

AS

SELECT \*

FROM employees

WHERE department\_id = 20

WITH CHECK OPTION CONSTRAINT empvu20\_ck ;



**Uso de la Cláusula WITH CHECK OPTION**

* Puede asegurarse de que las operaciones DML realizadas en la vista permanecerán en el dominio de la vista mediante la cláusula WITH CHECK OPTION:
* Cualquier intento de ejecutar INSERT con una fila con department\_id distinto de 20, o ejecutar UPDATE en el número de departamento para cualquier fila de la vista fallará porque viola la restricción WITH CHECK OPTION.

### Uso de la Cláusula WITH CHECK OPTION

Es posible realizar comprobaciones de integridad referencial a través de las vistas. También puede aplicar restricciones a nivel de base de datos. La vista se puede utilizar para proteger la integridad de los datos, pero su uso es muy limitado.

La cláusula **WITH CHECK OPTION** especifica que las operaciones **INSERT** y **UPDATE** realizadas a través de la vista no pueden crear filas que no pueda seleccionar la vista. Por lo tanto, permite aplicar restricciones de integridad y comprobaciones de validación de datos en los datos que se van a insertar o actualizar. Si se intenta realizar operaciones DML en filas que no haya seleccionado la vista, se muestra un error, junto con el nombre de la restricción si se ha especificado.

UPDATE empvu20

SET department\_id = 10

WHERE employee\_id = 201;

causa:



**Nota:** no se ha actualizado ninguna fila porque si el número de departamento se cambiara a 10, la vista ya no podría ver ese empleado. Con la cláusula **WITH CHECK OPTION**, por tanto, la vista sólo puede ver los empleados del departamento 20 y no permite cambiar el número de departamento de dichos empleados a través de la vista.



**Denegación de Operaciones DML**

* Para asegurarse de que no se realice ninguna operación DML, agregue la opción WITH READ ONLY a la definición de vista.
* Cualquier intento de realizar una operación DML en cualquier fila de la vista provocará un error del servidor de Oracle.



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Denegación de Operaciones DML

Puede asegurarse de que no se realice ninguna operación DML en la vista, créela con la opción **WITH READ ONLY**. En el ejemplo de la siguiente diapositiva se modifica la vista EMPVU10 para evitar operaciones DML en la vista.



CREATE OR REPLACE VIEW empvu10(

employee\_number,

employee\_name,

job\_title)

AS

SELECT employee\_id,

last\_name,

job\_id

FROM employees

WHERE department\_id = 10 WITH READ ONLY ;



**Denegación de Operaciones DML**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Cualquier intento de eliminar una fila de una vista con una restricción de sólo lectura provocará un error:

DELETE FROM empvu10

WHERE employee\_number = 200;

Igualmente, cualquier intento de insertar o modificar una fila mediante una vista con una restricción de sólo lectura provocará el mismo error.





DROP VIEW empvu80;

DROP VIEW *view*;



**Eliminación de Vistas**

Puede eliminar una vista sin perder los datos, ya que las vistas se basan en tablas subyacentes de la base de datos.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Eliminación de Vistas

Puede utilizar la sentencia **DROP**  **VIEW** para eliminar una vista. La sentencia elimina la definición de vista de la base de datos. Sin embargo, el borrado de vistas no tiene ningún efecto en las tablas en las que se basan las vistas. Por otro lado, las vistas y otras aplicaciones basadas en las vistas suprimidas se convierten en no válidas. Sólo el creador o un usuario con el privilegio **DROP ANY VIEW** puede eliminar una vista.

En la sintaxis, *view* es el nombre de la vista.

## SECUENCIA

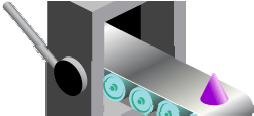


**Secuencias**

|  |  |
| --- | --- |
| **Objeto** | **Descripción** |
| Tabla | Unidad básica de almacenamiento; compuesta por filas. |
| Vista | Representa de forma lógica subjuegos de datos de una o más tablas. |
| Secuencia | Genera valores numéricos. |
| Índice | Mejora el rendimiento de algunas consultas. |
| Sinónimo | Ofrece nombres alternativos para los objetos. |

## Secuencias

Una secuencia es un objeto de base de datos que crea valores enteros. Puede crear secuencias y, a continuación, utilizarlas para generar números.



**9**

**7**

**5**

**3**

**1**

**10**

**8**

**6**

**4**

**2**

Una secuencia:

* Puede generar automáticamente números únicos
* Es un objeto que se puede compartir
* Se puede utilizar para crear un valor de clave primaria
* Sustituye el código de aplicación
* Acelera la eficacia del acceso a los valores de secuencia cuando están almacenados en caché

**Secuencias**

Una secuencia es un objeto de base de datos creado por el usuario que pueden compartir varios usuarios para generar enteros.

Puede definir una secuencia para generar valores únicos o reciclar y volver a utilizar los mismos números.

El uso normal de las secuencias es la creación de un valor de clave primaria, que debe ser único para cada fila. Una secuencia se genera y aumenta (o disminuye) mediante una rutina interna de Oracle. Éste puede ser un objeto de ahorro de tiempo, ya que reduce la cantidad de código de aplicación necesario para escribir una rutina de generación de secuencia.

Los números de secuencia se almacenan y generan independientemente de las tablas. Por lo tanto, la misma secuencia se puede utilizar para varias tablas.

CREATE SEQUENCE *sequence* [INCREMENT BY *n*] [START WITH *n*]

[{MAXVALUE *n* | NOMAXVALUE}] [{MINVALUE *n* | NOMINVALUE}] [{CYCLE | NOCYCLE}] [{CACHE *n* | NOCACHE}];



**Sentencia CREATE SEQUENCE: Sintaxis**

Definir una secuencia para generar números secuenciales automáticamente:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Sentencia CREATE SEQUENCE: Sintaxis

Genera números secuenciales automáticamente mediante la sentencia **CREATE SEQUENCE**. En la sintaxis:

*sequence* es el nombre del generador de secuencias.

**INCREMENT BY *n***especifica el intervalo entre números de secuencia, donde

*n* es un entero (Si se omite esta cláusula, la secuencia aumenta en 1).

**START WITH *n***especifica el primer número de secuencia que se va a generar

(Si se omite esta cláusula, la secuencia empieza con 1).

**MAXVALUE *n***especifica el valor máximo que puede generar la secuencia.

**NOMAXVALUE** especifica un valor máximo de 10^27 para una secuencia ascendente y –1 para una secuencia descendente (Ésta es la opción por defecto).

**MINVALUE *n***especifica el valor mínimo de secuencia.

**NOMINVALUE** especifica un valor mínimo de 1 para una secuencia ascendente y –(10^26) para una secuencia descendente (Ésta es la opción por defecto).



CREATE SEQUENCE dept\_deptid\_seq

INCREMENT BY 10

START WITH 120

MAXVALUE 9999 NOCACHE NOCYCLE;



**Creación de Secuencias**

* Crear una secuencia con el nombre DEPT\_DEPTID\_SEQ que se utilizará para la clave primaria de la tabla DEPARTMENTS.
* No utilizar la opción CYCLE.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

CYCLE | NOCYCLE especifica si la secuencia sigue generando

valores después de alcanzar su valor máximo o mínimo (NOCYCLE es la opción por defecto).

CACHE *n* | NOCACHE especifica cuántos valores preasigna el servidor de Oracle en la memoria (por defecto, el servidor de Oracle almacena en caché 20 valores).

En el ejemplo de la diapositiva se crea una secuencia con el nombre DEPT\_DEPTID\_SEQ que se utilizará para la columna DEPARTMENT\_ID de la tabla DEPARTMENTS. La secuencia se inicia en 120, no permite el almacenamiento en caché y no sigue ningún ciclo.

No utilice la opción **CYCLE** si la secuencia se utiliza para generar valores de clave primaria, a menos que disponga de un mecanismo fiable que depure las filas antiguas más rápido que los ciclos de secuencia.

**Nota:** la secuencia no está ligada a una tabla. Como norma general, debe asignar un nombre a la secuencia según el uso que desee darle. Sin embargo, la secuencia se puede utilizar en cualquier lugar, independientemente de su nombre.



**Pseudocolumnas NEXTVAL y CURRVAL**

* NEXTVAL devuelve el siguiente valor de secuencia disponible. Devuelve un valor único cada vez que se hace referencia a dicha columna, incluso para diferentes usuarios.
* CURRVAL obtiene el valor de secuencia actual.
* NEXTVAL se debe emitir para dicha secuencia antes de que CURRVAL contenga un valor.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Pseudocolumnas NEXTVAL y CURRVAL

Después de crear la secuencia, se generan números secuenciales para utilizarlos en las tablas. Haga referencia a los valores de secuencia mediante las pseudocolumnas **NEXTVAL** y **CURRVAL**.

La pseudocolumna **NEXTVAL** se utiliza para extraer números de secuencia sucesivos de una secuencia especificada. Debe cualificar a **NEXTVAL** con el nombre de secuencia. Al hacer referencia a *sequence*. **NEXTVAL**, se genera un nuevo número de secuencia y el actual se sustituye en **CURRVAL**.

La pseudocolumna **CURRVAL** se utiliza para hacer referencia al número de secuencia que acaba de generar el usuario actual. Sin embargo, **NEXTVAL** se debe utilizar para generar un número de secuencia en la sesión del usuario actual antes de que se pueda hacer referencia a **CURRVAL**. Debe cualificar a **CURRVAL** con el nombre de secuencia. Al hacer referencia a *sequence*.**CURRVAL**, se muestra el último valor devuelto al proceso de ese usuario.

Reglas para el Uso de NEXTVAL y CURRVAL

* + Puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:
  + La lista SELECT de una sentencia SELECT que no forme parte de una subconsulta
  + La lista SELECT de una subconsulta de la sentencia INSERT
  + La cláusula VALUES de una sentencia INSERT
  + La cláusula SET de la sentencia UPDATE

Puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:

* + La lista SELECT de una vista
  + Una sentencia SELECT con la palabra clave DISTINCT
  + Una sentencia SELECT con las cláusulas GROUP BY, HAVING o ORDER BY
  + Una subconsulta en una sentencia SELECT, DELETE o UPDATE
  + La expresión DEFAULT en una sentencia CREATE TABLE o ALTER TABLE



SELECT dept\_deptid\_seq.CURRVAL

FROM dual;

INSERT INTO departments(department\_id,

department\_name,

location\_id)

VALUES (dept\_deptid\_seq.NEXTVAL,

'Support', 2500);



**Uso de una Secuencia**

* Insertar un nuevo departamento denominado “Support” en la ubicación con el ID 2500:
* Consultar el valor actual de la secuencia

DEPT\_DEPTID\_SEQ:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Uso de una Secuencia

En el ejemplo de la diapositiva se inserta un nuevo departamento en la tabla DEPARTMENTS. Utiliza la secuencia DEPT\_DEPTID\_SEQ Para generar un nuevo número de departamento como se muestra a continuación.

Puede visualizar el valor actual de la secuencia mediante *sequence\_name*.CURRVAL, se muestra en el ejemplo de la diapositiva. La salida de la consulta se muestra a continuación:



Supongamos que ahora desea contratar empleados para el nuevo departamento. La sentencia **INSERT** que se debe ejecutar para los nuevos empleados puede incluir el siguiente código:

INSERT INTO employees ( employee\_id,

department\_id, ...)

VALUES( employees\_seq.NEXTVAL,

dept\_deptid\_seq.CURRVAL, ...);

**Nota:** en el ejemplo anterior, se asume que la secuencia denominada EMPLOYEE\_SEQ ya se ha creado para generar los números de los nuevos empleados.



**Almacenamiento en Caché de los Valores de Secuencia**

* El almacenamiento en caché de los valores de secuencia en la memoria proporciona un acceso más rápido a dichos valores.
* Los intervalos en los valores de secuencia se producen cuando:
  + Se realiza un rollback
  + El sistema falla
  + Una secuencia se utiliza en otra tabla

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Almacenamiento en Caché de los Valores de Secuencia

Puede almacenar las secuencias en caché en la memoria para proporcionar un acceso más rápido a los valores de dichas secuencias. La caché se rellena la primera vez que hace referencia a la secuencia. Las solicitudes del siguiente valor de secuencia se recuperan de la secuencia almacenada en caché. Después de utilizar el último valor de secuencia, la siguiente solicitud de la secuencia introduce otra caché de secuencias en la memoria.

**Intervalos en la Secuencia**

Aunque los generadores de secuencias emiten números secuenciales sin intervalos, esta acción se realiza independientemente de que se realice una confirmación o un rollback. Por lo tanto, si realiza un rollback de una sentencia que contiene una secuencia, se pierde el número.

Otro evento que puede provocar intervalos en la secuencia es un fallo del sistema. Si la secuencia almacena los valores en caché en la memoria, estos valores se pierden si se produce un fallo del sistema.

Puesto que las secuencias no están ligadas directamente a las tablas, se puede utilizar la misma secuencia para varias tablas. Si lo hace así, cada tabla puede contener intervalos en los números secuenciales.



ALTER SEQUENCE dept\_deptid\_seq

INCREMENT BY 20

MAXVALUE 999999 NOCACHE NOCYCLE;



**Modificación de una Secuencia**

Cambiar el valor incremental, el valor máximo, el valor mínimo, la opción de ciclo o la opción de caché:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Modificación de una Secuencia

Si alcanza el límite **MAXVALUE** para la secuencia, no se asignan valores adicionales de la secuencia y recibirá un error que indica que la secuencia excede el valor **MAXVALUE**. Para seguir utilizando la secuencia, puede modificarla mediante la sentencia **ALTER SEQUENCE**.

Sintaxis

ALTER SEQUENCE *secuencia*

[INCREMENT BY *n*]

[{MAXVALUE *n* | NOMAXVALUE}] [{MINVALUE *n* | NOMINVALUE}] [{CYCLE | NOCYCLE}] [{CACHE *n* | NOCACHE}];

En la sintaxis, ***sequence*** es el nombre del generador de secuencias.



DROP SEQUENCE dept\_deptid\_seq;



**Instrucciones para Modificar una Secuencia**

* Debe ser el propietario o tener el privilegio ALTER para la secuencia.
* Sólo se ven afectados los números de secuencia futuros.
* La secuencia se debe borrar y volver a crear para reiniciar la secuencia en un número diferente.
* Se realiza alguna validación.
* Para eliminar una secuencia, utilice la sentencia DROP:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Instrucciones para Modificar una Secuencia

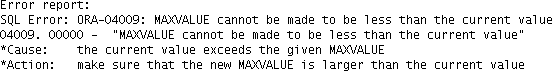
* Debe ser el propietario o tener el privilegio **ALTER** para la secuencia si desea modificarla. Debe ser el propietario o tener el privilegio **DROP ANY SEQUENCE** para eliminarla.
* Sólo se ven afectados por la sentencia **ALTER SEQUENCE** los números de secuencia futuros.
* La opción **START WITH** no se puede cambiar mediante **ALTER SEQUENCE**. La secuencia se debe borrar y volver a crear para reiniciar la secuencia en un número diferente.
* Se realiza alguna validación. Por ejemplo, no se puede imponer un nuevo valor **MAXVALUE** menor que el número de secuencia actual.

ALTER SEQUENCE dept\_deptid\_seq

INCREMENT BY 20

MAXVALUE 90 NOCACHE NOCYCLE;

* El error:



## ÍNDICE

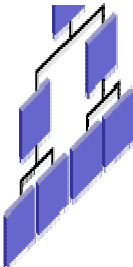


**Índices**

|  |  |
| --- | --- |
| **Objeto** | **Descripción** |
| Tabla | Unidad básica de almacenamiento; compuesta por filas. |
| Vista | Representa de forma lógica subjuegos de datos de una o más tablas. |
| Secuencia | Genera valores numéricos. |
| Índice | Mejora el rendimiento de algunas consultas. |
| Sinónimo | Ofrece nombres alternativos para los objetos. |

## Índices

Los índices son objetos de base de datos que puede crear para mejorar el rendimiento de algunas consultas. Los índices también los puede crear automáticamente el servidor al crear una restricción de clave primaria o única.



**Índices**

Un índice:

* Es un objeto de esquema
* Lo puede utilizar el servidor de Oracle para acelerar la recuperación de filas mediante un puntero
* Puede reducir la entrada/salida (E/S) de disco mediante un método de ruta de acceso rápido para buscar datos de forma rápida
* Es independiente de la tabla que indexa
* Lo utiliza y mantiene automáticamente el servidor de Oracle



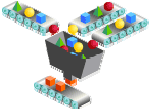
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Un índice de servidor de Oracle es un esquema de objeto que puede acelerar la recuperación de filas mediante un puntero. Los índices se pueden crear explícita o automáticamente. Si no hay un índice en la columna, se produce una exploración de tabla completa.

Un índice proporciona acceso directo y rápido a las filas de una tabla. Su finalidad es reducir la E/S de disco mediante una ruta de acceso indexada para buscar datos de forma rápida. Un índice lo utiliza y mantiene automáticamente el servidor de Oracle. Después de crear un índice, no será necesaria ninguna intervención directa por parte del usuario.

Los índices son lógica y físicamente independientes de la tabla que indexan. Esto significa que se pueden crear o borrar en cualquier momento sin que afecten a las tablas base o a otros índices.

**Nota:** al borrar una tabla, se borran también los índices correspondientes.



**¿Cómo se Crean los Índices?**

* Automáticamente: al definir una restricción PRIMARY KEY o UNIQUE en una definición de tabla, se crea un índice único automáticamente.
* Manualmente: los usuarios pueden crear índices no únicos en las columnas para acelerar el acceso a las filas.



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### ¿Cómo se Crean los Índices?



Puede crear dos tipos de índices.

* **Índice único:** el servidor de Oracle crea automáticamente este índice al definir una columna en la tabla para tener una restricción **PRIMARY** **KEY** o **UNIQUE**. El nombre del índice es el nombre proporcionado a la restricción.
* **Índice no único:** se trata de un índice que puede crear el usuario. Por ejemplo, puede crear un índice de columna **FOREIGN** **KEY** para una unión en una consulta para mejorar la velocidad de recuperación.

Nota: puede crear manualmente un índice único, pero se recomienda crear una restricción única, que implícitamente crea el índice único.



CREATE [UNIQUE][BITMAP]INDEX *index*

ON *table* (*column*[, *column*]...);

CREATE INDEX emp\_last\_name\_idx

ON employees(last\_name);



**Creación de Índices**

* Crear un índice en una o más columnas:
* Mejorar la velocidad de acceso de consulta a la columna

LAST\_NAME de la tabla EMPLOYEES:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Creación de Índices

Para crear un índice en una o más columnas, emita la sentencia **CREATE INDEX**. En la sintaxis:

* index es el nombre del índice.
* table es el nombre de la tabla.
* column es el nombre de la columna de la tabla que se debe rellenar.

Especifique **UNIQUE** para indicar que el valor de la columna (o columnas) en la que se basa el índice debe ser único. Especifique BITMAP para indicar que el índice se debe crear con un bitmap para cada clave distinta, en lugar de indexar cada fila por separado. Los índices de bitmap almacenan los rowids asociados con un valor de clave como bitmap.



**Instrucciones de Creación de Índices**

|  |  |
| --- | --- |
| **Crear un índice si:** | |
|  | Una columna contiene una amplia variedad de valores |
|  | Una columna contiene un gran número de valores nulos |
|  | Una o más columnas se utilizan con frecuencia en conjunto en una cláusula WHERE o una condición de unión |
|  | La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2% al 4% de las filas en la tabla |
| **No crear un índice si:** | |
|  | No se suelen utilizar las columnas como condición en la consulta |
|  | La tabla es pequeña o se espera que la mayoría de las consultas recuperen más del 2% al 4% de las filas de la tabla |
|  | La tabla se actualiza con frecuencia |
|  | Se hace referencia a las columnas indexadas como parte de una expresión |

**Instrucciones de Creación de Índices**

**Más No Siempre Es Sinónimo de Mejor**

El hecho de tener más índices en una tabla no significa que las consultas sean más rápidas. Cada operación DML que se confirma en una tabla con índices implica la actualización de los índices. Cuanto mayor sea el número de índices asociados a una tabla, mayor será el esfuerzo que debe realizar el servidor de Oracle para actualizar todos los índices después de la operación DML.

Cuándo Crear un Índice

Por lo tanto, sólo debe crear índices si:

* La columna contiene una amplia variedad de valores
* La columna contiene un gran número de valores nulos
* Una o más columnas se utilizan con frecuencia en conjunto en una cláusula **WHERE** o una condición de unión
* La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2% al 4% de las filas

Recuerde que si desea reforzar la unicidad, debe definir una restricción única en la definición de tabla. En ese caso, se crea un índice único automáticamente.



DROP INDEX *index*;

DROP INDEX emp\_last\_name\_idx;



**Eliminación de Índices**

* Eliminar un índice del diccionario de datos mediante el comando DROP INDEX:
* Eliminar el índice emp\_last\_name\_idx del diccionario de datos:
* Para borrar un índice, debe ser el propietario del mismo o tener el privilegio DROP ANY INDEX.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Eliminación de Índices

No puede modificar los índices. Para cambiar un índice, debe borrarlo y volver a crearlo.

Eliminar una definición de índice del diccionario de datos mediante la emisión de la sentencia **DROP INDEX**. Para borrar un índice, debe ser el propietario del mismo o tener el privilegio **DROP ANY INDEX**.

En la sintaxis, *índice* es el nombre del índice.

**Nota:** si borra una tabla, los índices y restricciones se borran automáticamente, pero permanecen las vistas y secuencias.

## SINÓNIMOS

# Sinónimos



**Sinónimos**

|  |  |
| --- | --- |
| **Objeto** | **Descripción** |
| Tabla | Unidad básica de almacenamiento; compuesta por filas. |
| Vista | Representa de forma lógica subjuegos de datos de una o más tablas. |
| Secuencia | Genera valores numéricos. |
| Índice | Mejora el rendimiento de algunas consultas. |
| Sinónimo | Ofrece nombres alternativos para los objetos. |

Sinónimos

Los sinónimos son objetos de base de datos que permiten llamar a una tabla por otro nombre. Puede crear sinónimos para proporcionar un nombre alternativo a la tabla.

CREATE [PUBLIC] SYNONYM *synonym*

FOR *object*;



**Creación de un Sinónimo para un Objeto**

Simplificar el acceso a los objetos mediante la creación de un sinónimo (otro nombre para un objeto). Con los sinónimos, puede:

* Crear una referencia más sencilla a una tabla propiedad de otro usuario
* Acortar nombres de objetos largos

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Creación de un Sinónimo para un Objeto**

Para hacer referencia a una tabla propiedad de otro usuario, debe anteponer al nombre de la tabla el nombre del usuario que la haya creado, seguido de un punto. Con la creación de un sinónimo se elimina la necesidad de cualificar el nombre del objeto con el esquema y se ofrece nombres alternativos para tablas, vistas, secuencias, procedimientos u otros objetos. Este método es especialmente útil con nombres de objetos largos, como las vistas.

En la sintaxis:

**PUBLIC** crea un sinónimo al que pueden acceder todos los usuarios.

***synonym***es el nombre del sinónimo que se va a crear.

***object***identifica el objeto para el que se crea el sinónimo.

Instrucciones

* El objeto no puede estar en un paquete.
* Un nombre sinónimo privado debe ser distinto de todos los demás objetos propiedad del mismo usuario.



CREATE SYNONYM d\_sum

FOR dept\_sum\_vu;

DROP SYNONYM d\_sum;



**Creación y Eliminación de Sinónimos**

* Crear un nombre abreviado para la vista DEPT\_SUM\_VU:
* Borrar un sinónimo:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Creación y Eliminación de Sinónimos

**Creación de Sinónimos**

En la diapositiva del ejemplo se crea un sinónimo para la vista DEPT\_SUM\_VU para una referencia más rápida.

El administrador de la base de datos puede crear un sinónimo público al que puedan acceder todos los usuarios. En el siguiente ejemplo se crea un sinónimo público con el nombre DEPT para la tabla DEPARTMENTS de Alice:

CREATE PUBLIC SYNONYM dept

FOR alice.departments;



Eliminación de Sinónimos

Para eliminar un sinónimo, utilice la sentencia **DROP SYNONYM**. Sólo el administrador de la base de datos puede borrar un sinónimo público.

DROP PUBLIC SYNONYM dept;

# RESTRICCIONES



**Inclusión de Restricciones**

* Las restricciones aplican reglas a nivel de tabla.
* Las restricciones impiden la supresión de una tabla si hay dependencias.
* Los siguientes tipos de restricciones son válidos:
  + NOT NULL
  + UNIQUE
  + PRIMARY KEY
  + FOREIGN KEY
  + CHECK



Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

## Restricciones

El servidor de Oracle utiliza restricciones para evitar la introducción de datos no válidos en las tablas. Puede utilizar restricciones para realizar lo siguiente:

* + Aplicar reglas a los datos de la tabla cuando se inserta, actualiza o suprime una fila de la misma.
  + La restricción se debe cumplir para que la operación sea correcta.
  + Evitar la supresión de una tabla si hay dependencias de otras tablas.
  + Proporcionar reglas para las herramientas de Oracle, como Oracle Developer.

Restricciones de Integridad de Datos

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| **NOT NULL** | Especifica que la columna no puede contener un valor nulo. |
| **UNIQUE** | Especifica una columna o combinación de columnas cuyos valores deben ser únicos para todas las filas de la tabla. |
| **PRIMARY KEY** | Identifica de forma única cada fila de la tabla. |
| **FOREIGN KEY** | Establece y aplica la integridad referencial entre la columna y la columna de la tabla a la que se hace referencia; por ejemplo, busca  los valores de una tabla que coinciden con los valores de la otra tabla. |
| **CHECK** | Especifica una condición que debe ser verdadera. |



**Instrucciones de Restricción**

* Puede asignar un nombre a una restricción o el servidor de Oracle genera un nombre con el formato SYS\_C*n*.
* Crear una restricción en uno de los siguientes momentos:
  + En el momento de la creación de la tabla
  + Después de la creación de la tabla
* Definir una restricción a nivel de columna o de tabla.
* Ver una restricción en el diccionario de datos.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Instrucciones de Restricción**

Todas las restricciones se almacenan en el diccionario de datos. Es fácil hacer referencia a las restricciones si les asigna un nombre significativo. Los nombres de las restricciones deben seguir las reglas de nomenclatura de objetos estándar, excepto que el nombre no puede ser el mismo que el de otro objeto propiedad del mismo usuario. Si no asigna ningún nombre a la restricción, el servidor de Oracle genera un nombre con el formato SYS\_C*n*, donde *n* es un entero para que el nombre de la restricción sea único.

Las restricciones se pueden definir en el momento de la creación de la tabla o después de haberla creado. Puede definir una restricción a nivel de columna o de tabla. Funcionalmente, una restricción a nivel de tabla es la mismo que una restricción a nivel de columna.

columna,...

[CONSTRAINT constraint\_name] constraint\_type

CREATE TABLE [*schema*.]*table*

(*column datatype* [DEFAULT *expr*] [*column\_constraint*],

...

[*table\_constraint*][,...]);

column [CONSTRAINT constraint\_name] constraint\_type,



**Definición de Restricciones**

* Sintaxis:
* Sintaxis de restricción a nivel de columna:
* Sintaxis de restricción a nivel de tabla:

Definición de Restricciones

En la diapositiva se proporciona la sintaxis para la definición de restricciones al crear una tabla. Puede crear las restricciones a nivel de columna o de tabla. Las restricciones definidas a nivel de columna se incluyen al definir la columna. Las restricciones a nivel de tabla se definen al final de la definición de tabla y debe hacer referencia a la columna o las columnas a las que pertenece la restricción en un juego de paréntesis. Se trata principalmente de la sintaxis que diferencia a las dos; de lo contrario, funcionalmente, una restricción a nivel de columna es lo mismo que una restricción a nivel de tabla.

Las restricciones **NOT NULL** se deben definir a nivel de columna.

Las restricciones que se aplican a más de una columna se deben definir a nivel de tabla. En la sintaxis:

**schema** es el mismo nombre que el del propietario

**table** es el nombre de la tabla

**DEFAULT expr** especifica un valor por defecto para utilizarlo si se omite un valor en la sentencia **INSERT**

**column** es el nombre de la columna

**datatype** es el tipo de dato y la longitud de la columna

**column\_constraint es** una restricción de integridad como parte de la definición de columna.

**table\_constraint** es una restricción de integridad como parte de la definición de tabla.

**CREATE TABLE employees(employee\_id first\_name...**

**job\_id NUMBER(6), VARCHAR2(20), VARCHAR2(10) NOT NULL,**

**CONSTRAINT emp\_emp\_id\_pk PRIMARY KEY (EMPLOYEE\_ID));**

CREATE TABLE employees(

employee\_id NUMBER(6)

CONSTRAINT emp\_emp\_id\_pk PRIMARY KEY,

first\_name VARCHAR2(20),

...);



**Definición de Restricciones**

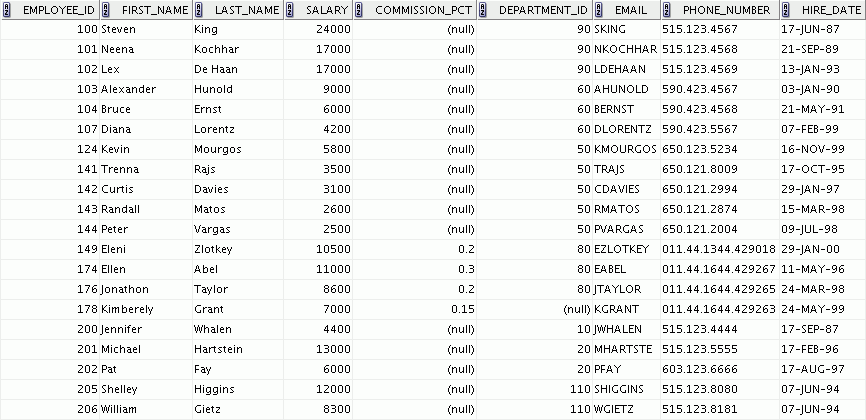
* Ejemplo de una restricción a nivel de columna: ¿
* Ejemplo de una restricción a nivel de tabla:

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Las restricciones se crean normalmente al mismo tiempo que la tabla. Las restricciones se pueden agregar a una tabla después de su creación y se pueden desactivar temporalmente.

Ambos ejemplos de la diapositiva crean una restricción de clave primaria en la columna EMPLOYEE\_ID de la tabla EMPLOYEES.

1. En el primer ejemplo se utiliza la sintaxis de nivel de columna para definir la restricción.
2. En el segundo ejemplo se utiliza la sintaxis de nivel de tabla para definir la restricción.



**Ausencia de la restricción NOT NULL (Cualquier fila puede contener un valor nulo para esta columna.)**

**NOT NULL**

**restricción**

**Restricción NOT NULL (La clave primaria aplica la restricción NOT NULL.)**

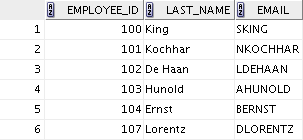
Garantiza que no se permiten los valores nulos para la columna:

**Restricción NOT NULL**

### Restricción NOT NULL

La restricción **NOT NULL** garantiza que la columna no contiene ningún valor nulo. Las columnas sin la restricción **NOT NULL** pueden contener valores nulos por defecto. Las restricciones **NOT NULL** se deben definir a nivel de columna. En la tabla EMPLOYEES, la columna EMPLOYEE\_ID hereda una restricción **NOT NULL** como se define en la clave primaria. De lo contrario, las columnas LAST\_NAME, EMAIL, HIRE\_DATE y JOB\_ID tienen la restricción **NOT NULL** aplicada.

**Nota:** la restricción de clave primaria se trata con detalle más adelante en esta misma lección.



**INSERT INTO**

**Permitido**

**No permitido: ya existe**

**…**

**Restricción UNIQUE**

**EMPLOYEES**

**Restricción UNIQUE**

### Restricción UNIQUE

Una restricción de integridad de clave **UNIQUE** necesita que todos los valores de una columna o juego de columnas (clave) sean únicos; es decir, que dos filas de una tabla no pueden tener valores duplicados en una columna o juego de columnas concreto. La columna (o juego de columnas) incluida en la definición de la restricción de clave **UNIQUE** se denomina *clave única.* Si la restricción **UNIQUE** comprende más de una columna, el grupo de columnas se denomina *clave única compuesta*.

Las restricciones **UNIQUE** permiten la entrada de valores nulos a menos que defina también las restricciones **NOT NULL** para las mismas columnas. De hecho, se puede incluir cualquier número de valores nulos para las columnas sin restricciones **NOT NULL** porque los valores nulos no se consideran iguales a ningún otro elemento. Un valor nulo en una columna (o en todas las columnas de una clave **UNIQUE** compuesta) cumple siempre una restricción **UNIQUE**.

**Nota:** debido al mecanismo de búsqueda de restricciones **UNIQUE** en una o más columnas, no puede tener valores idénticos en las columnas no nulas de una restricción de clave **UNIQUE** compuesta parcialmente nula.

CREATE TABLE employees( employee\_id NUMBER(6),

last\_name VARCHAR2(25) NOT NULL,

email VARCHAR2(25),

salary NUMBER(8,2),

commission\_pct NUMBER(2,2),

hire\_date DATE NOT NULL,

...

CONSTRAINT emp\_email\_uk UNIQUE(email));



**Restricción UNIQUE**

Definida a nivel de tabla o de columna:

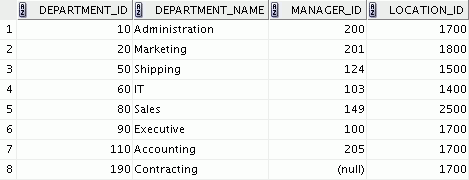
Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**Las restricciones UNIQUE** se pueden definir a nivel de tabla o de columna. Defina la restricción a nivel de tabla si desea crear una clave única compuesta. Una clave compuesta se define si no hay un único atributo que pueda identificar de forma única una fila. En ese caso, puede tener una clave única compuesta de dos o más columnas, el valor combinado de ambas es siempre único y puede identificar filas.

En el ejemplo de la diapositiva se aplica la restricción **UNIQUE** a la columna EMAIL de la tabla

EMPLOYEES. El nombre de la restricción es EMP\_EMAIL\_UK.

**Nota:** el servidor de Oracle aplica la restricción **UNIQUE** mediante la creación implícita de un índice único en la columna o columnas únicas.



**No permitido (ya existen 50)**

**INSERT INTO**

**No permitido (valor nulo)**

**PRIMARY KEY**

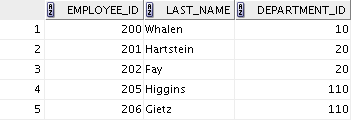
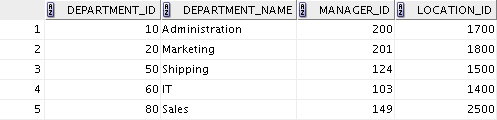
**DEPARTMENTS**

**Restricción PRIMARY KEY**

### Restricción PRIMARY KEY

La restricción **PRIMARY KEY** crea una clave primaria para la tabla. Sólo se puede crear una clave primaria para cada tabla. La restricción **PRIMARY KEY** es una columna o juego de columnas que identifica de forma única cada fila de tabla. Esta restricción aplica la unicidad de la columna o combinación de columnas y garantiza que ninguna columna que forme parte de la clave contenga un valor nulo.

**Nota:** puesto que la unicidad forma parte de la definición de restricción de clave primaria, el servidor de Oracle aplica la unicidad mediante la creación implícita de un índice único en la columna o columnas de clave primaria.



**No permitido (9 no existen)**

**Permitido**

**INSERT INTO**

**…**

**FOREIGN KEY**

**…**

**EMPLOYEES**

**PRIMARY KEY**

**Restricción PRIMARY KEY**

**DEPARTMENTS**

### Restricción FOREIGN KEY

La restricción **FOREIGN KEY** (o integridad referencial) designa una columna o combinación de columnas como clave ajena y establece una relación entre una clave primaria o única en la misma tabla o en otra diferente.

En el ejemplo de la diapositiva, DEPARTMENT\_ID se ha definido como clave ajena en la tabla EMPLOYEES (tabla dependiente o secundaria); hace referencia a la columna DEPARTMENT\_ID de la tabla DEPARTMENTS (tabla a la que se hace referencia o principal).

Instrucciones

* + El valor de clave ajena debe coincidir con un valor existente de la tabla principal o ser un valor **NULL**.
  + Las claves ajenas se basan en los valores de datos y son punteros puramente lógicos, en lugar de físicos.

CREATE TABLE employees( employee\_id NUMBER(6),

last\_name VARCHAR2(25) NOT NULL,

email VARCHAR2(25),

salary NUMBER(8,2),

commission\_pct NUMBER(2,2),

hire\_date DATE NOT NULL,

...

department\_id NUMBER(4),

CONSTRAINT emp\_dept\_fk FOREIGN KEY (department\_id) REFERENCES departments(department\_id),

CONSTRAINT emp\_email\_uk UNIQUE(email));



**Restricción PRIMARY KEY**

Definida a nivel de tabla o de columna:

**Las restricciones FOREIGN KEY** se pueden definir a nivel de restricción de columna o de tabla. Las claves ajenas compuestas se deben crear mediante la definición a nivel de tabla.

En el ejemplo de la diapositiva se define una restricción **FOREIGN KEY** en la columna DEPARTMENT\_ID de la tabla EMPLOYEES, mediante la sintaxis a nivel de tabla. El nombre de la restricción es EMP\_DEPT\_FK.

La clave ajena se puede definir también a nivel de columna, siempre que la restricción está basada en una sola columna. La sintaxis difiere en que las palabras clave **FOREIGN KEY** no aparecen. Por ejemplo:

CREATE TABLE employees (...

department\_id NUMBER(4)

CONSTRAINT emp\_deptid\_fk

REFERENCES departments(department\_id),

...)



**Restricción FOREIGN KEY: Palabras Clave**

* FOREIGN KEY: define la columna en la tabla secundaria a nivel de restricción de tabla
* REFERENCES: identifica la tabla y la columna en la tabla principal
* ON DELETE CASCADE: suprime las filas dependientes de la tabla secundaria cuando se suprime una fila de la tabla principal
* ON DELETE SET NULL: convierte los valores de clave ajena dependiente en nulos.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Restricción FOREIGN KEY: Palabras Clave

La clave ajena se define en la tabla secundaria y la tabla que contiene la columna a la que se hace referencia es la tabla principal. La clave ajena se define mediante una combinación de las siguientes palabras clave:

* **FOREIGN KEY** se utiliza para definir la columna en la tabla secundaria a nivel de restricción de tabla.
* **REFERENCES** identifica la tabla y la columna en la tabla principal.
* **ON DELETE CASCADE** indica que al suprimir una fila de la tabla principal, también se suprimen las filas dependientes de la tabla secundaria.
* **ON DELETE SET NULL** indica que al suprimir una fila de la tabla principal, los valores de clave ajena se definen como nulos.

El comportamiento por defecto se denomina *regla estricta*, que no permite la actualización o supresión de los datos a los que se hace referencia.

Sin las opciones **ON DELETE CASCADE** u **ON DELETE SET NULL**, la fila de la tabla principal no se puede suprimir si hace referencia a la misma en la tabla secundaria.

..., salary NUMBER(2) CONSTRAINT emp\_salary\_min

CHECK (salary > 0),...



**Restricción CHECK**

* Define una condición que debe cumplir cada fila.
* Las siguientes expresiones no están permitidas:
  + Referencias a las pseudocolumnas CURRVAL, NEXTVAL, LEVEL y ROWNUM
  + Llamadas a las funciones SYSDATE, UID, USER y USERENV
  + Consultas que hagan referencia a otros valores en otras filas

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

### Restricción CHECK

La restricción **CHECK** define una condición que debe cumplir cada fila. La condición puede utilizar las mismas construcciones que las condiciones de consulta, con las siguientes excepciones:

* + Referencias a las pseudocolumnas **CURRVAL**, **NEXTVAL**, **LEVEL** y **ROWNUM**
  + Llamadas a las funciones **SYSDATE**, **UID**, **USER** y **USERENV**
  + Consultas que hagan referencia a otros valores en otras filas

Una sola columna puede tener varias restricciones **CHECK** que hagan referencia a la columna en su definición. No hay ningún límite en cuanto al número de restricciones **CHECK** que puede definir en una columna.

Las restricciones **CHECK** se pueden definir a nivel de tabla o de columna.

CREATE TABLE employees (...

salary NUMBER(8,2)

CONSTRAINT emp\_salary\_min

CHECK (salary > 0),

...



**Sentencia CREATE TABLE: Ejemplo ¿**

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

**CONSTRAINT emp\_salary\_ck CHECK (salary>0)**

**, commission\_pct NUMBER(2,2)**

**, manager\_id NUMBER(6)**

**CONSTRAINT emp\_manager\_fk REFERENCES employees (employee\_id)**

**, department\_id NUMBER(4)**

**CONSTRAINT emp\_dept\_fk REFERENCES departments (department\_id));**

**NOT NULL**

**NOT NULL**

**NOT NULL UNIQUE**

**emp\_email\_nn emp\_email\_uk**

**VARCHAR2(20) DATE**

**emp\_hire\_date\_nn VARCHAR2(10)**

**emp\_job\_nn NUMBER(8,2)**

**VARCHAR2(25)**

**NOT NULL**

**emp\_last\_name\_nn**

**PRIMARY KEY**

**NUMBER(6)**

**emp\_employee\_id VARCHAR2(20) VARCHAR2(25)**

**CREATE TABLE employees ( employee\_id**

**CONSTRAINT**

**, first\_name**

**, last\_name**

**CONSTRAINT**

**, email**

**CONSTRAINT CONSTRAINT**

**, phone\_number**

**, hire\_date**

**CONSTRAINT**

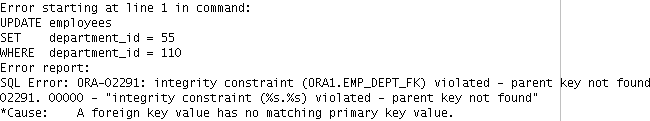
**, job\_id**

**CONSTRAINT**

**, salary**

Sentencia CREATE TABLE: Ejemplo

El ejemplo de la diapositiva muestra la sentencia que se utiliza para crear la tabla EMPLOYEES en el esquema HR.



UPDATE employees

SET department\_id = 55

WHERE department\_id = 110;



**Violación de Restricciones**

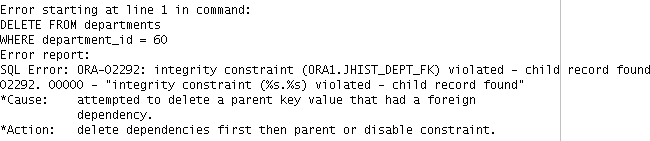
El departamento 55 no existe.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Violación de Restricciones

Si tiene restricciones en lugar de columnas, recibirá un error si intenta violar la regla de restricción. Por ejemplo, si intenta actualizar un registro con un valor ligado a una restricción de integridad, se devuelve un error.

En el ejemplo de la diapositiva, el departamento 55 no existe en la tabla principal, DEPARTMENTS y, por lo tanto, recibirá la violación de clave principal no encontrada ORA-02291.



DELETE FROM departments

WHERE department\_id = 60;



**Violación de Restricciones**

No puede suprimir una fila que contenga una clave primaria que se utilice como clave ajena en otra tabla.

Copyright © 2010, Oracle y/o sus filiales. Todos los derechos reservados.

Por ejemplo, si intenta suprimir un registro con un valor ligado a una restricción de integridad, se devuelve un error.

En el ejemplo de la diapositiva, se intenta suprimir el departamento 60 de la tabla DEPARTMENTS, pero se produce un error porque ese número de departamento se utiliza como clave ajena en la tabla EMPLOYEES. Si el registro principal que intenta suprimir tiene registros secundarios, recibirá la violación de registro secundario encontrado ORA-02292.

La siguiente sentencia funciona porque no hay ningún empleado en el departamento 70:

DELETE FROM departments

WHERE department\_id = 70;

****

# TRIGGERS

Utilice la CREATE TRIGGERdeclaración para crear y habilitar un **activador de base de datos** , que es:

* Un bloque PL / SQL almacenado asociado con una tabla, un esquema o la base de datos o
* Un bloque PL / SQL anónimo o una llamada a un procedimiento implementado en PL / SQL o Java

Oracle Database ejecuta automáticamente un disparador cuando ocurren condiciones específicas.

Cuando crea un disparador, la base de datos lo habilita automáticamente. Posteriormente, puede deshabilitar y habilitar un disparador con la cláusula DISABLEy ENABLEde la declaración ALTER TRIGGERo ALTER TABLE.

**Ver también:**

* [*Conceptos de base de datos de Oracle*](https://docs.oracle.com/cd/B19306_01/server.102/b14220/triggers.htm#CNCPT017) para obtener una descripción de los distintos tipos de activadores y[*la Guía del desarrollador de aplicaciones de base de datos Oracle: Fundamentos*](https://docs.oracle.com/cd/B19306_01/B14251_01/adfns_triggers.htm#ADFNS012) para obtener más información sobre cómo diseñar activadores.
* [ALTER TRIGGER](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4001.htm#BABFDCBJ) y [ALTER TABLE](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_3001.htm#CJAHHIBI) para obtener información sobre cómo habilitar, deshabilitar y compilar disparadores, y [DROP TRIGGER](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9005.htm#i2061473) para obtener información sobre cómo soltar un disparador

**Prerrequisitos**

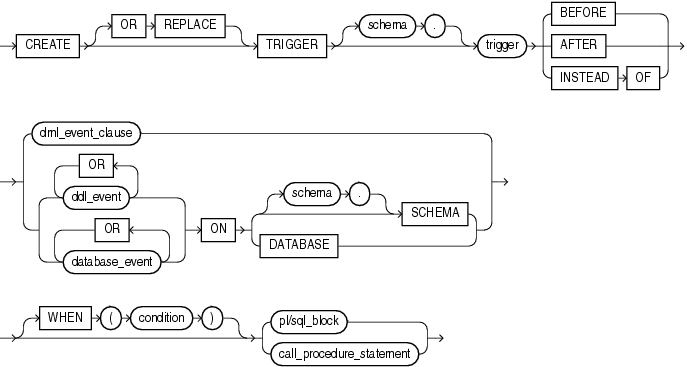
Antes de que se pueda crear un disparador, el usuario SYSdebe ejecutar un script SQL comúnmente llamado DBMSSTDX.SQL. El nombre exacto y la ubicación de este script dependen de su sistema operativo.

* Para crear un disparador en su propio esquema en una tabla en su propio esquema o en su propio esquema ( SCHEMA), debe tener CREATE TRIGGERprivilegios de sistema.
* Para crear un disparador en cualquier esquema en una tabla en cualquier esquema, o en el esquema de otro usuario ( *schema*. SCHEMA), Debe tener CREATE ANY TRIGGERprivilegios de sistema.
* Además de los privilegios anteriores, para crear un desencadenador DATABASE, debe tener el ADMINISTER DATABASE TRIGGERprivilegio del sistema.

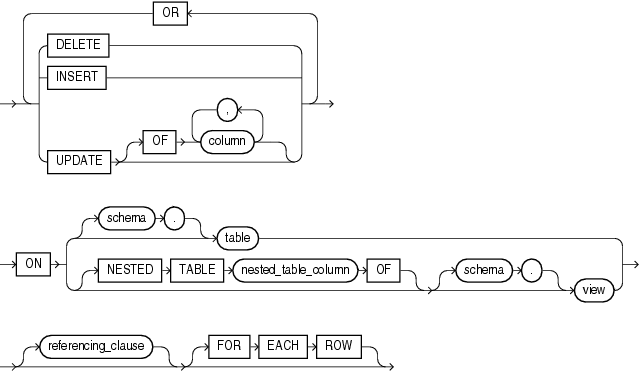
Si el disparador emite sentencias SQL o llama a procedimientos o funciones, entonces el propietario del disparador debe tener los privilegios necesarios para realizar estas operaciones. Estos privilegios deben otorgarse directamente al propietario en lugar de adquirirse a través de roles.

**Sintaxis**

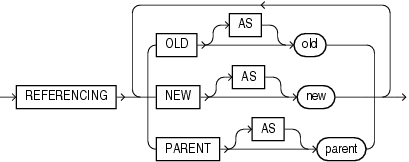
***create\_trigger* :: =**

  
[Descripción de la ilustración create\_trigger.gif](https://docs.oracle.com/cd/B19306_01/server.102/b14200/img_text/create_trigger.htm)

[***DML\_event\_clause***](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064026)**:: =**

  
[Descripción de la ilustración DML\_event\_clause.gif](https://docs.oracle.com/cd/B19306_01/server.102/b14200/img_text/DML_event_clause.htm)

[***cláusula\_de\_referencia***](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064286)**:: =**

  
[Descripción de la ilustración referencing\_clause.gif](https://docs.oracle.com/cd/B19306_01/server.102/b14200/img_text/referencing_clause.htm)

**Semántica**

**O REEMPLAZAR**

Especifique OR REPLACEvolver a crear el activador si ya existe. Utilice esta cláusula para cambiar la definición de un disparador existente sin primero eliminarlo.

***esquema***

Especifique el esquema para contener el disparador. Si omite *schema*, Oracle Database crea el disparador en su propio esquema.

***desencadenar***

Especifique el nombre del disparador que se creará.

Si un disparador produce errores de compilación, aún se crea, pero falla en la ejecución. Esto significa que bloquea efectivamente todas las declaraciones DML que se activan hasta que se deshabilita, se reemplaza por una versión sin errores de compilación o se elimina. Puede ver los mensajes de error del compilador asociados con el comando SQL \* Plus SHOW ERRORS.

**Nota:**

Si crea un disparador en una tabla base de una vista materializada, debe asegurarse de que el disparador no se dispare durante una actualización de la vista materializada. Durante la actualización, el DBMS\_MVIEWprocedimiento I\_AM\_A\_REFRESHvuelve TRUE.

**ANTES DE**

Especifique BEFOREpara hacer que la base de datos active el activador antes de ejecutar el evento activador. Para los activadores de fila, el activador se activa antes de que se cambie cada fila afectada.

**Restricciones en ANTES de disparadores**BEFORE Los disparadores están sujetos a las siguientes restricciones:

* No puede especificar un BEFOREdisparador en una vista o una vista de objeto.
* Puede escribir en el NEWvalor : pero no en el OLDvalor : .

**DESPUÉS**

Especifique AFTERpara hacer que la base de datos active el activador después de ejecutar el evento activador. Para los activadores de fila, el activador se activa después de que se cambia cada fila afectada.

**Restricciones en AFTER Triggers**AFTER Los disparadores están sujetos a las siguientes restricciones:

* No puede especificar un AFTERdesencadenante en una vista o una vista de objeto.
* No se puede escribir cualquiera de los: OLDo: NEWvalor.

**Nota:**

Cuando crea un registro de vista materializado para una tabla, Oracle Database crea implícitamente un AFTER ROWdisparador en la tabla. Este disparador inserta una fila en el registro de vista materializada cada vez que una INSERT, UPDATEo DELETEdeclaración modifica los datos en la tabla maestra. No puede controlar el orden en el que se disparan varias filas. Por lo tanto, no debe escribir desencadenantes destinados a afectar el contenido de la vista materializada.

**Ver también:**

[CREAR REGISTRO DE VISTA MATERIALIZADO](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6003.htm#i2064649) para obtener más información sobre los registros de vista materializada

**EN LUGAR DE**

Especifique INSTEAD OFpara hacer que Oracle Database active el disparador en lugar de ejecutar el evento disparador. INSTEAD OFLos desencadenantes son válidos para eventos DML en vistas. No son válidos para eventos de base de datos o DDL.

Si una vista es inherentemente actualizable y tiene INSTEAD OFactivadores, entonces los activadores tienen preferencia. En otras palabras, la base de datos activa los activadores en lugar de realizar DML en la vista.

Si la vista pertenece a una jerarquía, las subvistas no heredan el desencadenante.

**Nota:**

El control de acceso detallado de Oracle Database le permite definir políticas de seguridad a nivel de fila en las vistas. Estas políticas imponen reglas específicas en respuesta a las operaciones de DML. Si INSTEAD OFtambién se define un disparador en la vista, la base de datos no aplicará las políticas de seguridad de nivel de fila, porque la base de datos activa el INSTEAD OFdisparador en lugar de ejecutar el DML en la vista.

**EN LUGAR DE DETECTORES**

* INSTEAD OFlos activadores son válidos solo para las vistas. No puede especificar un INSTEAD OFdisparador en una mesa.
* Puede leer tanto el: OLDcomo el: NEWvalor, pero no puede escribir el: OLDo el: NEWvalor.

**Nota:**

Puede crear varios activadores del mismo tipo ( BEFORE, AFTERo INSTEAD OF) que se activan para la misma declaración en la misma tabla. El orden en el que Oracle Database activa estos activadores es indeterminado. Si su aplicación requiere que un disparador se active antes que otro del mismo tipo para la misma declaración, combine estos disparadores en un solo disparador cuya acción de disparador realiza las acciones de disparador de los disparadores originales en el orden apropiado.

**Ver también:**

["Creación de un disparador EN VEZ DE: Ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064426)

***DML\_event\_clause***

Le *DML\_event\_clause*permite especificar una de las tres declaraciones DML que pueden hacer que se active el disparador. Oracle Database dispara el disparador en la transacción de usuario existente.

No puede especificar la MERGEpalabra clave en el *DML\_event\_clause*. Si desea que se active un disparador en relación con una MERGEoperación, debe crear disparadores en las operaciones INSERTy UPDATEen las que se MERGEdescompone la operación.

**Ver también:**

["Creación de un activador DML: ejemplos"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2153487)

**ELIMINAR**

Especifique DELETEsi desea que la base de datos active el activador cada vez que una DELETEdeclaración elimina una fila de la tabla o elimina un elemento de una tabla anidada.

**INSERTAR**

Especifique INSERTsi desea que la base de datos active el disparador cada vez que una INSERTdeclaración agrega una fila a una tabla o agrega un elemento a una tabla anidada.

**ACTUALIZAR**

Especifique UPDATEsi desea que la base de datos active el disparador cada vez que una UPDATEdeclaración cambie un valor en una de las columnas especificadas a continuación OF. Si omite OF, la base de datos activa el disparador cada vez que una UPDATEdeclaración cambia un valor en cualquier columna de la tabla o tabla anidada.

Para un UPDATEactivador, puede especificar el tipo de objeto, varray y REFcolumnas después OFpara indicar que el activador debe activarse siempre que una UPDATEdeclaración cambie un valor en una de las columnas. Sin embargo, no puede cambiar los valores de estas columnas en el cuerpo del propio desencadenador.

**Nota:**

El uso de funciones OCI o el DBMS\_LOBpaquete para actualizar valores LOB o atributos LOB de columnas de objeto no hace que Oracle Database active los activadores definidos en la tabla que contiene las columnas o los atributos.

**Restricciones sobre los activadores de las operaciones UPDATE**La UPDATEcláusula está sujeta a las siguientes restricciones:

* No puede especificar UPDATE OFun INSTEAD OFdisparador. Oracle Database dispara INSTEAD OFactivadores cada vez que UPDATEcambia un valor en cualquier columna de la vista.
* No puede especificar una tabla anidada o una columna LOB en la UPDATE OFcláusula.

**Ver también:**

AS *subquery*cláusula de [CREATE VIEW](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8004.htm#i2065510) para obtener una lista de construcciones que impiden inserciones, actualizaciones o eliminaciones en una vista

La realización de operaciones DML directamente en las columnas de la tabla anidada no hace que Oracle Database active los activadores definidos en la tabla que contiene la columna de la tabla anidada.

***ddl\_event***

Especifique uno o más tipos de declaraciones DDL que pueden hacer que se active el disparador. Puede crear activadores para estos eventos en DATABASEo a SCHEMAmenos que se indique lo contrario. Puede crear BEFOREy AFTERactivadores para estos eventos. Oracle Database dispara el disparador en la transacción de usuario existente.

**Restricción de activadores en eventos DDL**No puede especificar como evento desencadenante ninguna operación DDL realizada mediante un procedimiento PL / SQL.

**Ver también:**

["Creación de un activador DDL: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2153503)

Los siguientes *ddl\_event*valores son válidos:

**ALTERAR**Especifique ALTERque se active el disparador siempre que una ALTERdeclaración modifique un objeto de base de datos en el diccionario de datos.

**Restricción de disparadores en operaciones ALTER**El gatillo no será disparado por una ALTER DATABASEdeclaración.

**ANALIZAR**Especifique ANALYZEque se active el disparador cada vez que la base de datos recopile o elimine estadísticas o valide la estructura de un objeto de base de datos.

**Ver también:**

[ANALIZAR](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4005.htm#i2086320) para obtener información sobre diversas formas de recopilar estadísticas

**ESTADÍSTICAS ASOCIADAS**Especifique ASSOCIATE STATISTICSque se active el disparador siempre que la base de datos asocie un tipo de estadística con un objeto de base de datos.

**AUDITORÍA**Especifique AUDITque se active el disparador siempre que la base de datos rastree la ocurrencia de una instrucción SQL o rastree las operaciones en un objeto de esquema.

**COMENTARIO**Especifique COMMENTque se active el disparador siempre que se agregue un comentario sobre un objeto de base de datos al diccionario de datos.

**CREAR**Especifique CREATEque se active el disparador siempre que una CREATEdeclaración agregue un nuevo objeto de base de datos al diccionario de datos.

**Restricción de disparadores en operaciones CREATE**El disparador no se activará con una declaración CREATE DATABASEo CREATE CONTROLFILE.

**ESTADÍSTICAS DESASOCIADAS**Especifique DISASSOCIATE STATISTICSque se active el activador siempre que la base de datos disocie un tipo de estadística de un objeto de base de datos.

**SOLTAR**Especifique DROPque se active el disparador cada vez que una DROPdeclaración elimine un objeto de base de datos del diccionario de datos.

**CONCEDER**Especifique GRANTque se active el activador siempre que un usuario otorgue privilegios o roles del sistema o privilegios de objeto a otro usuario o rol.

**NOAUDIT**Especifique NOAUDITque se active el disparador siempre que una NOAUDITdeclaración indique a la base de datos que deje de rastrear una declaración o operaciones SQL en un objeto de esquema.

**REBAUTIZAR**Especifique RENAMEque se active el disparador cada vez que una RENAMEdeclaración cambie el nombre de un objeto de base de datos.

**REVOCAR**Especifique REVOKEque se active el disparador siempre que una REVOKEdeclaración elimine privilegios o roles del sistema o privilegios de objeto de un usuario o rol.

**TRUNCAR**Especifique TRUNCATEque se active el disparador cada vez que una TRUNCATEdeclaración elimine las filas de una tabla o clúster y restablezca sus características de almacenamiento.

**DDL**Especifique DDLque se active el disparador siempre que se emita alguna de las declaraciones DDL anteriores.

***database\_event***

Especifique uno o más estados particulares de la base de datos que pueden hacer que se active el disparador. Puede crear activadores para estos eventos en DATABASEo a SCHEMAmenos que se indique lo contrario. Para cada uno de estos eventos desencadenantes, Oracle Database abre un alcance de transacción autónoma, activa el desencadenador y confirma cualquier transacción separada (independientemente de cualquier transacción de usuario existente).

**Ver también:**

["Creación de un activador de eventos de base de datos: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2153530)

**ERROR DEL SERVIDOR**Especifique SERVERERRORque se active el disparador siempre que se registre un mensaje de error del servidor.

Los siguientes errores no hacen que se SERVERERRORdispare un disparador:

* ORA-01403: datos no encontrados
* ORA-01422: la recuperación exacta devuelve más del número solicitado de filas
* ORA-01423: error encontrado al buscar filas adicionales en la recuperación exacta
* ORA-01034: ORACLE no disponible
* ORA-04030: sin memoria de proceso al intentar asignar *string*bytes ( *string*, *string*)

**INICIAR SESIÓN**Especifique LOGONque se active el disparador siempre que una aplicación cliente inicie sesión en la base de datos.

**DESCONECTARSE**Especifique LOGOFFque se active el disparador cada vez que una aplicación cliente cierre la sesión de la base de datos.

**PUESTA EN MARCHA**Especifique STARTUPque se active el disparador siempre que se abra la base de datos.

**APAGAR**Especifique SHUTDOWNque se active el disparador siempre que se cierre una instancia de la base de datos.

**SUSPENDER**Especifique SUSPENDque se active el disparador siempre que un error del servidor provoque la suspensión de una transacción.

**DB\_ROLE\_CHANGE**En una configuración de Data Guard, especifique DB\_ROLE\_CHANGEque se active el disparador siempre que se produzca un cambio de función de modo de espera a principal o de principal a modo de espera.

**Notas:**

* Solamente los AFTERdisparadores son relevantes para LOGON, STARTUP, SERVERERROR, SUSPEND, y DB\_ROLE\_CHANGE.
* Solo los BEFOREdesencadenantes son relevantes para LOGOFFy SHUTDOWN.
* AFTER STARTUPy los BEFORE SHUTDOWNdesencadenantes se aplican solo a DATABASE.

**Ver también:**

[*PL / SQL User's Guide and Reference*](https://docs.oracle.com/cd/B19306_01/appdev.102/b14261/toc.htm) para obtener más información sobre el alcance de las transacciones autónomas

**ON *table* | *ver***

La ONcláusula le permite determinar el objeto de base de datos en el que se creará el disparador. Especificar la *schema*y *table*o *view*nombre de uno de los siguientes en la que el gatillo se va a crear:

* Mesa o vista
* Tabla de objetos o vista de objetos
* Una columna de tipo tabla anidada

Si omite *schema*, Oracle Database asume que la tabla está en su propio esquema. Puede crear activadores en tablas organizadas por índices.

**Restricción en el esquema**No puede crear un disparador en una tabla en el esquema SYS.

**Cláusula de mesa anidada**

Especifique el *nested\_table\_column*de una vista en la que se define el disparador. Este disparador se activará solo si el DML opera en los elementos de la tabla anidada.

**Restricción de disparadores en tablas anidadas**Puede especificar NESTED TABLEsolo para INSTEAD OFdisparadores.

**BASE DE DATOS**

Especifique DATABASEpara definir el desencadenante en toda la base de datos. El disparador se activa siempre que cualquier usuario de la base de datos inicia el evento disparador.

**ESQUEMA**

Especifique SCHEMApara definir el disparador en el esquema actual. El disparador se *schema*activa siempre que cualquier usuario conectado como inicia el evento disparador.

**Ver también:**

["Creación de un activador de SCHEMA: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2110848)

***cláusula\_de\_referencia***

Le *referencing\_clause*permite especificar nombres de correlación. Puede usar nombres de correlación en el bloque PL / SQL y la WHENcondición de un desencadenador de fila para referirse específicamente a valores nuevos y antiguos de la fila actual. Los nombres de correlación predeterminados son OLDy NEW. Si su desencadenador de fila está asociado con una tabla denominada OLDo NEW, utilice esta cláusula para especificar diferentes nombres de correlación para evitar confusiones entre el nombre de la tabla y el nombre de correlación.

* Si el disparador se define en una tabla anidada, entonces OLDy hace NEWreferencia a la fila de la tabla anidada y PARENThace referencia a la fila actual de la tabla principal.
* Si el desencadenante se define en una tabla o vista de objetos, entonces OLDy NEWhaga referencia a instancias de objetos.

**Restricción de la cláusula de *referencia***No *referencing\_clause*es válido con INSTEAD OFdisparadores en CREATEeventos DDL.

**POR CADA FILA**

Especifique FOR EACH ROWpara designar el disparador como un disparador de fila. Oracle Database dispara un disparador de fila una vez por cada fila que se ve afectada por la declaración de disparo y cumple con la restricción de disparador opcional definida en la WHENcondición.

Excepto por los INSTEAD OFdisparadores, si omite esta cláusula, el disparador es un disparador de declaración. Oracle Database dispara un disparador de declaración solo una vez cuando se emite la declaración de disparo si se cumple la restricción de disparador opcional.

INSTEAD OF Las declaraciones de activación se activan implícitamente para cada fila.

**Restricción de activadores de fila**Esta cláusula es válida solo para activadores de eventos DML, no para activadores de eventos DDL o de base de datos.

**CUANDO Cláusula**

Especifique la condición del disparador, que es una condición de SQL que debe cumplirse para que la base de datos active el disparador. Consulte la descripción de la sintaxis de *condition*en el [Capítulo 7, "Condiciones"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/conditions.htm#g1077361) . Esta condición debe contener nombres de correlación y no puede contener una consulta.

Las palabras clave NEWy OLD, cuando se especifican en la WHENcláusula, no se consideran variables de vinculación, por lo que no están precedidas por dos puntos (:). Sin embargo, debe preceder NEWy OLDcon dos puntos en todas las referencias que no sean la WHENcláusula.

**Ver también:**

["Llamar a un procedimiento en un cuerpo de disparador: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064401)

**Restricciones sobre las condiciones desencadenantes**Las condiciones de activación están sujetas a las siguientes restricciones:

* Si especifica esta cláusula para un activador de evento DML, también debe especificar FOR EACH ROW. Oracle Database evalúa esta condición para cada fila afectada por la declaración de activación.
* No puede especificar condiciones de INSTEAD OFactivación para declaraciones de activación.
* Puede hacer referencia a columnas de objeto o sus atributos, o varray, tabla anidada o columnas LOB. No puede invocar funciones o métodos PL / SQL en la condición de activación.

***pl / sql\_block***

Especifique el bloque PL / SQL que Oracle Database ejecuta para disparar el disparador.

El bloque PL / SQL de un activador de base de datos puede contener una de una serie de funciones integradas en el SYSesquema diseñadas únicamente para extraer atributos de eventos del sistema. Estas funciones se pueden utilizar *solamente* en el bloque PL / SQL de un trigger.

**Restricciones en la implementación de activadores**La implementación de un disparador está sujeta a las siguientes restricciones:

* El bloque PL / SQL de un disparador no puede contener sentencias de control de transacciones de SQL ( COMMIT, ROLLBACK, SAVEPOINT, y SET CONSTRAINT) si el bloque se ejecuta dentro de la misma transacción.
* Puede hacer referencia y utilizar columnas LOB en la acción de activación dentro del bloque PL / SQL. Puede modificar los NEWvalores : pero no los OLDvalores : de las columnas LOB dentro de la acción de activación.

**Ver también:**

* [*PL / SQL User's Guide and Reference*](https://docs.oracle.com/cd/B19306_01/appdev.102/b14261/toc.htm) para obtener información sobre PL / SQL, incluido cómo escribir bloques PL / SQL
* [*Guía del desarrollador de aplicaciones de base de datos Oracle: fundamentos*](https://docs.oracle.com/cd/B19306_01/appdev.102/b14251/toc.htm) para obtener información sobre estas funciones
* ["Llamar a un procedimiento en un cuerpo de disparador: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064401)

***call\_procedure\_statement***

Le *call\_procedure\_statement*permite llamar a un procedimiento almacenado en lugar de especificar el código de activación en línea como un bloque PL / SQL. La sintaxis de esta declaración es la misma que la de [CALL](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_4008.htm#BABDEHHG) , con las siguientes excepciones:

* No puede especificar la INTOcláusula de CALL, porque se aplica solo a funciones.
* No puede especificar variables de enlace en *expr*.
* Para hacer referencia a columnas de tablas en las que se está definiendo el gatillo, debe especificar: NEWy: OLD.

**Ver también:**

["Llamar a un procedimiento en un cuerpo de disparador: ejemplo"](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7004.htm#i2064401)

**Ejemplos**

**Creación de un disparador DML: ejemplos**Este ejemplo muestra la sintaxis básica de un BEFOREactivador de declaración. Escribiría un activador de este tipo para imponer restricciones a las declaraciones DML emitidas en una tabla, por ejemplo, cuándo podrían emitirse dichas declaraciones.

CREATE TRIGGER *schema.trigger\_name*

ANTES DE

BORRAR O INSERTAR O ACTUALIZAR

ON *schema.table\_name*

*pl / sql\_block*

Oracle Database activa un activador de este tipo cada vez que una declaración DML afecta la tabla. Este activador es un BEFOREactivador de declaración, por lo que la base de datos lo activa una vez antes de ejecutar la declaración de activación.

El siguiente ejemplo muestra un BEFOREdisparador de fila parcial . El bloque PL / SQL puede especificar, por ejemplo, que el salario de un empleado debe estar dentro del rango de salario establecido para el trabajo del empleado:

CREAR TRIGGER hr.salary\_check

ANTES DE INSERTAR O ACTUALIZAR EL salario, job\_id ON hr.employees

POR CADA FILA

CUÁNDO (new.job\_id <> 'AD\_VP')

*pl / sql\_block*

Oracle Database activa este disparador siempre que se emite una de las siguientes declaraciones:

* Una INSERTdeclaración que agrega filas a la employeestabla.
* Una UPDATEdeclaración que cambia los valores de las columnas salaryo job\_idde la employeestabla.

salary\_checkes un BEFOREdisparador de fila, por lo que la base de datos lo activa antes de cambiar cada fila que actualiza la UPDATEdeclaración o antes de agregar cada fila que inserta la INSERTdeclaración.

salary\_checktiene una condición desencadenante que le impide verificar el salario del vicepresidente administrativo ( AD\_VP).

**Creación de un disparador DDL: ejemplo**Este ejemplo crea un AFTERdesencadenante de declaración en cualquier declaración DDL CREATE. Este desencadenante se puede utilizar para auditar la creación de nuevos objetos de diccionario de datos en su esquema.

CREAR TRIGGER audit\_db\_object DESPUÉS DE CREAR

EN ESQUEMA

*pl / sql\_block*

**Llamar a un procedimiento en un cuerpo de disparador: ejemplo**Puede crear el salary\_checkdisparador descrito en el ejemplo anterior llamando a un procedimiento en lugar de proporcionar el cuerpo del disparador en un bloque PL / SQL. Suponga que ha definido un procedimiento check\_salen el hresquema, que verifica que el salario de un empleado está en un rango apropiado. Entonces podría crear el disparador de la salary\_checksiguiente manera:

CREAR TRIGGER salario\_check

ANTES DE INSERTAR O ACTUALIZAR EL salario, job\_id ON empleados

POR CADA FILA

CUÁNDO (new.job\_id <> 'AD\_VP')

LLAME check\_sal (: new.job\_id,: new.salary,: new.last\_name)

El procedimiento check\_salse puede implementar en PL / SQL, C o Java. Además, puede especificar: OLDvalores en la CALLcláusula en lugar de: NEWvalores.

**Creación de un activador de eventos de base de datos: ejemplo**Este ejemplo muestra la sintaxis básica de un disparador para registrar todos los errores. El bloque hipotético PL / SQL realiza un procesamiento especial para un error particular (inicio de sesión no válido, número de error 1017). Este activador es un AFTERactivador de declaración, por lo que se activa después de una ejecución incorrecta de la declaración, como un inicio de sesión incorrecto.

CREATE TRIGGER log\_errors DESPUÉS DE SERVERERROR ON DATABASE

*BEGIN*

*IF (IS\_SERVERERROR (1017)) THEN*

*<procesamiento especial de error de inicio de sesión>*

*ELSE*

*<número de error de registro>*

*END IF;*

*FIN;*

**Creación de un disparador EN LUGAR DE: Ejemplo**En este ejemplo, oe.order\_infose crea una vista para mostrar información sobre los clientes y sus pedidos:

CREAR VER order\_info COMO

SELECCIONE c.customer\_id, c.cust\_last\_name, c.cust\_first\_name,

o.order\_id, o.order\_date, o.order\_status

DE clientes c, pedidos o

DONDE c.customer\_id = o.customer\_id;

Normalmente, esta vista no sería actualizable, porque la clave principal de la orderstabla ( order\_id) no es única en el conjunto de resultados de la vista de combinación. Para que esta vista sea actualizable, cree un INSTEAD OFdisparador en la vista para procesar INSERTdeclaraciones dirigidas a la vista. La implementación del disparador PL / SQL se muestra en cursiva.

CREAR O REEMPLAZAR TRIGGER order\_info\_insert

EN LUGAR DE INSERTAR EN order\_info

*DECLARAR*

*EXCEPCIÓN de duplicate\_info;*

*PRAGMA EXCEPTION\_INIT (duplicate\_info, -00001);*

*COMIENCE A*

*INSERTAR EN los clientes*

*(customer\_id, cust\_last\_name, cust\_first\_name)*

*VALUES (*

*: new.customer\_id*

*,:*

*new.cust\_last\_name,: new.cust\_first\_name);*

*INSERT INTO orders (order\_id, order\_date, customer\_id)*

*VALUES (*

*: new.order\_id*

*,:*

*new.order\_date,: new.customer\_id);*

*EXCEPCIÓN*

*CUANDO duplicate\_info ENTONCES*

*RAISE\_APPLICATION\_ERROR (*

*num => -20107,*

*msg => 'ID de pedido o cliente duplicado');*

*END order\_info\_insert;*

/

Ahora puede insertar en ambas tablas base a través de la vista (siempre que todas las NOT NULLcolumnas reciban valores):

INSERTAR EN LOS VALORES de order\_info

(999, 'Smith', 'John', 2500, '13 -MAR-2001 ', 0);

**Creación de un disparador SCHEMA: ejemplo**El siguiente ejemplo crea un BEFOREdesencadenante de declaración en el esquema de muestra hr. Cuando un usuario conectado como hrintenta soltar un objeto de la base de datos, la base de datos activa el disparador antes de soltar el objeto:

CREAR O REEMPLAZAR TRIGGER drop\_trigger

ANTES DE BAJAR EN hr.SCHEMA

EMPEZAR

RAISE\_APPLICATION\_ERROR (

num => -20000,

msg => 'No se puede soltar el objeto');

FIN;

/

## TRIGGERS COMPUESTOS

1. Se pueden crear triggers compuestos, que son como varios triggers definidos en uno.  
2. Se pueden crear los triggers inicialmente deshabilitados.  
3. Se puede establecer un orden de ejecución de triggers sobre tablas con mas de un trigger definido.  
  
En esta nota voy a mostrarles como crear y usar los triggers compuestos y además haré una comparativa de tiempos con respecto a los triggers tradicionales.  
  
Para comenzar me gustaría mostrarles mediante un ejemplo usando dos tablas inventadas, de forma de simular una base de libros de una biblioteca. Los libros pueden estar en 3 estados: D:"Disponible", P:"Prestado" y R:"En restauracion". Se mantiene un registro de cada cambio de estado en la tabla libros\_log que se llena mediante un trigger cada vez que se realiza un update del campo estado en la tabla libros.

rop@ROP111> create table libros (id\_libro int primary key,  
 autor varchar2(20),  
 titulo varchar2(30),  
 editorial varchar2(10),  
 estado char(1) check (estado in ('P','D','R')))  
/  
  
Tabla creada.  
Transcurrido: 00:00:00.15  
rop@ROP111>  
rop@ROP111> create table libros\_log (id\_operacion int,  
 id\_libro int references libros (id\_libro),  
 estado char(1) check (estado in ('P','D','R')),  
 fecha\_operacion date)  
/  
  
Tabla creada.  
  
Transcurrido: 00:00:00.01  
  
rop@ROP111> create sequence libros\_log\_seq;  
  
Secuencia creada.

Una vez creadas las dos tablas y la secuencia voy a proceder a cargar la tabla de libros con 1M de registros generados aleatoriamente, de forma tal de acercarme a la realidad, aunque sean datos irreales.

rop@ROP111> insert into libros  
 select rownum,  
 dbms\_random.string('a',trunc(dbms\_random.value(1,20))),  
 dbms\_random.string('l',trunc(dbms\_random.value(1,30))),  
 'EDITORIAL'trunc(dbms\_random.value(1,10)),  
 'D'  
 from dual  
 connect by rownum <= 1000000   
/   
1000000 filas creadas.   
Transcurrido: 00:02:02.97   
  
rop@ROP111> commit;  
Confirmación terminada.

Ahora voy a cambiar el estado a los libros registrados. La idea es cambiar el estado tratando de simular una distribución real. Se definirá un 40% del total de los libros como prestados (P) y el 2% en estado de restauración (R):

rop@ROP111> update libros  
 set estado = 'P'  
 where id\_libro in (select trunc(dbms\_random.value(1,200000))  
 from libros  
 where rownum <= 200000\*0.4)   
/   
66163 filas actualizadas.   
Transcurrido: 00:00:03.71   
rop@ROP111>  
rop@ROP111> update libros  
 set estado = 'R'  
 where id\_libro in (select trunc(dbms\_random.value(1,200000))  
 from libros  
 where rownum <= 200000\*0.02)   
/   
3966 filas actualizadas.   
Transcurrido: 00:00:02.45

Voy a crear un trigger simple, tal como lo creariamos en versiones 10g o inferiores:

rop@ROP111> create or replace trigger libros\_trg\_simple  
2 after update of estado on libros  
3 for each row  
4 begin  
5 insert into libros\_log values (libros\_log\_seq.nextval,  
6 :old.id\_libro,  
7 :new.estado,  
8 sysdate);  
9 end libros\_trg\_simple;  
10 /  
Disparador creado.

Ahora voy a crear el trigger compuesto introducido en 11g:

rop@ROP111> create or replace trigger libros\_trg\_compound  
2 for update of estado  
3 on libros  
4 compound trigger  
5 type libros\_log\_type is table of libros\_log%rowtype index by pls\_integer;  
6 l\_libros\_log\_array libros\_log\_type;  
7 l\_pos pls\_integer := 0;  
8 before statement is  
9 begin  
10 null;  
11 end before statement;  
12 before each row is  
13 begin  
14 null;  
15 end before each row;  
16 after each row is  
17 begin  
18 l\_pos := l\_pos+1;  
19 l\_libros\_log\_array(l\_pos).id\_operacion := libros\_log\_seq.nextval;  
20 l\_libros\_log\_array(l\_pos).id\_libro := :old.id\_libro;  
21 l\_libros\_log\_array(l\_pos).estado := :new.estado;  
22 l\_libros\_log\_array(l\_pos).fecha\_operacion := sysdate;  
23 end after each row;  
24 after statement is  
25 begin  
26 forall i in 1..l\_libros\_log\_array.count()  
27 insert into libros\_log  
28 values l\_libros\_log\_array(i);  
29 end after statement;  
30 end libros\_trg\_compound;  
31 /  
  
Disparador creado.  
  
Transcurrido: 00:00:00.06  
rop@ROP111>

Notar que se realiza un bulk insert lo cual deberia ser mas rapido que insertar por cada update. Tambien se puede ver que se obtiene el próximo valor de secuencia (linea 19) en forma directa desde PL/SQL sin necesidad de usar un select into (nuevo en 11g) .  
  
Los triggers fueron creados habilitados (comportamiento default), pero podrian haberse creado inicialmente deshabilitados aprovechando que en 11g se puede.  
Para realizar la primera prueba deshabilitamos el trigger compuesto y dejamos solo habilitado el trigger simple

rop@ROP111> alter trigger libros\_trg\_compound disable;  
Disparador modificado.

Vamos a simular que se recibieron todos los libros que estaban para restaurar (estado=R) y entonces hay que cambiarles el estado en la base para que queden nuevamente disponibles (estado=D).

rop@ROP111> update libros  
 set estado = 'D'  
 where estado = 'R'  
/  
3960 filas actualizadas.  
  
Transcurrido: **00:00:01.90**  
  
rop@ROP111> rollback;  
Rollback terminado.

Una vez tomado el tiempo, deshacemos el update con rollback y habilitamos el trigger  
compuesto y deshabilitamos el simple

rop@ROP111> alter trigger libros\_trg\_compound enable;  
Disparador modificado.  
  
Transcurrido: 00:00:00.01  
rop@ROP111> alter trigger libros\_trg\_simple disable;  
Disparador modificado.  
  
rop@ROP111> update libros  
 set estado = 'D'  
 where estado = 'R';  
3966 filas actualizadas.  
  
Transcurrido: **00:00:01.22**

Como observamos de la comparación, se nota una mejora en los tiempos al usar el trigger compuesto en relación con el trigger simple.  
  
Para notar mas la diferencia voy a modificar todos los registros de forma tal de  
setear todos los estados en disponible (estado=D). Primero pruebo con el trigger simple activo:

rop@ROP111> alter trigger libros\_trg\_simple enable;  
Disparador modificado.  
  
rop@ROP111> alter trigger libros\_trg\_compound disable;  
Disparador modificado.  
  
rop@ROP111> update libros  
 set estado = 'D';  
1000000 filas actualizadas.  
  
Transcurrido: **00:02:57.83**  
rop@ROP111> rollback;  
  
Rollback terminado.  
Transcurrido: 00:01:20.89

Por ultimo comparamos con los tiempos usando el trigger compuesto:

rop@ROP111> alter trigger libros\_trg\_simple disable;  
Disparador modificado.  
  
rop@ROP111> alter trigger libros\_trg\_compound enable;  
Disparador modificado.  
  
rop@ROP111> update libros  
 set estado = 'D';  
  
1000000 filas actualizadas.  
Transcurrido: **00:02:45.03**

Como vemos en la ultima comparación, al modificar mayor cantidad de registros se ve mas claramente la diferencia entre procesar todo al final de la sentencia en forma bulk (trigger compuesto) que con el procesamiento tradicional donde el trigger se dispara por cada registro.

# HINTS

Listado Alfabético de Sugerencias

Esta sección proporciona sintaxis y semántica para todos los consejos en orden alfabético.

### ALL\_ROWS Sugerencia

La descripción de all_rows_hint.gif sigue  
[Descripción de la ilustración '' all\_rows\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/all_rows_hint.htm)

La ALL\_ROWS sugerencia le indica al optimizador que optimice un bloque de declaración con el objetivo de obtener el mejor rendimiento, que es el consumo mínimo total de recursos. Por ejemplo, el optimizador utiliza el enfoque de optimización de consulta para optimizar esta declaración para el mejor rendimiento:

SELECT /\*+ ALL\_ROWS \*/ employee\_id, last\_name, salary, job\_id

FROM employees

WHERE employee\_id = 107;

Si especifica la sugerencia ALL\_ROWS o la FIRST\_ROWS en una declaración SQL, y si el diccionario de datos no tiene estadísticas sobre las tablas a las que accede la declaración, entonces el optimizador usa valores estadísticos predeterminados, como el almacenamiento asignado para dichas tablas, para estimar las estadísticas faltantes. y posteriormente elegir un plan de ejecución. Es posible que estas estimaciones no sean tan precisas como las que recopila el paquete DBMS\_STATS, por lo que debe usar el paquete DBMS\_STATS para recopilar estadísticas.

Si especifica sugerencias para rutas de acceso o operaciones de unión junto con la sugerencia ALL\_ROWSo FIRST\_ROWS, el optimizador da prioridad a las rutas de acceso y las operaciones de unión especificadas por las sugerencias.

### ANEXO Sugerencia

La descripción de append_hint.gif sigue  
[Descripción de la ilustración '' append\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/append_hint.htm)

La sugerencia APPEND le indica al optimizador que use la ruta directa INSERT con la sintaxis de la subconsulta de la declaración INSERT.

* Convencional INSERTes el valor predeterminado en modo serie. En el modo en serie, la ruta directa solo se puede usar si se incluye la sugerencia APPEND.
* La ruta directa INSERT es la predeterminada en modo paralelo. En modo paralelo, la inserción convencional solo se puede usar si especifica la sugerencia NOAPPEND.

La decisión de si INSERT irá paralela o no es independiente de la sugerencia APPEND.

En la ruta directa INSERT, los datos se agregan al final de la tabla, en lugar de utilizar el espacio existente asignado actualmente a la tabla. Como resultado, el camino directo INSERT puede ser considerablemente más rápido que el convencional INSERT.

La sugerencia APPEND solo se admite con la sintaxis de subconsulta de la declaración INSERT, no con la cláusula VALUES. Si especifica la sugerencia APPEND con la cláusula VALUES, se ignorará y se usará la inserción convencional. Para utilizar la ruta directa INSERT con la cláusula VALUES, consulte la ["Sugerencia APPEND\_VALUES"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABFIHGA) .

**Ver también:**

["Sugerencia NOAPPEND"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABJCJFG) para obtener información sobre esa sugerencia y [*la Guía del administrador de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e25494/tables.htm#ADMIN01509)para obtener información sobre inserciones de ruta directa

### APPEND\_VALUES Sugerencia

La descripción de append_values_hint.gif sigue  
[Descripción de la ilustración '' append\_values\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/append_values_hint.htm)

La sugerencia APPEND\_VALUES le indica al optimizador que use la ruta directa INSERT con la cláusula VALUES. Si no especifica esta sugerencia, entonces INSERT se usa convencional .

En la ruta directa INSERT, los datos se agregan al final de la tabla, en lugar de utilizar el espacio existente asignado actualmente a la tabla. Como resultado, el camino directo INSERT puede ser considerablemente más rápido que el convencional INSERT.

La sugerencia APPEND\_VALUES se puede utilizar para mejorar enormemente el rendimiento. Algunos ejemplos de sus usos son:

* En un programa de interfaz de llamada de Oracle (OCI), cuando se usan enlaces de matriz grande o enlaces de matriz con devoluciones de llamada de fila
* En PL / SQL, cuando se carga una gran cantidad de filas con un bucle FORALL que tiene una declaración INSERT con una cláusula VALUES

La sugerencia APPEND\_VALUES solo es compatible con la cláusula VALUES de la declaración INSERT. Si especifica la sugerencia APPEND\_VALUES con la sintaxis de subconsulta de la declaración INSERT, se ignorará y se usará la inserción convencional. Para usar la ruta directa INSERT con una subconsulta, consulte ["APÉNDICE Sugerencia"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABBBHCJ) .

**Ver también:**

[*Guía del administrador de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e25494/tables.htm#ADMIN01509) para obtener información sobre las inserciones de ruta directa

### CACHE Sugerencia

La descripción de cache_hint.gif sigue.  
[Descripción de la ilustración '' cache\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cache_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia CACHE indica al optimizador que coloque los bloques recuperados para la tabla en el extremo más recientemente utilizado de la lista LRU en la memoria caché del búfer cuando se realiza un escaneo completo de la tabla. Esta sugerencia es útil para tablas de búsqueda pequeñas.

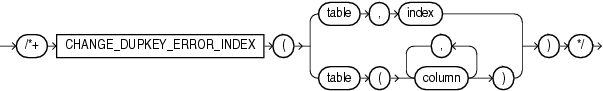
En el siguiente ejemplo, la sugerencia CACHE anula la especificación de almacenamiento en caché predeterminada de la tabla:

SELECT /\*+ FULL (hr\_emp) CACHE(hr\_emp) \*/ last\_name

FROM employees hr\_emp;

Las sugerencias CACHE y los puntos NOCACHE afectan a las estadísticas del sistema table scans (long tables)y table scans (short tables), como se muestra en la vista V$SYSSTAT del diccionario de datos.

### CHANGE\_DUPKEY\_ERROR\_INDEX Sugerencia

  
[Descripción de la ilustración '' change\_dupkey\_error\_index.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/change_dupkey_error_index.htm)

**Nota:**

Los CHANGE\_DUPKEY\_ERROR\_INDEX, IGNORE\_ROW\_ON\_DUPKEY\_INDEX y los consejos RETRY\_ON\_ROW\_CHANGE son diferentes a otros en que tienen un efecto semántico. La filosofía general explicada en ["Sugerencias"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#i35922) no se aplica a estas tres sugerencias.

La sugerencia CHANGE\_DUPKEY\_ERROR\_INDEX proporciona un mecanismo para identificar de forma inequívoca una violación de clave única para un conjunto específico de columnas o para un índice específico. Cuando se produce una violación de clave única para el índice especificado, se informa un error ORA-38911 en lugar de un ORA-001.

Esta sugerencia se aplica a operaciones INSERT, UPDATE. Si especifica un índice, entonces el índice debe existir y ser único. Si especifica una lista de columnas en lugar de un índice, entonces debe existir un índice único cuyas columnas coincidan con las columnas especificadas en número y orden.

Este uso de esta sugerencia da como resultado mensajes de error si se violan reglas específicas. Consulte [IGNORE\_ROW\_ON\_DUPKEY\_INDEX Hint](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDEGDDG) para obtener más información.

**Nota:**

Esta sugerencia deshabilita el modo APPEND y el DML paralelo.

**Ver también:**

[IGNORE\_ROW\_ON\_DUPKEY\_INDEX Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDEGDDG) para obtener información sobre esa sugerencia y [*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/hintsref.htm#PFGRF94941) para obtener más información sobre el uso de las sugerencias relacionadas con la actualización de la aplicación en línea

### Sugerencia de grupo

La descripción de cluster_hint.gif sigue.  
[Descripción de la ilustración '' cluster\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cluster_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia CLUSTER indica al optimizador que use un escaneo de clústeres para acceder a la tabla especificada. Esta sugerencia se aplica solo a las tablas en un clúster indexado.

### CURSOR\_SHARING\_EXACT Sugerencia

La descripción de cursor_sharing_exact_hint.gif sigue  
[Descripción de la ilustración '' cursor\_sharing\_exact\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/cursor_sharing_exact_hint.htm)

Oracle puede reemplazar los literales en las sentencias de SQL con variables de enlace, cuando sea seguro hacerlo. Este reemplazo se controla con el parámetro CURSOR\_SHARING de inicialización. La sugerencia CURSOR\_SHARING\_EXACT indica al optimizador que desactive este comportamiento. Cuando especifica esta sugerencia, Oracle ejecuta la instrucción SQL sin ningún intento de reemplazar los literales con variables de enlace.

### DRIVING\_SITE Sugerencia

Descripción de driving_site_hint.gif sigue  
[Descripción de la ilustración '' driving\_site\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/driving_site_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia DRIVING\_SITE le indica al optimizador que ejecute la consulta en un sitio diferente al seleccionado por la base de datos. Esta sugerencia es útil si está utilizando la optimización de consultas distribuidas.

Por ejemplo:

SELECT /\*+ DRIVING\_SITE(departments) \*/ \*

FROM employees, departments@rsite

WHERE employees.department\_id = departments.department\_id;

Si esta consulta se ejecuta sin la sugerencia, las filas de departments se envían al sitio local y la unión se ejecuta allí. Con la sugerencia, las filas de employees se envían al sitio remoto, y la consulta se ejecuta allí y el conjunto de resultados se devuelve al sitio local.

### DYNAMIC\_SAMPLING Sugerencia

La descripción de dynamic_sampling_hint.gif sigue  
[Descripción de la ilustración '' dynamic\_sampling\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dynamic_sampling_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia DYNAMIC\_SAMPLING indica al optimizador cómo controlar el muestreo dinámico para mejorar el rendimiento del servidor mediante la determinación de estadísticas de selectividad de predicción más precisas para tablas e índices.

Puede establecer el valor de DYNAMIC\_SAMPLING un valor de 0 a 10. Cuanto más alto sea el nivel, más esfuerzo hará el compilador en el muestreo dinámico y más ampliamente se aplicará. El muestreo se establece de forma predeterminada en el nivel del cursor a menos que especifique *tablespec*.

El *integer*valor es 0 a 10,lo que indica el grado de muestreo.

Si ya existe una estadística de cardinalidad para la tabla, entonces el optimizador la usa. De lo contrario, el optimizador permite el muestreo dinámico para estimar la estadística de cardinalidad.

Si especifica *tablespec* y la estadística de cardinalidad ya existe, entonces:

* Si no hay un predicado de tabla única (una cláusula WHERE que evalúa solo una tabla), el optimizador confía en las estadísticas existentes e ignora esta sugerencia. Por ejemplo, la siguiente consulta no dará como resultado ningún muestreo dinámico si employees se analiza:
* SELECT /\*+ DYNAMIC\_SAMPLING(e 1) \*/ count(\*)
* FROM employees e;
* Si hay un predicado de una sola tabla, el optimizador utiliza la estadística de cardinalidad existente y estima la selectividad del predicado utilizando las estadísticas existentes.

Para aplicar el muestreo dinámico a una tabla específica, use la siguiente forma de la sugerencia:

SELECT /\*+ DYNAMIC\_SAMPLING(employees 1) \*/ \*

FROM employees

WHERE ...

**Ver también:**

[*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/stats.htm#PFGRF30101) para obtener información sobre el muestreo dinámico y los niveles de muestreo que puede configurar

### Fact int

Descripción de fact_hint.gif sigue  
[Descripción de la ilustración '' fact\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/fact_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La FACTpista se utiliza en el contexto de la transformación de la estrella. Le indica al optimizador que la tabla especificada en *tablespec* debe considerarse como una tabla de hechos.

### FIRST\_ROWS Hint

La descripción de first_rows_hint.gif sigue  
[Descripción de la ilustración '' first\_rows\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/first_rows_hint.htm)

La sugerencia FIRST\_ROWS le indica a Oracle que optimice una instrucción SQL individual para una respuesta rápida, eligiendo el plan que devuelve las primeras *n*filas de manera más eficiente. Para *integer*, especifique el número de filas a devolver.

Por ejemplo, el optimizador utiliza el enfoque de optimización de consultas para optimizar la siguiente declaración para el mejor tiempo de respuesta:

SELECT /\*+ FIRST\_ROWS(10) \*/ employee\_id, last\_name, salary, job\_id

FROM employees

WHERE department\_id = 20;

En este ejemplo, cada departamento contiene muchos empleados. El usuario desea que los primeros 10 empleados del departamento 20 se muestren lo más rápido posible.

El optimizador ignora esta pista en bloques de instrucciones DELETE y UPDATE y en bloques de instrucciones SELECT que incluyen cualquier operación de bloqueo, tales como las clases o grupos. Dichas declaraciones no pueden optimizarse para el mejor tiempo de respuesta, ya que Oracle Database debe recuperar todas las filas a las que accede la declaración antes de devolver la primera fila. Si especifica esta sugerencia en una declaración de este tipo, la base de datos se optimiza para obtener el mejor rendimiento.

**Ver también:**

["Sugerencia ALL\_ROWS"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABHIAIG) para obtener información adicional sobre la sugerencia FIRST\_ROWS y las estadísticas

### FULL Hint

Descripción de full_hint.gif sigue  
[Descripción de la ilustración '' full\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/full_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia FULL indica al optimizador que realice un análisis completo de la tabla para la tabla especificada. Por ejemplo:

SELECT /\*+ FULL(e) \*/ employee\_id, last\_name

FROM hr.employees e

WHERE last\_name LIKE :b1;

La base de datos Oracle realiza una exploración completa de la tabla en la tabla employees para ejecutar esta declaración, incluso si hay un índice en la columna last\_name que está disponible por la condición en la cláusula WHERE.

La tabla employees tiene un alias e en la cláusula FROM, por lo que la sugerencia debe referirse a la tabla por su alias en lugar de por su nombre. No especifique nombres de esquema en la sugerencia, incluso si están especificados en la cláusula FROM.

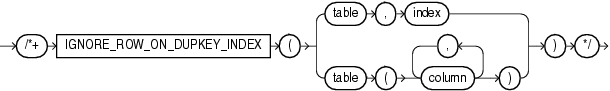
### HASH Hint

Descripción de hash_hint.gif sigue  
[Descripción de la ilustración '' hash\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/hash_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia HASH le indica al optimizador que use un escaneo hash para acceder a la tabla especificada. Esta sugerencia se aplica solo a las tablas en un clúster hash.

### IGNORE\_ROW\_ON\_DUPKEY\_INDEX Hint

  
[Descripción de la ilustración '' ignore\_row\_on\_dupkey\_index.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/ignore_row_on_dupkey_index.htm)

**Nota:**

Los CHANGE\_DUPKEY\_ERROR\_INDEX, IGNORE\_ROW\_ON\_DUPKEY\_INDEX y los consejos RETRY\_ON\_ROW\_CHANGE son diferentes a otros en que tienen un efecto semántico. La filosofía general explicada en ["Sugerencias"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#i35922) no se aplica a estas tres sugerencias.

La sugerencia IGNORE\_ROW\_ON\_DUPKEY\_INDEX se aplica solo a las operaciones INSERT de una sola tabla . No se admite para operaciones UPDATE, DELETE, MERGE, o de inserción de varias tablas. IGNORE\_ROW\_ON\_DUPKEY\_INDEX hace que la declaración ignore una violación de clave única para un conjunto específico de columnas o para un índice específico. Cuando se encuentra una violación de clave única, se produce una reversión a nivel de fila y la ejecución se reanuda con la siguiente fila de entrada. Si especifica esta sugerencia cuando inserte datos con el registro de errores DML habilitado, entonces la violación de la clave única no se registra y no causa la terminación de la declaración.

El efecto semántico de esta sugerencia da como resultado mensajes de error si se violan reglas específicas:

* Si lo especifica *index*, entonces el índice debe existir y ser único. De lo contrario, la declaración provoca ORA-38913.
* Debe especificar exactamente un índice. Si no especifica ningún índice, la declaración provoca ORA-38912. Si especifica más de un índice, entonces la declaración causa ORA-38915.
* Puede especificar una CHANGE\_DUPKEY\_ERROR\_INDEX o una IGNORE\_ROW\_ON\_DUPKEY\_INDEXsugerencia en una declaración INSERT, pero no ambas. Si especifica ambos, entonces la declaración causa ORA-38915.

Al igual que con todas las sugerencias, un error de sintaxis en la sugerencia hace que se ignore silenciosamente. El resultado será que se generará ORA-00001, como si no se usara ninguna pista.

**Nota:**

Esta sugerencia deshabilita el modo APPEND y el paralelo DML.

**Ver también:**

[CHANGE\_DUPKEY\_ERROR\_INDEX Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDIFFJE) para obtener información sobre esa sugerencia y [*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/hintsref.htm#PFGRF94941) para obtener más información sobre el uso de las sugerencias relacionadas con la actualización de la aplicación en línea

### INDEX Hint

Descripción de index_hint.gif sigue  
[Descripción de la ilustración '' index\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX le indica al optimizador que use un escaneo de índice para la tabla especificada. Puede usar la sugerencia INDEX para índices de unión basados ​​en función, dominio, árbol B, mapa de bits y mapa de bits.

El comportamiento de la pista depende de la especificación *indexspec*:

* Si la sugerencia INDEX especifica un único índice disponible, entonces la base de datos realiza una exploración en este índice. El optimizador no considera un análisis completo de la tabla o un análisis de otro índice en la tabla.
* Para una sugerencia sobre una combinación de índices múltiples, Oracle recomienda usar INDEX\_COMBINE en lugar de INDEX, porque es una sugerencia más versátil. Si la sugerencia INDEX especifica una lista de índices disponibles, el optimizador considera el costo de una exploración en cada índice de la lista y luego realiza la exploración del índice con el costo más bajo. La base de datos también puede optar por escanear múltiples índices de esta lista y combinar los resultados, si dicha ruta de acceso tiene el menor costo. La base de datos no considera un escaneo completo de la tabla o un escaneo en un índice no listado en la sugerencia.
* Si la sugerencia INDEX no especifica índices, el optimizador considera el costo de una exploración en cada índice disponible en la tabla y luego realiza la exploración del índice con el costo más bajo. La base de datos también puede elegir escanear múltiples índices y combinar los resultados, si dicha ruta de acceso tiene el costo más bajo. El optimizador no considera un escaneo completo de la tabla.

Por ejemplo:

SELECT /\*+ INDEX (employees emp\_department\_ix)\*/ employee\_id, department\_id

FROM employees

WHERE department\_id > 50;

### INDEX\_ASC Hint

La descripción de index_asc_hint.gif sigue.  
[Descripción de la ilustración '' index\_asc\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_asc_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_ASC le indica al optimizador que use un escaneo de índice para la tabla especificada. Si la declaración utiliza un escaneo de rango de índice, la base de datos Oracle escanea las entradas de índice en orden ascendente de sus valores indexados. Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) .

El comportamiento predeterminado para una exploración de rango es escanear las entradas de índice en orden ascendente de sus valores indexados, o en orden descendente para un índice descendente. Esta sugerencia no cambia el orden predeterminado del índice y, por lo tanto, no especifica nada más que la sugerencia INDEX. Sin embargo, puede usar la sugerencia INDEX\_ASC para especificar exploraciones de rango ascendente explícitamente si el comportamiento predeterminado cambia.

### INDEX\_COMBINE Hint

Descripción de index_combine_hint.gif sigue  
[Descripción de la ilustración '' index\_combine\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_combine_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_COMBINE le indica al optimizador que use una ruta de acceso de mapa de bits para la tabla. Si *indexspec* se omite en la sugerencia INDEX\_COMBINE suerencia, entonces el optimizador utiliza cualquier combinación booleana de índices que tenga el mejor costo estimado para la tabla. Si lo especifica indexspec, entonces el optimizador intenta usar alguna combinación booleana de los índices especificados. Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo:

SELECT /\*+ INDEX\_COMBINE(e emp\_manager\_ix emp\_department\_ix) \*/ \*

FROM employees e

WHERE manager\_id = 108

OR department\_id = 110;

### INDEX\_DESC Hint

La descripción de index_desc_hint.gif sigue  
[Descripción de la ilustración '' index\_desc\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_desc_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_DESC le indica al optimizador que use un escaneo de índice descendente para la tabla especificada. Si la declaración utiliza un escaneo de rango de índice y el índice es ascendente, entonces Oracle escanea las entradas de índice en orden descendente de sus valores indexados. En un índice particionado, los resultados están en orden descendente dentro de cada partición. Para un índice descendente, esta sugerencia cancela efectivamente el orden descendente, dando como resultado un escaneo de las entradas del índice en orden ascendente. Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo:

SELECT /\*+ INDEX\_DESC(e emp\_name\_ix) \*/ \*

FROM employees e;

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10103) para obtener información sobre análisis completos

### INDEX\_FFS Hint

La descripción de index_ffs_hint.gif sigue  
[Descripción de la ilustración '' index\_ffs\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_ffs_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_FFS indica al optimizador que realice una exploración rápida de índice completo en lugar de una exploración de tabla completa.

Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo:

SELECT /\*+ INDEX\_FFS(e emp\_name\_ix) \*/ first\_name

FROM employees e;

### INDEX\_JOIN Sugerencia

Descripción de index_join_hint.gif sigue  
[Descripción de la ilustración '' index\_join\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_join_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_JOIN le indica al optimizador que use una unión de índice como una ruta de acceso. Para que la sugerencia tenga un efecto positivo, debe existir un número suficientemente pequeño de índices que contengan todas las columnas necesarias para resolver la consulta.

Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo, la siguiente consulta utiliza una unión de índice para acceder a las columnas manager\_id y department\_id, ambas indexadas en la tabla employees.

SELECT /\*+ INDEX\_JOIN(e emp\_manager\_ix emp\_department\_ix) \*/ department\_id

FROM employees e

WHERE manager\_id < 110

AND department\_id < 50;

### INDEX\_SS Sugerencia

Descripción de index_ss_hint.gif sigue  
[Descripción de la ilustración '' index\_ss\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_ss_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_SS indica al optimizador que realice una exploración de omisión de índice para la tabla especificada. Si la declaración utiliza un escaneo de rango de índice, entonces Oracle escanea las entradas de índice en orden ascendente de sus valores indexados. En un índice particionado, los resultados están en orden ascendente dentro de cada partición.

Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo:

SELECT /\*+ INDEX\_SS(e emp\_name\_ix) \*/ last\_name

FROM employees e

WHERE first\_name = 'Steven';

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10105) para obtener información sobre las exploraciones de salto de índice

### INDEX\_SS\_ASC Sugerencia

La descripción de index_ss_asc_hint.gif sigue  
[Descripción de la ilustración '' index\_ss\_asc\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_ss_asc_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_SS\_ASC indica al optimizador que realice una exploración de omisión de índice para la tabla especificada. Si la declaración utiliza un escaneo de rango de índice, la base de datos Oracle escanea las entradas de índice en orden ascendente de sus valores indexados. En un índice particionado, los resultados están en orden ascendente dentro de cada partición. Cada parámetro cumple el mismo propósito que en ["Indicación de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) .

El comportamiento predeterminado para una exploración de rango es escanear las entradas de índice en orden ascendente de sus valores indexados, o en orden descendente para un índice descendente. Esta sugerencia no cambia el orden predeterminado del índice y, por lo tanto, no especifica nada más que la sugerencia INDEX\_SS. Sin embargo, puede usar la sugerencia INDEX\_SS\_ASC para especificar exploraciones de rango ascendente explícitamente si el comportamiento predeterminado cambia.

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10105) para obtener información sobre las exploraciones de salto de índice

### INDEX\_SS\_DESC Sugerencia

La descripción de index_ss_desc_hint.gif sigue  
[Descripción de la ilustración '' index\_ss\_desc\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/index_ss_desc_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia INDEX\_SS\_DESC indica al optimizador que realice una exploración de omisión de índice para la tabla especificada. Si la declaración utiliza un escaneo de rango de índice y el índice es ascendente, entonces Oracle escanea las entradas de índice en orden descendente de sus valores indexados. En un índice particionado, los resultados están en orden descendente dentro de cada partición. Para un índice descendente, esta sugerencia cancela efectivamente el orden descendente, dando como resultado un escaneo de las entradas del índice en orden ascendente.

Cada parámetro cumple el mismo propósito que en la ["Sugerencia de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) . Por ejemplo:

SELECT /\*+ INDEX\_SS\_DESC(e emp\_name\_ix) \*/ last\_name

FROM employees e

WHERE first\_name = 'Steven';

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10105) para obtener información sobre las exploraciones de salto de índice

### LEADING Hint

A continuación se encuentra la descripción de leading_hint.gif.  
[Descripción de la ilustración '' leading\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/leading_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia LEADING indica al optimizador que use el conjunto de tablas especificado como el prefijo en el plan de ejecución. Esta sugerencia es más versátil que la sugerencia ORDERED. Por ejemplo:

SELECT /\*+ LEADING(e j) \*/ \*

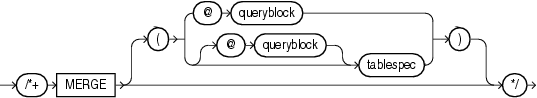
FROM employees e, departments d, job\_history j

WHERE e.department\_id = d.department\_id

AND e.hire\_date = j.start\_date;

La sugerencia LEADING se ignora si las tablas especificadas no se pueden unir primero en el orden especificado debido a las dependencias en el gráfico de unión. Si especifica dos o más sugerencia LEADING en conflicto , se ignorarán todas. Si especifica la sugerencia ORDERED, anula todas las sugerencia LEADIN.

### MERGE Sugerencia

  
[Descripción de la ilustración '' merge\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/merge_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia MERGE le permite combinar vistas en una consulta.

Si el bloque de consulta de una vista contiene una cláusula GROUP BY u operador DISTINCT en la lista SELECT, el optimizador puede fusionar la vista en la declaración de acceso solo si la combinación de vistas complejas está habilitada. La fusión compleja también se puede usar para fusionar una  IN en la declaración de acceso si la subconsulta no está correlacionada.

Por ejemplo:

SELECT /\*+ MERGE(v) \*/ e1.last\_name, e1.salary, v.avg\_salary

FROM employees e1,

(SELECT department\_id, avg(salary) avg\_salary

FROM employees e2

GROUP BY department\_id) v

WHERE e1.department\_id = v.department\_id

AND e1.salary > v.avg\_salary

ORDER BY e1.last\_name;

Cuando MERGE se utiliza la sugerencia sin un argumento, se debe colocar en el bloque de consulta de vista. Cuando MERGE se usa con el nombre de la vista como un argumento, se debe colocar en la consulta que lo rodea.

### MODEL\_MIN\_ANALYSIS Hint

La descripción de model_min_analysis_hint.gif sigue  
[Descripción de la ilustración '' model\_min\_analysis\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/model_min_analysis_hint.htm)

La sugerencia MODEL\_MIN\_ANALYSIS indica al optimizador que omita algunas optimizaciones en tiempo de compilación de las reglas de la hoja de cálculo, principalmente un análisis detallado del gráfico de dependencia. El optimizador aún utiliza otras optimizaciones de la hoja de cálculo, como la creación de filtros para rellenar de forma selectiva las estructuras de acceso a la hoja de cálculo y la eliminación limitada de reglas.

Esta sugerencia reduce el tiempo de compilación porque el análisis de la hoja de cálculo puede ser largo si el número de reglas de la hoja de cálculo es más que varios cientos.

### MONITOR Hint

Descripción de monitor_hint.gif sigue  
[Descripción de la ilustración '' monitor\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/monitor_hint.htm)

La sugerencia MONITOR obliga a la supervisión de SQL en tiempo real para la consulta, incluso si la instrucción no se ejecuta durante mucho tiempo. Esta sugerencia es válida solo cuando el parámetro CONTROL\_MANAGEMENT\_PACK\_ACCESS está establecido en DIAGNOSTIC+TUNING.

**Ver también:**

[*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/instance_tune.htm#PFGRF94543) para obtener más información sobre la supervisión de SQL en tiempo real

### NATIVE\_FULL\_OUTER\_JOIN Sugerencia

La descripción de native_foj_hint.gif sigue.  
[Descripción de la ilustración '' native\_foj\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/native_foj_hint.htm)

La sugerencia NATIVE\_FULL\_OUTER\_JOIN indica al optimizador que use una combinación externa completa nativa, que es un método de ejecución nativo basado en una combinación hash.

**Ver también:**

* [NO\_NATIVE\_FULL\_OUTER\_JOIN Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABDGDFD)
* [*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF94662) para obtener más información acerca de las combinaciones externas completas nativas

### Sugerencia NOAPPEND

Descripción de noappend_hint.gif sigue  
[Descripción de la ilustración '' noappend\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/noappend_hint.htm)

La sugerencia NOAPPEND indica al optimizador que use el modo convencional INSERT deshabilitando el modo paralelo durante la duración de la declaración INSERT. Convencional INSERT es el valor predeterminado en modo serie, y la ruta directa INSERT es el valor predeterminado en modo paralelo.

### Sugerencia NOCACHE

A continuación se encuentra la descripción de nocache_hint.gif.  
[Descripción de la ilustración '' nocache\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/nocache_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NOCACHE le indica al optimizador que coloque los bloques recuperados para la tabla en el extremo de la lista LRU que se usó más recientemente en la memoria caché del búfer cuando se realiza un escaneo completo de la tabla. Este es el comportamiento normal de los bloques en el caché del búfer. Por ejemplo:

SELECT /\*+ FULL(hr\_emp) NOCACHE(hr\_emp) \*/ last\_name

FROM employees hr\_emp;

Las sugerencias CACHE y NOCACHE afectan las estadísticas del sistema table scans(long tables)y table scans(short tables), como se muestra en la vista V$SYSSTAT.

**Ver también:**

[*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10107) para obtener información sobre el almacenamiento automático en caché de tablas, según su tamaño

### NO\_EXPAND Sugerencia

La descripción de no_expand_hint.gif sigue  
[Descripción de la ilustración '' no\_expand\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_expand_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia NO\_EXPAND le indica al optimizador que no  tenga en cuenta OR\_-expand para consultas que tengan condiciones OR o IN-listas en la cláusula WHERE. Por lo general, el optimizador considera usar la OR\_expand y usa este método si decide que el costo es menor que no usarlo. Por ejemplo:

SELECT /\*+ NO\_EXPAND \*/ \*

FROM employees e, departments d

WHERE e.manager\_id = 108

OR d.department\_id = 110;

**Ver también:**

La ["Sugerencia USE\_CONCAT"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIAFIB) , que es lo opuesto a esta sugerencia

### NO\_FACT Sugerencia

La descripción de no_fact_hint.gif sigue  
[Descripción de la ilustración '' no\_fact\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_fact_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La pista NO\_FACT se utiliza en el contexto de la transformación de la estrella. Indica al optimizador que la tabla consultada no debe considerarse como una tabla de hechos.

### Sugerencia NO\_INDEX

Descripción de no_index_hint.gif sigue  
[Descripción de la ilustración '' no\_index\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_index_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia NO\_INDEX le indica al optimizador que no use uno o más índices para la tabla especificada. Por ejemplo:

SELECT /\*+ NO\_INDEX(employees emp\_empid) \*/ employee\_id

FROM employees

WHERE employee\_id > 200;

Cada parámetro cumple el mismo propósito que en ["Sugerencia de INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEFDFC) con las siguientes modificaciones:

* Si esta sugerencia especifica un solo índice disponible, el optimizador no considera una exploración en este índice. Otros índices no especificados todavía son considerados.
* Si esta sugerencia especifica una lista de índices disponibles, el optimizador no considera una exploración en ninguno de los índices especificados. Otros índices no especificados en la lista todavía se consideran.
* Si esta sugerencia no especifica índices, el optimizador no considera una exploración en ningún índice de la tabla. Este comportamiento es el mismo que una NO\_INDEXsugerencia que especifica una lista de todos los índices disponibles para la tabla.

La sugerencia NO\_INDEX se aplica a los índices basados ​​en función, árbol B, mapa de bits, clúster o dominio. Si una pista NO\_INDEX y una sugerencia de índice ( INDEX, INDEX\_ASC, INDEX\_DESC, INDEX\_COMBINE, o INDEX\_FFS) tanto especifican los mismos índices, a continuación, la base de datos no tiene en cuenta la sugerencia NO\_INDEX y la sugerencia de índice para los índices previstos y estima los índices para su uso durante la ejecución de la instrucción.

### NO\_INDEX\_FFS Sugerencia

La descripción de no_index_ffs_hint.gif sigue  
[Descripción de la ilustración '' no\_index\_ffs\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_index_ffs_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia NO\_INDEX\_FFS indica al optimizador que excluya una exploración rápida de índice completo de los índices especificados en la tabla especificada. Cada parámetro cumple el mismo propósito que en la ["Sugerencia NO\_INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABHJBIB) . Por ejemplo:

SELECT /\*+ NO\_INDEX\_FFS(items item\_order\_ix) \*/ order\_id

FROM order\_items items;

### NO\_INDEX\_SS Sugerencia

La descripción de no_index_ss_hint.gif sigue  
[Descripción de la ilustración '' no\_index\_ss\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_index_ss_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia NO\_INDEX\_SS indica al optimizador que excluya una exploración de omisión de los índices especificados en la tabla especificada. Cada parámetro cumple el mismo propósito que en la ["Sugerencia NO\_INDEX"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABHJBIB) .

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10105) para obtener información sobre las exploraciones de salto de índice

### NO\_MERGE Sugerencia

  
[Descripción de la ilustración '' no\_merge\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_merge_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_MERGE le indica al optimizador que no combine la consulta externa y ninguna consulta de vista en línea en una sola consulta.

Esta sugerencia le permite tener más influencia sobre la forma en que se accede a la vista. Por ejemplo, la siguiente declaración hace que la vista seattle\_dept no se fusione:

SELECT /\*+ NO\_MERGE(seattle\_dept) \*/ e1.last\_name, seattle\_dept.department\_name

FROM employees e1,

(SELECT location\_id, department\_id, department\_name

FROM departments

WHERE location\_id = 1700) seattle\_dept

WHERE e1.department\_id = seattle\_dept.department\_id;

Cuando utilice la sugerencia NO\_MERGE en el bloque de consulta de vista, especifíquelo sin un argumento. Cuando especifique NO\_MERGE en la consulta que lo rodea, especifíquelo con el nombre de la vista como argumento.

### Sugerencia NO\_MONITOR

La descripción de no_monitor_hint.gif sigue  
[Descripción de la ilustración '' no\_monitor\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_monitor_hint.htm)

La sugerencia NO\_MONITOR deshabilita la supervisión de SQL en tiempo real para la consulta, incluso si la consulta es de larga ejecución.

### NO\_NATIVE\_FULL\_OUTER\_JOIN Sugerencia

La descripción de no_native_foj_hint.gif sigue  
[Descripción de la ilustración '' no\_native\_foj\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_native_foj_hint.htm)

La sugerencia NO\_NATIVE\_FULL\_OUTER\_JOIN indica al optimizador que excluya el método de ejecución nativo al unir cada tabla especificada. En su lugar, la combinación externa completa se ejecuta como una unión de combinación externa izquierda y anti-join.

**Ver también:**

[NATIVE\_FULL\_OUTER\_JOIN Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABDFDGH)

### Sugerencia NO\_PARALLEL

La descripción de no_parallel_hint.gif sigue  
[Descripción de la ilustración '' no\_parallel\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_parallel_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_PARALLEL indica al optimizador que ejecute la instrucción en serie. Esta sugerencia anula el valor del parámetro PARALLEL\_DEGREE\_POLICY de inicialización. También anula un parámetro PARALLEL en el DDL que creó o modificó la tabla. Por ejemplo, la siguiente declaración SELECT se ejecutará en serie:

ALTER TABLE employees PARALLEL 8;

SELECT /\*+ NO\_PARALLEL(hr\_emp) \*/ last\_name

FROM employees hr\_emp;

**Ver también:**

* ["Nota sobre sugerencias paralelas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDGEGFI) para obtener más información sobre las sugerencias paralelas
* [*Referencia de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams181.htm#REFRN10310) para obtener más información sobre el parámetro de inicialización PARALLEL\_DEGREE\_POLICY

### Sugerencia NOPARALLEL

La pista NOPARALLEL ha sido desaprobada. Utilice la sugerencia NO\_PARALLEL en su lugar.

### Sugerencia NO\_PARALLEL\_INDEX

La descripción de no_parallel_index_hint.gif sigue  
[Descripción de la ilustración '' no\_parallel\_index\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_parallel_index_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia NO\_PARALLEL\_INDEX anula un parámetro PARALLEL en el DDL que creó o modificó el índice, evitando así una operación de exploración de índice paralela.

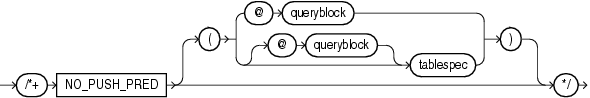
**Ver también:**

["Nota sobre sugerencias paralelas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDGEGFI) para obtener más información sobre las sugerencias paralelas

### Sugerencia NOPARALLEL\_INDEX

La pista NOPARALLEL\_INDEX ha sido desaprobada. Utilice la sugerencia NO\_PARALLEL\_INDEX en su lugar.

### NO\_PUSH\_PRED Sugerencia

  
[Descripción de la ilustración '' no\_push\_pred\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_push_pred_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_PUSH\_PRED indica al optimizador que no introduzca un predicado de unión en la vista. Por ejemplo:

SELECT /\*+ NO\_MERGE(v) NO\_PUSH\_PRED(v) \*/ \*

FROM employees e,

(SELECT manager\_id

FROM employees) v

WHERE e.manager\_id = v.manager\_id(+)

AND e.employee\_id = 100;

### Sugerencia NO\_PUSH\_SUBQ

La descripción de no_push_subq_hint.gif sigue  
[Descripción de la ilustración '' no\_push\_subq\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_push_subq_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia NO\_PUSH\_SUBQ le indica al optimizador que evalúe las subconsultas no fusionadas como el último paso en el plan de ejecución. Si lo hace, puede mejorar el rendimiento si la subconsulta es relativamente costosa o no reduce significativamente el número de filas.

### NO\_PX\_JOIN\_FILTER Sugerencia

La descripción de no_px_join_filter_hint.gif sigue  
[Descripción de la ilustración '' no\_px\_join\_filter\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_px_join_filter_hint.htm)

Esta sugerencia evita que el optimizador utilice el filtrado de mapa de bits de combinación paralela.

### NO\_QUERY\_TRANSFORMATION Sugerencia

La descripción de no_query_transformatn_hint.gif sigue  
[Descripción de la ilustración '' no\_query\_transformatn\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_query_transformatn_hint.htm)

La sugerencia NO\_QUERY\_TRANSFORMATION le indica al optimizador que omita todas las transformaciones de consulta, incluidas, entre OR otras, las siguientes: expansión, fusión de vistas, desinformación de subconsultas, transformación de estrellas y reescritura de vistas materializadas. Por ejemplo:

SELECT /\*+ NO\_QUERY\_TRANSFORMATION \*/ employee\_id, last\_name

FROM (SELECT \* FROM employees e) v

WHERE v.last\_name = 'Smith';

### NO\_RESULT\_CACHE Sugerencia

La descripción de no_result_cache_hint.gif sigue  
[Descripción de la ilustración '' no\_result\_cache\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_result_cache_hint.htm)

El optimizador almacena en caché los resultados de la consulta en el caché de resultados si el parámetro de inicialización RESULT\_CACHE\_MODE está configurado en FORCE. En este caso, la sugerencia NO\_RESULT\_CACHE deshabilita dicho almacenamiento en caché para la consulta actual.

Si la consulta se ejecuta desde el cliente OCI y la memoria caché de resultados del cliente OCI está habilitada, la sugerencia NO\_RESULT\_CACHE deshabilita el almacenamiento en caché para la consulta actual.

### Sugerencia NO\_REWRITE

Descripción de no_rewrite_hint.gif sigue  
[Descripción de la ilustración '' no\_rewrite\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_rewrite_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia NO\_REWRITE le indica al optimizador que deshabilite la reescritura de consultas para el bloque de consulta, anulando la configuración del parámetro QUERY\_REWRITE\_ENABLED. Por ejemplo:

SELECT /\*+ NO\_REWRITE \*/ sum(s.amount\_sold) AS dollars

FROM sales s, times t

WHERE s.time\_id = t.time\_id

GROUP BY t.calendar\_month\_desc;

### Sugerencia de NOREWRITE

La pista NOREWRITE ha sido desaprobada. Utilice la sugerencia NO\_REWRITE en su lugar.

### NO\_STAR\_TRANSFORMATION Sugerencia

La descripción de no_star_transformation_hint.gif sigue  
[Descripción de la ilustración '' no\_star\_transformation\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_star_transformation_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia NO\_STAR\_TRANSFORMATION indica al optimizador que no realice la transformación de consulta en estrella.

### NO\_STATEMENT\_QUEUING Sugerencia

La descripción de no_statement_queuing_hint.gif sigue  
[Descripción de la ilustración '' no\_statement\_queuing\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_statement_queuing_hint.htm)

La sugerencia NO\_STATEMENT\_QUEUING influye en si una declaración se pone en cola o no con una cola de instrucciones paralela.

Cuando PARALLEL\_DEGREE\_POLICY se establece en AUTO, esta sugerencia permite que una instrucción omita la cola de la declaración paralela. Sin embargo, una declaración que pasa por alto la cola de instrucciones puede hacer que el sistema supere el número máximo de servidores de ejecución paralelos definidos por el valor del parámetro de inicialización PARALLEL\_SERVERS\_TARGET, que determina el límite en el que se inicia la puesta en cola paralela de la declaración.

No hay garantía de que la declaración que pasa por alto la cola de instrucciones paralelas reciba la cantidad de servidores de ejecución paralelos solicitados porque solo PARALLEL\_MAX\_SERVERS se puede asignar la cantidad de servidores de ejecución paralelos disponibles en el sistema, hasta el valor del parámetro de inicialización.

Por ejemplo:

SELECT /\*+ NO\_STATEMENT\_QUEUING \*/ emp.last\_name, dpt.department\_name

FROM employees emp, departments dpt

WHERE emp.department\_id = dpt.department\_id;

**Ver también:**

["STATEMENT\_QUEUING Hint"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIBCFA)

### NO\_UNNEST Sugerencia

La descripción de no_unnest_hint.gif sigue  
[Descripción de la ilustración '' no\_unnest\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_unnest_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

El uso de la pista NO\_UNNEST se desactiva.

### NO\_USE\_HASH Sugerencia

La descripción de no_use_hash_hint.gif sigue  
[Descripción de la ilustración '' no\_use\_hash\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_use_hash_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_USE\_HASH le indica al optimizador que excluya las combinaciones hash al unir cada tabla especificada a otra fuente de fila utilizando la tabla especificada como la tabla interna. Por ejemplo:

SELECT /\*+ NO\_USE\_HASH(e d) \*/ \*

FROM employees e, departments d

WHERE e.department\_id = d.department\_id;

### NO\_USE\_MERGE Sugerencia

La descripción de no_use_merge_hint.gif sigue  
[Descripción de la ilustración '' no\_use\_merge\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_use_merge_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_USE\_MERGE indica al optimizador que excluya las combinaciones de combinación de ordenación al unir cada tabla especificada a otro origen de fila utilizando la tabla especificada como la tabla interna. Por ejemplo:

SELECT /\*+ NO\_USE\_MERGE(e d) \*/ \*

FROM employees e, departments d

WHERE e.department\_id = d.department\_id

ORDER BY d.department\_id;

### NO\_USE\_NL Sugerencia

La descripción de no_use_nl_hint.gif sigue  
[Descripción de la ilustración '' no\_use\_nl\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_use_nl_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia NO\_USE\_NL le indica al optimizador que excluya las combinaciones de bucles anidados al unir cada tabla especificada a otra fuente de fila utilizando la tabla especificada como la tabla interna. Por ejemplo:

SELECT /\*+ NO\_USE\_NL(l h) \*/ \*

FROM orders h, order\_items l

WHERE l.order\_id = h.order\_id

AND l.order\_id > 2400;

Cuando se especifica esta sugerencia, solo las combinaciones hash join y sort-merge se consideran para las tablas especificadas. Sin embargo, en algunos casos, las tablas solo se pueden unir mediante el uso de bucles anidados. En tales casos, el optimizador ignora la sugerencia de esas tablas.

### NO\_XML\_QUERY\_REWRITE Sugerencia

La descripción de no_xml_query_rewrite_hint.gif sigue  
[Descripción de la ilustración '' no\_xml\_query\_rewrite\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_xml_query_rewrite_hint.htm)

La sugerencia NO\_XML\_QUERY\_REWRITE indica al optimizador que prohíba la reescritura de expresiones XPath en sentencias de SQL. Al prohibir la reescritura de expresiones XPath, esta sugerencia también prohíbe el uso de cualquier Indice XML para la consulta actual. Por ejemplo:

SELECT /\*+NO\_XML\_QUERY\_REWRITE\*/ XMLQUERY('<A/>' RETURNING CONTENT)

FROM DUAL;

**Ver también:**

["NO\_XMLINDEX\_REWRITE Sugerencia"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABFAHGE)

### Sugerencia NO\_XMLINDEX\_REWRITE

La descripción de no_xmlindex_rewrite_hint.gif sigue  
[Descripción de la ilustración '' no\_xmlindex\_rewrite\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/no_xmlindex_rewrite_hint.htm)

La sugerencia NO\_XMLINDEX\_REWRITE le indica al optimizador que no use ningún índice XMLIndex para la consulta actual. Por ejemplo:

SELECT /\*+NO\_XMLINDEX\_REWRITE\*/ count(\*)

FROM warehouses

WHERE existsNode(warehouse\_spec, '/Warehouse/Building') = 1;

**Ver también:**

["NO\_XML\_QUERY\_REWRITE Hint"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABGDHBD) para otra forma de deshabilitar el uso de XMLIndexes

### Sugerencia OPT\_PARAM

La descripción de opt_param_hint.gif sigue.  
[Descripción de la ilustración '' opt\_param\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/opt_param_hint.htm)

La sugerencia OPT\_PARAM le permite establecer un parámetro de inicialización solo para la duración de la consulta actual. Esta sugerencia es válida sólo para los siguientes parámetros: OPTIMIZER\_DYNAMIC\_SAMPLING, OPTIMIZER\_INDEX\_CACHING, OPTIMIZER\_INDEX\_COST\_ADJ, OPTIMIZER\_SECURE\_VIEW\_MERGING, y STAR\_TRANSFORMATION\_ENABLED. Por ejemplo, la indirecta establece el parámetro STAR\_TRANSFORMATION\_ENABLED a TRUE para a la que se añade la declaración:

SELECT / \* + OPT\_PARAM ('star\_transformation\_enabled' 'true') \* / \*

DESDE ... ;

Los valores de los parámetros que son cadenas se incluyen entre comillas simples. Los valores de los parámetros numéricos se especifican sin las comillas.

### CONSEJO ORDENADO

A continuación se encuentra la descripción de order_hint.gif.  
[Descripción de la ilustración '' order\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/ordered_hint.htm)

La sugerencia ORDERED le indica a Oracle que una las tablas en el orden en que aparecen en la FROMcláusula. Oracle recomienda que utilice la sugerencia LEADING, que es más versátil que la sugerencia ORDERED.

Cuando se omite la sugerencia ORDERED de una instrucción SQL que requiere una combinación, el optimizador elige el orden en el que se unen las tablas. Es posible que desee utilizar la sugerencia ORDERED para especificar un orden de unión si sabe algo que el optimizador no conoce sobre el número de filas seleccionadas de cada tabla. Dicha información le permite elegir una tabla interna y externa mejor que el optimizador.

La siguiente consulta es un ejemplo del uso de la pista ORDERED:

SELECT /\*+ ORDERED \*/ o.order\_id, c.customer\_id, l.unit\_price \* l.quantity

FROM customers c, order\_items l, orders o

WHERE c.cust\_last\_name = 'Taylor'

AND o.customer\_id = c.customer\_id

AND o.order\_id = l.order\_id;

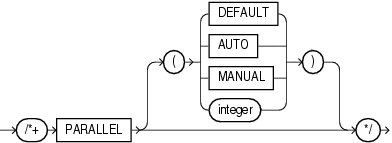
### CONSEJO PARALELO

**Nota sobre consejos paralelos**

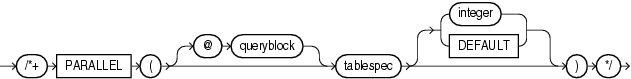
A partir de base de datos Oracle 11 *g* Release 2 (11.2.0.1), el PARALLEL y NO\_PARALLEL sustituyen los consejos de nivel de objeto anteriores:  PARALLEL\_INDEX, NO\_PARALLEL\_INDEX y especificadas previamente PARALLEL y consejos NO\_PARALLEL. Para PARALLEL, si lo especifica *integer*, ese grado de paralelismo se usará para la declaración. Si omites *integer*, entonces la base de datos calcula el grado de paralelismo. Todas las vías de acceso que pueden usar el paralelismo usarán el grado de paralelismo especificado o computado.

En los siguientes diagramas de sintaxis, se muestra la sintaxis *parallel\_hint\_statement* para las sugerencias a nivel de enunciado y *parallel\_hint\_object* la sintaxis para las sugerencias a nivel de objeto. Las sugerencias a nivel de objeto son compatibles para la compatibilidad con versiones anteriores, y son sustituidas por sugerencias a nivel de declaración.

***parallel\_hint\_statement* :: =**

  
[Descripción de la ilustración '' parallel\_hint\_statement.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/parallel_hint_statement.htm)

***parallel\_hint\_object* :: =**

  
[Descripción de la ilustración '' parallel\_hint\_object.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/parallel_hint_object.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia PARALLEL le indica al optimizador que use la cantidad especificada de servidores concurrentes para una operación paralela. Esta sugerencia anula el valor del parámetro de inicialización PARALLEL\_DEGREE\_POLICY. La sugerencia se aplica al SELECT, INSERT, MERGE, UPDATE, y porciones de un comunicado DELETE, así como a la porción de recorrido de tabla. Si se violan las restricciones paralelas, la sugerencia se ignora.

**Nota:**

La cantidad de servidores que se pueden usar es el doble del valor de la sugerencia PARALLEL, si también se realizan operaciones de clasificación o agrupación.

Para una **sugerencia PARALELA a nivel de declaración** :

* PARALLEL: La instrucción siempre se ejecuta en paralelo, y la base de datos calcula el grado de paralelismo, que puede ser 2 o mayor.
* PARALLEL( DEFAULT): El optimizador calcula un grado de paralelismo igual al número de CPU disponibles en todas las instancias participantes por el valor del parámetro de inicialización PARALLEL\_THREADS\_PER\_CPU.
* PARALLEL( AUTO): La base de datos calcula el grado de paralelismo, que puede ser 1 o mayor. Si el grado de paralelismo calculado es 1, la instrucción se ejecuta en serie.
* PARALLEL( MANUAL): El optimizador está obligado a usar la configuración paralela de los objetos en la declaración.
* PARALLEL( *integer*): El optimizador utiliza el grado de paralelismo especificado por *integer*.

En el siguiente ejemplo, el optimizador calcula el grado de paralelismo. La declaración siempre se ejecuta en paralelo.

SELECT /\*+ PARALLEL \*/ last\_name

FROM employees;

En el siguiente ejemplo, el optimizador calcula el grado de paralelismo, pero ese grado puede ser 1, en cuyo caso la instrucción se ejecutará en serie.

SELECT /\*+ PARALLEL (AUTO) \*/ last\_name

FROM employees;

En el siguiente ejemplo, la sugerencia PARALLEL recomienda al optimizador que use el grado de paralelismo actualmente vigente para la tabla, que es 5:

CREATE TABLE parallel\_table (col1 number, col2 VARCHAR2(10)) PARALLEL 5;

SELECT /\*+ PARALLEL (MANUAL) \*/ col2

FROM parallel\_table;

Para una **sugerencia PARALELA a nivel de objeto** :

* PARALLEL: El coordinador de consultas debe examinar la configuración de los parámetros de inicialización para determinar el grado predeterminado de paralelismo.
* PARALLEL( *integer*): El optimizador utiliza el grado de paralelismo especificado por *integer*.
* PARALLEL( DEFAULT): El optimizador calcula un grado de paralelismo igual al número de CPU disponibles en todas las instancias participantes por el valor del parámetro de inicialización PARALLEL\_THREADS\_PER\_CPU.

En el siguiente ejemplo, la sugerencia PARALLEL anula el grado de paralelismo especificado en la definición de la tabla employees:

SELECT /\*+ FULL(hr\_emp) PARALLEL(hr\_emp, 5) \*/ last\_name

FROM employees hr\_emp;

En el siguiente ejemplo, la sugerencia PARALLEL anula el grado de paralelismo especificado en la definición de la tabla employees e indica al optimizador que calcule un grado de paralelismo igual al número de CPU disponibles en todas las instancias participantes por el valor del parámetro de inicialización PARALLEL\_THREADS\_PER\_CPU.

SELECT /\*+ FULL(hr\_emp) PARALLEL(hr\_emp, DEFAULT) \*/ last\_name

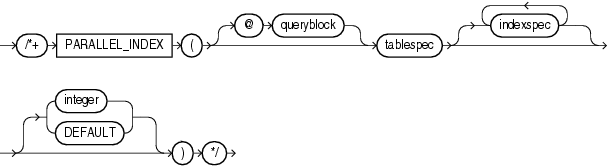
FROM employees hr\_emp;

Consulte [CREATE TABLE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_7002.htm#i2095331) y [*Oracle Database Concepts*](https://docs.oracle.com/cd/E11882_01/server.112/e40540/process.htm#CNCPT1475) para obtener más información sobre la ejecución paralela.

**Ver también:**

* [CREATE TABLE](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_7002.htm#i2095331) y [*Oracle Database Concepts*](https://docs.oracle.com/cd/E11882_01/server.112/e40540/process.htm#CNCPT1475) para obtener más información sobre la ejecución paralela.
* [*Oracle Database PL / SQL Packages and Types Reference*](https://docs.oracle.com/cd/E11882_01/appdev.112/e40758/d_parallel_ex.htm#ARPLS233) para obtener información sobre el paquete DBMS\_PARALLEL\_EXECUTE, que proporciona métodos para aplicar cambios de tabla en trozos de filas. Los cambios en cada parte se realizan de forma independiente cuando no hay errores.
* [*Referencia de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams181.htm#REFRN10310) para obtener más información sobre el parámetro de inicialización PARALLEL\_DEGREE\_POLICY
* [Sugerencia NO\_PARALLEL](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABFGFCB)

### Sugerencia PARALLEL\_INDEX

  
[Descripción de la ilustración '' parallel\_index\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/parallel_index_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)*[:: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH) )

La sugerencia PARALLEL\_INDEX le indica al optimizador que use la cantidad especificada de servidores concurrentes para paralelizar las exploraciones de rango de índice, las exploraciones completas y las exploraciones completas rápidas para los índices particionados.

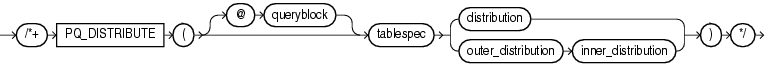
El valor *integer* indica el grado de paralelismo para el índice especificado. Especificar DEFAULT o ningún valor significa que el coordinador de consultas debe examinar la configuración de los parámetros de inicialización para determinar el grado predeterminado de paralelismo. Por ejemplo, la siguiente sugerencia indica que se deben usar tres procesos de ejecución paralelos:

SELECT /\*+ PARALLEL\_INDEX(table1, index1, 3) \*/

**Ver también:**

["Nota sobre sugerencias paralelas"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDGEGFI) para obtener más información sobre las sugerencias paralelas

### PQ\_DISTRIBUTE Sugerencia

  
[Descripción de la ilustración '' pq\_distribute\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/pq_distribute_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia PQ\_DISTRIBUTE indica al optimizador cómo distribuir filas entre los servidores de consulta del productor y del consumidor. Puede controlar la distribución de filas para uniones o para carga.

**Control de distribución para carga**Puede controlar la distribución de filas para las declaraciones paralelas INSERT... SELECT y paralelas CREATE TABLE... AS SELECT para indicar cómo deben distribuirse las filas entre el productor (consulta) y los servidores de consumo (carga). Utilice la rama superior de la sintaxis especificando un solo método de distribución. Los valores de los métodos de distribución y su semántica se describen en la [Tabla 3-22](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDCAGHI) .

***Tabla 3-22 Valores de distribución para carga***

| **Distribución** | **Descripción** |
| --- | --- |
| NONE | No hay distribución. Esa es la consulta y la operación de carga se combinan en cada  servidor de consultas. Todos los servidores cargarán todas las particiones. Esta falta  de distribución es útil para evitar la sobrecarga de distribuir filas donde no hay sesgo.  La inclinación puede ocurrir debido a segmentos vacíos o a un predicado en la declaración  que filtra todas las filas evaluadas por la consulta. Si se produce sesgo debido al uso de  este método, entonces use uno RANDOMo RANDOM\_LOCAL distribución en su lugar.  **Nota** : Utilice esta distribución con cuidado. Cada partición cargada requiere un mínimo  de 512 KB por proceso de memoria PGA. Si también utiliza compresión, entonces  aproximadamente 1.5 MB de memoria PGA es consumidor por servidor. |
| PARTITION | Este método utiliza la información de partición *tablespec*para distribuir las filas de los  servidores de consulta a los servidores de carga. Utilice este método de distribución  cuando no sea posible o deseable combinar las operaciones de consulta y carga, cuando  el número de particiones que se carguen sea mayor o igual al número de servidores de  carga, y los datos de entrada se distribuirán de manera uniforme entre las particiones  que se están distribuyendo. cargado, es decir, no hay sesgo. |
| RANDOM | Este método distribuye las filas de los productores de forma rotativa a los consumidores.  Utilice este método de distribución cuando los datos de entrada estén muy sesgados. |
| RANDOM\_LOCAL | Este método distribuye las filas de los productores a un conjunto de servidores que  son responsables de mantener un conjunto dado de particiones. Dos o más servidores  pueden cargar la misma partición, pero ningún servidor está cargando todas las particiones.  Utilice este método de distribución cuando los datos de entrada están sesgados y la  combinación de las operaciones de consulta y carga no es posible debido a las limitaciones  de memoria. |

Por ejemplo, en la siguiente operación de inserción de carga directa, las partes de consulta y carga de la operación se combinan en cada servidor de consulta:

INSERT /\*+ APPEND PARALLEL(target\_table, 16) PQ\_DISTRIBUTE(target\_table, NONE) \*/

INTO target\_table

SELECT \* FROM source\_table;

En el siguiente ejemplo de creación de tabla, el optimizador usa la partición de target\_table para distribuir las filas:

CREATE /\*+ PQ\_DISTRIBUTE(target\_table, PARTITION) \*/ TABLE target\_table

NOLOGGING PARALLEL 16

PARTITION BY HASH (l\_orderkey) PARTITIONS 512

AS SELECT \* FROM source\_table;

**Control de distribución para uniones**Usted controla el método de distribución para uniones al especificar dos métodos de distribución, como se muestra en la rama inferior del diagrama de sintaxis, una distribución para la tabla externa y una distribución para la tabla interna.

* *outer\_distribution* Es la distribución de la tabla exterior.
* *inner\_distribution* Es la distribución de la tabla interna.

Los valores de las distribuciones son HASH, BROADCAST, PARTITION, y NONE. Solo son válidas seis distribuciones de tablas de combinaciones, como se describe en la [Tabla 3-23](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABCCEJI) :

***Tabla 3-23 Valores de distribución para uniones***

| **Distribución** | **Descripción** |
| --- | --- |
| HASH, HASH | Las filas de cada tabla se asignan a los servidores de consulta del consumidor,  utilizando una función hash en las claves de combinación. Cuando se completa  la asignación, cada servidor de consultas realiza la unión entre un par de particiones  resultantes. Esta distribución se recomienda cuando las tablas son de tamaño  comparable y la operación de unión se implementa mediante la combinación hash  o la combinación de combinación de clasificación. |
| BROADCAST, NONE | Todas las filas de la tabla externa se transmiten a cada servidor de consultas.  Las filas de la tabla interna se dividen aleatoriamente. Esta distribución se  recomienda cuando la tabla exterior es muy pequeña en comparación con la tabla  interna. Como regla general, use esta distribución cuando el tamaño de la tabla  interna multiplicado por el número de servidores de consulta sea mayor que el  tamaño de la tabla externa. |
| NONE, BROADCAST | Todas las filas de la tabla interna se transmiten a cada servidor de consultas del  consumidor. Las filas de la tabla externa se dividen aleatoriamente.  Esta distribución se recomienda cuando la tabla interna es muy pequeña en  comparación con la tabla externa. Como regla general, use esta distribución cuando  el tamaño de la tabla interna multiplicado por el número de servidores de consulta  sea menor que el tamaño de la tabla externa. |
| PARTITION, NONE | Las filas de la tabla externa se asignan utilizando la partición de la tabla interna.  La tabla interna debe estar particionada en las teclas de unión. Esta distribución  se recomienda cuando el número de particiones de la tabla externa es igual o casi  igual a un múltiplo del número de servidores de consultas; por ejemplo, 14  particiones y 15 servidores de consulta.  **Nota** : El optimizador ignora esta sugerencia si la tabla interna no está particionada  o no está configurada en la clave de partición. |
| NONE, PARTITION | Las filas de la tabla interna se asignan utilizando la partición de la tabla externa.  La tabla externa debe estar particionada en las claves de unión. Esta distribución  se recomienda cuando el número de particiones de la tabla externa es igual o casi  igual a un múltiplo del número de servidores de consultas; por ejemplo, 14  particiones y 15 servidores de consulta.  **Nota** : El optimizador ignora esta sugerencia si la tabla externa no está particionada  o no está configurada en la clave de partición. |
| NONE, NONE | Cada servidor de consultas realiza la operación de unión entre un par de particiones  coincidentes, una de cada tabla. Ambas tablas deben estar provistas de particiones  en las teclas de unión. |

Por ejemplo, dadas dos tablas r y s que se unen mediante una combinación hash, la siguiente consulta contiene una sugerencia para usar la distribución de hash:

SELECT /\*+ORDERED PQ\_DISTRIBUTE(s HASH, HASH) USE\_HASH (s)\*/ *column\_list*

FROM r,s

WHERE r.c=s.c;

Para difundir la tabla exterior r, la consulta es:

SELECT /\*+ORDERED PQ\_DISTRIBUTE(s BROADCAST, NONE) USE\_HASH (s) \*/ *column\_list*

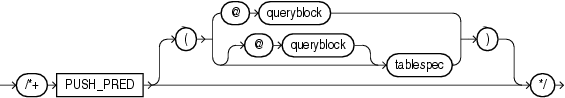
FROM r,s

WHERE r.c=s.c;

**Ver también:**

[*Guía de ajuste de rendimiento de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/hintsref.htm#PFGRF94942) para obtener más información sobre cómo Oracle paraliza las operaciones de unión

### PUSH\_PRED Sugerencia

  
[Descripción de la ilustración '' push\_pred\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/push_pred_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia PUSH\_PRED indica al optimizador que introduzca un predicado de unión en la vista. Por ejemplo:

SELECT /\*+ NO\_MERGE(v) PUSH\_PRED(v) \*/ \*

FROM employees e,

(SELECT manager\_id

FROM employees) v

WHERE e.manager\_id = v.manager\_id(+)

AND e.employee\_id = 100;

### PUSH\_SUBQ Sugerencia

La descripción de push_subq_hint.gif sigue  
[Descripción de la ilustración '' push\_subq\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/push_subq_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia PUSH\_SUBQ le indica al optimizador que evalúe las subconsultas no fusionadas en el primer paso posible del plan de ejecución. Generalmente, las subconsultas que no están fusionadas se ejecutan como el último paso en el plan de ejecución. Si la subconsulta es relativamente barata y reduce el número de filas significativamente, entonces evaluar la subconsulta antes puede mejorar el rendimiento.

Esta sugerencia no tiene ningún efecto si la subconsulta se aplica a una tabla remota o una que se une mediante una combinación de combinación.

### PX\_JOIN\_FILTER Sugerencia

La descripción de px_join_filter_hint.gif sigue  
[Descripción de la ilustración '' px\_join\_filter\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/px_join_filter_hint.htm)

Esta sugerencia obliga al optimizador a utilizar el filtrado de mapa de bits de combinación paralela.

### QB\_NAME Sugerencia

La descripción de qb_name_hint.gif sigue  
[Descripción de la ilustración '' qb\_name\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/qb_name_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

Use la sugerencia QB\_NAME para definir un nombre para un bloque de consulta. Este nombre se puede usar en una sugerencia en la consulta externa o incluso en una vista en línea para afectar la ejecución de la consulta en las tablas que aparecen en el bloque de consulta nombrado.

Si dos o más bloques de consulta tienen el mismo nombre, o si el mismo bloque de consulta se insinúa dos veces con nombres diferentes, el optimizador ignora todos los nombres y las sugerencias que hacen referencia a ese bloque de consulta. Los bloques de consulta que no tienen nombre usando esta sugerencia tienen nombres únicos generados por el sistema. Estos nombres pueden mostrarse en la tabla del plan y también pueden usarse en sugerencias dentro del bloque de consulta, o en sugerencias de bloque de consulta. Por ejemplo:

SELECT /\*+ QB\_NAME(qb) FULL(@qb e) \*/ employee\_id, last\_name

FROM employees e

WHERE last\_name = 'Smith';

### RESULTADO\_CACHE Sugerencia

La descripción de result_cache_hint.gif sigue  
[Descripción de la ilustración '' result\_cache\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/result_cache_hint.htm)

La sugerencia RESULT\_CACHE indica a la base de datos que almacene en caché los resultados de la consulta actual o el fragmento de consulta en la memoria y luego utilice los resultados almacenados en caché en futuras ejecuciones de la consulta o del fragmento de consulta. La sugerencia se reconoce en la consulta de nivel superior, la cláusula *subquery\_factoring\_clause* o la vista en línea FROM. Los resultados almacenados en caché residen en la porción de memoria caché de resultados del grupo compartido.

Un resultado en caché se invalida automáticamente cada vez que un objeto de base de datos utilizado en su creación se modifica con éxito. Esta sugerencia tiene prioridad sobre la configuración del parámetro de inicialización RESULT\_CACHE\_MODE.

La consulta es elegible para el almacenamiento en caché de resultados solo si todas las funciones implicadas en la consulta, por ejemplo, las funciones incorporadas o definidas por el usuario o las columnas virtuales, son deterministas.

Si la consulta se ejecuta desde el cliente OCI y el caché de resultados del cliente OCI está habilitado, la sugerencia RESULT\_CACHE habilita el almacenamiento en caché del cliente para la consulta actual.

**Ver también:**

[*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/memory.htm#PFGRF987) para obtener información sobre el uso de esta sugerencia,*Oracle Database Reference*para obtener información sobre el[RESULT\_CACHE\_MODE](https://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams221.htm#REFRN10270)parámetro de inicialización, y[*Oracle Call Interface Programmer's Guide*](https://docs.oracle.com/cd/E11882_01/appdev.112/e10646/oci10new.htm#LNOCI10103) para obtener más información sobre el caché de resultados OCI y las pautas de uso

### RETRY\_ON\_ROW\_CHANGE Sugerencia

La descripción de retry_on_row_change.gif sigue  
[Descripción de la ilustración '' retry\_on\_row\_change.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/retry_on_row_change.htm)

**Nota:**

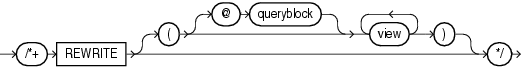
Los CHANGE\_DUPKEY\_ERROR\_INDEX, IGNORE\_ROW\_ON\_DUPKEY\_INDEXy los consejos RETRY\_ON\_ROW\_CHANGE son diferentes a otros en que tienen un efecto semántico. La filosofía general explicada en ["Sugerencias"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#i35922) no se aplica a estas tres sugerencias.

Esta sugerencia es válida solo para operaciones UPDATEy DELETE. No es compatible INSERT ni para operaciones MERGE. Cuando especifica esta sugerencia, la operación se reintenta cuando la ORA\_ROWSCN de una o más filas en el conjunto ha cambiado desde el momento en que se determina el conjunto de filas a modificar hasta el momento en que el bloque se modifica realmente.

**Ver también:**

[IGNORE\_ROW\_ON\_DUPKEY\_INDEX Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDEGDDG) y [CHANGE\_DUPKEY\_ERROR\_INDEX Sugerencia](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#CHDIFFJE) para obtener información sobre esas sugerencias y [*Oracle Database Performance Tuning Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/hintsref.htm#PFGRF94941) para obtener más información sobre el uso de las sugerencias relacionadas con la actualización de la aplicación en línea

### Sugerencia de reescritura

  
[Descripción de la ilustración '' rewrite\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/rewrite_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia REWRITE le indica al optimizador que vuelva a escribir una consulta en términos de vistas materializadas, cuando sea posible, sin tener en cuenta el costo. Utilice la REWRITEpista con o sin una lista de vista. Si utiliza REWRITE con una lista de vistas y la lista contiene una vista materializada elegible, entonces Oracle utiliza esa vista independientemente de su costo.

Oracle no considera las vistas fuera de la lista. Si no especifica una lista de vistas, entonces Oracle busca una vista materializada elegible y siempre la usa, independientemente del costo del plan final.

**Ver también:**

* [*Oracle Database Concepts*](https://docs.oracle.com/cd/E11882_01/server.112/e40540/schemaob.htm#CNCPT411) y[*Oracle Database Advanced Replication*](https://docs.oracle.com/cd/E11882_01/server.112/e10706/repplan.htm#REPLN101) para obtener más información sobre vistas materializadas
* [*Oracle Database Data Warehousing Guide*](https://docs.oracle.com/cd/E11882_01/server.112/e25554/qrbasic.htm#DWHSG018) para obtener más información sobre el usoREWRITEcon vistas materializadas

### STAR\_TRANSFORMATION Sugerencia

La descripción de star_transformation_hint.gif sigue.  
[Descripción de la ilustración '' star\_transformation\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/star_transformation_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia STAR\_TRANSFORMATION le indica al optimizador que use el mejor plan en el que se ha usado la transformación. Sin la sugerencia, el optimizador podría tomar una decisión de optimización de la consulta para utilizar el mejor plan generado sin la transformación, en lugar del mejor plan para la consulta transformada. Por ejemplo:

SELECT /\*+ STAR\_TRANSFORMATION \*/ s.time\_id, s.prod\_id, s.channel\_id

FROM sales s, times t, products p, channels c

WHERE s.time\_id = t.time\_id

AND s.prod\_id = p.prod\_id

AND s.channel\_id = c.channel\_id

AND c.channel\_desc = 'Tele Sales';

Incluso si se especifica la sugerencia, no hay garantía de que la transformación tendrá lugar. El optimizador genera las subconsultas solo si parece razonable hacerlo. Si no se generan subconsultas, entonces no hay una consulta transformada, y se utiliza el mejor plan para la consulta no transformada, independientemente de la sugerencia.

**Ver también:**

* [*Guía de almacenamiento de datos de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e25554/schemas.htm#DWHSG019) para una discusión completa de la transformación estrella.
* [*Referencia de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams253.htm#REFRN10213) para obtener más información sobre el parámetro de inicialización STAR\_TRANSFORMATION\_ENABLED.

### STATEMENT\_QUEUING Sugerencia

La descripción de statement_queuing_hint.gif sigue  
[Descripción de la ilustración '' statement\_queuing\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/statement_queuing_hint.htm)

La sugerencia NO\_STATEMENT\_QUEUING influye en si una declaración se pone en cola o no con una cola de instrucciones paralela.

Cuando PARALLEL\_DEGREE\_POLICY no se establece en AUTO, esta sugerencia permite que una declaración se considere para la cola de instrucciones paralelas, pero que se ejecute solo cuando haya suficientes procesos paralelos disponibles para ejecutarse en el DOP solicitado. El número de servidores de ejecución paralelos disponibles, antes de habilitar la cola, es igual a la diferencia entre el número de servidores de ejecución paralelos en uso y el número máximo permitido en el sistema, que está definido por el parámetro de inicialización PARALLEL\_SERVERS\_TARGET.

Por ejemplo:

SELECT /\*+ STATEMENT\_QUEUING \*/ emp.last\_name, dpt.department\_name

FROM employees emp, departments dpt

WHERE emp.department\_id = dpt.department\_id;

**Ver también:**

["NO\_STATEMENT\_QUEUING Hint"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABCJCEC)

### Sugerencia UNNEST

Descripción de unnest_hint.gif sigue  
[Descripción de la ilustración '' unnest\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/unnest_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia UNNEST indica al optimizador que anule y fusione el cuerpo de la subconsulta en el cuerpo del bloque de consulta que lo contiene, lo que permite al optimizador considerarlos juntos al evaluar las rutas de acceso y las combinaciones.

Antes de desestimar una subconsulta, el optimizador primero verifica si la declaración es válida. La declaración debe pasar las pruebas de optimización heurística y de consulta. La sugerencia UNNEST indica al optimizador que compruebe la validez del bloque de subconsulta. Si el bloque de subconsultas es válido, entonces se habilita la anulación de subconsultas sin verificar las heurísticas o los costos.

**Ver también:**

* ["Collection Unnesting: Example"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_10002.htm#i2071637) para obtener más información sobre las subconsultas anidadas y las condiciones que hacen que un bloque de subconsulta sea válido.
* [*Guía de ajuste de rendimiento de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/optimops.htm#PFGRF10104) para obtener información adicional sobre la falta de consulta de subconsultas

### Sugerencia USE\_CONCAT

La descripción de use_concat_hint.gif sigue  
[Descripción de la ilustración '' use\_concat\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/use_concat_hint.htm)

(Consulte ["Especificar un bloque de consulta en una pista"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) )

La sugerencia USE\_CONCAT le indica al optimizador que transforme las condiciones combinadas  OR en la cláusula WHERE de una consulta en una consulta compuesta utilizando el UNION ALLoperador de conjunto. Sin esta sugerencia, esta transformación ocurre solo si el costo de la consulta que usa las concatenaciones es más barato que el costo sin ellas. La sugerencia USE\_CONCAT anula la consideración de costo. Por ejemplo:

SELECT /\*+ USE\_CONCAT \*/ \*

FROM employees e

WHERE manager\_id = 108

OR department\_id = 110;

**Ver también:**

La ["Sugerencia NO\_EXPAND"](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABHBECB) , que es lo opuesto a esta sugerencia

### USE\_HASH Sugerencia

La descripción de use_hash_hint.gif sigue  
[Descripción de la ilustración '' use\_hash\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/use_hash_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia USE\_HASH indica al optimizador que combine cada tabla especificada con otra fuente de fila mediante una combinación hash. Por ejemplo:

SELECT /\*+ USE\_HASH(l h) \*/ \*

FROM orders h, order\_items l

WHERE l.order\_id = h.order\_id

AND l.order\_id > 2400;

### USE\_MERGE Sugerencia

La descripción de use_merge_hint.gif sigue  
[Descripción de la ilustración '' use\_merge\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/use_merge_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia USE\_MERGE le indica al optimizador que combine cada tabla especificada con otra fuente de fila usando una combinación de ordenación y combinación. Por ejemplo:

SELECT /\*+ USE\_MERGE(employees departments) \*/ \*

FROM employees, departments

WHERE employees.department\_id = departments.department\_id;

El uso del USE\_NL e USE\_MERGE indirectas se recomienda con el LEADING e ORDERED indirectas. El optimizador usa esas sugerencias cuando la tabla a la que se hace referencia se ve obligada a ser la tabla interna de una unión. Las sugerencias se ignoran si la tabla a la que se hace referencia es la tabla externa.

### Sugerencia USE\_NL

La sugerencia USE\_NL indica al optimizador que combine cada tabla especificada con otra fuente de fila con una combinación de bucles anidados, utilizando la tabla especificada como la tabla interna.

La descripción de use_nl_hint.gif sigue  
[Descripción de la ilustración '' use\_nl\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/use_nl_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) )

La sugerencia USE\_NL indica al optimizador que combine cada tabla especificada con otra fuente de fila con una combinación de bucles anidados, utilizando la tabla especificada como la tabla interna.

El uso del USE\_NL e USE\_MERGE indirectas se recomienda con el LEADINGe ORDEREDindirectas. El optimizador usa esas sugerencias cuando la tabla a la que se hace referencia se ve obligada a ser la tabla interna de una unión. Las sugerencias se ignoran si la tabla a la que se hace referencia es la tabla externa.

En el siguiente ejemplo, cuando un bucle anidado se fuerza a través de una sugerencia, orders se accede a través de un escaneo completo de la tabla y la condición del filtro l.order\_id = h.order\_id se aplica a cada fila. Para cada fila que cumple con la condición de filtro, order\_items se accede a través del índice order\_id.

SELECT /\*+ USE\_NL(l h) \*/ h.customer\_id, l.unit\_price \* l.quantity

FROM orders h, order\_items l

WHERE l.order\_id = h.order\_id;

Agregar una sugerencia INDEX a la consulta podría evitar el escaneo completo de la tabla orders, lo que resultaría en un plan de ejecución similar al que se usa en sistemas más grandes, aunque podría no ser particularmente eficiente aquí.

### USE\_NL\_WITH\_INDEX Sugerencia

La descripción de use_nl_with_index_hint.gif sigue  
[Descripción de la ilustración '' use\_nl\_with\_index\_hint.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/use_nl_with_index_hint.htm)

(Consulte ["Especificación de un bloque de consulta en una](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA)[*sugerencia*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB)["](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABEBCAA) , *[tablespec](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABIEJEB)*[*:: =*](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#BABIEJEB) , *[indexspec :: =](https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm" \l "BABGFHCH)* )

La sugerencia USE\_NL\_WITH\_INDEX indica al optimizador que combine la tabla especificada con otra fuente de fila con una combinación de bucles anidados utilizando la tabla especificada como la tabla interna. Por ejemplo:

SELECT /\*+ USE\_NL\_WITH\_INDEX(l item\_product\_ix) \*/ \*

FROM orders h, order\_items l

WHERE l.order\_id = h.order\_id

AND l.order\_id > 2400;

Se aplican las siguientes condiciones:

* Si no se especifica ningún índice, entonces el optimizador debe poder usar algún índice con al menos un predicado de unión como clave de índice.
* Si se especifica un índice, entonces el optimizador debe poder usar ese índice con al menos un predicado de unión como clave de índice.

# REFERENCIAS A OBJETOS EN OTROS ESQUEMAS

## Objetos no escamosos

Otros tipos de objetos también se almacenan en la base de datos y se pueden crear y manipular con SQL, pero no están contenidos en un esquema:

Contextos   
Directorios   
Ediciones   
Puntos de restauración   
Roles   
Rollback segmentos   
Tablespaces   
Usuarios

En esta referencia, cada tipo de objeto se describe en el [Capítulo 10](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_1.htm#g2257928) al [Capítulo 19](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_10.htm#g2232147) , en la sección dedicada a la declaración que crea el objeto de la base de datos. Estas declaraciones comienzan con la palabra clave CREATE. Por ejemplo, para la definición de un cluster, vea [CREATE CLUSTER](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_5001.htm#BABDBDEE) .

Debe proporcionar nombres para la mayoría de los tipos de objetos de base de datos cuando los cree. Estos nombres deben seguir las reglas enumeradas en las secciones que siguen.

## Referencias a objetos en otros esquemas

Para referirse a objetos en esquemas diferentes al suyo, prefije el nombre del objeto con el nombre del esquema:

schema.object

Por ejemplo, esta declaración elimina la tabla employees en el esquema de muestra hr:

DROP TABLE hr.employees;

## Referencias a objetos en bases de datos remotas

Para referirse a objetos en bases de datos que no sean su base de datos local, siga el nombre del objeto con el nombre del enlace de la base de datos a esa base de datos. Un enlace de base de datos es un objeto de esquema que hace que Oracle se conecte a una base de datos remota para acceder a un objeto allí. Esta sección te dice:

* + Cómo crear enlaces de base de datos
  + Cómo usar los enlaces de la base de datos en tus sentencias SQL

### Creación de enlaces de base de datos

Crea un enlace de base de datos con la declaración [CREATE DATABASE LINK](https://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_5005.htm#i2061505) . La declaración le permite especificar esta información sobre el enlace de la base de datos:

* + El nombre del enlace de la base de datos.
  + La cadena de conexión de la base de datos para acceder a la base de datos remota.
  + El nombre de usuario y la contraseña para conectarse a la base de datos remota.

Oracle almacena esta información en el diccionario de datos.

**Nombres de enlace de base de datos**

Cuando creas un enlace de base de datos, debes especificar su nombre. Los nombres de los enlaces de la base de datos son diferentes de los nombres de otros tipos de objetos. Pueden tener hasta 128 bytes y pueden contener puntos (.) Y el signo "en" (@).

El nombre que le dé a un enlace de base de datos debe corresponder al nombre de la base de datos a la que se refiere el enlace de la base de datos y la ubicación de esa base de datos en la jerarquía de nombres de bases de datos. El siguiente diagrama de sintaxis muestra la forma del nombre de un enlace de base de datos:

***dblink* :: =**

La descripción de dblink.gif sigue  
[Descripción de la ilustración '' dblink.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/dblink.htm)

dónde:

* *Database* debe especificar la parte del nombre global *name* de la base de datos remota a la que se conecta el enlace de la base de datos. Este nombre global se almacena en el diccionario de datos de la base de datos remota. Puedes ver este nombre en la GLOBAL\_NAME vista del diccionario de datos.
* *Domain* debe especificar la parte del nombre global *domain* de la base de datos remota a la que se conecta el enlace de la base de datos. Si omite el nombre *domain* de un enlace de base de datos, entonces Oracle califica el nombre del enlace de base de datos con el dominio de su base de datos local tal como existe actualmente en el diccionario de datos.
* *connection\_qualifier* le permite calificar un enlace de base de datos. Utilizandocalificadores de conexión, puede crear múltiples enlaces de base de datos a la misma base de datos. Por ejemplo, puede usar calificadores de conexión para crear múltiples enlaces de base de datos a diferentes instancias de Oracle Real Application Clusters que acceden a la misma base de datos.

La combinación *database.domain* a veces se llama el **nombre del servicio** .

#### Nombre de usuario y contraseña

Oracle utiliza el nombre de usuario y la contraseña para conectarse a la base de datos remota. El nombre de usuario y la contraseña para un enlace de base de datos son opcionales.

#### Base de datos de conexión de cadena

La cadena de conexión de la base de datos es la especificación utilizada por Oracle Net para acceder a la base de datos remota. Para obtener información sobre cómo escribir cadenas de conexión de base de datos, consulte la documentación de Oracle Net para su protocolo de red específico. La cadena de conexión de base de datos para un enlace de base de datos es opcional.

### Referencias a enlaces de base de datos

Los enlaces de la base de datos están disponibles solo si está utilizando la funcionalidad distribuida de Oracle. Cuando emite una declaración SQL que contiene un enlace de base de datos, puede especificar el nombre del enlace de la base de datos en una de estas formas:

* El nombre completo enlace de la base de datos almacenada en el diccionario de datos, incluyendo los *database*, *domain* y componentes opcionales *connection\_qualifier*.
* El nombre *partial* del enlace de la base de datos es el *database* y los componentes opcionales *connection\_qualifier*, pero no el componente *domain*.

Oracle realiza estas tareas antes de conectarse a la base de datos remota:

1. Si el nombre del enlace de la base de datos especificado en la declaración es parcial, entonces Oracle expande el nombre para que contenga el dominio de la base de datos local tal como se encuentra en el nombre de la base de datos global almacenado en el diccionario de datos. (Puede ver el nombre de la base de datos global actual en la vista GLOBAL\_NAME del diccionario de datos).
2. Oracle primero busca un enlace de base de datos privada en su propio esquema con el mismo nombre que el enlace de la base de datos en la declaración. Luego, si es necesario, busca un enlace de base de datos pública con el mismo nombre.
   * Oracle siempre determina el nombre de usuario y la contraseña desde el primer enlace de la base de datos coincidente (ya sea privado o público). Si el primer enlace de la base de datos coincidente tiene un nombre de usuario y una contraseña asociados, entonces Oracle lo utiliza. Si no tiene un nombre de usuario y contraseña asociados, entonces Oracle utiliza su nombre de usuario y contraseña actuales.
   * Si el primer enlace de base de datos coincidente tiene una cadena de base de datos asociada, entonces Oracle lo utiliza. De lo contrario, Oracle busca el siguiente enlace de la base de datos (pública). Si no se encuentra un enlace de base de datos coincidente, o si ningún enlace coincidente tiene una cadena de base de datos asociada, Oracle devuelve un error.
3. Oracle utiliza la cadena de base de datos para acceder a la base de datos remota. Después de acceder a la base de datos remota, si el valor del GLOBAL\_NAMES parámetro es true, Oracle verifica que la *database.domain* parte del nombre del enlace de la base de datos coincida con el nombre global completo de la base de datos remota. Si esta condición es verdadera, entonces Oracle continúa con la conexión, utilizando el nombre de usuario y la contraseña elegidos en el Paso 2. Si no, Oracle devuelve un error.
4. Si la conexión que utiliza la cadena de la base de datos, el nombre de usuario y la contraseña es exitosa, entonces Oracle intentará acceder al objeto especificado en la base de datos remota utilizando las reglas para resolver las referencias de los objetos y hacer referencia a los objetos en otros esquemas descritos anteriormente en esta sección.

Puede deshabilitar el requisito de que la parte del nombre *database.domain* del enlace de la base de datos debe coincidir con el nombre global completo de la base de datos remota configurando FALSE el parámetro de inicialización GLOBAL\_NAMES o el GLOBAL\_NAMES parámetro de la declaración ALTER SYSTEM o ALTER SESSION.

## Referencias a Tablas e Índices Particionados

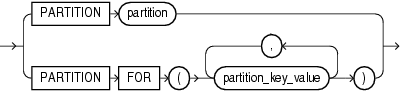
Tablas e índices pueden ser particionados. Cuando se particionan, estos objetos de esquema constan de una serie de partes llamadas **particiones** , todas las cuales tienen los mismos atributos lógicos. Por ejemplo, todas las particiones en una tabla comparten la misma columna y definiciones de restricciones, y todas las particiones en un índice comparten las mismas columnas de índice.

Los nombres de partición extendida y subpartición le permiten realizar algunas operaciones de nivel de partición y de partición, como eliminar todas las filas de una partición o subpartición, en una sola partición o subpartición. Sin nombres extendidos, tales operaciones requerirían que especifique un predicado ( cláusula WHERE). Para las tablas particionadas por rango y por lista, tratar de expresar una operación de nivel de partición con un predicado puede resultar engorroso, especialmente cuando la clave de partición de rango usa más de una columna. Para particiones y subparticiones hash, usar un predicado es aún más difícil, ya que estas particiones y subparticiones se basan en una función hash definida por el sistema.

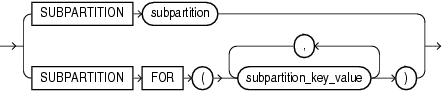
Los nombres de partición extendida le permiten usar particiones como si fueran tablas. Una ventaja de este método, que es más útil para tablas con particiones de rango, es que puede crear mecanismos de control de acceso a nivel de partición otorgando (o revocando) privilegios en estas vistas a (o desde) otros usuarios o roles. Para usar una partición como una tabla, cree una vista seleccionando datos de una sola partición y luego use la vista como una tabla.

**Sintaxis**Puede especificar nombres de tabla de partición extendida o subpartición extendida en cualquier declaración SQL en la que el elemento *partition\_extended\_nam* o *subpartition\_extended\_name* aparezca en la sintaxis.

***partition\_extended\_name* :: =**

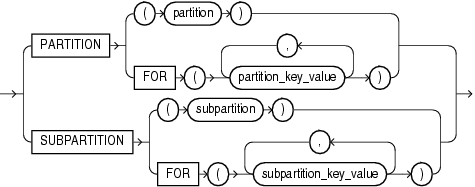
  
[Descripción de la ilustración '' partition\_extended\_name.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/partition_extended_name.htm)

***subpartition\_extended\_name* :: =**

  
[Descripción de la ilustración '' subpartition\_extended\_name.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/subpartition_extended_name.htm)

Las instrucciones DML INSERT, UPDATE y DELETE y la afirmación ANALYZE requiere paréntesis alrededor del nombre de la partición o subpartición. Esta pequeña distinción se refleja en el *partition\_extension\_clause*:

***partition\_extension\_clause* :: =**

  
[Descripción de la ilustración '' partition\_extension\_clause.gif ''](https://docs.oracle.com/cd/E11882_01/server.112/e41084/img_text/partition_extension_clause.htm)

En *partition\_extended\_name*, *subpartition\_extended\_name* y *partition\_extension\_clause*, las cláusulas PARTITION FOR y SUBPARTITION FOR que se refieren vamos a una partición sin usar su nombre. Son válidos con cualquier tipo de partición y son especialmente útiles para particiones de intervalo. Las particiones de intervalo se crean automáticamente según sea necesario cuando los datos se insertan en una tabla.

Para el respectivo *partition\_key\_value* o *subpartition\_key\_value*, especifique un valor para cada columna de clave de partición. Para las claves de partición de varias columnas, especifique un valor para cada clave de partición. Para las particiones compuestas, especifique un valor para cada clave de partición, seguido de un valor para cada clave de subpartición. Todos los valores de clave de partición están separados por comas. Para las particiones de intervalo, puede especificar solo una *partition\_key\_value*, y debe ser un valor NUMBER válido o de fecha y hora. Su declaración SQL operará en la partición o subparticiones que contienen los valores que especifique.

**Restricciones en los nombres extendidos**Actualmente, el uso de nombres de tablas extendidas por partición y extendidas por subpartición tiene las siguientes restricciones:

* No hay tablas remotas: un nombre de tabla de partición extendida o subpartición extendida no puede contener un enlace de base de datos (dblink) o un sinónimo que se traduce en una tabla con un enlace doble. Para usar particiones y subparticiones remotas, cree una vista en el sitio remoto que use la sintaxis del nombre de la tabla extendida y luego refiérase a la vista remota.
* Sin sinónimos: una extensión de partición o subpartición debe especificarse con una tabla base. No puedes usar sinónimos, vistas o cualquier otro objeto.
* Las cláusulas PARTITION FOR y SUBPARTITION FOR no son válidas para operaciones DDL en vistas.
* En los PARTITION FOR y las cláusulas SUBPARTITION FOR, no se puede especificar las palabras clave DEFAULT o MAXVALUE o una variable de vinculación para el *partition\_key\_value* o *subpartition\_key\_value*.
* En las cláusulas PARTITION y SUBPARTITION, no puede especificar una variable de enlace para el nombre de partición o subparte.

**Ejemplo**En la siguiente declaración, sales es una tabla particionada con partición sales\_q1\_2000. Puede crear una vista de la partición única sales\_q1\_2000 y luego usarla como si fuera una tabla. Este ejemplo elimina filas de la partición.

CREATE VIEW Q1\_2000\_sales AS

SELECT \*

FROM sales PARTITION (SALES\_Q1\_2000);

DELETE FROM Q1\_2000\_sales

WHERE amount\_sold < 0;

## Referencias a atributos y métodos de tipo de objeto

Para hacer referencia a los atributos o métodos de tipo de objeto en una declaración SQL, debe calificar completamente la referencia con un alias de tabla.

Considere el siguiente ejemplo del esquema de muestra oe, que contiene un tipo cust\_address\_typ y una tabla customers con una columna cust\_address basada en cust\_address\_typ:

CREATE TYPE cust\_address\_typ

OID '82A4AF6A4CD1656DE034080020E0EE3D'

AS OBJECT

(street\_address VARCHAR2(40),

postal\_code VARCHAR2(10),

city VARCHAR2(30),

state\_province VARCHAR2(10),

country\_id CHAR(2));

/

CREATE TABLE customers

(customer\_id NUMBER(6),

cust\_first\_name VARCHAR2(20) CONSTRAINT cust\_fname\_nn NOT NULL,

cust\_last\_name VARCHAR2(20) CONSTRAINT cust\_lname\_nn NOT NULL,

cust\_address cust\_address\_typ,

. . .

En una declaración SQL, la referencia al atributo postal\_code debe estar completamente calificada utilizando un alias de tabla, como se ilustra en el siguiente ejemplo:

SELECT c.cust\_address.postal\_code

FROM customers c;

UPDATE customers c

SET c.cust\_address.postal\_code = '14621-2604'

WHERE c.cust\_address.city = 'Rochester'

AND c.cust\_address.state\_province = 'NY';

Para hacer referencia a un método miembro que no acepta argumentos, debe proporcionar paréntesis vacíos. Por ejemplo, el esquema de muestra oe contiene una tabla de objetos categories\_tab, basada en catalog\_typ, que contiene la función miembro getCatalogName. Para poder llamar a este método en una declaración SQL, debe proporcionar paréntesis vacíos como se muestra en este ejemplo:

SELECT TREAT(VALUE(c) AS

catalog\_typ).getCatalogName() "Catalog Type"

FROM categories\_tab c

WHERE category\_id = 90;

Catalog Type

------------------------------------

online catalog

# DATABASE LINK

# EXATADA

**Introducción**

El siguiente documento explica de una forma muy superficial las principales características de Oracle Exadata, ya sea su arquitectura y estructuras internas.

Además se mencionan las principales características de su componente estrella, el Exadata Storage Server, que en el fondo marca la diferencia con las bases de datos actuales.

Se mencionan los componentes técnicos y al final del documento se hace una breve reseña , como se estructuran jerárquicamente los distintos Storage Server para llegar a construir una base de datos.

¿Qué es Oracle Exadata?

Oracle Exadata es una máquina (rack) que da soporte de alta performance para aplicaciones OLTP y cargas OLAP.

En un principio fue un trabajo entre Oracle Corporation y Hewlett Packard, Oracle diseñaba todo lo que era base de datos y colocaba el sistema operativo (OEL) , lo que correspondía a Storage era parte de HP junto con la arquitectura de esta máquina. Esto fue un primer release, a los meses Oracle compraba Sun , con lo cual anuncia una versión de su Oracle Exadata , donde deja fuera a HP y ocupa todas las tecnologías de Sun Microsystems.

Hoy en día Oracle Exadata se distribuye con elección de Sistema Operativo, ya sea , OEL o Solaris 11 Express.

Oracle Exadata luce de esta forma.



Oracle Exadata es un rack que junta una serie de componentes, los cuales conforman la gran infraestructura de base de datos, entre esos componentes se encuentran discos, servidores, networking, etc.

Siendo un poco más técnicos, los servidores de base de datos tienen la siguiente composición

## Componentes técnicos

### Database servers X2-2

(Sun Fire X4170 M2)

2 × Six-Core Intel Xeon X5675 Processors (3.06 GHz)

96 GB Memory (expandable to 144 GB with optional memory expansion kit)

Disk Controller HBA with 512MB Battery Backed Write Cache

4 × 300 GB 10,000 RPM SAS Disks

2 × QDR (40Gbit/s) Ports

2 × 10 Gb Ethernet Ports based on the Intel 82599 10GbE Controller

4 × 1 Gb Ethernet Ports

1. × ILOM Ethernet Port
2. × Redundant Hot-Swappable Power Supplies
3. × 36 port QDR (40 Gbit/s) InfiniBand Switches (2 x in Quarter Rack configuration)

Los servidores de Storage

### Exadata Storage servers X2-2

2 × Six-Core Intel Xeon L5640 (2.26 GHz) Processors

Exadata Smart Flash Cache 384 GB

System Memory 24 GB

Disk Controller Disk Controller HBA with 512MB Battery Backed Write Cache

InfiniBand Connectivity Dual-Port QDR (40Gbit/s) InfiniBand Host Channel Adapter

Power Supplies Dual-redundant, hot-swappable power supply

Remote Management Sun Embedded Integrated Lights Out Manager (ILOM)

Disk Drives 12 × 600 GB 15,000 RPM High Performance SAS or

12 × 3 TB 7,200 RPM High Capacity SAS

Integrated Lights Out Manager (ILOM) Ethernet port

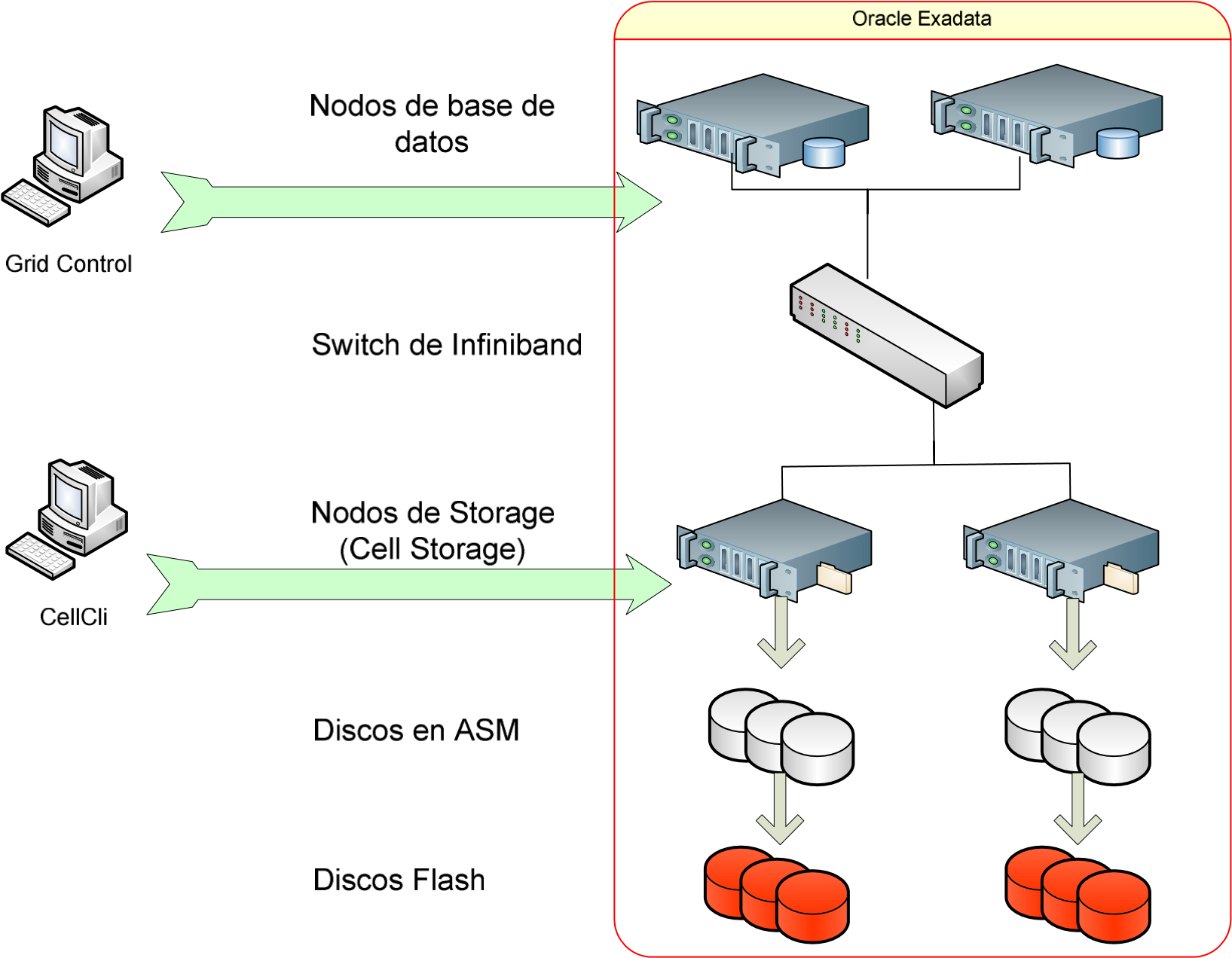
Y los switches presentes en el Exadata

### Infiniband switches

Sun Datacenter Infiniband Switch 36

36 ports

Si quisiéramos mostrar un dibujo de los componentes, sería algo así



Grid Control es el componente para la administración de Oracle Exadata y Cellcli es un utilitario que está en el lado de los nodos de Storage

Para lo anterior existen 3 configuraciones en el mercado , que combinan nodos de base de datos, nodos de storage y switches, esas configuraciones son Full Rack , half rack o quarter rack, para los tres casos son los mismos componentes , sólo varia la cantidad.

## Descripción de componentes

### Nodos de bases de datos (Database nodes)

La base de datos y el cluster corren en los nodos conocidos como Databases nodes, en estas máquinas por defecto corre un RAC versión 11gr2 . Un full rack contiene 8 nodos de base de datos, un half rack tiene 4 nodos y un quarter rack posee 2 nodos.

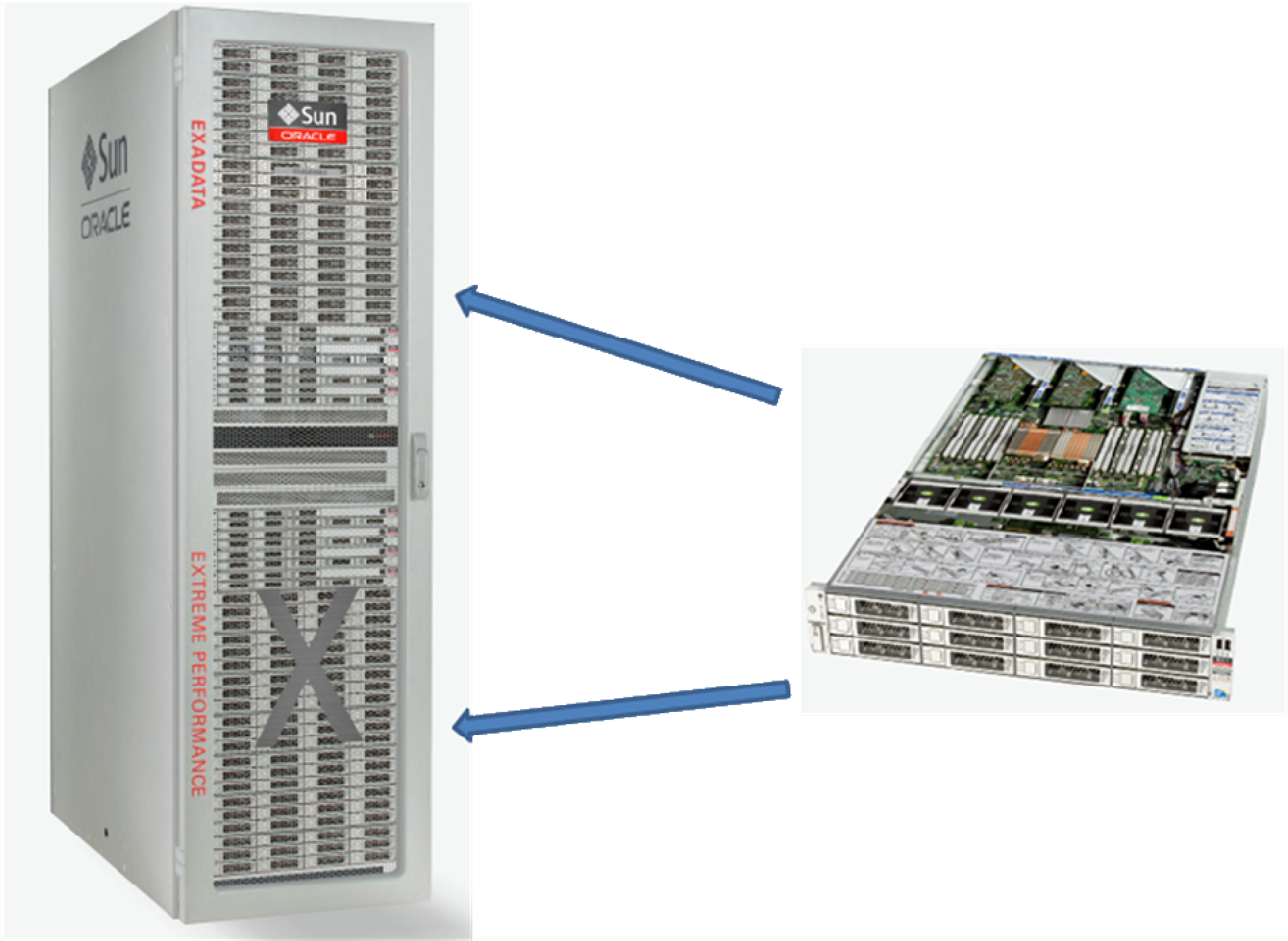
Para visualizarlo dentro del Exadata



### Nodos de Storage (Storage Cells)

En estas máquinas es donde se encuentran atachados los discos que se utilizarán para almacenar la información de nuestras bases de datos, sobre estas máquinas corre todo el software que maneja los discos

Para visualizarlo dentro del exadata



### Disks

Cada celda de Storage contiene 12 discos (dependiendo de la configuración), estos discos pueden ser de 600Gb o 2Tb.

### Flash Disks

Cada celda de storage , contiene además 384Gb de flash disk, estos discos pueden ser presentados a los nodos (Database nodes) como storage o usados como cache secundaria por el cluster de base de datos, esto último es conocido como Smart Cache.

### Infiniband Switch

Los databases nodes y las storage cell están conectadas a través de infiniband (para saber lo que es infiniband puede consultar el siguiente link)

Existe 3 switches de infiniband para nuestro Oracle Exadata, lo cual claramente da una seguridad pues elimina puntos de falla único al momento de realizar las comunicaciones.

### Ethernet Switch

Los usuarios y administradores de Oracle Exadata se comunican a través de Infiniband o a través de Ethernet.

## Exadata Storage Server

He aquí el gran secreto de Oracle y por lo cual ha alcanzado niveles de performance increíbles, este Storage es el encargado de procesar la información solicitada desde los Database nodes y entregar la información de la manera más rápida y consistente. Es un conjunto de software que trabaja de una forma tal que trata de disminuir los tiempos de respuesta de las aplicaciones.

Los storage tradicionales tienen serios problemas a nivel de desarrollo de IT, o sea, cada vez que hay una base de datos de por medio, siempre habrá problemas de performance, sobre todo en arquitecturas clásicas de arreglos de Storage.

El problema principal , son los anchos de banda al momento de que todas las bases de datos conectadas a un storage , comienzan a realizar consultas, obviamente el recurso de ancho de banda queda corto y siempre irá en perjuicio de los tiempos de respuesta

La gran cualidad de exadata es que está orientado a obtener los mejores resultados posibles a nivel de tiempos de respuesta e I/O, para lo anterior los Exadata Storage Server utilizan las siguientes cualidades

## Offloading of Data Search

Esta característica tiene relación con la capacidad de procesamiento de las celdas de Storage, esto implica que ellas son capaces de pre-procesar la información para minimizar la cantidad de información que va a ser traspasada a través de la red que conectar los databases nodes y los cell storage, las características que vienen a continuación , mejoran de una gran forma la información que se procesa, lo que obviamente impacta en el tráfico y en los tiempos de respuesta de las aplicaciones.

## Smart Scan

En una base de datos común y corriente , cada vez que se hace una consulta por ejemplo de una columna en una fila, el bloque completo es extraído desde los datafiles y llevado a ls SGA (buffer cache), indistinto de la cantidad de columnas que tenga la fila e indistinto de la cantidad de bloques que tenga el bloque Oracle.

En Oracle Exadata, lo anterior no cambia en casi nada , pero si hay algunas cosas muy interesantes y extremadamente útiles que sí marcan la diferencia, por ejemplo Direct Path Access, full table scans y full index scans. En lo anterior se puede colocar una fila o específicamente una columna directamente desde disco y enviada a las bases de datos. Lo anterior es conocido como Smart Scan, lo anterior implica una reducción increíble de I/O.

## iDB

iDB es la abreviación de Inteligent Database, el Smart Scan es la capacidad de envíar solo la información solicitada, pero para ello se debe saber que es lo que se envía, el iDB es un protocolo de comunación existente entre los databases nodes y los cell storage, para los casos en que se puede enviar solamente un pequeño grupo de información (o sea, que se puede aplicar un Smart Scan) el iDB envía los nombres de las tablas, las columnas, los predicados entre otros datos, con esta información las cell storage determinan de mejor forma la información a enviar más que solamente la dirección de los bloques a enviar, con lo anterior se envía solamente la fila o la columna en vez de enviar los bloques Oracle .

## Storage Indexes

Cada Cell Exadata mantiene un Storage Index que contiene un resumen de toda la distribución de data en los discos. Esta información es mantenida de de forma automática y es totalmente transparente para la base de datos.

Por cada región indexada, el storage index mantiene el mínimo y máximo valor de las columnas de una tabla (región de disco casi siempre de 1MB) , como son regiones distintas para cada Cell Storage , redunda en que el sistema es altamente escalable y nunca se producen esperas por contensiones de latch , puesto que a mayor cantidad de información , mayor la cantidad de Storage Indexes, los cuales no se deben confundir con los índices normales, que son estructuras totalmente distintas.

¿De qué sirve mantener el máximo y mínimo de las columnas indexadas? Pues bien, esto ayuda eliminando el I/O innecesario, este efecto se conoce como I/O filtering. Cada I/O que se produce en la celda de storage es almacenado en la vista V$SYS\_STAT y muestra el número de bytes de I/O que son “ahorrados” usando los Storage Index.

¿Qué consultas son mejoradas por los Storage Index? Pues todas las consultas que ejecuten cualquiera de las siguientes instrucciones :

* Igualdad (=)
* No iguales (< , ¡= o >)
* Menor igual que (<=)
* Mayor o igual que (>=)
* Is null
* Is not null

## Storage centralizado

Se puede usar Oracle Exadata Storage para centralizar todos los requerimientos de storage de una compañía, no importando la cantidad de bases de datos que lo utilicen.

Las celdas de exadata con ASM distribuyen toda la carga de I/O a través de todos los discos disponibles en el storage. Cada base de datos puede usar todos los discos disponibles con lo cual se alcanzan niveles de performance muy completos.

## I/O Resource Management

I/O Resource Management de aquí en adelante IORM y Database Resource Management, permiten que multiples bases de datos compartan el mismo storage, mientras se asegura que todos los recursos de I/O sean ocupados de buena forma a través de todas las bases de datos.

Lo anterior da como resultado que una base de datos no puede, entiéndase bien, no puede monopolizar los recursos de I/O cuando se accede a la información de las Cell Storage.

IORM es implementado y manejado a través del las políticas definidas en el Database resource management, Database resource management en una instancia de base de datos se comunica con el software de IORM en las storage cell, para manejar todas las políticas declaradas por el DBA (negocio) . Los database resource plan son manejados por la base de datos, mientras los interdatabase plans son manejados por las storage cell

## Offloading of Incremental Backup

También se optimizan los respaldos a través de RMAN, dejando fuera los bloques que no sean necesarios respaldar.

Esto implica que no solamente deja fuera los bloques vacíos, si no que también deja fuera los bloques que no son necesarios en una restauración, esto se hace de forma automática y no requiere intervención del DBA.

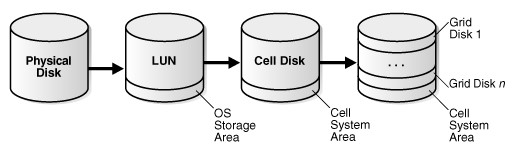
## Smart Cache

El Database buffer cache es el lugar donde los bloques de datos son leídos en una primera instancia, si no se encuentran allí, pues sencillamente se leen desde disco y allí se produce el inefable I/O. Este es el caso común de la totalidad de las bases de datos Oracle pre-Exadata.

Pues bien Oracle Exadata crea un cache intermedio , entre el storage y el buffer cache de la SGA, esta cache se llama Smart Cache, esta porción de memoria almacena los datos más frecuentemente usados, claramente esto puede redundar en que muchas veces se reduce el I/O dado que la información si no esta en el buffer cache, se encuentra disponible en esta segunda área de cache.

## Estructura jerarquíca de los discos

Para entender un poco como se estructuran los discos hasta llegar a trabajar con ASM, se presenta el siguiente cuadro



El anterior cuadro refleja lo siguiente

* Oracle Exadata se compone de múltiples servidores de storage, si hablamos de un Full Rack son 14 servidores de Storage, para un half rack son 7 servidores de storage y para un quarte rack son 3 servidores de Storage

* Cada Oracle Exadata Server, tiene 12 discos (SAS o SATA de 3”5), cada uno de los tipos de discos tiene una capacidad distinta y performance distintos.

* A partir de los anteriores discos se generan LUNs (particiones)

* Las LUNs son identificadas y presentadas como Cell Disk

* Cada Cell Disk es presentada como Grid Disk (las cell disks se pueden dividir en múltiples Grid Disks)

* Los Grid Disk, son usados como ASM Disks

* Y los ASM Disks, pues usados para construir ASM Diskgroups

* Y con los ASM Diskgroups, yo puedo generar mi base de datos

Si los Diskgroups de ASM tienen redundancia normal o alta, los grupos de falla de los diskgroups, siempre se van a ubicar en distintas celdas del storage, esto implica que si una celda falla, la información estará disponible en la otra celda

También hay una variación, de cuando se requiere montar un filesystem

* Oracle Exadata se compone de múltiples servidores de storage, si hablamos de un Full Rack son 14 servidores de Storage, para un half rack son 7 servidores de storage y para un quarte rack son 3 servidores de Storage

* Cada Oracle Exadata Server, tiene 12 discos (SAS o SATA de 3”5), cada uno de los tipos de discos tiene una capacidad distinta y performance distintos.

* A partir de los anteriores discos se generan LUNs (particiones)

* Cada partición es presentada como un VOLUME

* Cada volumen es con lo que se construye un punto de montaje

# SQL DEVELOPER

## 1 Conceptos y uso del desarrollador SQL

Oracle SQL Developer es una versión gráfica de SQL \* Plus que brinda a los desarrolladores de bases de datos una manera conveniente de realizar tareas básicas. Puede explorar, crear, editar y eliminar (soltar) objetos de base de datos; ejecutar instrucciones y scripts SQL; editar y depurar código PL / SQL; manipular y exportar datos; y ver y crear informes.

Puede conectarse a cualquier esquema de base de datos Oracle de destino mediante la autenticación estándar de la base de datos Oracle. Una vez conectado, puede realizar operaciones en objetos en la base de datos.

Puede conectarse a esquemas para bases de datos de terceros seleccionados (no Oracle), como MySQL, Microsoft SQL Server, Sybase Adaptive Server y Microsoft Access, y ver metadatos y datos en estas bases de datos; y puede migrar bases de datos de terceros a Oracle.

Este capítulo contiene las siguientes secciones principales:

[Instalar y comenzar con el desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CEGGHEFC)

[Interfaz de usuario de desarrollador SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAHIFGB)

[Objetos de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAJFCIJ)

[Conexiones de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHIFGJ)

[Entrar y modificar datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHFJBE)

[Ejecutar y depurar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIGHA)

[Usando la hoja de cálculo SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI)

[Uso de fragmentos para insertar fragmentos de código](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEJCHH)

[Usando el objeto Find DB para encontrar objetos de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEFHFE)

[Uso de la búsqueda extendida](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGFGFJ)

[Utilizando versioning](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH)

[Informes del desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBJIBJ)

[Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD)

[Ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD)

[Soporte de base de datos en memoria de Oracle TimesTen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJFIIH)

[Usando la ayuda](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGCAAA)

### Instalar y comenzar con el desarrollador de SQL

Para instalar e iniciar SQL Developer, simplemente descargue un archivo ZIP y descomprímalo en el directorio o carpeta principal que desee, y luego escriba un comando o haga doble clic en el nombre de un archivo. Debe leer la [*Guía de instalación de Oracle Database SQL Developer*](http://www.oracle.com/pls/topic/lookup?ctx=db112&id=RPTIG) antes de realizar la instalación. Después de leer la guía de instalación, los pasos básicos son:

1. Descomprima el kit de desarrollador de SQL en un directorio (carpeta) de su elección. Esta ubicación del directorio será referida como *<sqldeveloper\_install>*.

Al descomprimir el kit de desarrollador de SQL, sqldeveloperse creará un directorio nombrado en el *<sqldeveloper\_install>*directorio. También hace que muchos archivos y carpetas se coloquen en y debajo de ese directorio.

**Si también se instala Oracle Database (versión 11 o posterior), también se incluye** una versión de SQL Developer y se puede acceder a ella a través del sistema de menú bajo Oracle. Esta versión de SQL Developer es independiente de cualquier kit de desarrollador de SQL que descargue y descomprima por su cuenta, así que no confunda los dos y no descomprima un kit sobre los archivos de desarrollador de SQL que se incluyen con la base de datos Oracle. Sugerencia: cree un acceso directo para el archivo ejecutable de SQL Developer que instale y siempre utilícelo para iniciar SQL Developer.

1. Para iniciar SQL Developer, vaya al sqldeveloperdirectorio debajo del *<sqldeveloper\_install>*directorio y realice una de las siguientes acciones:

En sistemas Linux y Mac OS X, ejecute *sh sqldeveloper.sh* .

En los sistemas Windows, haga doble clic sqldeveloper.exe.

Si se le solicita que ingrese la ruta de acceso completa para java.exe, haga clic en **Examinar** y busque java.exe. Por ejemplo, en un sistema Windows, la ruta podría tener un nombre similar a C:\Program Files\Java\jdk1.6.0\_06\bin\java.exe.

1. Si desea familiarizarse con los conceptos de SQL Developer antes de usar la interfaz, lea el resto de este capítulo antes de continuar con el siguiente paso.
2. Cree al menos una conexión de base de datos (o importe algunas conexiones exportadas anteriormente), de modo que pueda ver y trabajar con objetos de base de datos, usar la hoja de cálculo SQL y otras características.

Para crear una nueva conexión de base de datos, haga clic con el botón derecho en el nodo **Conexiones** en el navegador de Conexiones, seleccione **Nueva conexión** y complete las entradas necesarias en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) .

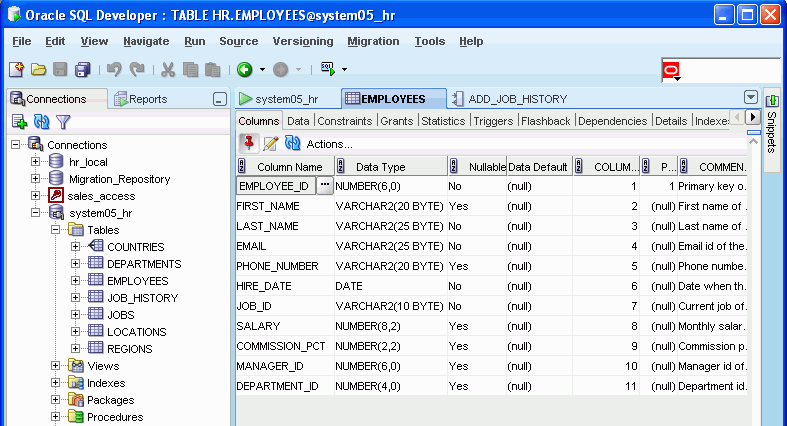
1. Si desea comenzar rápidamente con SQL Developer, realice el breve tutorial en el [Capítulo 3, "Tutorial: Crear objetos para una pequeña base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHDFAF) , o trabaje con sus objetos de base de datos existentes.

### Interfaz de usuario de desarrollador SQL

La ventana del desarrollador de SQL generalmente usa el lado izquierdo para la navegación para buscar y seleccionar objetos, y el lado derecho para mostrar información sobre los objetos seleccionados.

[La figura 1-1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BABCHFGH) muestra la ventana principal.

***Figura 1-1 Ventana principal de SQL Developer***



**Nota:**

Este texto explica la interfaz por defecto. Sin embargo, puede personalizar muchos aspectos de la apariencia y el comportamiento de SQL Developer configurando las preferencias (consulte [Preferencias de SQL Developer](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) ). Si alguna vez necesita restaurar la interfaz predeterminada, consulte [Restauración de la apariencia original](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHFIA) .

**Nota:**

Para la migración de bases de datos de terceros a Oracle, consulte también la [Interfaz de usuario de desarrollador de SQL para la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCEH) .

Los menús en la parte superior contienen entradas estándar, más las entradas para funciones específicas de SQL Developer (ver [Menús para SQL Developer](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHJBA) ), como se muestra en la siguiente figura.

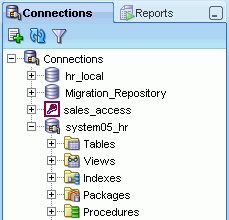
Parte superior de la ventana: menús e iconos

Puede usar las **teclas** de acceso **directo** para acceder a los menús y elementos del menú: por ejemplo, Alt + F para el menú Archivo y Alt + E para el menú Editar; o Alt + H, luego Alt + S para ayuda, luego búsqueda de texto completo. También puede mostrar el menú Archivo presionando la tecla F10.

Los iconos debajo de los menús realizan varias acciones, incluyendo las siguientes:

* **Nuevo** crea un nuevo y nuevo objeto de base de datos (consulte [Crear / Editar nuevo objeto (Nueva galería)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBBDCB) ).
* **Abrir** abre un archivo (ver [Abrir archivo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCGAAG) ).
* **Guardar** guarda cualquier cambio en el objeto seleccionado actualmente.
* **Guardar todo** guarda cualquier cambio en todos los objetos abiertos.
* **Atrás se** mueve al panel que has visitado más recientemente. (O use la flecha desplegable para especificar una vista de pestaña).
* **Forward se** mueve al panel después del actual en la lista de paneles visitados. (O use la flecha desplegable para especificar una vista de pestaña).
* **Abrir hoja de trabajo SQL** abre la hoja de trabajo SQL (consulte [Uso de la hoja de trabajo SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) ). Si no usa la flecha desplegable para especificar la conexión de la base de datos a usar, se le solicitará que seleccione una conexión.

El lado izquierdo de la ventana del desarrollador de SQL tiene pestañas y paneles para los navegadores de Conexiones e Informes, los iconos para realizar acciones y una visualización de árbol jerárquico para el navegador seleccionado actualmente, como se muestra en la siguiente figura.



El **navegador de conexiones** enumera las conexiones de base de datos que se han creado. Para crear una nueva conexión de base de datos, importe un archivo XML con definiciones de conexión, o exporte o edite las conexiones actuales, haga clic con el botón derecho en el nodo Conexiones y seleccione el elemento de menú apropiado. (Para obtener más información, consulte [Conexiones de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHIFGJ) ).

El **navegador de archivos** (marcado por un ícono de carpeta; no se muestra en la figura anterior) muestra su sistema de archivos local utilizando una jerarquía estándar de carpetas y archivos. Puede hacer doble clic o arrastrar y soltar archivos para abrirlos, y puede editar y guardar los archivos. Por ejemplo, si abre un archivo .sql, se mostrará en una ventana de hoja de cálculo SQL. El navegador de archivos es especialmente útil si está utilizando el control de versiones con SQL Developer (consulte [Uso de control de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) ).

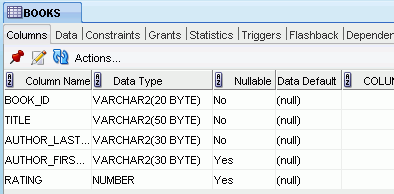
El **navegador de informes** enumera los informes informativos proporcionados por el desarrollador de SQL, como una lista de tablas sin claves principales para cada conexión de base de datos, así como los informes definidos por el usuario. (Para obtener más información, consulte [Informes de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBJIBJ) ).

Los iconos debajo de la pestaña Conexiones (sobre el árbol de metadatos) realizan las siguientes acciones en el objeto seleccionado actualmente:

* **Actualizar** consulta a la base de datos los detalles actuales sobre el objeto seleccionado (por ejemplo, una conexión o solo una tabla).
* **Aplicar filtro** restringe la visualización de objetos usando un filtro que usted especifique. Por ejemplo, puede hacer clic con el botón derecho en el nodo Tablas y especificar un filtro de EM% para ver solo las tablas que comienzan con *EM* y hacer que la etiqueta del nodo Tablas se cambie a *Tablas (EM%)* . Para eliminar los efectos de aplicar un filtro, haga clic con el botón derecho en el nodo y seleccione **Borrar filtro** .

El árbol de metadatos en el panel de Conexiones muestra todos los objetos (categorizados por tipo de objeto) accesibles a las conexiones definidas. Para seleccionar un objeto, expanda el nodo del árbol o los nodos apropiados, luego haga clic en el objeto.

El lado derecho de la ventana del Desarrollador de SQL tiene pestañas y paneles para los objetos que selecciona o abre, como se muestra en la siguiente figura, que muestra información sobre una tabla llamada LIBROS. (Si mantiene el puntero del mouse sobre la etiqueta de la pestaña - LIBROS en esta figura - una información sobre herramientas muestra el propietario del objeto y la conexión de la base de datos).



Para objetos que no sean subprogramas, los iconos proporcionan las siguientes opciones:

* **Congelar vista** (el **pin** ) mantiene la pestaña y la información de ese objeto en la ventana cuando hace clic en otro objeto en el navegador de Conexiones; se crean una pestaña y una pantalla separadas para ese otro objeto. Si vuelve a hacer clic en el pin, la pantalla del objeto está disponible para su reutilización.
* **Editar** muestra un cuadro de diálogo para editar el objeto.
* **Actualizar** actualiza la pantalla consultando la base de datos para obtener la información más reciente.
* **Acciones** muestra un menú con las acciones apropiadas para el objeto. Las acciones son las mismas que cuando hace clic con el botón derecho en un objeto de ese tipo en el navegador de Conexiones, excepto que el menú Acciones no incluye Editar.

Para cambiar entre objetos, haga clic en las pestañas deseadas; para cerrar una pestaña, haga clic en la X en la pestaña. Si realiza cambios en un objeto y hace clic en la X, se le preguntará si desea guardar los cambios.

Para tablas y vistas, esta información se agrupa en pestañas, que están etiquetadas cerca de la parte superior. Por ejemplo, para las tablas, las pestañas son Columnas, Datos (para ver y modificar los datos en sí), Índices, Restricciones, etc. y puede hacer clic en un encabezado de columna debajo de una pestaña para ordenar las filas de la cuadrícula por los valores de esa columna. Para la mayoría de los objetos, las pestañas incluyen SQL, que muestra la declaración SQL para crear el objeto.

Puede exportar datos desde un panel de detalles o desde los resultados de una operación de hoja de cálculo SQL o un informe utilizando el menú contextual y seleccionando **Exportar** .

El área **Mensajes - Registro** se usa para obtener información de retroalimentación según corresponda (por ejemplo, resultados de una acción o mensajes de error o advertencia). Si esta área aún no está visible, puede mostrarla haciendo clic en Ver y luego en Iniciar sesión.

El área del **compilador - Registro** se usa para cualquier mensaje que se muestre como resultado de una operación de compilación o compilación para depuración.

### Menus para SQL Developer

Este tema explica los elementos de menú que son específicos de SQL Developer.

**Menú de edición**

**Pegado extendido** : muestra el cuadro de diálogo [Pegar](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCHFGG) , en el que selecciona un elemento del portapapeles (de potencialmente muchos) para pegarlo en la ubicación actual.

**Selección duplicada** : cuando ha seleccionado texto mientras edita una función o procedimiento, crea una copia del texto seleccionado en la ubicación actual.

**Ajustar selección** : cuando haya seleccionado texto mientras edita una función o un procedimiento, ajusta el texto seleccionado.

**Ver menú**

Contiene opciones que afectan lo que se muestra en la interfaz del desarrollador de SQL.

**Conexiones** : muestra el navegador de conexiones.

**Archivos** : muestra el navegador de archivos, que está marcado con un icono de carpeta. Puede usar el navegador de archivos para explorar, abrir, editar y guardar archivos accesibles desde el sistema local.

**Informes** : muestra el navegador de informes (consulte [Informes de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBJIBJ) ).

**Modelos capturados** : muestra el navegador de modelos capturados (ver [Interfaz de usuario de desarrollador de SQL para la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCEH) ).

**Modelos convertidos** : muestra el navegador de modelos convertidos (consulte [Interfaz de usuario de desarrollador de SQL para la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCEH)).

**Buscar objeto DB** : muestra el panel Buscar objeto de base de datos (consulte [Uso del objeto Buscar base de datos para buscar objetos de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEFHFE) ).

**Navegador de versiones** : muestra el navegador de versiones (consulte [Uso de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) ).

**Registro** : muestra el panel Mensajes - Registro, que puede contener errores, advertencias y mensajes informativos.

**Depurador** : muestra los paneles relacionados con la depuración (vea [Ejecución y depuración de funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIGHA) ).

**Gestor de ejecución** : muestra el panel Gestor de ejecución, que contiene entradas para las sesiones de depuración activas.

**Historial de SQL** : muestra información sobre las sentencias de SQL que ha ejecutado. Puede seleccionar las declaraciones y agregarlas o sobrescribirlas en la hoja de trabajo (ver [Historial de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDGDHF) ).

**Fragmentos** : muestra los fragmentos (véase [Uso de fragmentos de código para insertar fragmentos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEJCHH) ).

**Barra de estado** : controla la visualización de la barra de estado en la parte inferior de la ventana del desarrollador de SQL.

**Barras de herramientas** : controla la visualización de la barra de herramientas principal (en los menús de SQL Developer) y la barra de herramientas del navegador de conexiones.

**Actualizar** : actualiza la pantalla actual para cualquier conexión abierta que use los objetos actuales en la base de datos o bases de datos afectadas.

**Menú de navegación**

Contiene opciones para navegar a paneles y en la ejecución de subprogramas.

**Atrás** : se desplaza al panel que ha visitado más recientemente.

**Adelante** : se desplaza al panel después del actual en la lista de paneles visitados.

**Alternar marcador** : si está editando una función o un procedimiento, crea o elimina un marcador (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Eliminar marcadores del archivo** : elimina los marcadores de la ventana de edición actualmente activa para una función o procedimiento (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Eliminar todos los marcadores** : elimina los marcadores de las ventanas de edición abiertas para funciones y procedimientos (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Ir a marcador** : muestra un cuadro de diálogo para que pueda ir a un marcador específico (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Ir al siguiente marcador** : va al siguiente marcador en la ventana de edición actualmente activa para una función o procedimiento (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Ir al marcador anterior** : va al marcador anterior en la ventana de edición actualmente activa para una función o procedimiento (consulte [Uso de marcadores al editar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCHGD) ).

**Ir a línea** : va al número de línea especificado y resalta la línea en la ventana de edición para la función o el procedimiento seleccionado. (Para mostrar los números de línea, habilite **Mostrar números de línea** en el [Editor de código:](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFCIBD) Preferencias de [canal de línea](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFCIBD) ).

**Ir a la última edición** : va a la última línea que se editó en la ventana de edición para una función o procedimiento.

**Ir a archivos recientes** : muestra el cuadro de diálogo [Archivos recientes](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJEEBC) , en el que puede especificar una función o un procedimiento al que ir.

**Ejecutar menú**

Contiene opciones relevantes cuando se selecciona una función o un procedimiento o cuando está abierto para la depuración.

**Ejecutar [nombre]** : inicia la ejecución de la función o el procedimiento especificado.

**Depurar [nombre]** : inicia la ejecución de la función o el procedimiento especificado en el modo de depuración.

Los elementos restantes en el menú Depurar coinciden con los comandos en la barra de herramientas de depuración, que se describe en [Ejecutar y depurar funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIGHA) .

**Menú fuente**

Contiene opciones para usar al editar funciones y procedimientos.

**Información detallada sobre la finalización** : hace que se muestre una ventana emergente cuando escribe el código PL / SQL, enumerando los elementos de los que puede seleccionar para completar automáticamente. Ver las opciones del [Editor de código](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDDEBF) en [Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) .

**Alternar comentarios de línea** : inserta y elimina los indicadores de comentarios al inicio de las líneas de código seleccionadas.

**Bloqueo de sangría** : mueve las instrucciones seleccionadas a la derecha.

**Unindent Block** : mueve las declaraciones seleccionadas a la izquierda.

**Menú de versiones**

Contiene opciones relacionadas con el soporte para el sistema de control de fuente y gestión de versiones CVS y Subversion; Consulte [Uso de control](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) de [versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) para obtener más información.

Los comandos en el menú de versiones dependen de si el sistema de versiones está especificado como CVS, Subversion o ninguno.

**Menú de migración**

Contiene opciones relacionadas con la migración de bases de datos de terceros a Oracle; Vea el [Menú de Migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFFEIC) para más información.

**Menú de herramientas**

Invoca herramientas de desarrollador de SQL.

**Copia de base de datos** : le permite copiar objetos de un esquema de base de datos a otro (consulte la interfaz de [Copia de base de datos (objetos de esquema)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDECCG) ).

**Exportación de base de datos** : le permite exportar algunos o todos los objetos de uno o más tipos de objetos para una conexión de base de datos a un archivo que contiene sentencias de SQL para crear estos objetos y, opcionalmente, exportar datos de la tabla (consulte la interfaz [Exportar (Objetos de la base de datos y Datos)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJFIFF) ).

**Base de** datos diferencial: le permite comparar dos esquemas para encontrar diferencias entre objetos del mismo tipo y nombre (por ejemplo, tablas llamadas CLIENTES) en dos esquemas diferentes, y opcionalmente actualizar los objetos en el esquema de destino para reflejar las diferencias en el esquema de origen (Ver la interfaz de [diferencias de esquema de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIGDDD) ).

**Monitorear sesiones** : muestra el estado de una o más sesiones, utilizando la información de la vista V $ RSRC\_SESSION\_INFO, que muestra cómo el Administrador de recursos de base de datos de Oracle ha afectado la sesión. Para obtener más información sobre la supervisión de la sesión, consulte [*la Guía del administrador de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e25494/dbrm.htm#ADMIN11906) .

**Monitor SQL** : muestra información sobre cualquier consulta que se esté ejecutando actualmente y consultas que se ejecutan para una conexión seleccionada. Para ver información detallada sobre una consulta, haga clic con el botón derecho en su fila y seleccione Mostrar detalles de SQL. La información es especialmente útil para el monitoreo en tiempo real de sentencias SQL de larga ejecución. Las estadísticas del cursor (como los tiempos de CPU y IO) y las estadísticas del plan de ejecución (como el número de filas de salida, la memoria y el espacio temporal utilizado) se actualizan casi en tiempo real durante la ejecución de la declaración. (Internamente, esta función llama al subprograma DBMS\_SQLTUNE.REPORT\_SQL\_MONITOR).

**Hoja de trabajo de SQL** : muestra una hoja de trabajo en la que puede ingresar y ejecutar sentencias de SQL y PL / SQL usando una conexión específica (consulte [Uso de la](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) hoja de [trabajo de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) ).

**Herramientas externas** : muestra el cuadro de diálogo [Herramientas externas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHHCFE) , con información sobre las herramientas externas definidas por el usuario que están integradas con la interfaz del desarrollador de SQL. Desde este cuadro de diálogo puede agregar herramientas externas (consulte [Crear / Editar herramienta externa](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAAHG) ). El menú Herramientas también contiene elementos para cualquier herramienta externa definida por el usuario.

**Preferencias** : le permite personalizar el comportamiento de SQL Developer (consulte [Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) ).

**Menú de ayuda**

Muestra ayuda sobre SQL Developer y le permite verificar las actualizaciones de SQL Developer.

**Buscar** : muestra la ventana del Centro de ayuda.

**Tabla de contenido** : muestra la ventana del Centro de ayuda.

**Consejo del día** : muestra una sugerencia para el uso eficiente de SQL Developer.

**Buscar actualizaciones** : comprueba si hay actualizaciones de las extensiones opcionales seleccionadas de SQL Developer, así como de cualquier extensión obligatoria de SQL Developer. Las actualizaciones disponibles pueden incluir el Controlador JTBC JTDS para Microsoft SQL Server y el Controlador JDBE MySQL, que le permiten crear conexiones a bases de datos de terceros. (Si el sistema que está utilizando está detrás de un servidor de seguridad, consulte las preferencias de usuario del desarrollador de SQL para [el navegador web y el proxy](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBFJA) ).

**Acerca de** : muestra información relacionada con la versión sobre SQL Developer y sus componentes.

### Restaurando el "Look and Feel" original

Si ha realizado cambios en la interfaz de usuario de SQL Developer ("apariencia"), como el reposicionamiento accidental de los navegadores y los paneles, puede restaurar la interfaz a la forma en que estaba después de instalar SQL Developer siguiendo estos pasos:

1. Si está ejecutando SQL Developer, salga.
2. Cree una copia de respaldo de la carpeta o directorio donde se almacena la información de usuario de su Desarrollador de SQL, en caso de que quiera restaurar cualquier informe antiguo, fragmentos de código, plantillas de código o historial de SQL definidos por el usuario. La ubicación por defecto es:
   * Windows: C: \ Documents and Settings \ *<nombre de usuario>* \ Application Data \ SQL Developer
   * Linux o Mac OS X: ~ / .sqldeveloper

Si ha especificado una ubicación no predeterminada para su información de usuario de desarrollador de SQL (consulte [Ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD) ), cree la copia de respaldo de esa carpeta o directorio.

(Si no desea utilizar ninguna información o configuración antiguas, puede omitir la creación de una copia de respaldo).

1. Elimine la carpeta o el directorio original (no la copia de seguridad) donde se almacena su información de usuario (se explica en el paso 2).
2. Inicie SQL Developer.

Esto crea una carpeta o directorio donde se almacena su información de usuario (explicada en el paso 2), que tiene el mismo contenido que cuando se instaló SQL Developer.

### Objetos de base de datos

Puede crear, editar y eliminar (eliminar) la mayoría de los tipos de objetos en una base de datos Oracle utilizando el menú contextual (clic derecho o Mayús + F10) en el navegador de Conexiones o haciendo clic en el botón **Acciones** en la pantalla del panel de detalles. Para algunos objetos, puede realizar otras operaciones, según sea apropiado para el tipo de objeto.

**Nota:**

Las acciones disponibles desde los menús contextuales y los botones de Acciones dependen del número de versión de la Base de Datos Oracle para la conexión de la base de datos especificada. Si una acción mencionada en el texto no está disponible con una conexión, es posible que la función no esté disponible en esa versión de Oracle Database.

Puede buscar objetos específicos asociados con una conexión de base de datos de Oracle haciendo clic en **Ver** , luego en **Buscar objeto DB**. Para obtener más información, consulte [Uso de objetos de base de datos de búsqueda para buscar objetos de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEFHFE) .

Si se ha conectado a bases de datos de terceros (que no sean de Oracle), como MySQL, Microsoft SQL Server, Sybase Adaptive Server o Microsoft Access, puede ver sus objetos utilizando el navegador de Connections. (Para obtener información sobre cómo conectarse a bases de datos de terceros, consulte las preferencias de usuario del desarrollador de SQL para [Base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) de [datos: controladores JDBC de terceros](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) ).

### Aplicaciones (Application Express 3.0.1 y posteriores)

En vigencia con Oracle Application Express 3.0.1, si usa SQL Developer para conectarse a un esquema que posee cualquier aplicación de Application Express, el navegador de Connections tiene un nodo de **Application Express** . Puede hacer clic en el nombre de una aplicación para mostrar las pestañas (Aplicación, Páginas, LOV, Listas, Plantillas, Breadcrumbs, etc.) con información sobre la aplicación.

Puede realizar las siguientes operaciones en una aplicación de Application Express haciendo clic con el botón derecho en el nombre de la aplicación en el navegador de Conexiones y seleccionando un elemento del menú:

* **Importar aplicación** : **importa** una aplicación desde un archivo específico e instala la aplicación.
* **Implementar aplicación** : implementa una aplicación en un esquema de destino específico.
* **Drop** : borra la aplicación.
* **Modificar aplicación** : le permite cambiar el alias, el nombre (Renombrar), el estado, la notificación global y el servidor proxy para la aplicación.
* **Exportar DDL** : guarda las declaraciones DDL para crear la aplicación (o el componente seleccionado) en un archivo, un archivo .zip, una hoja de trabajo o el portapapeles del sistema.

Las siguientes operaciones están disponibles solo haciendo clic derecho en el nodo de Application Express en el navegador de Conexiones y seleccionando un elemento del menú:

* **Iniciar EPG** : inicia la puerta de enlace PL / SQL incorporada para Application Express. Muestra un cuadro de diálogo para ejecutar las siguientes declaraciones: BEGIN DBMS\_EPG.map\_dad ('APEX', '/ apex / \*'); fin;
* **Detener EPG** : detiene la puerta de enlace PL / SQL incorporada para Application Express. Muestra un cuadro de diálogo para ejecutar las siguientes declaraciones: BEGIN DBMS\_EPG.unmap\_dad ('APEX'); fin;

### Grupos de caché (base de datos en memoria de Oracle TimesTen)

Un grupo de caché describe una colección de tablas de base de datos en memoria que se asignan a todas o un subconjunto de las tablas en una base de datos Oracle. Un grupo de caché puede constar de todas o de un subconjunto de las filas y columnas en estas tablas. Se pueden usar varios grupos de caché para almacenar en caché diferentes conjuntos de tablas relacionadas en la base de datos Oracle.

### Enlaces de base de datos (públicos y privados)

Un enlace de base de datos es un objeto de base de datos en una base de datos que le permite acceder a objetos en otra base de datos. La otra base de datos no necesita ser un sistema de base de datos Oracle; sin embargo, para acceder a sistemas que no sean de Oracle, debe usar servicios heterogéneos de Oracle. Una vez que haya creado un enlace de base de datos, puede usarlo para referirse a tablas y vistas en la otra base de datos. El navegador de conexiones tiene una **base de datos Enlaces** nodo para todos los enlaces de bases de datos (públicas y privadas) pertenecientes al usuario asociado con la conexión especificada, y una **base de datos pública Enlaces** nodo para todos los enlaces de bases de datos públicas sobre la base de datos asociada a la conexión. Para obtener ayuda con opciones específicas para crear un enlace de base de datos, consulte [Crear / Editar enlace de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIAAAE) .

Puede realizar las siguientes operaciones en un enlace de base de datos haciendo clic con el botón derecho en el nombre del enlace de base de datos en el navegador de Conexiones y seleccionando un elemento del menú:

* **Probar el enlace de** la base de datos: valida el enlace de la base de datos.
* **Drop** : borra el enlace de la base de datos.

### Directorios

Un objeto de directorio especifica un alias para un directorio (llamado carpeta en los sistemas Windows) en el sistema de archivos del servidor donde se encuentran los LOB de archivos binarios externos (BFILE) y los datos de la tabla externa. Para crear un directorio (es decir, un objeto de directorio), puede usar SQL Developer o la instrucción SQL CREATE DIRECTORY.

Puede usar nombres de directorio cuando haga referencia a BFILE en su código PL / SQL y a las llamadas OCI, en lugar de codificar el nombre de la ruta del sistema operativo, para la flexibilidad de la administración. Todos los directorios se crean en un solo espacio de nombres y no son propiedad de un esquema individual. Puede asegurar el acceso a los BFILE almacenados dentro de la estructura de directorios otorgando privilegios de objeto en los directorios a usuarios específicos.

### Funciones

Una función es un tipo de subprograma PL / SQL, que es un objeto de programación que se puede almacenar y ejecutar en el servidor de base de datos, y se puede llamar desde otros objetos de programación o aplicaciones. (Las funciones devuelven un valor; los procedimientos no devuelven un valor). Para obtener ayuda con las opciones específicas para crear un subprograma PL / SQL, consulte [Crear](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBIBAI) subprograma [PL / SQL (Función o Procedimiento)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBIBAI) .

Puede realizar las siguientes operaciones en una función haciendo clic con el botón derecho en el nombre de la función en el navegador de Conexiones y seleccionando un elemento del menú:

* **Abrir** : muestra el texto de la función para que pueda verlo y editarlo.
* **Compilar** : realiza una compilación PL / SQL de la función.
* **Compilar para depuración** : realiza una compilación de PL / SQL del procedimiento, con las unidades de la biblioteca de PL / SQL compiladas para la depuración.
* **Ejecutar** : muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) , y luego ejecuta la función en modo normal (no en depuración).
* **Depurar** : muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) y luego ejecuta la función en modo depuración.
* **Perfil** (para una conexión Oracle Database versión 11.1 o posterior): muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) , y luego ejecuta la función y recopila estadísticas de ejecución.
* **Renombrar** : **Renombra** la función.
* **Drop** : borra la función.

### Índices

Un índice es un objeto de base de datos que contiene una entrada para cada valor que aparece en la (s) columna (s) indexada (s) de la tabla o clúster y brinda acceso directo y rápido a las filas. Los índices se crean automáticamente en columnas de clave primaria; sin embargo, debe crear índices en otras columnas para obtener los beneficios de la indexación. Para obtener ayuda con las opciones específicas para crear un índice, consulte [Crear / Editar índice](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEAJAD) .

Puede realizar las siguientes operaciones en un índice haciendo clic con el botón derecho en el nombre del índice en el navegador de Conexiones y seleccionando un elemento del menú:

* **Drop** : borra el índice.
* **Reconstruir índice** : vuelve a crear el índice o una de sus particiones o subparticiones. Si el índice es inutilizable, una operación de reconstrucción exitosa hace que el índice sea utilizable. Para un índice basado en funciones, la reconstrucción también habilita el índice; sin embargo, si la función en la que se basa el índice no existe, la operación de reconstrucción falla.
* **Renombrar índice** : cambia el nombre del índice.
* **Hacer inutilizable** : evita que el índice sea utilizado por Oracle en la ejecución de consultas. Un índice inutilizable debe ser reconstruido, o descartado y recreado, antes de que pueda ser usado nuevamente.
* **Coalesce** : fusiona el contenido de los bloques de índice, donde sea posible, para liberar bloques para su reutilización.
* **Calcular estadísticas** : para un índice basado en funciones, recopila estadísticas sobre el índice y su tabla base mediante el paquete DBMS\_STATS. Dichas estadísticas permitirán que Oracle Database decida correctamente cuándo usar el índice.
* **Exportar DDL** : guarda la declaración DDL para crear el índice en un archivo, una hoja de cálculo SQL o el portapapeles del sistema.

### Fuentes de Java

Las fuentes Java se pueden crear y administrar en la base de datos. Puede crear un objeto fuente Java haciendo clic con el botón derecho en el nodo Java en el navegador de Conexiones, seleccionando **Cargar Java** y especificando el nombre y el código fuente de Java. (Se ejecuta una declaración CREAR O REEMPLAZAR Y RESOLVER JAVA SOURCE utilizando la información que especifique). Para obtener información sobre los conceptos de Java y los procedimientos almacenados, consulte [*la Guía para desarrolladores de bases de datos Oracle*](https://docs.oracle.com/cd/E11882_01/java.112/e10588/toc.htm) .

### Vistas materializadas

Una vista materializada es un objeto de base de datos que contiene los resultados de una consulta. La cláusula FROM de la consulta puede nombrar tablas, vistas y otras vistas materializadas. En conjunto, estos objetos se denominan tablas maestras (un término de replicación) o tablas de detalles (un término de almacenamiento de datos). Esta referencia utiliza "tablas maestras" para la consistencia. Las bases de datos que contienen las tablas maestras se denominan bases de datos maestras. Para obtener ayuda con las opciones específicas para crear una vista materializada, consulte [Crear / Editar vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJIGG) , especialmente el [panel Ver información de Propiedades de la vista materializada](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBJIBD) .

### Registros de vistas materializadas

Un registro de vista materializada es una tabla asociada con la tabla maestra de una vista materializada. Cuando se realizan cambios en DML a los datos de la tabla maestra, la base de datos Oracle almacena las filas que describen esos cambios en el registro de la vista materializada y luego utiliza el registro de la vista materializada para actualizar las vistas materializadas basadas en la tabla maestra. Este proceso se llama actualización incremental o rápida. Sin un registro de vista materializada, Oracle Database debe volver a ejecutar la consulta de vista materializada para actualizar la vista materializada. Este proceso se llama una actualización completa. Por lo general, una actualización rápida lleva menos tiempo que una actualización completa.

### Paquetes

Un paquete es un objeto que contiene subprogramas, que son objetos de programación que se pueden almacenar y ejecutar en el servidor de base de datos, y se pueden llamar desde otros objetos de programación o aplicaciones. Un paquete puede contener funciones o procedimientos, o ambos. Para obtener ayuda con opciones específicas para crear un paquete, consulte [Crear](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFHCIJ) paquete [PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFHCIJ) .

Puede realizar las siguientes operaciones en un paquete haciendo clic derecho en el nombre del paquete en el navegador de Conexiones y seleccionando un elemento del menú:

* **Editar** : muestra una vista con pestañas de solo lectura de la información sobre el paquete.
* **Abrir** : abre el paquete en una ventana, donde puede modificar el contenido y otra información.
* **Ejecutar** : le permite seleccionar un miembro en el paquete y ejecutarlo.
* **Compilar** : realiza una compilación PL / SQL de los miembros en el paquete.
* **Compilar para depuración** : realiza una compilación de PL / SQL de los miembros en el paquete, con unidades de biblioteca de PL / SQL compiladas para la depuración
* **Ordenar miembros por** : Ordena a los miembros del paquete por ubicación en la fuente, por nombre o por tipo y por nombre dentro de cada tipo.
* **Drop Package** : borra el paquete.
* **Crear cuerpo** : muestra un panel en el que puede ingresar texto para el cuerpo del paquete.
* **Otorgar** : le permite otorgar privilegios en el paquete
* **Revocar** : le permite revocar privilegios en el paquete.
* **Guardar especificación y cuerpo** del paquete: guarda la especificación y el cuerpo del paquete en un archivo que usted especifique.
* **Exportar DDL** : guarda la declaración DDL para crear el paquete en un archivo, una hoja de cálculo SQL o el portapapeles del sistema.

### Procedimientos

Un procedimiento es un tipo de subprograma PL / SQL, que es un objeto de programación que se puede almacenar y ejecutar en el servidor de base de datos, y se puede llamar desde otros objetos de programación o aplicaciones. (Los procedimientos no devuelven un valor; las funciones devuelven un valor). Para obtener ayuda con opciones específicas para crear un subprograma PL / SQL, consulte [Crear](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm" \l "BACBIBAI)subprograma [PL / SQL (Función o Procedimiento)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBIBAI) .

Puede realizar las siguientes operaciones en un procedimiento haciendo clic con el botón derecho en el nombre del procedimiento en el navegador de Conexiones y seleccionando un elemento del menú:

* **Abrir** : muestra el texto del procedimiento para que pueda verlo y editarlo.
* **Ejecutar** : muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) , y luego ejecuta el procedimiento en modo normal (no depuración).
* **Compilar** : realiza una compilación PL / SQL del procedimiento.
* **Compilar para depuración** : realiza una compilación de PL / SQL del procedimiento, con las unidades de la biblioteca de PL / SQL compiladas para la depuración.
* **Depurar** : muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) y luego ejecuta el procedimiento en modo depuración.
* **Perfil** (para una conexión Oracle Database Versión 11.1 o posterior): muestra el cuadro de diálogo [Ejecutar / Depurar / Perfil PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHBIGH) , y luego ejecuta el procedimiento y recopila las estadísticas de ejecución.
* **Otorgar** : le permite otorgar privilegios en el procedimiento
* **Revocar** : le permite revocar privilegios en el procedimiento.
* **Drop** : borra el procedimiento.
* **Compilar dependientes** : realiza una compilación de PL / SQL del procedimiento y de los subprogramas dependientes relevantes (consulte la pestaña Dependencias).
* **Exportar DDL** : guarda la declaración DDL para crear el procedimiento en un archivo, una hoja de cálculo SQL o el portapapeles del sistema.

### Colas

Una cola es un objeto en el que los mensajes se ponen en cola y se retiran de la cola. Las colas son administradas por Oracle Streams Advanced Queuing (AQ). Para obtener información sobre el uso de colas, consulte *la Guía del usuario de Oracle Streams Advanced Queuing* .

### Tablas de cola

Una tabla de cola es una tabla que contiene los mensajes que se utilizarán con Oracle Streams Advanced Queuing (AQ). Para obtener información sobre el uso de tablas de cola, consulte *la Guía del usuario de Oracle Streams Advanced Queuing* , especialmente la información sobre la administración de tablas de cola en el capítulo que describe la interfaz administrativa de Oracle Streams AQ.

### Papelera de reciclaje

La Papelera de reciclaje (aplicable solo a Oracle Database Release *10g* y posteriores) contiene objetos que se han eliminado (eliminado). Los objetos no se eliminan realmente hasta que se realiza una operación de confirmación. Antes de eliminar realmente los objetos, puede "recuperarlos" seleccionándolos en la Papelera de reciclaje y seleccionando **Deshacer** en el menú contextual.

Puede realizar las siguientes operaciones en un objeto en la Papelera de reciclaje haciendo clic derecho en el nombre del objeto en la Papelera de reciclaje en el navegador de Conexiones y seleccionando un elemento del menú:

* **Purgar** : elimina el objeto de la Papelera de reciclaje y lo elimina.
* **Flashback to Before Drop** : Mueve el objeto de la Papelera de reciclaje a su lugar apropiado en la pantalla del navegador de Conexiones.

### Esquemas de replicación (base de datos en memoria de Oracle TimesTen)

Un esquema de replicación es una configuración que utiliza sentencias de SQL y un registro basado en transacciones, en el que los cambios confirmados se copian desde su origen a una o más bases de datos de suscriptores. El objetivo es permitir una alta eficiencia y una baja sobrecarga durante la replicación.

### Secuencias

Las secuencias se utilizan para generar enteros únicos. Puedes usar secuencias para generar automáticamente valores de clave primaria. Para obtener ayuda con las opciones específicas para crear y editar una secuencia, consulte [Crear / Editar secuencia](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJCCFJ) .

### Sinónimos (públicos y privados)

Los sinónimos proporcionan nombres alternativos para tablas, vistas, secuencias, procedimientos, funciones almacenadas, paquetes, vistas materializadas, objetos de base de datos de clase Java, tipos de objetos definidos por el usuario u otros sinónimos. El navegador de conexiones tenga un **Sinónimos** nodo para todos los sinónimos (públicos y privados) son propiedad del usuario asociado con la conexión especificada, y una **Públicas sinónimos** nodo para todos los sinónimos públicos sobre la base de datos asociada a la conexión. Para obtener ayuda con las opciones específicas para crear y editar un sinónimo, consulte [Crear / Editar sinónimo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCBFHJ) .

### Tablas

Las tablas se utilizan para mantener los datos. Normalmente, cada tabla tiene varias columnas que describen los atributos de la entidad de base de datos asociada con la tabla, y cada columna tiene un tipo de datos asociado. Puede elegir entre muchas opciones de creación de tablas y organizaciones de tablas (como tablas particionadas, tablas organizadas por índice y tablas externas), para satisfacer una variedad de necesidades empresariales. Para crear una tabla, puede hacer una de las siguientes acciones:

* Cree la tabla rápidamente agregando columnas y especificando las funciones de uso frecuente. Para hacer esto, *no marque* el cuadro Avanzado en el cuadro de diálogo Crear tabla. Para obtener ayuda con las opciones para crear una tabla utilizando este enfoque rápido, consulte [Crear tabla (creación rápida)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAEJI) .
* Cree la tabla agregando columnas y seleccionando de un conjunto más grande de características. Para hacer esto, *marque* la casilla Avanzado en el cuadro de diálogo Crear tabla. Para obtener ayuda con las opciones para crear una tabla con funciones avanzadas, consulte [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .
* Crea la tabla automáticamente desde una hoja de cálculo de Microsoft Excel. Para hacer esto, haga clic con el botón derecho en Tablas bajo una conexión en el navegador de Conexiones y seleccione **Importar datos** . Cuando se le solicite el archivo, seleccione un archivo de tipo .xls o .csv.

Puede realizar las siguientes operaciones en una tabla haciendo clic con el botón derecho en el nombre de la tabla en el navegador de Conexiones y seleccionando un elemento del menú:

* **Editar** : muestra el cuadro de diálogo [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .
* **Tabla** : las acciones de la tabla incluyen Renombrar, Copiar (crear una copia usando un nombre diferente), Eliminar (eliminar la tabla), Truncar (eliminar datos existentes sin afectar la definición de la tabla), Bloquear (configurar el modo de bloqueo de la tabla: compartir fila, exclusivo, y así sucesivamente), Comentario (comentario descriptivo que explica el uso o propósito de la tabla), Paralelo (cambie el grado predeterminado de paralelismo para consultas y DML en la tabla), No paralelo (especifique la ejecución en serie), y cuente las filas (devuelva el número de filas).
* **Columna** : las acciones de la columna incluyen Comentario (comentario descriptivo sobre una columna), Agregar, Eliminar y Normalizar.
* **Restricción** : incluye opciones para agregar, eliminar, habilitar y deshabilitar restricciones.
* **Índice** : las opciones incluyen Crear (crear un índice en columnas especificadas), Crear texto (crear un índice de texto de Oracle en una columna), Crear texto (crear un índice basado en funciones en una columna) y soltar.
* **Restricción** : las opciones incluyen Habilitar o Deshabilitar único, Eliminar (eliminar una restricción), Agregar cheque (agregar una restricción de verificación), Agregar clave externa y Agregar único.
* **Privilegios** : si está conectado como un usuario de base de datos con suficientes privilegios, puede otorgar o revocar privilegios en la mesa a otros usuarios.
* **Estadísticas** : las opciones incluyen Gather Statistics (calcular estadísticas exactas de tablas y columnas y almacenarlas en el diccionario de datos) y Validar la estructura (verifica la integridad de cada fila y bloque de datos, y para una tabla organizada por índices también genera el recuento de compresión de prefijos óptimo para el índice de clave principal en la tabla). El optimizador de la base de datos Oracle utiliza las estadísticas para elegir el plan de ejecución de las sentencias de SQL que acceden a los objetos analizados.
* **Almacenamiento** : las opciones incluyen Tabla de reducción (espacio de reducción en una tabla, para segmentos en espacios de tabla con administración automática de segmentos) y Tabla de movimiento (a otro espacio de tabla). Las opciones de Encoger tabla incluyen Compacto (solo desfragmenta el espacio del segmento y compacta las filas de la tabla para su posterior lanzamiento, pero no reajusta la marca de límite superior y no libera el espacio inmediatamente) y Cascada (realiza las mismas operaciones en todos los objetos dependientes del tabla, incluidos los índices secundarios en tablas organizadas por índice).
* **Desencadenador** : las opciones incluyen Crear, Crear PK desde Secuencia (crear un desencadenador de inserción anterior para completar la clave principal utilizando valores de una secuencia específica), Habilitar o Deshabilitar todo, Habilitar o Deshabilitar una tecla y Desechar (eliminar el desencadenante).
* **Importar datos** : le permite importar datos de una hoja de cálculo de Microsoft Excel (archivo .xls o .csv).
* **Exportar datos** : le permite exportar algunos o todos los datos de la tabla a un archivo o al portapapeles del sistema, en cualquiera de los siguientes formatos: XML (etiquetas y datos XML), CSV (valores separados por comas que incluyen una fila de encabezado para la columna) identificadores), SQL Insert (instrucciones INSERT) o SQL Loader (SQL \* Loader control file). Después de seleccionar un formato, se muestra el cuadro de diálogo [Exportar datos de tabla](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDEEDI) .

Puede realizar las siguientes operaciones en una columna de una tabla haciendo clic con el botón derecho en el nombre de la columna en el navegador de Conexiones y seleccionando un elemento del menú:

* **Renombrar** : **Renombra** la columna.
* **Descartar** : elimina la columna (incluidos todos los datos de esa columna) de la tabla.
* **Comentario** : Agrega un comentario descriptivo sobre la columna.
* **Cifrar** (para Oracle Database Versión 10.2 y superior, y solo si la función Cifrado de datos transparentes está habilitada para la base de datos): muestra un cuadro de diálogo en el que se especifica un algoritmo de cifrado compatible para usar para cifrar todos los datos en la columna. Los datos actuales y los datos insertados posteriormente están encriptados.
* **Descifrar** (para Oracle Database Release 10.2 y superior, y solo si la función Cifrado de datos transparentes está habilitada para la base de datos): descifra los datos en la columna que se había cifrado y hace que los datos que se insertan posteriormente no se cifren.
* **Normalizar** : crea una nueva tabla utilizando los valores distintos en la columna especificada. Debe especificar los nombres de la nueva tabla y su columna de clave principal, así como un nombre de secuencia y un nombre de activador.

Soporte de tabla de flashback

Para las tablas en Oracle Database Release 11.1 y posteriores, la visualización de la tabla incluye la pestaña **Flashback** , que proporciona una vista de los datos modificados y originales de la tabla. Si tiene los privilegios adecuados, puede hacer clic en la subpestaña **Deshacer de SQL** para seleccionar y revisar la sintaxis necesaria para deshacer los cambios. Para obtener información sobre el uso de la función de tabla de Flashback, consulte [*la Guía del usuario de copia de seguridad y recuperación de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/backup.112/e10642/toc.htm) .

### Triggers

Los activadores son bloques PL / SQL almacenados asociados con una tabla, un esquema o la base de datos, o bloques PL / SQL anónimos o llamadas a un procedimiento implementado en PL / SQL o Java. La base de datos Oracle ejecuta automáticamente un disparador cuando ocurren condiciones específicas. Para obtener ayuda con opciones específicas para crear un disparador, consulte [Crear disparador](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIGFBJ) .

### Los tipos

Un tipo de datos asocia un conjunto fijo de propiedades con los valores que se pueden usar en una columna de una tabla o en un argumento de una función o procedimiento. Estas propiedades hacen que Oracle Database trate los valores de un tipo de datos de manera diferente a los valores de otro tipo de datos. La mayoría de los tipos de datos son suministrados por Oracle, aunque los usuarios pueden crear tipos de datos.

Para obtener ayuda con las opciones específicas para crear un tipo definido por el usuario, consulte [Crear tipo (definido por el usuario)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEAEDJ) .

### Usuarios (Otros Usuarios)

Los usuarios de la base de datos son cuentas a través de las cuales puede iniciar sesión en la base de datos. En el navegador de Conexiones, puede ver los **Otros usuarios** en la base de datos asociada con una conexión, pero los objetos de la base de datos que puede ver para cada usuario están determinados por los privilegios del usuario de la base de datos asociado con la conexión actual de la base de datos.

Si está conectado como usuario con el rol DBA, puede crear un usuario de base de datos haciendo clic con el botón derecho en Otros usuarios y seleccionando **Crear usuario** , y puede editar un usuario de base de datos existente haciendo clic con el botón derecho del mouse en Otros usuarios y seleccionando **Editar usuario** . Para obtener ayuda sobre las opciones para crear y editar usuarios, consulte [Crear / Editar usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGEIDJ) .

### Vistas

Las vistas son tablas virtuales (análogas a las consultas en algunos productos de base de datos) que seleccionan datos de una o más tablas subyacentes. Oracle Database ofrece muchas opciones de creación de vistas y tipos de vistas especializadas (como las vistas materializadas, descritas en [Vistas materializadas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGAAAI) ), para satisfacer una variedad de necesidades empresariales. Para obtener ayuda con opciones específicas para crear y editar una vista, consulte [Crear / Editar vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJIGG) .

Puede realizar las siguientes operaciones en una vista haciendo clic con el botón derecho en el nombre de la vista en el navegador de Conexiones y seleccionando un elemento del menú:

* **Editar** : muestra el cuadro de diálogo [Crear / Editar vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJIGG) .
* **Drop** : borra la vista.
* **Compilar** : Recompila la vista, para permitirle localizar posibles errores antes del tiempo de ejecución. Es posible que desee recompilar una vista después de modificar una de sus tablas base para asegurarse de que el cambio no afecte a la vista u otros objetos que dependen de ella.

### Esquemas XML

Los esquemas XML son definiciones de esquema, escritas en XML, que describen la estructura y varias otras semánticas de los documentos XML de instancia conformes. Para obtener información conceptual y de uso sobre esquemas XML, consulte [*la Guía del desarrollador de Oracle XML DB*](https://docs.oracle.com/cd/E11882_01/appdev.112/e23094/toc.htm) en la biblioteca de documentación de la base de datos Oracle.

Puede editar un esquema XML haciendo clic con el botón derecho en el nombre del esquema XML en el navegador de Conexiones y seleccionando **Editar** en el menú.

### Objetos de base de datos capturados y convertidos (para migración)

Si está migrando una base de datos de terceros a Oracle, los navegadores Modelos capturados y Modelos convertidos pueden mostrar modelos que incluyen objetos de base de datos, como tablas y procedimientos. Un objeto capturado representa un objeto en la base de datos de terceros capturados, y un objeto convertido representa un modelo de Oracle de ese objeto tal como se creará en la base de datos de Oracle.

El menú contextual para cada objeto capturado incluye **Convertir a Oracle** , que crea un objeto convertido correspondiente. El menú contextual para cada objeto convertido incluye **Generar** , que crea el objeto de base de datos de Oracle correspondiente. (Los menús contextuales contendrán otros elementos según sea apropiado para el objeto.)

Para obtener información sobre los objetos relacionados de la base de datos Oracle, consulte lo siguiente:

* [Funciones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDEICJ)
* [Índices](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGABDH)
* [Procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBAIIB)
* [Secuencias](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIBBAC)
* [Mesas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEGJJD)
* [Gatillos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGEAEC)
* [Usuarios (Otros Usuarios)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDCCBGJ)
* [Puntos de vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGBHDE)

### Conexiones de base de datos

Una **conexión** es un objeto de desarrollador de SQL que especifica la información necesaria para conectarse a una base de datos específica como un usuario específico de esa base de datos. Debe tener al menos una conexión de base de datos (existente, creada o importada) para usar SQL Developer.

Puede conectarse a cualquier esquema de base de datos Oracle de destino mediante la autenticación estándar de la base de datos Oracle. Una vez conectado, puede realizar operaciones en objetos en la base de datos. También puede conectarse a esquemas para bases de datos de terceros (no Oracle) seleccionadas, como MySQL, Microsoft SQL Server, Sybase Adaptive Server y Microsoft Access, y ver metadatos y datos.

Cuando inicie SQL Developer y cada vez que muestre el cuadro de diálogo de conexiones de base de datos, SQL Developer automáticamente lee cualquier conexión definida en el archivo tnsnames.ora en su sistema, si ese archivo existe. De forma predeterminada, tnsnames.ora se encuentra en el directorio $ ORACLE\_HOME / network / admin, pero también puede estar en el directorio especificado por la variable de entorno TNS\_ADMIN o el valor del registro o (en sistemas Linux) el directorio de configuración global. En los sistemas Windows, si el archivo tnsnames.ora existe pero SQL Developer no está utilizando sus conexiones, defina TNS\_ADMIN como una variable de entorno del sistema. Para obtener información sobre el archivo tnsnames.ora, consulte el capítulo "Parámetros de nombres locales (tnsnames.ora)" en la *Referencia de servicios de la base de datos de Oracle* .

Puede crear conexiones adicionales (por ejemplo, para conectarse a la misma base de datos pero como usuarios diferentes, o para conectarse a bases de datos diferentes). Cada conexión de base de datos se enumera en la jerarquía del navegador de conexiones.

Para crear una nueva conexión de base de datos, haga clic con el botón derecho en el nodo Conexiones y seleccione **Nueva conexión de base de datos** . Use el cuadro de diálogo para especificar información sobre la conexión (vea [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) ). También puede crear una nueva conexión de base de datos seleccionando una conexión existente en ese cuadro de diálogo, cambiando el nombre de la conexión, cambiando otros atributos de conexión según sea necesario y haciendo clic en Guardar o Conectar.

Para editar la información sobre una conexión de base de datos existente, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Propiedades** . Use el cuadro de diálogo para modificar la información sobre la conexión, pero no cambie el nombre de la conexión. (Consulte [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) ).

Para organizar los grupos de conexión mediante carpetas, consulte [Uso de carpetas para agrupar conexiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDGIJ) .

Para exportar información sobre las conexiones de base de datos existentes a un archivo XML que puede usar más adelante para importar conexiones, haga clic con el botón derecho en Conexiones en la pantalla del navegador de Conexiones y seleccione **Exportar conexiones** . Use el cuadro de diálogo para especificar las conexiones que se exportarán (consulte [Exportar / Importar descriptores de conexión](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBDDIA) ).

Para importar las conexiones que se habían exportado anteriormente (agregándolas a cualquier conexión que ya exista en SQL Developer), haga clic con el botón derecho en Conexiones en la pantalla del navegador de Conexiones y seleccione **Importar conexiones** . Use el cuadro de diálogo para especificar las conexiones que se importarán (consulte [Exportar / Importar descriptores de conexión](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBDDIA) ).

Para realizar operaciones limitadas de administración de la base de datos si está conectado COMO SYSDBA, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Administrar base de datos** . Puede hacer clic para actualizar la visualización de solo lectura de la memoria (SGA y PGA) y la información del espacio de tabla. Si un escucha se está ejecutando con un escucha estático configurado para la base de datos, también puede hacer clic para iniciar y detener la base de datos.

Para realizar la depuración remota si está utilizando la arquitectura de depuración de la plataforma Java (JPDA) de Sun Microsystem y desea que el depurador escuche para que un depurador se pueda conectar al depurador, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Remoto depuración** . Use el cuadro de diálogo para especificar información de depuración remota (vea [Depurador - Adjuntar a JPDA](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHGIHJ) ).

Para estimar o calcular estadísticas de objetos en un esquema de base de datos, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Recopilar estadísticas de esquema** . Las estadísticas se utilizan para optimizar la ejecución de SQL.

Para generar documentación (en formato HTML (comparable a Javadoc para las clases de Java) sobre un esquema, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Connections y seleccione **Generar DB Doc** . Para ver la documentación generada, abra el archivo index.html en el directorio de salida que ha especificado.

Para cambiar el nombre de una conexión, haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Cambiar nombre de conexión** .

Para eliminar una conexión (es decir, eliminarla de SQL Developer, *no* simplemente desconectarse de la conexión actual), haga clic con el botón derecho en el nombre de la conexión en la pantalla del navegador de Conexiones y seleccione **Eliminar** . Eliminar una conexión no elimina el usuario asociado con esa conexión.

Para conectarse utilizando una conexión existente, expanda su nodo en el navegador de conexiones, o haga clic con el botón derecho en su nombre y seleccione **Conectar** . También se abre una ventana de la hoja de trabajo de SQL para la conexión (consulte [Uso de la hoja de trabajo de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) ).Para crear una hoja de cálculo no compartida separada para una conexión, use Ctrl + Shift + N.

Para desconectarse de la conexión actual, haga clic con el botón derecho en su nombre en el navegador de Conexiones y seleccione **Desconectar** .

Para especificar una preferencia para usar un controlador OCI (grueso, Tipo 2) (si está disponible) en lugar de un controlador JDBC (delgado) para conexiones de base de datos básicas y TNS (alias de red), habilite la opción **Usar controlador OCI / Thick** en la [Base de datos: Parámetros avanzados de](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFGCAC) preferencias del usuario.

### Uso de carpetas para agrupar conexiones

Puede usar las carpetas en el navegador de conexiones para organizar las conexiones en grupos: por ejemplo, una carpeta para las conexiones en su sistema local, otra para las conexiones en el sistema de prueba y otra para las conexiones en el sistema de producción.

Para crear una carpeta para mantener las conexiones, haga clic con el botón derecho en el nombre en el navegador de conexiones de una conexión que se agregará a la carpeta, seleccione **Agregar a la carpeta** y luego **Nueva carpeta** , y especifique el nombre de la carpeta (como Conexiones locales).

Para agregar más conexiones a una carpeta, haga clic con el botón derecho en el nombre en el navegador de conexiones de una conexión que se agregará a la carpeta, seleccione **Agregar a la carpeta** y luego el nombre de la carpeta en la que desea agregar la conexión.

Para mover una conexión de una carpeta a otra, haga clic con el botón derecho en el nombre de la conexión debajo de su carpeta actual, seleccione **Agregar a la carpeta** y luego el nombre de la carpeta de destino o **Nueva carpeta** para mover la conexión a una nueva carpeta que se creará .

Para eliminar una conexión de la carpeta, haga clic con el botón derecho en el nombre de la conexión debajo de la carpeta y seleccione **Eliminar de la carpeta** . (Esto *no* elimina la conexión; se mueve al nivel superior en la pantalla de jerarquía del navegador de Conexiones).

Para eliminar una carpeta, haga clic con el botón derecho en el nombre de la carpeta y seleccione **Eliminar carpeta** . (Esto *no* elimina ninguna conexión que esté en la carpeta; estas conexiones se mueven al nivel superior en la pantalla de jerarquía del navegador de conexiones).

Para cambiar el nombre de una carpeta, haga clic con el botón derecho en el nombre de la carpeta, seleccione **Cambiar nombre de carpeta**y especifique el nuevo nombre.

### Intercambio de conexiones

De forma predeterminada, cada conexión en SQL Developer se comparte cuando es posible. Por ejemplo, si abre una tabla en el navegador de conexiones y dos hojas de trabajo de SQL que usan la misma conexión, los tres paneles usan una conexión compartida a la base de datos. En este ejemplo, una operación de confirmación en una hoja de cálculo SQL se realiza en los tres paneles. Si desea una sesión dedicada, debe duplicar su conexión y darle otro nombre. Las sesiones se comparten por nombre, no por información de conexión, por lo que esta nueva conexión se mantendrá separada de la original.

### Seguridad avanzada para la conexión JDBC a la base de datos

Se recomienda utilizar Oracle Advanced Security para asegurar una conexión JDBC a la base de datos. Tanto los controladores JDBC OCI como JDBC Thin son compatibles con al menos algunas de las funciones de seguridad avanzada de Oracle. Si está utilizando el controlador OCI, puede establecer parámetros relevantes de la misma manera que lo haría en cualquier configuración de cliente de Oracle. El controlador JDBC Thin es compatible con las funciones de seguridad avanzada de Oracle a través de un conjunto de clases Java incluidas con las clases JDBC en un archivo de archivo Java (JAR) y admite la configuración de parámetros de seguridad a través de los objetos de propiedades Java.

Para obtener más información sobre el uso de Oracle Advanced Security, consulte [*la Guía y referencia para desarrolladores de Oracle Database JDBC*](https://docs.oracle.com/cd/E11882_01/java.112/e16548/toc.htm) .

### Conexiones con autenticación de sistema operativo (SO)

Cuando crea una conexión a una base de datos Oracle que utiliza la autenticación del sistema operativo (SO), puede omitir el nombre de usuario y la contraseña; es decir, especifique un nombre de conexión y toda la otra información necesaria, excepto que no especifique un nombre de usuario o contraseña. Para obtener información sobre el uso de la autenticación externa, incluido el uso de los parámetros de inicialización de la base de datos OS\_AUTHENT\_PREFIX y REMOTE\_OS\_AUTHENT, consulte *la Guía de seguridad de la base de datos Oracle* .

Si omite el nombre de usuario y la contraseña que intentan crear una conexión a un sistema que no está configurado para la autenticación externa, se muestra un mensaje de error.

### Conexiones con autenticación de proxy

La autenticación de proxy permite que una conexión JDBC actúe como un proxy para otras conexiones JDBC. Si utiliza la opción Conexión de proxy cuando crea una conexión de base de datos, la conexión se utilizará para conectarse como el usuario especificado para la conexión, pero se autentificará utilizando el nombre de usuario y la contraseña o el nombre distinguido del usuario proxy. Para obtener información sobre el uso de un servidor de nivel medio para la autenticación proxy, consulte *la Guía de seguridad de la base de datos Oracle* .

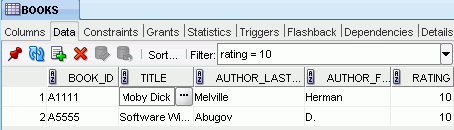
Por ejemplo, para crear una conexión para un usuario llamado PROXY\_USER pero conectándose con el nombre de usuario y la contraseña del usuario de la base de datos existente SCOTT, siga estos pasos.

1. Cree el usuario proxy y concédale los privilegios apropiados:
2. CREAR USUARIO proxy\_user IDENTIFICADO POR <contraseña>;
3. ALTER USER proxy\_user GRANT CONECTAR A TRAVÉS DE scott AUTENTICADO USANDO LA CONTRASEÑA;
4. GRANT crea sesión TO proxy\_user;
5. . . . <Concede otros privilegios según sea necesario.>
6. Crear una nueva conexión de base de datos. Por ejemplo: nombre de conexión = proxy\_conn, nombre de usuario = scott, contraseña = *<contraseña para scott>* .
7. Habilitar (marcar) la **Conexión Proxy** .
8. En el cuadro de diálogo Conexión de proxy de Oracle, seleccione **Nombre de usuario** para Tipo de proxy
9. Para el Usuario Proxy, ingrese **PROXY\_USER** ; y para Proxy Password, ingrese la contraseña para el usuario de la base de datos PROXY\_USER.
10. Haga clic en Aceptar para cerrar el cuadro de diálogo Conexión de proxy de Oracle.
11. Complete cualquier otra información de conexión necesaria y haga clic en Conectar para crear la conexión.

En este ejemplo, cuando se conecta utilizando la conexión proxy\_conn, el nombre de usuario y la contraseña del usuario SCOTT se usan para conectarse a la base de datos, pero la conexión ve los objetos de la base de datos que son propiedad del esquema PROXY\_USER.

### Entrar y modificar datos

Puede usar SQL Developer para introducir datos en tablas y editar y eliminar datos de tablas existentes. Para realizar cualquiera de estas operaciones, seleccione la tabla en el navegador de Conexiones, luego haga clic en la pestaña **Datos** en la pantalla de detalles de la tabla. La siguiente figura muestra el panel de Datos para una tabla llamada BOOKS, con un filtro aplicado para mostrar solo los libros cuya calificación es 10, y luego de que el usuario haya hecho clic en la celda de Título del primer libro.



Los iconos y otros controles en la pestaña Datos proporcionan las siguientes opciones:

* **Congelar vista** (el **pin** ) mantiene la pestaña y la información de ese objeto en la ventana cuando hace clic en otro objeto en el navegador de Conexiones; se crean una pestaña y una pantalla separadas para ese otro objeto. Si vuelve a hacer clic en el pin, la pantalla del objeto está disponible para su reutilización.
* **Actualizar** consulta la base de datos para actualizar la visualización de datos. Si se especifica un filtro, la operación de actualización utiliza el filtro.
* **Insertar fila** agrega una fila vacía después de la fila seleccionada, para que ingrese nuevos datos.
* **Eliminar fila (s) seleccionada (s)** marca las filas seleccionadas para eliminarlas. La eliminación real no se produce hasta que confirme los cambios.
* **Commit Changes** finaliza la transacción actual y hace permanentes todos los cambios realizados en la transacción.
* **Los cambios de reversión** deshacen cualquier trabajo realizado en la transacción actual.
* **Ordenar** muestra un cuadro de diálogo para seleccionar columnas para ordenar. Para cada columna, puede especificar el orden ascendente o descendente, y puede especificar que los valores nulos se muestren primero.
* **El filtro le** permite ingresar un predicado SQL (texto de la cláusula WHERE sin la palabra clave WHERE) para limitar la visualización de los datos. Por ejemplo, para mostrar solo las filas donde el valor de la columna RATING es igual a 10, especifique:rating = 10
* **Acciones** muestra un menú con acciones relevantes para la tabla.

Cuando ingresa una celda en la cuadrícula, puede editar directamente los datos para muchos tipos de datos, y para todos los tipos de datos puede hacer clic en el botón de puntos suspensivos (...) para editar los datos. Para datos binarios no puede editar los datos en la celda, pero debe usar el botón de puntos suspensivos.

En la cuadrícula de datos, el menú contextual (clic con el botón derecho) incluye los siguientes comandos:

* **La vista de registro único** muestra el cuadro de diálogo [Vista de registro único](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDGFFH) , que le permite editar datos para una tabla o vista, un registro a la vez.
* **Ajustar automáticamente todas las columnas** ajusta el ancho de todas las columnas según su especificación (por encabezado de columna, por datos de columna o por el mejor ajuste).
* **Ajustar automáticamente las columnas seleccionadas** ajusta el ancho de las columnas seleccionadas de acuerdo con su especificación (por encabezado de columna, por datos de columna o por el mejor ajuste).
* **Contar filas** muestra el número de filas en la tabla.
* **Publicar en Apex** (si se instala Application Express) crea una pequeña aplicación Application Express basada en los datos. Muestra un cuadro de diálogo en el que se especifica lo siguiente para la aplicación que se creará: espacio de trabajo, nombre de la aplicación, tema, nombre de la página y declaración SQL para generar el informe.
* **Exportar datos le** permite exportar algunos o todos los datos de la tabla a un archivo o al portapapeles del sistema, en cualquiera de los siguientes formatos: XML (etiquetas y datos XML), CSV (valores separados por comas que incluyen una fila de encabezado para los identificadores de columna) ), SQL Insert (instrucciones INSERT), o SQL Loader (SQL \* Loader control file). Después de seleccionar un formato, se muestra el cuadro de diálogo [Exportar datos de tabla](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDEEDI) .

Puede copiar y pegar datos entre celdas de celdas de tabla y celdas en una hoja de cálculo de Microsoft Excel.

Para copiar los datos de la tabla al portapapeles, haga clic en el encabezado de la columna (para todos los datos de la columna) o seleccione celdas específicas y presione Ctrl + C; para copiar el texto del encabezado de la columna junto con los datos de la tabla, presione Ctrl + Mayús + C.

Para ordenar la visualización de filas por valores dentro de una columna, haga doble clic en el encabezado de la columna; para cambiar entre la clasificación ascendente y descendente, haga doble clic en la flecha arriba / abajo en el encabezado de la columna.

En el panel Datos de una tabla o vista, puede **dividir** la pantalla vertical u horizontalmente para ver dos (o más) partes de manera independiente al usar el cuadro dividido (rectángulo azul fino), ubicado a la derecha de la barra de desplazamiento inferior y sobre la barra de desplazamiento inferior. barra de desplazamiento derecha

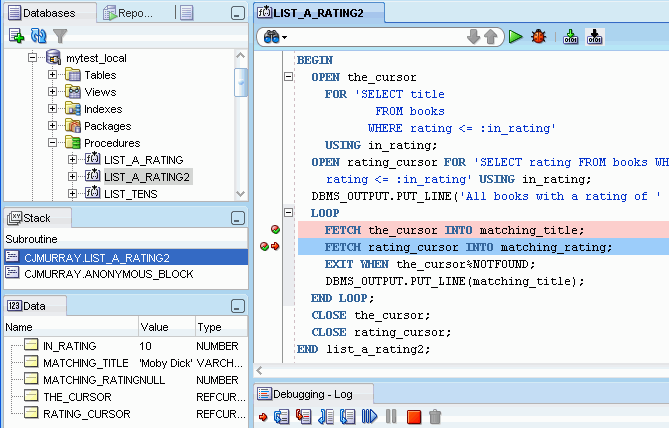
En el panel de Datos, el formato o formatos aceptables para ingresar fechas pueden ser diferentes del formato de fecha requerido por SQL \* Plus.

### Ejecutar y depurar funciones y procedimientos

Puede usar SQL Developer para ejecutar y depurar subprogramas PL / SQL (funciones y procedimientos).

* Para ejecutar un subprograma, haga clic en su nombre en el navegador de Conexiones; luego haga clic con el botón derecho y seleccione Ejecutar, o haga clic en el ícono Editar y luego haga clic en el ícono Ejecutar sobre su lista de fuentes.
* Para depurar un subprograma, haga clic en su nombre en el navegador de Conexiones. Si el procedimiento en su forma actual aún no se ha compilado para la depuración, haga clic con el botón derecho y seleccione Compilar para la depuración. Luego haga clic en el ícono Editar y haga clic en el ícono Depurar sobre su lista de fuentes.

En ambos casos, se muestra una ventana de edición de código. La siguiente figura muestra la ventana de edición de código que se usa para depurar un procedimiento llamado LIST\_A\_RATING2, que se usa con fines de tutoría en [Depurar un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACAEJG) .



En la ventana de edición de código, debajo de la pestaña con el nombre del subprograma, hay una barra de herramientas, y debajo de ella está el texto del subprograma, que puede editar. Puede establecer y anular puntos de interrupción para la depuración haciendo clic a la izquierda de la línea vertical delgada al lado de cada declaración con la que desea asociar un punto de interrupción. (Cuando se establece un punto de interrupción, se muestra un círculo rojo.)

La barra de herramientas debajo de la pestaña para el nombre del subprograma tiene una barra de herramientas que incluye los íconos que se muestran en la siguiente figura.

Iconos de la barra de herramientas de la pestaña Fuente

* **Ejecutar** inicia la ejecución normal del subprograma y muestra los resultados en la pestaña **Ejecución - Registro** .
* **La depuración** inicia la ejecución del subprograma en el modo de depuración y muestra la pestaña **Registro de depuración** , que incluye la barra de herramientas de depuración para controlar la ejecución.
* **La compilación** realiza una **compilación** PL / SQL del subprograma.
* **Compile for Debug** realiza una compilación de PL / SQL del subprograma para que pueda ser depurado.

La pestaña **Debugging - Log** en el área de texto del código contiene la barra de herramientas de depuración y los mensajes informativos. La barra de herramientas de depuración tiene los iconos que se muestran en la siguiente figura.

Iconos de la barra de herramientas de depuración

* **Encontrar punto de ejecución** va al punto de ejecución (la siguiente línea de código fuente que ejecutará el depurador).
* **Over Over** pasa por alto el siguiente subprograma (a menos que el subprograma tenga un punto de interrupción) y pasa a la siguiente declaración después del subprograma. Si el punto de ejecución se encuentra en una llamada de subprograma, ejecuta ese subprograma sin detenerse (en lugar de entrar en él), luego coloca el punto de ejecución en la declaración que sigue a la llamada. Si el punto de ejecución se encuentra en la última declaración de un subprograma, Paso sobre regresa del subprograma, colocando el punto de ejecución en la línea de código que sigue a la llamada al subprograma desde el cual está regresando.
* **Step Into** ejecuta una sola instrucción de programa a la vez. Si el punto de ejecución se encuentra en una llamada a un subprograma, Paso a paso en ese subprograma y coloca el punto de ejecución en su primera declaración. Si el punto de ejecución se encuentra en la última declaración de un subprograma, Paso a retorno regresa del subprograma, colocando el punto de ejecución en la línea de código que sigue a la llamada al subprograma desde el cual está regresando.
* **Salir** deja el subprograma actual y pasa a la siguiente declaración.
* **Step to End of Method** va a la última declaración del subprograma actual.
* **Reanudar** continúa la ejecución.
* **La pausa** detiene la ejecución pero no sale, lo que le permite reanudar la ejecución.
* **Terminar** detiene y sale de la ejecución. No puede reanudar la ejecución desde este punto; en su lugar, para comenzar a ejecutar o depurar desde el principio del subprograma, haga clic en el icono Ejecutar o Depurar en la barra de herramientas de la pestaña Fuente.

La pestaña **Puntos de** interrupción muestra puntos de interrupción, tanto definidos por el sistema como definidos por el usuario.

La pestaña **Smart Data** muestra información sobre las variables, usando su [Depurador:](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBADII) preferencias de [Smart Data](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBADII) . También puede especificar estas preferencias haciendo clic derecho en la ventana Smart Data y seleccionando **Preferencias** .

La pestaña **Datos** muestra información sobre las variables, utilizando su [Depurador:](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGCJDD) Preferencias de [datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGCJDD) . También puede especificar estas preferencias haciendo clic derecho en la ventana de Datos y seleccionando **Preferencias** .

La pestaña **Relojes** muestra información sobre los relojes (consulte [Configuración de relojes de expresión](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDJEGC) ).

Si la función o el procedimiento que se va a depurar está en un sistema remoto, vea también [Depuración remota](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDCIEC) .

### Uso de marcadores al editar funciones y procedimientos

Cuando edita una función o procedimiento largo, puede que le resulte conveniente crear marcadores en el código para que pueda navegar fácilmente a los puntos de interés.

Para crear o eliminar un marcador, haga clic en **Navegar** , luego en **Activar marcador** . Cuando se crea un marcador, aparece un icono a la izquierda de la delgada línea vertical.

Para ir a un marcador específico, haga clic en **Navegar** , luego en **Ir a marcador** . Para ir al marcador siguiente o anterior, haga clic en **Navegar** , luego **Ir al siguiente marcador** o **Ir al marcador anterior** , respectivamente.

Para eliminar todos los marcadores de la ventana de edición actualmente activa para una función o procedimiento o de todas las ventanas de edición abiertas, haga clic en **Navegar** , luego **Eliminar marcadores de archivo** o **Eliminar todos los marcadores** , respectivamente.

También puede ir a una línea específica oa su última edición haciendo clic en **Navegar** , luego **Ir a línea** o **Ir a Última edición** , respectivamente.

### Depuración remota

Para depurar una función o procedimiento para una conexión en la que la base de datos se encuentra en un host diferente al que está ejecutando SQL Developer, puede realizar una depuración remota. La depuración remota implica muchos de los pasos para la depuración local; sin embargo, haga lo siguiente antes de iniciar la depuración remota:

1. Utilice un cliente de Oracle como SQL \* Plus para emitir el comando de conexión del depurador. Independientemente del cliente que utilice, asegúrese de que la sesión que emite los comandos de conexión del depurador sea la misma sesión que ejecuta su programa PL / SQL que contiene los puntos de interrupción. Por ejemplo, si el nombre del sistema remoto es remoto1, use el siguiente comando SQL \* Plus para abrir una conexión TCP / IP a ese sistema y el puerto para la sesión JDWP:
2. EXEC DBMS\_DEBUG\_JDWP.CONNECT\_TCP ('remote1', '4000');

El primer parámetro es la dirección IP o el nombre de host del sistema remoto, y el segundo parámetro es el número de puerto en ese sistema remoto en el que el depurador está escuchando.

1. Haga clic con el botón derecho en la conexión de la base de datos remota, seleccione **Depuración remota** y complete la información en el cuadro de diálogo [Depurar - Adjuntar a JPDA](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHGIHJ) .

Luego, siga los pasos que seguiría para la depuración local (por ejemplo, vea [Depurar un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACAEJG) ).

### Visualización de archivos de seguimiento de SQL (.trc)

Si tiene archivos de salida de seguimiento de SQL (.trc), puede mostrarlos en el Desarrollador de SQL como una alternativa al uso del programa TKPROF para dar formato al contenido del archivo de seguimiento. Para abrir un archivo .trc en SQL Developer y ver una visualización atractiva y efectiva de la información, haga clic en **Archivo** , luego en **Abrir** , y especifique el archivo; o arrastre el nombre o el icono del archivo a la ventana del desarrollador de SQL.

Luego puede examinar la información en los paneles Vista de lista, Vista de estadísticas e Historial, con cada panel que incluye opciones para filtrar y controlar la pantalla.

Para obtener información sobre SQL Trace y TKPROF, consulte la [*Guía de ajuste de rendimiento de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/sqltrace.htm#PFGRF01010) .

### Usando el Perfilador Jerárquico PL / SQL

Para una conexión Oracle Database Release 11.1 o posterior, puede usar el perfilador jerárquico PL / SQL para identificar cuellos de botella y oportunidades de ajuste de rendimiento en aplicaciones PL / SQL. La creación de perfiles consta de dos pasos: ejecutar el módulo PL / SQL en modo de creación de perfiles y analizar los informes. Además, se requiere un trabajo de configuración de una sola vez la primera vez que use la creación de perfiles en SQL Developer.

Para iniciar la creación de perfiles, haga clic con el botón derecho en el nombre de la función o el procedimiento en la jerarquía del navegador de Conexiones y seleccione **Perfil** , o haga clic en el botón Perfil en la barra de herramientas del editor de fuentes PL / SQL. Después de ejecutar la función o el procedimiento en el modo de creación de perfiles, los informes del generador de perfiles se encuentran en la pestaña Perfiles de ejecución de la ventana del visor de objetos. Puede revisar la información de resumen de ejecución a nivel de subprograma, como:

* Número de llamadas al subprograma.
* Tiempo empleado en el propio subprograma (tiempo de función o tiempo libre)
* Tiempo empleado en el subprograma en sí y en sus subprogramas descendientes (tiempo de subárbol)
* Información detallada de padres e hijos, incluidos todos los subprogramas a los que llamó un subprograma dado (es decir, hijos del subprograma dado)

Para obtener más información sobre el uso del perfilador jerárquico PL / SQL, consulte [*la Guía del desarrollador de aplicaciones avanzadas de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_profiler.htm#ADFNS023) .

### Configuración de relojes de expresión

Un reloj le permite monitorear los valores cambiantes de las variables o expresiones a medida que se ejecuta su programa. Después de ingresar una expresión de observación, la ventana Relojes muestra el valor actual de la expresión. A medida que su programa se ejecuta, el valor del reloj cambia a medida que su programa actualiza los valores de las variables en la expresión del reloj.

Un reloj evalúa una expresión de acuerdo con el contexto actual que está controlado por la selección en la ventana Pila. Si te mueves a un nuevo contexto, la expresión se vuelve a evaluar para el nuevo contexto. Si el punto de ejecución se mueve a una ubicación donde alguna de las variables en la expresión de observación no está definida, la expresión de observación completa se vuelve indefinida. Si el punto de ejecución regresa a una ubicación donde se puede evaluar la expresión de observación, la ventana de Vigilancia muestra nuevamente el valor de la expresión de observación.

Para abrir la ventana de Relojes, haga clic en **Ver** , luego en **Depurador** , luego en **Relojes** .

Para agregar un reloj, haga clic con el botón derecho en la ventana Relojes y seleccione **Agregar reloj** . Para editar un reloj, haga clic con el botón derecho en la ventana Relojes y seleccione **Editar reloj** .

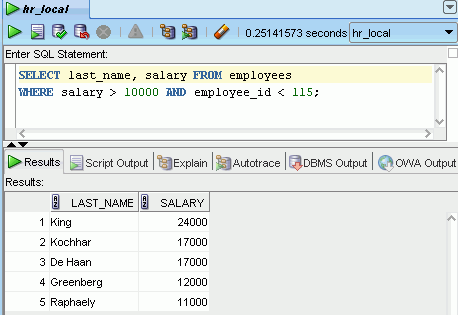
### Usando la hoja de cálculo SQL

Puede usar la hoja de trabajo de SQL para ingresar y ejecutar sentencias de SQL, PL / SQL y SQL \* Plus. Puede especificar cualquier acción que pueda ser procesada por la conexión de la base de datos asociada con la hoja de trabajo, como crear una tabla, insertar datos, crear y editar un disparador, seleccionar datos de una tabla y guardar esos datos en un archivo.

Puede mostrar una hoja de trabajo SQL haciendo clic con el botón derecho en una conexión en el navegador de Conexiones y seleccionando **Abrir hoja de trabajo SQL** , seleccionando **Herramientas** y luego la **hoja de trabajo SQL** , o haciendo clic en el ícono **Usar hoja de trabajo SQL** debajo de la barra de menú. En el cuadro de diálogo Seleccionar conexión, seleccione la conexión de base de datos que usará para su trabajo con la hoja de trabajo. También puede usar ese cuadro de diálogo para crear y editar conexiones de base de datos. (Puede abrir una ventana de la hoja de trabajo SQL automáticamente cuando abre una conexión de base de datos habilitando las preferencias de usuario de desarrollador de SQL apropiadas en Conexiones de base de datos).

Para crear una hoja de cálculo no compartida separada para una conexión, use Ctrl + Shift + N.

La hoja de trabajo SQL tiene la interfaz de usuario que se muestra en la siguiente figura:



**Barra de herramientas de** la hoja de trabajo SQL (en la pestaña de la hoja de trabajo SQL): contiene iconos para las siguientes operaciones:

* **Execute Statement** ejecuta la instrucción en el puntero del mouse en el cuadro Enter SQL Statement. Las sentencias de SQL pueden incluir variables de enlace y variables de sustitución de tipo VARCHAR2 (aunque en la mayoría de los casos, VARCHAR2 se convierte automáticamente a NÚMERO si es necesario); se muestra un cuadro emergente para ingresar valores variables.
* **Ejecutar secuencia de comandos** ejecuta todas las declaraciones en el cuadro Ingresar instrucción SQL utilizando el [Script Runner](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIJIEF) . Las instrucciones SQL pueden incluir variables de sustitución (pero no variables de enlace) de tipo VARCHAR2 (aunque en la mayoría de los casos, VARCHAR2 se convierte automáticamente en NÚMERO si es necesario); se muestra un cuadro emergente para ingresar valores de variables de sustitución.
* **Confirmar** escribe cualquier cambio en la base de datos y finaliza la transacción; también borra cualquier salida en los paneles Resultados y Script de salida.
* **La reversión** descarta cualquier cambio sin escribirlos en la base de datos y finaliza la transacción; también borra cualquier salida en los paneles Resultados y Script de salida.
* **Cancelar** detiene la ejecución de cualquier sentencia que se esté ejecutando actualmente.
* **Monitor SQL Status** (Oracle Database versión 11.1 y posterior solamente) llama a la función de monitoreo de SQL en tiempo real de Oracle Database, lo que le permite monitorear el rendimiento de las sentencias de SQL mientras se ejecutan.
* **Execute Explain Plan** genera el plan de ejecución para la sentencia (ejecutando internamente la sentencia EXPLAIN PLAN). Para ver el plan de ejecución, haga clic en la pestaña Explicar. Para más información, ver [Plan de Ejecución](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEJJDB) .
* **Autotrace** genera información de rastreo para la declaración. Para ver el plan de ejecución, haga clic en la pestaña Autotrace. Para más información, ver [Plan de Ejecución](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEJJDB) .
* **Borrar** borra la declaración o declaraciones en el cuadro Ingresar instrucción SQL.
* A la derecha de estos iconos hay una lista desplegable para cambiar la **conexión de** la **base de datos** para usar con la hoja de trabajo.

El menú contextual (clic con el botón derecho o Mayús + F10) incluye las operaciones de la barra de herramientas de la hoja de trabajo de SQL anteriores, más las siguientes operaciones:

* **Imprimir archivo** imprime el contenido del cuadro Introducir declaración SQL.
* **Cortar** , **Copiar** , **Pegar** y **Seleccionar todo** tienen los mismos significados que para las operaciones de edición de texto normales.
* **Query Builder** abre el cuadro de diálogo [Query Builder](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm" \l "BACCEAIG) , donde puede crear una declaración SELECT arrastrando y soltando nombres de tablas y vistas y especificando gráficamente columnas y otros elementos de la consulta.
* **La refactorización le** permite hacer lo siguiente en el texto seleccionado: cambiar el caso del carácter (a superior / inferior / initcap), extraer la secuencia de sentencias PL / SQL a un procedimiento, o cambiar el nombre de la variable local.
* **Formato** formatea la declaración SQL (en mayúsculas los nombres de declaraciones, cláusulas, palabras clave, etc.).
* **Ventana emergente Describe** , si el nombre de un objeto de la base de datos está completamente seleccionado, muestra una ventana con pestañas e información apropiada para ese tipo de objeto (ver [Ventana Describir Objeto](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJFFDI) ).
* **Guardar fragmento** abre el cuadro de diálogo [Guardar fragmento (definido por el usuario)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBADJA) con el texto seleccionado como texto de fragmento.

**Ingrese la declaración SQL** : la declaración o declaraciones que pretende ejecutar. Para varias declaraciones, cada instrucción que no sea PL / SQL debe terminar con un punto y coma o (en una nueva línea) una barra (/), y cada instrucción PL / SQL debe terminar con una barra (/) en una nueva línea . Las palabras clave de SQL se resaltan automáticamente. Para formatear la declaración, haga clic con el botón derecho en el área de la declaración y seleccione **Formatear SQL** .

Puede arrastrar algunos tipos de objetos desde el navegador de Connections y soltarlos en el cuadro Enter SQL Statement:

* Si arrastra y suelta una tabla o vista, se construye una instrucción SELECT con todas las columnas en la tabla o vista. Luego puede editar la declaración, por ejemplo, modificando la lista de columnas o agregando una cláusula WHERE.
* Si arrastra y suelta una función o un procedimiento, se construye un bloque de texto tipo fragmento de código para que lo edite al incluir ese objeto en una declaración.

Para ver los detalles de cualquier objeto, puede seleccionar su nombre en el cuadro Ingresar declaración SQL y seleccionar **Desplegar emergente** en el menú contextual (o presionar Mayús + F4). Por ejemplo, si selecciona un nombre de tabla y presiona Mayús + F4, se muestra información sobre Columnas, Restricciones, Subvenciones, etc. o si selecciona un nombre de procedimiento y presiona Mayús + F4, se muestra información sobre Código, Subvenciones, Dependencias, Referencias y Detalles.

**Las pestañas** muestran paneles con la siguiente información:

* **Resultados** : muestra los resultados de la operación Execute Statement más reciente.
* **Explicar** : muestra el resultado si hizo clic en el icono Explicar plan de ejecución (ver [Plan de ejecución](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEJJDB) ).
* **Salida de secuencia de comandos** : muestra la salida si hizo clic en el icono Ejecutar secuencia de comandos (consulte [Corredor de secuencias de comandos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIJIEF) ).
* **Salida de DBMS** : muestra la salida de las declaraciones del paquete DBMS\_OUTPUT (ver [Panel de salida de DBMS](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHJEGF) ).
* **Salida de OWA** : muestra la **salida del** Agente web de Oracle (MOD\_PLSQL) (vea el [Panel de salida de OWA](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFEICH) ).

Para alternar las alturas relativas del área Ingresar instrucción SQL y el área para pestañas y paneles de visualización, presione Ctrl + Alt + L. También puede ajustar manualmente las alturas.

### Declaraciones SQL \* Plus admitidas y no admitidas en la hoja de cálculo SQL

La hoja de trabajo de SQL admite algunas sentencias de SQL \* Plus. Las sentencias de SQL \* Plus deben ser interpretadas por la hoja de trabajo de SQL antes de pasarlas a la base de datos; cualquier SQL \* Plus que no sea compatible con la hoja de cálculo de SQL se ignora y no se pasa a la base de datos.

Las siguientes sentencias de SQL \* Plus son compatibles con la hoja de trabajo de SQL:

@

@@

aceptar]

conectar]

claro]

definir]

describir]

documento]

ejecutar]

exit (detiene la ejecución y restablece la conexión especificada)

anfitrión]

pausa]

rápido]

quit (detiene la ejecución y restablece la conexión especificada)

observación]

set pau [se] {ON | APAGADO}

comienzo]

sincronización]

undef [ine]

cuando

xquery

Las siguientes sentencias de SQL \* Plus *no* son compatibles con la hoja de trabajo de SQL:

adjuntar]

archivo

atributo]

descanso]

bti [tle]

cambio]

col [ulmn]

calcular]

dupdo

del

desconectar]

editar]

obtener

ayuda

entrada]

lista]

nueva pagina

oradebug

contraseña]

impresión

correr]

recuperar

repf [ooter]

reph [eader]

salvar]

espectáculo]

apagar]

carrete]

puesta en marcha

almacenar

tti [tle]

variable]

### Script Runner

El corredor de secuencias de comandos emula un conjunto limitado de características de SQL \* Plus. A menudo, puede ingresar sentencias de SQL y SQL \* Plus y ejecutarlas haciendo clic en el ícono **Ejecutar script** . El panel de salida de script muestra la salida.

Las características de SQL \* Plus disponibles en el corredor de secuencias de comandos incluyen @, @@, CONECTAR, SALIR, SALIR, DEFINICIÓN, SIEMPRE y variables de sustitución. Por ejemplo, para ejecutar un script llamado c: \ myscripts \ mytest.sql, escriba @c: \ myscripts \ mytest en el cuadro Ingrese la declaración SQL, y haga clic en el menú desplegable junto al icono Execute Statement y seleccione Ejecutar script.

Las siguientes consideraciones se aplican al uso del corredor de scripts SQL Developer:

* No puede utilizar variables de enlace. (La función Execute SQL Statement le permite usar variables de enlace de tipo VARCHAR2, NUMBER y DATE).
* Para las variables de sustitución, la sintaxis && asigna un valor de variable permanente, y la sintaxis y la variable asigna un valor de variable temporal (no almacenado).
* Para EXIT y QUIT, commit es el comportamiento predeterminado, pero puede especificar rollback. En cualquier caso, el contexto se restablece: por ejemplo, se borran la información del comando WHENEVER y los valores de las variables de sustitución.
* DESCRIBIR funciona para la mayoría de los tipos de objetos, pero no para todos, para los que se admite en SQL \* Plus.
* Para los comandos de SQL \* Plus que no son compatibles, se muestra un mensaje de advertencia.
* Los comentarios de SQL \* Plus se ignoran.
* Para los datos XMLType, los datos en la columna se muestran como "SYS.XMLDATA" si la conexión de la base de datos usa un controlador JDBC Thin, pero los valores XML expandidos se muestran si la conexión usa un controlador OCI (grueso, Tipo 2).

Si tiene SQL \* Plus disponible en su sistema, es posible que desee utilizarlo en lugar del ejecutor de scripts.

### Plan de ejecución

El icono Ejecutar plan de explicación genera el plan de ejecución, que puede ver haciendo clic en la pestaña Explicar. El plan de ejecución es la secuencia de operaciones que se realizarán para ejecutar la instrucción. Un plan de ejecución muestra un árbol de origen de fila con la jerarquía de operaciones que conforman la declaración. Para cada operación, muestra el orden de las tablas a las que hace referencia la declaración, el método de acceso para cada tabla mencionada en la declaración, el método de combinación para las tablas afectadas por las operaciones de unión en la declaración y las operaciones de datos como el filtro, la clasificación o la agregación.

Además del árbol de origen de la fila, la tabla del plan muestra información sobre la optimización (como el costo y la cardinalidad de cada operación), la partición (como el conjunto de particiones a las que se accede) y la ejecución paralela (como el método de distribución de entradas combinadas). ). Para obtener más información, consulte el capítulo sobre el uso de EXPLAIN PLAN en la *Guía de ajuste de rendimiento de la base de datos de Oracle* .

### Panel de autotrace

El panel Autotrace muestra información relacionada con la traza cuando ejecuta la instrucción SQL haciendo clic en el icono **Autotrace** . La mayor parte de la información específica que se muestra está determinada por las [Preferencias del desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) para la [base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIJCCC) de [datos: parámetros de seguimiento automático](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIJCCC) . Si cancela una declaración de larga duración, se muestran las estadísticas de ejecución parcial.

Esta información puede ayudarlo a identificar las sentencias de SQL que se beneficiarán de la optimización. Por ejemplo, es posible que pueda optimizar el manejo de los predicados agregando de forma transitiva, reescribiendo los predicados utilizando los principios del álgebra booleana, moviendo los predicados en el plan de ejecución, etc. Para obtener más información sobre el seguimiento y el seguimiento automático, consulte el capítulo sobre el ajuste en [*la Guía del usuario y referencia de SQL \* Plus*](https://docs.oracle.com/cd/E11882_01/server.112/e16604/ch_eight.htm#SQPUG018) .

Para usar la función de seguimiento automático, el usuario de la base de datos para la conexión debe tener el privilegio SELECT\_CATALOG\_ROLE.

### Panel de salida DBMS

El paquete PL / SQL DBMS\_OUTPUT le permite enviar mensajes desde procedimientos almacenados, paquetes y activadores. Los procedimientos PUT y PUT\_LINE en este paquete le permiten colocar información en un búfer que puede ser leído por otro desencadenante, procedimiento o paquete. En un procedimiento PL / SQL separado o un bloque anónimo, puede mostrar la información almacenada en el búfer llamando al procedimiento GET\_LINE. El panel de salida de DBMS se usa para mostrar la salida de ese búfer. Este panel contiene iconos y otros controles para las siguientes operaciones:

* **Habilitar / deshabilitar la salida de DBMS** : alterna la configuración de CONFIGURACIÓN DEL SERVIDOR entre ENCENDIDO y APAGADO. La configuración de la salida del servidor ENCENDIDA comprueba la salida que se coloca en el búfer DBMS\_OUTPUT, y cualquier salida se muestra en el panel.
* **Borrar** : borra el contenido del panel.
* **Guardar** : guarda el contenido del panel en un archivo que especifique.
* **Imprimir** : Imprime el contenido del panel.
* **Tamaño del búfer** : para las bases de datos anteriores a Oracle Database 10.2, se limita la cantidad de datos que se pueden almacenar en el búfer DBMS\_OUTPUT. El tamaño del búfer puede estar entre 1 y 1000000 (1 millón).
* **Encuesta** : el intervalo (en segundos) en el que SQL Developer comprueba el búfer DBMS\_OUTPUT para ver si hay datos para imprimir. La tasa de votación puede estar entre 1 y 15.

### Panel de salida OWA

OWA (Oracle Web Agent) o MOD\_PLSQL es un módulo de extensión de Apache (servidor web) que le permite crear páginas web dinámicas a partir de paquetes PL / SQL y procedimientos almacenados. El panel de salida de OWA le permite ver la salida HTML de las acciones MOD\_PLSQL que se han ejecutado en la hoja de cálculo de SQL. Este panel contiene iconos para las siguientes operaciones:

* **Habilitar / deshabilitar salida de OWA** : habilita y deshabilita la verificación del búfer de salida de OWA y la visualización de la salida de OWA en el panel.
* **Borrar** : borra el contenido del panel.
* **Guardar** : guarda el contenido del panel en un archivo que especifique.
* **Imprimir** : Imprime el contenido del panel.

### Historial de SQL

Puede hacer clic en **Ver** , luego en **Historial de SQL** (o presionar F8) para ver una ventana acoplable con las sentencias de SQL y los scripts que ha ejecutado, y opcionalmente seleccionar una o más sentencias para que reemplacen las sentencias actualmente en la Hoja de cálculo de SQL o para agregarlas a las declaraciones actualmente en la hoja de trabajo de SQL.

Puede hacer clic en el encabezado de una columna para ordenar las filas por los valores de esa columna.

La lista de historial de SQL no contendrá ninguna declaración que pueda incluir una contraseña. Dichas declaraciones incluyen (pero no se limitan necesariamente a) CONECTAR, ALTERAR USUARIO y CREAR ENLACE DE BASE DE DATOS.

**Anexar** : anexa la declaración o declaraciones seleccionadas a cualquier declaración que se encuentre actualmente en la hoja de trabajo de SQL. También puede agregar la declaración o las declaraciones seleccionadas arrastrándolas desde la ventana del historial de SQL y soltándolas en la ubicación deseada en la hoja de cálculo de SQL.

**Reemplazar** : reemplaza cualquier declaración actualmente en la hoja de cálculo SQL con la declaración o declaraciones seleccionadas.

**Borrar** : Elimina todas las declaraciones del historial de SQL.

**Filtro** : si escribe una cadena en el cuadro de texto y hace clic en Filtrar, solo se muestran las declaraciones SQL que contienen esa cadena.

### Indicadores: en la hoja de trabajo SQL e informes definidos por el usuario

Puede usar indicadores gráficos para mostrar los resultados de la consulta en la hoja de trabajo de SQL y en los informes definidos por el usuario. En ambos casos, debe especificar el nombre de la columna de valores para los datos del medidor, y los valores mínimo y máximo en el medidor, y los valores que se mostrarán como bajos y altos en el medidor (generalmente entre los valores mínimo y máximo) . En la hoja de trabajo SQL, la estructura requerida para el valor que se seleccionará es:

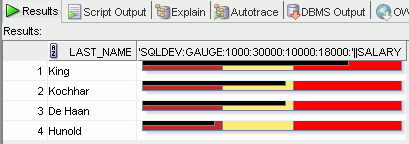
'SQLDEV: GAUGE: <min>: <max>: <low>: <high>:' || <value-column>

Por ejemplo, para mostrar el apellido y el salario en formato de indicador, donde el indicador muestra de 1000 a 30000 con menos de 10000 como mínimo y más de 18000 como máximo, para empleados con números de identificación inferiores a un número que debe especificarse, conéctese a la suministró el esquema de recursos humanos y ejecute la siguiente consulta:

SELECCIONE last\_name, 'SQLDEV: GAUGE: 1000: 30000: 10000: 18000:' || salario

DE los empleados WHERE employee\_id <: employee\_id

Si especifica 104 como el valor de la variable de enlace, la salida aparece como se muestra en la siguiente figura:



Para un informe de calibre definido por el usuario, la consulta debe especificar solo la columna de valores, los valores mínimo y máximo, y los valores alto y bajo, y opcionalmente una cláusula WHERE. La estructura requerida para la consulta (antes de cualquier cláusula WHERE opcional) es:

SELECCIONE <value-column>, <min>, <max>, <low>, <high> FROM <table-name>

Por ejemplo, para crear un informe de salarios en formato de marcación de indicador, con los mismos valores y la cláusula WHERE que en la consulta anterior, haga clic con el botón derecho en Informes definidos por el usuario en el navegador de Informes y seleccione **Agregar informe** . En el cuadro de diálogo Agregar informe, especifique un nombre de informe; para **Estilo** , seleccione **Calibrador** ; y para **SQL** , ingrese lo siguiente:

SELECCIONE salario, 1000, 30000, 10000, 18000 DE empleados

WHERE employee\_id <: EMPLOYEE\_ID;

Haga clic en la pestaña **Detalles** del **gráfico** cerca de la parte inferior del cuadro; para **Tipo de gráfico** , seleccione DIAL; para **Query Based** , seleccione verdadero; y haga clic en **Aplicar** .

Utilice el navegador de informes para ver el informe definido por el usuario recién creado. Para **Conexión** , especifique uno que se conecte al esquema de muestra de HR. Para el valor de la variable de enlace, especifique 104. El informe muestra cuatro cuadrantes semicirculares, cada uno con una etiqueta que contiene el monto del salario y una "aguja" que apunta a un lugar apropiado en el cuadrante.

### Uso de fragmentos para insertar fragmentos de código

Los fragmentos de código son fragmentos de código, como funciones SQL, sugerencias de optimizador y diversas técnicas de programación PL / SQL. Algunos fragmentos son solo una sintaxis, y otros son ejemplos. Puede insertar y editar fragmentos de código cuando utiliza la hoja de trabajo de SQL o cuando crea o edita una función o procedimiento de PL / SQL.

Para mostrar fragmentos, en el menú **Ver** , seleccione **Fragmentos** . En la ventana de fragmentos (en el lado derecho), use el menú desplegable para seleccionar un grupo (como Funciones de agregado o Funciones de caracteres). En la mayoría de los casos, los fragmentos en cada grupo no representan todos los objetos disponibles en esa agrupación lógica, o todos los formatos y opciones de cada fragmento mostrado. Para obtener información completa y detallada, consulte la documentación de la base de datos Oracle.

Se coloca un botón **Fragmentos** en el margen derecho de la ventana, de modo que puede mostrar la ventana de fragmentos si se oculta.

Para insertar un fragmento en su código en una hoja de cálculo SQL o en una función o procedimiento PL / SQL, arrastre el fragmento desde la ventana de fragmentos y suéltelo en el lugar deseado en su código; luego edite la sintaxis para que la función SQL sea válida en el contexto actual. Para ver una breve descripción de una función SQL en una información sobre herramientas, mantenga el puntero sobre el nombre de la función.

Por ejemplo, podría escribir SELECT y luego arrastrar CONCAT (char1, char2) desde el grupo de funciones de caracteres. Luego, edite la sintaxis de la función CONCAT y escriba el resto de la declaración, como en el siguiente:

SELECCIONE CONCAT (título, 'es un libro en la biblioteca') DE LOS libros;

### Fragmentos de usuario definidos

Puedes crear y editar fragmentos. Los fragmentos definidos por el usuario están destinados principalmente a permitirle complementar los fragmentos proporcionados por Oracle, aunque también se le permite reemplazar un fragmento proporcionado por Oracle por su propia versión.

Cuando crea un fragmento de código definido por el usuario, puede agregarlo a una de las categorías de fragmento de código suministradas por Oracle (como Funciones de agregado) o a una categoría que cree. Si agrega un fragmento a una categoría suministrada por Oracle y si su fragmento tiene el mismo nombre que un fragmento existente, su definición de fragmento reemplaza la existente. (Si posteriormente actualiza a una nueva versión de SQL Developer y elige conservar sus configuraciones anteriores, sus antiguos fragmentos definidos por el usuario reemplazarán cualquier fragmento de código proporcionado por Oracle con el mismo nombre en la nueva versión de SQL Developer).

Para crear un fragmento, realice una de las siguientes acciones:

* Abra la ventana Fragmentos y haga clic en el icono **Agregar fragmentos de usuario** .
* Seleccione texto para el fragmento en la ventana de la hoja de trabajo SQL, haga clic con el botón derecho y seleccione **Guardar fragmento** .
* Haga clic en el icono **Agregar**[fragmento de usuario en el](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJJDDF) cuadro de diálogo [Editar fragmentos (definido por el usuario)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJJDDF) .

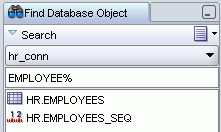
Para editar un fragmento de código definido por el usuario existente, haga clic en el icono **Editar fragmentos de usuario** en la ventana Fragmentos.

La información sobre los fragmentos de código definidos por el usuario se almacena en un archivo denominado UserSnippets.xml en el directorio para obtener información específica del usuario. Para obtener información sobre la ubicación de esta información, consulte [Ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD) .

### Usando el objeto Find DB para encontrar objetos de base de datos

Puede usar la función Buscar objeto de base de datos para buscar objetos de base de datos asociados con una conexión de base de datos Oracle y para abrir paneles de edición para trabajar con esos objetos. Para ir al panel Buscar objeto de base de datos o mostrarlo si no está visible, en el menú **Ver** , seleccione **Buscar objeto DB** .

La siguiente figura muestra el panel Buscar objeto de base de datos con resultados de una búsqueda de todos los objetos asociados con una conexión llamada hr\_conn que comienzan con EMPLOYEE. (El panel puede mostrarse en el lado derecho de la ventana del desarrollador de SQL o en la parte inferior).



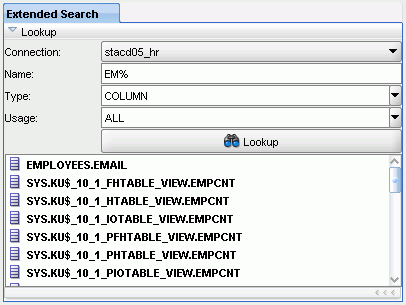
Para buscar objetos para una conexión de Oracle, haga clic en **Buscar** , seleccione el nombre de la conexión, ingrese un nombre de objeto o una cadena que contenga uno o más caracteres comodín, y presione la tecla Intro. Para ver o editar uno de los objetos, haga doble clic en su nombre en el panel Buscar objeto de base de datos.

Puede separar, mover y acoplar el panel Buscar objeto de base de datos haciendo clic y manteniendo presionada la pestaña, y arrastrando y posicionando el panel.

### Uso de la búsqueda extendida

Puede usar la función extendida para encontrar varios tipos de objetos (tablas, columnas, declaraciones dentro de funciones o procedimientos, etc.) asociados con una conexión de base de datos Oracle y para abrir paneles de edición para trabajar con esos objetos. (Por lo tanto, la búsqueda extendida es más completa que la función Buscar objeto de base de datos). Para ir al panel Búsqueda extendida o para mostrarlo si no está visible, en el menú **Ver** , seleccione **Búsqueda extendida** .

La siguiente figura muestra el panel Búsqueda ampliada con los resultados de una búsqueda de columnas asociadas con una conexión llamada stacd05\_hr que comienzan con EM. (El panel puede mostrarse en el lado derecho de la ventana del desarrollador de SQL o en la parte inferior).



**Conexión** : **Conexión de** base de datos a utilizar para la búsqueda.

**Nombre** : un nombre de objeto o una cadena que contiene uno o más caracteres comodín. Por ejemplo: EM%para todos los nombres que comiencen con EM.

**Tipo** : Tipo de objeto para el cual restringir la búsqueda.

**Uso** : Uso del objeto. Puede o no ser relevante, dependiendo del tipo de objeto.

Haga clic en el icono de **búsqueda** para mostrar los objetos que cumplen con los criterios especificados. Para ver o editar uno de los objetos (o el objeto principal que contiene el objeto especificado), haga doble clic en su nombre en el panel Búsqueda ampliada

Puede separar, mover y acoplar el panel Búsqueda extendida haciendo clic y manteniendo presionada la pestaña, y arrastrando y posicionando el panel.

### Utilizando versioning

SQL Developer proporciona soporte integrado para los siguientes sistemas de control de versiones y control de fuente: CVS (Sistema de versiones concurrentes) y Subversion. La documentación del Desarrollador de SQL no proporciona información detallada sobre los conceptos y operaciones de dichos sistemas; asume que los conoce o puede leer sobre ellos en la documentación del producto.

* Para el manual de CVS (por Per Cederqvist y otros), vea <http://ximbiot.com/cvs/manual/>.
* Para obtener información sobre Subversion, consulte <http://subversion.tigris.org/>. Para la documentación de Subversion, ver <http://svnbook.red-bean.com/>.

Para acceder a las funciones de control de versiones de SQL Developer, use el [menú Control de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEIHEA) .

Si crea cualquier repositorio del sistema de versiones, puede usar la visualización jerárquica en el navegador de Archivos, que está marcado con un icono de carpeta. (Si ese navegador no está visible, haga clic en **Ver** , luego en **Archivos** ).

### Sobre CVS y SQL Developer

Los repositorios CVS se pueden crear en una PC local o en un servidor remoto. Puede haber más de un repositorio de CVS. Necesitas crear uno o más repositorios CVS.

Los archivos de origen se guardan en un repositorio de CVS. Los archivos de origen en un repositorio de CVS se agrupan en módulos. Si tiene archivos nuevos, un asistente en SQL Developer lo ayudará a importarlos al repositorio de CVS y los colocará bajo el control de versiones. Se hace una copia de sus archivos originales y se coloca en un subdirectorio (.backup) del que los importó.

Los archivos en los que se trabajará se extraen del repositorio de CVS. Esto hace una copia local de los archivos. Puede ver el contenido del repositorio de CVS en el desarrollador de CVS Navigator de CVS y abrir versiones de archivos de solo lectura. A continuación, puede decidir qué archivos desea desproteger y trabajar.

CVS crea un nuevo directorio con la copia de los archivos de origen. Puedes ver los archivos en el System Navigator. También puedes abrirlos desde aquí.

Los archivos de origen tienen un estado, según las operaciones que se hayan realizado en ellos. Una preferencia le permite elegir si el estado del control de versión de un archivo se muestra en el System Navigator.

Cambios pendientes (CVS)

La ventana Cambios pendientes se muestra si hace clic en Control de versiones, luego en CVS, luego en Cambios pendientes, o cuando inicia una acción que cambia el estado del control de fuente local de un archivo. Esta ventana muestra los archivos que se agregaron, modificaron o eliminaron (local o remotamente), los archivos cuyo contenido está en conflicto con otras versiones de los mismos archivos que no se agregaron a los archivos de control de origen que se están viendo, y los archivos para los cuales los editores sido obtenido. Puede utilizar esta información para detectar conflictos y resolverlos cuando sea posible.

El panel Saliente muestra los cambios realizados localmente, el panel Entrante muestra los cambios realizados de forma remota y el panel Candidatos muestra los archivos que se crearon localmente pero aún no se han agregado al control de origen. Puede hacer doble clic en los nombres de los archivos para editarlos y puede usar el menú contextual para realizar las operaciones disponibles.

### Sobre Subversion y SQL Developer

Antes de poder trabajar con un repositorio de Subversion a través de SQL Developer, debe crear una conexión con él. Cuando crea un repositorio local de Subversion, se crea automáticamente una conexión, y esto se puede ver en el navegador de Subversion. Posteriormente puede editar los detalles de la conexión.

Los archivos existentes deben importarse al repositorio de Subversion para ponerlos bajo el control de versiones. Luego, los archivos se extraen del repositorio de Subversion a una carpeta local conocida como la "copia de trabajo de Subversion". Los archivos creados en (o trasladados a) SQL Developer deben almacenarse en la copia de trabajo de Subversion.

Los archivos recién creados en SQL Developer deben agregarse al control de versiones. Los archivos nuevos y modificados se ponen a disposición de otros usuarios al enviarlos al repositorio de SQL Developer. La copia de trabajo de Subversion se puede actualizar con el contenido del repositorio de Subversion para incorporar los cambios realizados por otros usuarios.

### Informes del desarrollador de SQL

SQL Developer proporciona muchos informes sobre la base de datos y sus objetos. También puede crear sus propios informes definidos por el usuario. Para mostrar informes, haga clic en la pestaña Informes en el lado izquierdo de la ventana (consulte la [Interfaz de usuario de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAHIFGB) ). Si esta pestaña no está visible, seleccione **Ver** y luego **Informes** .

Los informes individuales se muestran en paneles con pestañas en el lado derecho de la ventana; y para cada informe, puede seleccionar (en un control desplegable) la conexión de la base de datos para la cual se muestra el informe. Para los informes sobre objetos, los objetos que se muestran son solo aquellos visibles para el usuario de la base de datos asociado con la conexión de la base de datos seleccionada, y las filas generalmente están ordenadas por el propietario. El panel de visualización de detalles para un informe incluye los siguientes iconos en la parte superior:

* **Congelar vista** (el **pin** ) mantiene ese informe en la ventana del Desarrollador SQL cuando hace clic en otro informe en el navegador de Informes; se crean una pestaña separada y un panel de vista detallada para ese otro informe. Si vuelve a hacer clic en el pin, el panel de vista de detalles del informe estará disponible para su reutilización.
* **Ejecutar informe** actualiza la visualización del panel de vista de detalles consultando la base de datos para obtener la información más reciente.
* **Ejecutar informe en la hoja de trabajo SQL** muestra la instrucción SQL utilizada para recuperar la información de un informe en un panel de hoja de trabajo SQL, donde puede ver, editar y ejecutar la declaración (consulte [Uso de la hoja de trabajo SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) ).

El tiempo requerido para mostrar informes específicos variará, y puede verse afectado por la cantidad y complejidad de los objetos involucrados y por la velocidad de la conexión de red a la base de datos.

Para la mayoría de los informes que contienen nombres de objetos de base de datos, puede hacer doble clic en el nombre del objeto en el panel de visualización del informe (o hacer clic con el botón derecho en el nombre del objeto y seleccionar Ir a) para mostrar ese objeto en un panel de vista de detalles, como si Había seleccionado ese objeto utilizando el navegador de conexiones.

Para exportar un informe a un archivo XML que puede importarse más tarde, haga clic con el botón derecho en el nombre del informe en la pantalla del navegador de Informes y seleccione **Exportar** . Para importar un informe que se había exportado anteriormente, seleccione el nombre del nombre de la carpeta del informe (como una carpeta definida por el usuario) en la que desea almacenar el informe importado, haga clic con el botón derecho y seleccione **Importar** .

Puede crear un **informe compartido a** partir de un informe exportado haciendo clic en Herramientas, luego en Preferencias y utilizando el panel [Base de datos: Extensiones definidas por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIHFJA) para agregar una fila con Tipo como INFORME y Ubicación que especifique el archivo XML exportado. La próxima vez que reinicie SQL Developer, el navegador de informes tendrá una carpeta de informes compartidos que contiene ese informe.

Los informes se agrupan en las siguientes categorías:

[Los informes de Acerca de su base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDDHJ) contienen información sobre la versión de la base de datos asociada con la conexión.

[Todos los informes de Objetos muestran](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDJFAG) información sobre todos los objetos accesibles para el usuario asociados con la conexión de base de datos especificada, no solo los objetos que son propiedad del usuario.

[Los informes de Application Express](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGGJAF) contienen información sobre aplicaciones, páginas, esquemas, valores por defecto de UI y áreas de trabajo de Oracle Application Express 3.0.1 (o posterior).

[Los informes de ASH y AWR](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHBCCF) contienen información de la lista proporcionada por las características de Active Session History (ASH) y Automated Workload Repository (AWR).

[Los informes de administración de la base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDCBEJJ) contienen información sobre el uso de los recursos del sistema.

[El diccionario de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDFCIE) muestra información sobre las vistas del diccionario de datos a las que se puede acceder en la base de datos. Ejemplos de vistas del diccionario de datos son ALL\_OBJECTS y USER\_TABLES.

[Los informes de trabajos muestran](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDCFEBD) información sobre los trabajos que se ejecutan en la base de datos.

[Los informes PL / SQL muestran](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDAAC) información sobre sus objetos PL / SQL y le permiten buscar el origen de esos objetos.

[Los informes de seguridad](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDJAGC) enumeran información relacionada con privilegios sobre la base de datos.

[Los informes](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEGJAE) de secuencias muestran información sobre las reglas de transmisión.

[Los informes](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCICB) de tablas muestran información sobre las tablas que son propiedad del usuario asociadas con la conexión especificada. Estos informes pueden ayudarlo a comprender mejor los metadatos y los datos. Los informes de la tabla incluyen informes de control de [calidad](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBJIB) que indican posibles fallas de diseño lógicas y fuentes de problemas de rendimiento en tiempo de ejecución.

[Los informes XML](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDDBE) listan información sobre objetos XML.

[Los informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGGBDG) son cualquier informe personalizado que haya creado.

**Vincular variables para informes**

Para algunos informes, se le solicitarán las **variables de enlace** antes de que se genere el informe. Estas variables de enlace le permiten restringir aún más la salida. El valor predeterminado para todas las variables de enlace es nulo, lo que implica que no hay más restricciones. Para especificar una variable de enlace, seleccione el nombre de la variable y escriba una entrada en el campo Valor. Cualquier valor de la variable de vinculación que ingrese no distingue entre mayúsculas y minúsculas, todas las coincidencias se devuelven donde la cadena de valor aparece en cualquier lugar en el nombre del tipo de objeto relevante.

### Acerca de sus informes de base de datos

Los informes de Acerca de su base de datos contienen información sobre la versión de la base de datos asociada con la conexión seleccionada. Los informes incluyen Banner de versión (configuración de la base de datos) y Parámetros de compatibilidad de idioma nacional (valores de parámetros NLS\_xxx para compatibilidad con la globalización).

### Todos los informes de objetos

Todos los informes de Objetos muestran información sobre los objetos visibles para el usuario asociados con la conexión de la base de datos.

**Todos los objetos** : para cada objeto, muestra el propietario, el nombre, el tipo (tabla, vista, índice, etc.), el estado (válido o no válido), la fecha en que se creó y la fecha en que se usó el último idioma de definición de datos (DDL). ) la operación se realizó en él. La fecha del último DDL puede ayudarlo a encontrar si se han realizado cambios en las definiciones de los objetos en o después de una hora específica.

**Tipos de colección** : muestra información sobre cada tipo de colección. La información incluye el propietario del tipo, el nombre del tipo de elemento y el propietario, y la información específica que depende del tipo.

**Dependencias** : para cada objeto con referencias a él, enumera información sobre las referencias a (usos de) ese objeto.

**Objetos no válidos** : enumera todos los objetos que tienen un estado de no válido.

**Recuento de objetos por tipo** : para cada tipo de objeto asociado con un propietario específico, muestra el número de objetos. Este informe podría ayudarlo a identificar a los usuarios que han creado una gran cantidad de objetos, especialmente objetos de un tipo específico.

**Enlaces de bases de datos públicas** : enumera todos los enlaces de bases de datos públicas.

**Sinónimos públicos** : enumera todos los sinónimos públicos.

### Informes de Application Express

Si selecciona una conexión para un esquema que sea propietario de alguna aplicación Oracle Application Express 3.0.1 (o posterior), Application Express proporciona información de la lista de aplicaciones, páginas, esquemas, valores predeterminados de UI y áreas de trabajo. Para obtener información sobre Oracle Application Express, consulte la documentación de ese producto.

### Informes ASH y AWR

Los informes de ASH y AWR enumeran la información proporcionada por el historial de sesión activa (ASH) y las funciones del repositorio de carga de trabajo automatizada (AWR), que requieren una licencia especial. Para obtener información sobre el uso de AWR, incluido cómo usar los informes de ASH, consulte la información sobre estadísticas de rendimiento automáticas en la [*Guía de ajuste de rendimiento de la base de datos de Oracle*](https://docs.oracle.com/cd/E11882_01/server.112/e41573/autostat.htm#PFGRF027) .

### Informes gráficos

Los informes de gráficos incluyen un gráfico que muestra la distribución de objetos de diversos tipos de objetos (número de tablas, índices, etc.).

### Informes de administración de bases de datos

Los informes de administración de bases de datos muestran información de uso sobre los recursos del sistema Esta información puede ayudarlo a administrar el almacenamiento, las cuentas de usuario y las sesiones de manera eficiente. (El usuario para la conexión de la base de datos debe tener el rol de DBA para ver la mayoría de los informes de Administración de la base de datos).

**Todas las tablas** : contiene los informes que también se agrupan en [informes de tabla](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJCICB) , incluidos [los informes de control de calidad](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBJIB) .

**Cursores** : proporcionan información sobre los cursores, incluidos los cursores por sesión (incluidos los cursores abiertos y los detalles del cursor).

**Parámetros de la base de datos** : proporcione información sobre todos los parámetros de la base de datos o solo aquellos parámetros que no están configurados en sus valores predeterminados.

**Bloqueos** : proporciona información sobre los bloqueos, incluido el usuario asociado a cada uno.

**Sesiones** : Proporcionar información sobre las sesiones, seleccionadas y ordenadas por diversos criterios.

**Almacenamiento** : Proporcione información de uso y asignación para espacios de tablas y archivos de datos.

**Top SQL** : proporciona información sobre las sentencias de SQL, seleccionadas y ordenadas por varios criterios. Esta información puede ayudarlo a identificar las sentencias de SQL que se ejecutan con más frecuencia de lo esperado o que llevan más tiempo del esperado.

**Usuarios** : Proporcionan información sobre los usuarios de la base de datos, seleccionados y ordenados por diversos criterios. Por ejemplo, puede averiguar qué usuarios se crearon más recientemente, qué cuentas de usuarios han caducado, y qué usuarios usan los tipos de objetos y cuántos objetos posee cada uno.

**Esperas y eventos** : proporcione información sobre esperas y eventos, seleccionados por criterios relacionados con el tiempo y otros factores. Para eventos en los últimos x minutos, especifique la cantidad de minutos en el cuadro de diálogo [Ingresar valores de enlace](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFAIJH) .

### Informes del diccionario de datos

El diccionario de datos muestra información sobre las vistas del diccionario de datos a las que se puede acceder en la base de datos. Ejemplos de vistas del diccionario de datos son ALL\_OBJECTS y USER\_TABLES.

**Columnas de vista de diccionario** : para cada vista de diccionario de datos de Oracle, muestra información sobre las columnas en la vista.

**Vistas de diccionario** : muestra cada vista de diccionario de datos de Oracle y (en la mayoría de los casos) un comentario que describe su contenido o propósito.

### Informes de empleo

Los informes de trabajos muestran información sobre los trabajos que se ejecutan en la base de datos.

**Todos los trabajos** : muestra información sobre todos los trabajos que se ejecutan en la base de datos. La información incluye la hora de inicio de su última ejecución, la ejecución actual y la próxima ejecución programada.

**Trabajos de DBA** : muestra información sobre cada trabajo para el cual un usuario de DBA está asociado con la conexión de la base de datos. La información incluye la hora de inicio de su última ejecución, la ejecución actual y la próxima ejecución programada.

**Sus trabajos** : muestra información sobre cada trabajo para el que el usuario asociado con la conexión de la base de datos es el usuario de registro, el usuario con privilegios o el usuario de esquema. La información incluye la hora de inicio de su última ejecución, la ejecución actual y la próxima ejecución programada.

### Informes PL / SQL

Los informes de PL / SQL muestran información sobre los paquetes, funciones y procedimientos de PL / SQL y sobre los tipos definidos en ellos.

**Argumentos de la unidad de programa** : para cada argumento (parámetro) en una unidad de programa, enumera el nombre de la unidad de programa, la posición del argumento (1, 2, 3, etc.), el nombre del argumento y si el argumento es solo de entrada (In ), solo salida (salida), o entrada y salida (entrada / salida).

**Buscar código fuente** : para cada objeto PL / SQL, enumera el código fuente de cada línea y permite buscar en la fuente las incidencias de la variable especificada.

**Unidades de línea de unidad** : para cada objeto PL / SQL, muestra el número de líneas de código fuente. Esta información puede ayudarlo a identificar objetos complejos (por ejemplo, para identificar un código que puede ser necesario simplificar o dividir en varios objetos).

### Informes de seguridad

Los informes de seguridad incluyen información sobre los usuarios a los que se les han otorgado privilegios y, en algunos casos, sobre los usuarios que otorgaron los privilegios. Esta información puede ayudarlo a usted (o al administrador de la base de datos si no es un DBA) a comprender los posibles problemas y vulnerabilidades de seguridad, y a decidir la acción apropiada que debe tomar (por ejemplo, revocar ciertos privilegios de los usuarios que no necesitan esos privilegios) .

**Auditoría** : Enumera información sobre las políticas de auditoría.

**Cifrado** : Enumera información sobre columnas cifradas.

**Subvenciones y privilegios** : Incluye los siguientes informes:

* **Privilegios de columna** : para cada privilegio otorgado en una columna específica en una tabla específica, enumera el usuario que otorgó el privilegio, el usuario al que se otorgó el privilegio, la tabla, el privilegio y si el usuario al que se otorgó el privilegio puede otorga ese privilegio a otros usuarios.
* **Object Grants** : para cada privilegio otorgado en una tabla específica, enumera el usuario que otorgó el privilegio, el usuario al que se otorgó el privilegio, la tabla, el privilegio y si el usuario al que se otorgó el privilegio puede otorgar ese privilegio otros usuarios
* **Privilegios de rol** : para cada rol otorgado, enumera el usuario al que se le otorgó el rol, el rol, si el rol se otorgó con la opción ADMIN, y si el rol se designa como rol predeterminado para el usuario.
* **Privilegios del sistema** : para cada privilegio otorgado al usuario asociado con la conexión de la base de datos, se enumeran los privilegios y si se otorgó con la opción ADMIN.

**Políticas** : Enumera información sobre políticas.

**Subvenciones públicas** : muestra información sobre los privilegios otorgados a la función PUBLIC.

### Informes de transmisiones

Los informes de secuencias muestran información sobre las reglas de transmisión.

**Todas las reglas de** transmisión: muestra información sobre todas las reglas de transmisión. La información incluye el tipo y nombre de la secuencia, el propietario y el nombre del conjunto de reglas, el nombre y el propietario de la regla, el tipo de conjunto de reglas, el tipo de regla de las secuencias y la operación de subconjunto.

**Sus reglas de** transmisión: muestra información sobre cada regla de transmisión para la cual el usuario asociado con la conexión de la base de datos es el propietario de la regla o el propietario del conjunto de reglas. La información incluye el tipo y nombre de la secuencia, el propietario y el nombre del conjunto de reglas, el nombre y el propietario de la regla, el tipo de conjunto de reglas, el tipo de regla de las secuencias y la operación de subconjunto.

### Informes de mesa

Los informes de tablas muestran información sobre las tablas que son propiedad del usuario asociadas con la conexión especificada. Esta información no está diseñada específicamente para identificar áreas problemáticas; sin embargo, dependiendo de sus recursos y requisitos, parte de la información puede indicar cosas que debe supervisar o abordar.

Para los informes de tablas, el propietario es el usuario asociado con la conexión de la base de datos.

**Columns**: For each table, lists each column, its data type, and whether it can contain a null value. Also includes **Datatype Occurrences**: For each table owner, lists each data type and how many times it is used.

**Comments** for tables and columns: For each table and for each column in each table, lists the descriptive comments (if any) associated with it. Also includes a report of tables without comments. If database developers use the COMMENT statement when creating or modifying tables, this report can provide useful information about the purposes of tables and columns

**Constraints**: Includes the following reports related to constraints:

* **Todas las restricciones** : para cada tabla, enumera cada restricción asociada, incluido su tipo (restricción única, restricción de verificación, clave principal, clave externa) y estado (habilitado o deshabilitado).
* **Restricciones de verificación** : para cada restricción de verificación, muestra información que incluye al propietario, el nombre de la tabla, el nombre de la restricción, el estado de la restricción (habilitado o deshabilitado) y la especificación de la restricción.
* **Restricciones habilitadas** y **restricciones inhabilitadas** : para cada restricción con un estado habilitado o deshabilitado, enumera el nombre de la tabla, el nombre de la restricción, el tipo de restricción (restricción única, restricción de verificación, clave principal, clave externa) y el estado. Una restricción deshabilitada no se aplica cuando se agregan o modifican filas; para que se aplique una restricción deshabilitada, debe editar la tabla y establecer el estado de la restricción en Habilitado (consulte las pestañas correspondientes del cuadro de diálogo [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) ).
* **Restricciones de clave externa** : para cada restricción de clave externa, enumera la información que incluye el propietario, el nombre de la tabla, el nombre de la restricción, la columna contra la que se encuentra la restricción, la tabla a la que hace referencia la restricción y la restricción en la tabla a la que se hace referencia.
* **Restricciones de clave principal** : para la restricción de clave principal, enumera la información que incluye el propietario, el nombre de la tabla, el nombre de la restricción, el estado de la restricción (habilitado o deshabilitado) y el nombre de la columna.
* **Restricciones únicas** : para cada restricción única, enumera la información que incluye el propietario, el nombre de la tabla, el nombre de la restricción, el estado de la restricción (habilitado o deshabilitado) y el nombre de la columna.

**Índices** : incluye información sobre todos los índices, índices por estado, índices por tipo e índices no utilizados.

**Organización** : los informes especializados muestran información sobre tablas particionadas, tablas agrupadas y tablas organizadas por índice.

**Garantía de calidad** : (Ver [informes de garantía de calidad](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBJIB) ).

**Estadísticas** : para cada tabla, muestra información estadística, incluida la fecha en que se analizó por última vez, el número total de filas, la longitud de fila promedio y el tipo de tabla. Además, los informes especializados ordenan los resultados por la mayoría de las filas y la mayor longitud de fila promedio.

**Almacenamiento** : muestra información sobre el recuento de tablas por espacio de tablas y las tablas en cada espacio de tablas.

**Desencadenadores** : muestra información sobre todos los desencadenantes, desencadenantes deshabilitados y desencadenadores habilitados.

**Sinónimos del usuario** : muestra información sobre todos los sinónimos de usuario o los sinónimos de usuario que contienen la cadena que especificó en el cuadro de diálogo Ingresar variables de vinculación (anule la selección de Nulo en esa casilla para ingresar una cadena).

**Tablas de usuario** : muestra información sobre todas las tablas o las tablas que contienen la cadena que especificó en el cuadro de diálogo Ingresar variables de vinculación (deseleccione Nulo en esa casilla para ingresar una cadena).

**Informes de garantía de calidad**

Los informes de control de calidad son informes de tablas que identifican condiciones que no son errores técnicos, pero que generalmente indican fallas en el diseño de la base de datos. Estas fallas pueden dar lugar a varios problemas, como errores lógicos y la necesidad de una codificación de aplicación adicional para solucionar los errores, así como un rendimiento deficiente en las consultas en tiempo de ejecución.

**Tablas sin claves principales** : enumera las tablas que no tienen una clave principal definida. Una clave principal es una columna (o conjunto de columnas) que identifica de forma única cada fila en la tabla. Aunque no se requiere que las tablas tengan una clave principal, se recomienda encarecidamente que cree o designe una clave primaria para cada tabla. Las columnas de la clave principal están indexadas, lo que mejora el rendimiento de las consultas, y deben ser únicas y no nulas, lo que proporciona una validación "automática" de los datos de entrada. Las claves primarias también se pueden usar con claves externas para proporcionar integridad referencial.

**Tables without Indexes**: Lists tables that do not have any indexes. If a column in a table has an index defined on it, queries that use the column are usually much faster and more efficient than if there is no index on the column, especially if there are many rows in the table and many different data values in the column.

**Tables with Unindexed Foreign Keys**: Lists any foreign keys that do not have an associated index. A foreign key is a column (or set of columns) that references a primary key: that is, each value in the foreign key must match a value in its associated primary key. Foreign key columns are often joined in queries, and an index usually improves performance significantly for queries that use a column. If an unindexed foreign key is used in queries, you may be able to improve run-time performance by creating an index on that foreign key.

### XML reports

XML reports list information about XML objects.

**XML Schemas**: For each user that owns any XML objects, lists information about each object, including the schema URL of the XSD file containing the schema definition.

### Migration reports

Migration reports list information related to migrating third-party databases to Oracle. For more information, see [Using Migration Reports](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm" \l "CHDJGAIH).

### User Defined reports

User Defined reports are any reports that are created by SQL Developer users. To create a user-defined report, right-click the User Defined node under Reports and select **Add Report**. A dialog box is displayed in which you specify the report name and the SQL query to retrieve information for the report (see [Create/Edit User Defined Report](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm" \l "BACFEIDI)).

Puede organizar informes definidos por el usuario en carpetas y puede crear una jerarquía de carpetas y subcarpetas. Para crear una carpeta para informes definidos por el usuario, haga clic con el botón derecho en el nodo Definido por el usuario o en el nombre de cualquier carpeta debajo de ese nodo y seleccione **Agregar carpeta** (vea [Crear / Editar carpeta de informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBDFJB) ).

La información sobre los informes definidos por el usuario, incluidas las carpetas para estos informes, se almacena en un archivo denominado UserReports.xml en el directorio para obtener información específica del usuario. Para obtener información sobre la ubicación de esta información, consulte [Ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD) .

Para ver ejemplos de cómo crear informes definidos por el usuario, consulte:

* [Ejemplo de informe definido por el usuario: gráfico](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHBIDD)
* [Ejemplo de informe definido por el usuario: HTML dinámico](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEBHHB)
* [Indicadores: en la hoja de trabajo SQL e informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDIBD)

Ejemplo de informe definido por el usuario: gráfico

Este ejemplo crea un informe que se muestra como un gráfico. Utiliza la definición de la tabla EMPLEADOS del esquema de recursos humanos, que es un esquema de ejemplo proporcionado.

Haga clic con el botón derecho en Informes definidos por el usuario y seleccione **Agregar informe** . En el cuadro de diálogo Agregar informe, especifique un nombre de informe; para **Estilo** , seleccione **Gráfico** ; y para **SQL** , ingrese lo siguiente:

seleccione m.department\_id, e.last\_name, e.salary

de empleados m, empleados e

donde e.employee\_id = m.employee\_id

orden por 1

La consulta anterior enumera el apellido y el salario de cada empleado en cada departamento, agrupando los resultados por ID de departamento (10, 20, 30, ... 110). Tenga en cuenta que la sintaxis esperada para la declaración SQL para un informe de gráfico es:

SELECCIONE *<grupo>* , *<series>* , *<valor>* DE *<tabla (s)>*

Haga clic en la pestaña **Detalles** del **gráfico** cerca de la parte inferior del cuadro; para **Tipo de gráfico** , seleccione BAR\_VERT\_STACK (gráfico de barras, apilado verticalmente); y haga clic en **Aplicar** .

Utilice el navegador de informes para ver el informe definido por el usuario recién creado. Para **Conexión** , especifique uno que se conecte al esquema de muestra de HR.

El informe se muestra como un gráfico, parte del cual se muestra en la siguiente ilustración. Por ejemplo, como puede ver, el departamento 50 tiene principalmente empleados con los salarios más bajos, y el departamento 90 consta de los tres empleados mejor pagados.



Usuario definido Ejemplo de informe: HTML dinámico

Este ejemplo crea un informe utilizando una o más sentencias PL / SQL DBMS\_OUTPUT, para que el informe se muestre como HTML dinámico.

Haga clic con el botón derecho en Informes definidos por el usuario y seleccione **Agregar informe** . En el cuadro de diálogo Agregar informe, especifique un nombre de informe; para **Style** , seleccione **plsql-dbms\_output** ; y para **SQL** , ingrese lo siguiente:

empezar

dbms\_output.put\_line ('<H1> Este es el encabezado de nivel 1 </H1>');

dbms\_output.put\_line ('<H2> Este es un encabezado de nivel 2 </H2>');

dbms\_output.put\_line ('<p> Este es un texto de párrafo regular. </p>');

fin;

Haga clic en **Aplicar** .

Utilice el navegador de informes para ver el informe definido por el usuario recién creado. Para la **conexión** , especifique cualquiera de la lista. (Este informe no depende de una conexión específica de la tabla).

El informe se muestra como salida HTML formateada.

### Preferencias de desarrollador de SQL

Puede personalizar muchos aspectos de la interfaz y el entorno de SQL Developer modificando las preferencias de SQL Developer de acuerdo con sus preferencias y necesidades. Para modificar las preferencias de SQL Developer, seleccione **Herramientas** , luego **Preferencias** .

La información sobre las preferencias de SQL Developer se almacena en el directorio para obtener información específica del usuario. Para obtener información sobre la ubicación de esta información, consulte [Ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD) .

La mayoría de las preferencias se explican por sí mismas, y este tema explica solo aquellas cuyo significado e implicaciones no son evidentes. Algunas preferencias involucran el rendimiento o las compensaciones de recursos del sistema (por ejemplo, habilitar una función que agrega tiempo de ejecución), y otras preferencias solo involucran el gusto estético personal. Las preferencias se agrupan en las siguientes categorías.

### Ambiente

El panel de entorno contiene opciones que afectan el inicio y el comportamiento general y la apariencia de SQL Developer. Puede especificar que ciertas operaciones se realicen automáticamente en momentos específicos, y la compensación suele ser el tiempo extra para la operación, en lugar de la posibilidad de problemas si la operación no se realiza automáticamente (por ejemplo, si olvida realizarla). cuando deberías).

Los valores de deshacer (número de operaciones anteriores que se pueden deshacer) y nivel de navegación (número de archivos abiertos) implican aumentos o disminuciones leves en el uso de recursos del sistema para valores más altos o más bajos.

**Recargar automáticamente archivos modificados externamente** : si esta opción está marcada, todos los archivos abiertos en SQL Developer que hayan sido modificados por una aplicación externa se actualizan cuando vuelves a SQL Developer, sobrescribiendo los cambios que hayas realizado. Si esta opción no está marcada, los cambios que realice en SQL Developer sobrescribirán los cambios que puedan haber sido realizados por aplicaciones externas.

**Volver a cargar de forma silenciosa cuando el archivo no está modificado** : si esta opción está marcada, no se le preguntará si desea volver a cargar los archivos que se han modificado externamente, pero no en SQL Developer. Si esta opción no está marcada, se le preguntará si desea volver a cargar cada archivo que se haya modificado externamente, independientemente de si se ha modificado en SQL Developer.

**Entorno: Windows acoplable**

El panel Ventanas acoplables configura el comportamiento de las ventanas acoplables y las formas de las cuatro áreas de acoplamiento de SQL Developer: superior, inferior, izquierda y derecha.

**Ventanas acoplables siempre en la parte superior** : si esta opción está marcada, las ventanas acoplables siempre permanecen visibles frente a otras ventanas.

**Diseño de Windows** : haga clic en las flechas de las esquinas para alargar o acortar la forma de cada área de acoplamiento.

**Medio Ambiente: Historia Local**

El panel Historial local controla si se conserva la información sobre las operaciones de edición en los archivos abiertos en SQL Developer. Si el historial local está habilitado, puede especificar cuánto tiempo se retiene la información y el número máximo de revisiones para cada archivo.

**Ambiente: Log**

El panel de Registro configura los colores de ciertos tipos de mensajes de registro y el guardado de los mensajes de registro en los archivos de registro.

**Guardar registros en archivo** : si esta opción está marcada, todos los resultados en la ventana Mensajes - Registro se guardan en archivos de registro, donde el nombre del archivo refleja la operación y una marca de tiempo. También se le solicita que especifique un **Directorio de registro** ; y si el directorio especificado no existe, se crea. Tenga en cuenta que si guarda la información de registro en los archivos, la cantidad de estos archivos puede aumentar.

**Máximo de líneas de registro** : el número máximo de líneas para almacenar en cada archivo de registro.

### Aceleradores (atajos de teclado)

El panel Aceleradores le permite ver y personalizar las asignaciones de teclas del acelerador (atajos de teclado) para SQL Developer.

**Categoría** : seleccione Todas o una categoría específica (Editor de código, Base de datos, Depurar, Editar, etc.) para controlar qué acciones se muestran.

**Acciones** : Las acciones para la categoría seleccionada. Cuando selecciona una acción, se muestran las asignaciones de teclas del acelerador existentes.

**Aceleradores** : cualquier asignación de teclas existente para la acción seleccionada. Para eliminar una asignación de clave existente, selecciónela y haga clic en Eliminar.

**Nuevo acelerador** : la nueva tecla del acelerador que se asociará con la acción. Mantenga presionada la tecla modificadora deseada, luego presione la otra tecla. Por ejemplo, para asociar Ctrl + J con una acción, mantenga presionada la tecla Ctrl, luego presione la tecla j. Si alguna acción está actualmente asociada con esa tecla de aceleración, se enumeran en el cuadro Asignación actual.

**Asignación actual** : una visualización de solo lectura de la acción actual, si corresponde, que se asigna a la tecla de aceleración que especificó en el cuadro Nuevo acelerador.

**Cargar predefinido** : muestra el cuadro de diálogo [Cargar asignaciones de claves preestablecidas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFBHCJ) , donde puede cargar un conjunto de asignaciones de claves predefinidas (incluidos los valores predeterminados del desarrollador SQL) para ciertos sistemas y aplicaciones de edición externas. Si carga cualquier asignación de teclas preestablecida que entre en conflicto con los cambios que haya realizado, se sobrescribirán.

### Editor de código

El panel Editor de código contiene opciones generales que afectan la apariencia y el comportamiento de SQL Developer cuando edita funciones, procedimientos y paquetes.

**Editor de código: Marcadores**

El panel Marcadores contiene opciones que determinan la persistencia y el comportamiento de búsqueda de los marcadores que crea al usar el editor de código.

**Editor de Código: Comportamiento Caret**

El panel Comportamiento de la careta contiene opciones que determinan la forma, el color y las características de parpadeo de la careta (cursor) en el editor de código.

**Editor de código: Code Insight**

El panel de Code Insight contiene opciones para la finalización lógica (opciones de autocompletar) de palabras clave y nombres mientras codifica en la hoja de cálculo de SQL.

Cuando presiona **Ctrl + Espacio** , la información de código proporciona una ventana emergente sensible al contexto que puede ayudarlo a seleccionar nombres de parámetros. La información de finalización le proporciona una lista de posibles terminaciones en el punto de inserción que puede utilizar para completar automáticamente el código que está editando. Esta lista se basa en el contexto del código en el punto de inserción. Para salir del código en cualquier momento, presione Esc.

Puede habilitar o inhabilitar tanto la finalización como la información de parámetros, así como establecer el retraso de tiempo para las ventanas emergentes.

**Editor de código: Code Insight: Finalización**

El panel Code Insight: Completion contiene opciones para refinar el comportamiento cuando se encuentran elementos coincidentes. Para obtener más información, consulte la explicación para el [Editor de código: Code Insight](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIIEH) .

**Editor de código: Display**

El panel de Pantalla contiene opciones generales para la apariencia y el comportamiento del editor de código.

**El texto Anti-Aliasing** permite caracteres de bordes suaves siempre que sea posible.

**El Margen de Plegado de Código** permite que los bloques de programa en procedimientos y funciones se expandan y colapsen en la pantalla.

**El margen derecho visible** representa un margen derecho que puede configurar para controlar la longitud de las líneas de código.

**El ajuste automático de corsé** controla el resaltado de paréntesis y paréntesis de apertura y de bloques cuando se escribe un paréntesis o paréntesis de cierre.

**Editor de código: Fuentes**

El panel de Fuentes especifica las opciones de fuente de texto para el editor de código.

**Mostrar solo fuentes de ancho fijo** : si esta opción está marcada, la visualización de los nombres de fuentes disponibles se restringe a las fuentes donde todos los caracteres tienen el mismo ancho. (Las fuentes de ancho fijo se contrastan con las fuentes de ancho proporcional).

**Editor de código: Line Gutter**

El panel Canal de línea especifica las opciones para el canal de línea (margen izquierdo del editor de código).

**Mostrar números de línea** : si esta opción está marcada, las líneas están numeradas. (Para ir a un número de línea mientras está usando la hoja de cálculo SQL, presione Ctrl + G).

**Habilitar selección de línea haciendo clic y arrastrando** : Si esta opción está marcada, puede seleccionar líneas consecutivas en el editor haciendo clic en el margen y arrastrando el cursor sin soltar el botón del mouse.

**Editor de código: Impresión**

El panel de impresión especifica las opciones para imprimir el contenido del editor de código. La pantalla de muestra del panel Vista previa cambia a medida que selecciona y deselecciona las opciones.

**Editor de Código: Imprimiendo HTML**

El panel Imprimir HTML especifica opciones para imprimir archivos HTML desde el editor de código.

**Editor de código: guardar acciones**

El panel Acciones de guardado especifica las acciones que se realizarán automáticamente durante una operación de guardado.

**Editor de código: colores de sintaxis**

El panel Colores de sintaxis especifica colores para diferentes tipos de elementos de sintaxis.

**Editor de código: Deshacer comportamiento**

El panel Comportamiento de deshacer especifica opciones para el comportamiento de las operaciones de deshacer (Ctrl + Z, o Editar, luego Deshacer). Solo se consideran ediciones consecutivas *del mismo tipo* ; por ejemplo, la inserción de caracteres y la eliminación de caracteres son dos tipos de operaciones diferentes.

**Permitir que los cambios de solo navegación se puedan deshacer** : si esta opción está marcada, las acciones de navegación con el teclado o el mouse se pueden deshacer. Si esta opción no está marcada, las acciones de navegación no se pueden deshacer, y solo se pueden deshacer los cambios reales en el texto.

### Comparar y fusionar

El panel Comparar y fusionar define las opciones para comparar y combinar dos archivos de origen. Para obtener más información, consulte, consulte [Comparación de archivos de origen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHDCC) .

Para cada tipo de opción, puede especificar un **Tamaño máximo de archivo (KB)** : el tamaño máximo del archivo (número de kilobytes) para el cual se realizará la operación.

**Ignorar espacios en blanco** : si esta opción está habilitada, las pestañas iniciales y finales y el espacio entre letras se ignoran al comparar archivos. Los retornos de carro no son ignorados. Habilitar esta opción facilita la comparación de dos archivos cuando ha reemplazado todo el espacio con pestañas duras, o viceversa. De lo contrario, cada línea en los dos documentos puede mostrarse como diferente en la ventana Comparar.

**Mostrar diferencias de caracteres** : si esta opción está habilitada, se resaltan los caracteres que están presentes en un archivo y no en otro. El resaltado en rojo indica un carácter que se ha eliminado. El resaltado verde indica un carácter que se ha agregado. El resaltado se muestra solo cuando hace clic en un bloque de comparación que contiene diferencias de caracteres.

**Habilitar comparación de XML** : si esta opción está habilitada, los archivos XML se pueden comparar.

**Habilitar combinación de XML** : si esta opción está habilitada, los archivos XML se pueden combinar.

**Resultado de reformateo** : si esta opción está habilitada, los archivos XML combinados pueden reformatearse.

**Validar resultado (puede requerir acceso a Internet)** : si esta opción está habilitada, se validarán los archivos XML combinados.

**Comparación de archivos de origen**

Puede comparar los archivos de origen de las siguientes maneras:

* Un archivo que se está editando actualmente con su versión guardada: coloque el foco en la versión actual abierta en el editor, luego seleccione la pestaña Historial en la ventana del editor. El archivo guardado se abre al lado del archivo en el búfer del editor.
* Un archivo con otro archivo fuera del proyecto: coloque el foco en el archivo en el editor que se va a comparar; en el menú principal, seleccione **Archivo** , luego **Comparar con otro archivo** ; en el cuadro de diálogo Seleccionar archivo para comparar, navegue hasta el archivo y haga clic en Abrir.
* Dos archivos dentro del mismo proyecto: en el navegador, seleccione los dos archivos que se compararán; luego, desde el menú principal, elija **Archivo** , luego **Compare entre sí** .

### Base de datos

El panel Base de datos establece las propiedades para la conexión de la base de datos.

**Valide los valores predeterminados de fecha y hora** : si esta opción está marcada, la validación de fecha y hora se utiliza al abrir tablas.

**Ruta predeterminada para almacenar la exportación** : Ruta predeterminada del directorio o carpeta bajo la cual se almacenan los archivos de salida cuando realiza una operación de exportación. Para ver el valor predeterminado actual de su sistema, haga clic en el botón Examinar junto a este campo.

**Ejecutar el script de inicio en cada nueva conexión de base de datos** : si esta opción está marcada, el script especificado en el siguiente campo se ejecuta cada vez que se abre una conexión a una base de datos Oracle.

**Nombre de archivo para la secuencia de comandos de inicio** : nombre de archivo para que se ejecute la secuencia de comandos de inicio cuando se abre una conexión de base de datos Oracle. Puede hacer clic en **Examinar** para especificar la ubicación. La ubicación predeterminada es la ruta predeterminada para los scripts (consulte el panel de preferencias [Base de datos: Parámetros de la](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDGCI) hoja de trabajo).

**Base de datos: Parámetros avanzados**

El panel Parámetros avanzados especifica opciones como el tamaño de búsqueda de matriz SQL y las opciones de visualización para valores nulos.

También puede especificar los parámetros de configuración del controlador delgado de Kerberos, lo que le permite crear conexiones de base de datos utilizando la autenticación Kerberos y especificando el nombre de usuario y la contraseña. Para obtener más información, consulte la explicación de la Autenticación Kerberos en la [pestaña Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEDDJF) en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) . Para obtener información sobre la configuración de la autenticación Kerberos, consulte [*la Guía del administrador de seguridad avanzada de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/network.112/e40393/asokerb.htm#ASOAG060) .

**Use el controlador OCI / Thick** : si esta opción está marcada, y si está disponible un controlador OCI (grueso, Tipo 2), se usará ese controlador en lugar de un controlador JDBC (delgado) para conexiones de base de datos básicas y TNS (alias de red).

**Kerberos Thin Config: archivo de configuración** : archivo de configuración de Kerberos (por ejemplo, krb5.conf).

**Kerberos Thin Config: Archivo de caché de credenciales: Archivo de** caché de credenciales de Kerberos (por ejemplo, krb5\_cc\_cache).

**Base de datos: parámetros de autotrace**

El panel Parámetros de seguimiento automático especifica la información que se mostrará en el panel Seguimiento automático de la hoja de cálculo de SQL.

**Base de datos: Parámetros generales de exportación**

**Delimitador de exportación personalizado** : el carácter que se utilizará como delimitador al exportar datos de tablas en formato CSV. Esta opción le permite utilizar un carácter distinto de la coma predeterminada (,) como delimitador CSV.

**Base de datos: Parámetros NLS**

El panel Parámetros NLS especifica valores para los parámetros de compatibilidad con la globalización, como el idioma, el territorio, la preferencia de ordenación y el formato de fecha. Estos valores de parámetros se utilizan para las operaciones de sesión del Desarrollador de SQL, como las declaraciones ejecutadas con la hoja de cálculo de SQL y para el informe de Parámetros de compatibilidad de idioma nacional. La especificación de valores en este panel de preferencias *no* aplica esos valores a la base de datos subyacente. Para cambiar la configuración de la base de datos, debe cambiar los parámetros de inicialización apropiados y reiniciar la base de datos.

Tenga en cuenta que SQL Developer no utiliza los valores predeterminados del sistema actual para los parámetros de soporte de globalización; en su lugar, SQL Developer, cuando se instala inicialmente, usa valores de parámetros que incluyen lo siguiente:

NLS\_LANG, "AMERICAN"

NLS\_TERR, "AMERICA"

NLS\_CHAR, "AL32UTF8"

NLS\_SORT, "BINARIO"

NLS\_CAL, "GREGORIAN"

NLS\_DATE\_LANG, "AMERICAN"

NLS\_DATE\_FORM, "DD-MON-RR"

**Base de datos: parámetros de ObjectViewer**

El panel Parámetros de ObjectViewer especifica si se deben congelar las ventanas del visor de objetos y mostrar las opciones para la salida. Las opciones de visualización afectarán al DDL generado en la pestaña SQL. Las opciones del editor de datos afectan el comportamiento cuando está utilizando la pestaña Datos para editar datos de la tabla.

**Opciones del editor de datos**

**Publicar ediciones en el cambio de fila** : si esta opción está marcada, se publican cambios en DML cuando realiza ediciones utilizando la pestaña Datos (y la opción Establecer compromiso automático activado determina si los cambios se confirman automáticamente). Si esta opción no está marcada, los cambios se publican y confirman cuando presiona el botón de la barra de herramientas Confirmar.

**Activar** confirmación automática activada (disponible solo si la opción Editar edición en cambios de fila está habilitada): si esta opción está marcada, los cambios en DML se publicarán y confirmarán automáticamente cuando realice ediciones utilizando la pestaña Datos.

**Borre las configuraciones de ancho, columna, orden y filtro de las columnas de la tabla persistentes** : si hace clic en **Borrar** , todas las personalizaciones en la pestaña Datos se mostrarán para los anchos de columna, orden, clasificación y filtrado de la tabla, pero no se guardarán para las siguientes aperturas de la pestaña, sino que La configuración predeterminada se utiliza para las aperturas posteriores.

**Base de datos: Opciones de compilador PL / SQL**

El panel de Opciones del compilador de PL / SQL especifica las opciones para la compilación de subprogramas de PL / SQL.

**Generar información de depuración de PL / SQL** : Si esta opción está marcada, la información de depuración de PL / SQL se incluye en el código compilado; Si esta opción no está marcada, esta información de depuración no se incluye. La capacidad de detenerse en líneas de código individuales y el acceso del depurador a las variables se permite solo en el código compilado con la información de depuración generada.

**Tipos de mensajes** : puede controlar la visualización **de mensajes** informativos, graves y relacionados con el rendimiento. (El tipo ALL anula cualquier especificación individual para los otros tipos de mensajes). Para cada tipo de mensaje, puede especificar cualquiera de los siguientes:

* Sin entrada (en **blanco** ): Use cualquier valor especificado para TODOS; y si no se especifica ninguno, use el valor predeterminado de Oracle.
* **Habilitar** : habilite la visualización de todos los mensajes de esta categoría.
* **Deshabilitar** : deshabilita la visualización de todos los mensajes de esta categoría.
* **Error** : habilite la visualización de solo los mensajes de error de esta categoría.

**Nivel de optimización** : 1, 2 o 3, que refleja la cantidad y el tipo de cambios realizados (desde el impacto más bajo hasta el impacto más alto). Los niveles más altos generalmente resultan en un mejor rendimiento, pero un tiempo de compilación más largo.

**PLScope Identifiers** : especifica la cantidad de datos de identificación de PL / Scope que se deben recopilar y utilizar (todos o ninguno).

**Base de datos: Informes**

El panel Informes especifica las opciones relacionadas con los informes de SQL Developer.

**Cerrar todos los informes de desconexión** : si esta opción está marcada, todos los informes de cualquier conexión de base de datos se cierran automáticamente cuando se desconecta esa conexión.

**Base de datos: Plantillas de código del editor SQL**

El panel Plantillas de código del editor de SQL le permite ver, agregar y eliminar plantillas para editar código SQL y PL / SQL. Las plantillas de código lo ayudan a escribir el código de manera más rápida y eficiente al insertar texto para las declaraciones de uso común. A continuación, puede modificar el texto insertado.

Para insertar el contenido de una plantilla de código en la hoja de trabajo SQL, coloque el cursor en el punto donde se va a insertar la plantilla, escriba la ID asociada con la plantilla y luego presione Ctrl + Mayús + T. (Nota: Ctrl + Shift + T es el acelerador asignado en el mapa de teclas predeterminado, pero puede asignar otra asignación).

Para incluir una instrucción SELECT en un bucle FOR, seleccione (resalte) la instrucción SELECT y presione Ctrl + Shift + T.

**Agregar plantilla** : agrega una fila vacía en la pantalla de plantilla de código. Ingrese un valor de ID, luego mueva a la celda Plantilla; puede ingresar el contenido de la plantilla en esa celda o hacer clic en el botón de puntos suspensivos (...) para abrir el editor de código para ingresar el contenido de la plantilla.

**Eliminar plantilla** : elimina la plantilla de código seleccionada.

**Base de datos: formateador de SQL**

El panel del formateador de SQL controla cómo se formatean las declaraciones en la hoja de cálculo de SQL al hacer clic en Formatear SQL. Las opciones incluyen si se deben insertar caracteres de espacio o caracteres de tabulación al presionar la tecla Tab (y cuántos caracteres), mayúsculas o minúsculas para palabras clave e identificadores, si se deben conservar o eliminar las líneas vacías, y si se deben colocar elementos comparables o iguales. línea (si hay espacio) o en líneas separadas.

**Base de datos: Controladores JDBC de terceros**

El panel Controladores JDBC de terceros especifica los controladores que se utilizarán para las conexiones a bases de datos de terceros (no Oracle), como MySQL, Microsoft SQL Server o Sybase Adaptive Server. (No es necesario agregar un controlador para las conexiones a las bases de datos de Microsoft Access). Para agregar un controlador, haga clic en **Agregar entrada** y seleccione la ruta para el controlador (por ejemplo, un archivo con un nombre similar a mysql-connector-java-5.0.4-bin.jar, en un directorio debajo del en el que descomprimió la descarga del controlador MySQL, o jtds-1.2.jar, que se incluye en la jtds-1.2-dist.zipdescarga, para Microsoft SQL Server o Sybase Adaptive Server).

**Alternativa:**

Como alternativa al uso de esta preferencia, puede hacer clic en **Ayuda** , luego en **Buscar actualizaciones** para instalar el controlador JTBC JTDS para Microsoft SQL Server y el controlador JDBE de MySQL como extensiones.

Para encontrar un controlador JDBC de terceros específico, consulte el sitio web apropiado (por ejemplo, [http://www.mysql.com](http://www.mysql.com/)para el controlador MySQL Connector / J JDBC para MySQL, o <http://jtds.sourceforge.net/>para el controlador jTDS para Microsoft SQL Server y Sybase Adaptive Server). Para MySQL, use el controlador MySQL 5.0, no 5.1 o posterior, con la versión 1.5 del desarrollador de SQL.

Debe especificar un controlador JDBC de terceros o instalar un controlador utilizando la función Buscar actualizaciones antes de poder crear una conexión de base de datos a una base de datos de terceros de ese tipo asociado. (Consulte las pestañas para crear conexiones a bases de datos de terceros en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) ).

**Base de datos: Extensiones definidas por el usuario**

El panel de extensiones definidas por el usuario especifica las extensiones definidas por el usuario que se han agregado. Puede usar este panel para agregar extensiones que no están disponibles a través de la función Buscar actualizaciones. (Para obtener más información acerca de las extensiones y la comprobación de actualizaciones, consulte [Extensiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBBBID) ).

Un uso del panel Base de datos: Extensiones definidas por el usuario es crear una carpeta de Informes compartidos e incluir un informe exportado en esa carpeta: haga clic en Agregar fila, especifique Tipo como INFORME, y para Ubicación, especifique el archivo XML que contiene el informe exportado. La próxima vez que reinicie SQL Developer, el navegador de informes tendrá una carpeta de informes compartidos que contiene ese informe

**Base de datos: Parámetros de la hoja de trabajo**

**Confirmar automáticamente en la hoja de trabajo de SQL** : si esta opción está marcada, una operación de confirmación se realiza automáticamente después de cada instrucción INSERT, UPDATE o DELETE ejecutada utilizando la hoja de trabajo de SQL. Si esta opción no está marcada, no se realiza una operación de confirmación hasta que ejecute una instrucción COMMIT.

**Abra una hoja de trabajo al conectarse** : si esta opción está marcada, se abrirá automáticamente una ventana de **hoja de trabajo** SQL para la conexión cuando abra una conexión de base de datos. Si esta opción no está marcada, debe usar el comando del botón derecho Abrir hoja de trabajo SQL o el icono de la barra de herramientas para abrir una hoja de trabajo SQL.

**Cerrar todas las hojas de trabajo al desconectar** : si esta opción está marcada, todas las ventanas de la **hoja de trabajo** SQL para cualquier conexión de base de datos se cierran automáticamente cuando se desconecta esa conexión.

Número **máximo de filas para imprimir en un script** : limita el número de filas mostradas.

**Ruta predeterminada para buscar secuencias de comandos** : el directorio predeterminado donde se ve SQL Developer cuando ejecuta una secuencia de comandos (con @).

**Guardar las variables de vinculación en el disco al salir** : si esta opción está marcada, las variables de vinculación que ingresa al ejecutar un script se guardan en el disco para su reutilización. Si no desea que los valores de las variables de enlace se almacenen en el disco (por razones de seguridad u otras), asegúrese de no marcar esta opción.

**Efectos de arrastrar y soltar:** determina el tipo de instrucción SQL creada en la hoja de trabajo de SQL cuando arrastra un objeto desde el navegador de conexiones a la hoja de trabajo de SQL. La preferencia de desarrollador de SQL establece el valor predeterminado, que puede anular en el cuadro de diálogo Efectos de arrastrar y soltar.

El tipo de declaración (INSERTAR, BORRAR, ACTUALIZAR o SELECCIONAR) se aplica solo a los tipos de objetos para los cuales es posible dicha declaración. Por ejemplo, SELECT tiene sentido para una tabla, pero no para un disparador. Para los objetos para los cuales no se aplica el tipo de declaración, el nombre del objeto se inserta en la hoja de trabajo de SQL.

### Depurador

El panel del depurador contiene opciones generales para el depurador del desarrollador de SQL. Otros paneles contienen tipos específicos adicionales de opciones de depurador.

**Depurador: puntos de interrupción**

El panel Puntos de interrupción establece las columnas para que aparezcan en el panel Puntos de interrupción y el alcance de cada punto de interrupción.

**Depurador: Puntos de interrupción: Acciones predeterminadas**

El panel Puntos de interrupción: Acciones predeterminadas establece valores predeterminados para que las acciones ocurran en los puntos de interrupción. Estas acciones son las mismas que en la [pestaña Acciones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEIIDG) en el cuadro de diálogo [Crear / Editar punto de interrupción](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJIFJD) .

**Depurador: Datos**

El panel Datos le permite controlar las columnas para que aparezcan en el panel Datos del depurador y aspectos de cómo se muestran los datos.

**Depurador: inspector**

El panel Inspector le permite controlar las columnas para que aparezcan en el panel Inspector del depurador y aspectos de cómo se muestran los datos.

**Depurador: Smart Data**

El panel Datos inteligentes le permite controlar las columnas que aparecen en el panel Datos inteligentes del depurador y aspectos de cómo se muestran los datos.

**Depurador: pila**

El panel Pila le permite controlar las columnas para que aparezcan en el panel Pila del depurador y otras opciones.

**Depurador: Relojes**

El panel Relojes le permite controlar las columnas que aparecen en el panel Relojes del depurador y aspectos de cómo se muestran los datos.

### Extensiones

El panel de Extensiones determina qué extensiones opcionales utiliza SQL Developer cuando se inicia. (SQL Developer también usa algunas extensiones obligatorias, que los usuarios no pueden eliminar ni desactivar). Si cambia alguna configuración, debe salir de SQL Developer y reiniciarla para que la nueva configuración tenga efecto.

Para el soporte de versiones, la configuración (seleccionada o no, y las opciones de configuración si se seleccionan) afectan si se muestra el menú de versiones y los elementos en ese menú.

**Extensiones de uso** : controla las extensiones específicas opcionales de desarrollador de SQL para usarlas en el inicio.

**Buscar actualizaciones** : comprueba si hay actualizaciones para las extensiones opcionales seleccionadas de SQL Developer, así como también cualquier extensión obligatoria. (Si el sistema que está utilizando está detrás de un servidor de seguridad, consulte las preferencias de usuario del desarrollador de SQL para [el navegador web y el proxy](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBFJA) ).

**Buscar actualizaciones automáticamente** : si esta opción está marcada, SQL Developer busca automáticamente las actualizaciones de las extensiones opcionales seleccionadas de SQL Developer y las extensiones obligatorias al inicio.

### Tipos de archivo

El panel Tipos de archivos determina qué tipos de archivos y extensiones se abrirán de forma predeterminada por el Desarrollador de SQL. La pantalla muestra cada extensión de archivo, el tipo de archivo asociado y una marca de verificación si los archivos con esa extensión deben ser abiertos por el Desarrollador de SQL por defecto, como cuando un usuario hace doble clic en el nombre del archivo.

Área de **detalles** en la parte inferior: puede modificar el tipo de archivo, el tipo de contenido (texto o binario), y si desea abrir archivos con esta extensión automáticamente por el Desarrollador de SQL.

Para que los archivos con una extensión específica sean abiertos de manera predeterminada por el Desarrollador de SQL, haga clic en la extensión del archivo en la lista, luego marque Abrir con el Desarrollador de SQL en el área de Detalles. Esto anula cualquier asociación de aplicación anterior que pueda haber estado vigente para esa extensión de archivo.

Para agregar una extensión de archivo, haga clic en **Agregar** y especifique la extensión del archivo (incluido el período). Después de agregar la extensión, puede modificar su información asociada seleccionándola y usando el área de Detalles.

### Lista global de ignorados

El panel Lista de ignorados global especifica filtros que determinan qué archivos y tipos de archivos no se utilizarán en ningún procesamiento.

**Nuevo filtro** : un nombre de archivo o tipo de archivo que desea agregar a la lista de archivos y tipos de archivos (en el cuadro Filtro) que el desarrollador de SQL ignorará durante todo el procesamiento (si el filtro está habilitado o marcado). Puede excluir un archivo en particular ingresando su nombre de archivo completo, como mumble.txt, o puede excluir todos los archivos del mismo tipo ingresando una construcción que describa el tipo de archivo, como \*.txt.

**Agregar** : agrega el nuevo filtro a la lista en el cuadro Filtro.

**Eliminar** : elimina el filtro seleccionado de la lista en el cuadro Filtro.

**Restaurar valores predeterminados** : restaura el contenido del cuadro Filtro a los valores predeterminados del desarrollador de SQL.

**Filtro** : Contiene la lista de archivos y tipos de archivos. Para cada elemento, si está habilitado (marcado), el filtro se aplica y el desarrollador o el tipo de archivo es ignorado por SQL Developer; pero si está deshabilitado (sin marcar), el filtro no se aplica.

### Migración

El panel Migración contiene opciones que afectan el comportamiento de SQL Developer cuando migra objetos de esquema y datos de bases de datos de terceros a una base de datos Oracle.

**Repositorio predeterminado** : repositorio de migración que se utilizará para almacenar los modelos capturados y los modelos convertidos. Para obtener información sobre la migración de bases de datos de terceros a Oracle, incluido cómo crear un repositorio de migración, consulte el [Capítulo 2](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDECEIA) .

**Migración: Opciones de traslado de datos**

El panel Opciones de traslado de datos contiene opciones que afectan el comportamiento cuando migra datos de bases de datos de terceros a tablas de base de datos Oracle generadas por la migración.

**En línea para todos. Sin conexión para MySQL, SQL Server y Sybase Adaptive Server:** opciones que se pueden usar para la migración de datos en línea para todas las bases de datos de terceros compatibles y para la migración de datos sin conexión para MySQL, SQL Server y Sybase Adaptive Server.

**Representación para 0 serie de longitud** : el valor al que Oracle convierte las cadenas de longitud cero en los datos de origen. Puede ser un espacio ('') o un valor nulo (NULL). Notas específicas:

* Para las migraciones sin conexión de Microsoft Access, un valor nulo y un espacio se consideran lo mismo.
* Para las migraciones sin conexión de Sybase, '' se considera lo mismo que un espacio ('').
* Para las migraciones sin conexión de MySQL, un valor nulo se exporta como 'NULL', que se maneja como tipo VARCHAR2. Puede especificar otro carácter de escape utilizando la opción --fields-escaped-by con el comando mysqldump (por ejemplo, especificando \Npara nulo o \\para \). Para obtener información sobre el comando mysqldump, consulte [Crear archivos de datos desde MySQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDDCBC) .

Para las migraciones sin conexión de MySQL, los datos se exportan a un archivo llamado *nombre-tabla* .txt; por lo tanto, si está moviendo datos de dos o más tablas con el mismo nombre pero en diferentes esquemas, cambie el nombre de los archivos según sea necesario para que sean todos únicos, y modifique el archivo .ctl de SQL \* Loader en consecuencia.

**En línea** : las opciones de movimiento de datos en línea determinan los resultados de los archivos creados al hacer clic en **Migración** y luego en **Migrar datos** .

**Número de flujos de movimiento de datos paralelos** : el número de conexiones internas creadas para el movimiento simultáneo de datos desde la base de datos de origen a las tablas de Oracle. Los valores más altos pueden acortar el tiempo total requerido, pero usarán más recursos de base de datos durante ese tiempo.

**Número de filas para confirmar después** : durante la operación de traslado de datos, Oracle hace una pausa para realizar una operación de confirmación interna automática después de que cada número de filas que especifique se mueva de la base de datos de origen a las tablas de Oracle.

Los valores más bajos harán que una operación de movimiento exitosa tarde más tiempo; pero si se produce un error, es probable que existan más registros de origen en las tablas de Oracle y que si se reanuda la operación de traslado, será necesario mover menos registros de origen. Los valores más altos harán que una operación de movimiento exitosa tome menos tiempo; pero si se produce un error, es probable que existan menos registros de origen en las tablas de Oracle y, si se reanuda la operación de traslado, será necesario mover más registros de origen.

**Fuera de línea** : las opciones de movimiento de datos fuera de línea determinan los resultados de los archivos creados al hacer clic en **Migración** , luego en **Generar secuencias de comandos** y luego en **Generar datos de secuencias de comandos de movimiento** .

**Directorio de secuencias de datos sin conexión** : ubicación predeterminada para secuencias de comandos para operaciones de movimiento de datos sin conexión.

**Delimitador de final de columna** : cadena para indicar el final de la columna.

**Delimitador de fin de fila** : cadena para indicar el fin de fila.

**Máscara de fecha** : Formato de máscara para fechas.

**Máscara de marca de tiempo** : Formato de máscara para marcas de tiempo.

**Migración: Opciones de generación**

El panel Opciones de generación contiene opciones que determinan los resultados de los archivos creados al hacer clic en **Migración** , luego **Generar secuencias de comandos** y luego **Generar DDL de Oracle** .

**One single file**, **A file per object**, or **A file per database**: Determines how many files are created and their relative sizes. Having more files created might be less convenient, but may allow more flexibility with complex migration scenarios. (See also the Maximum Number of Lines option.)

**Output Directory**: Default location in which the files will be created.

I**mplement 'CREATE' as 'CREATE OR REPLACE'**: Causes CREATE statements in source database objects to be implemented using CREATE OR REPLACE when the Oracle syntax allows this.

**Generate Comments**: Generates comments in the Oracle SQL statements.

**Generar secuencia de comandos de control** : Genera una secuencia de comandos "maestra" para ejecutar todos los archivos necesarios.

**Número máximo de líneas** : establece un número máximo de líneas para cada archivo; a continuación, especifique el número.

**Migración de esquema con privilegios mínimos** : para migrar objetos de esquema en un modelo convertido a Oracle, las sentencias CREATE USER, GRANT y CONNECT *no* se generan en los scripts de salida. A continuación, debe asegurarse de que los scripts se ejecutan utilizando una conexión con privilegios suficientes. Puede seleccionar esta opción si el usuario de la base de datos y la conexión que desea usar para ejecutar los scripts ya existen, o si planea crearlos.

**Generar usuario de movimiento de datos** : para las operaciones de movimiento de datos, crea un usuario de base de datos adicional con privilegios adicionales para realizar la operación. Se recomienda que elimine este usuario después de la operación. Esta opción se proporciona para su comodidad y se sugiere a menos que desee realizar migraciones con menos privilegios o que desee otorgar privilegios manualmente a un usuario para las operaciones de traslado de datos. Esta opción se recomienda especialmente para migraciones multisquemas, como cuando no todas las tablas pertenecen a un solo usuario.

**Generar objetos con error** : hace que los objetos que no se hayan convertido se incluyan en el script de generación, de modo que pueda realizar los cambios deseados y luego ejecutar el script. Si esta opción no está marcada, los objetos que no se pudieron convertir no se incluyen en el script de generación.

**Generar procedimiento almacenado para migrar blobs sin conexión** : hace que se cree un procedimiento almacenado llamado CLOBtoBLOB\_sqldeveloper (con acceso de ejecución otorgado al público) si el esquema contiene un BLOB (objeto binario grande); este procedimiento se llama automáticamente si realiza una captura fuera de línea. Si esta opción no está marcada, deberá usar la solución manual descrita en Cómo completar [la base de datos de destino con los archivos de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDEAJCE) . (Después de la captura fuera de línea, puede eliminar el procedimiento CLOBtoBLOB\_sqldeveloper o eliminar el acceso de ejecución del público).

**Migración: Opciones de identificador**

El panel Opciones de identificador contiene opciones que se aplican a los identificadores de objetos durante las migraciones.

**Preparado para todos los nombres de identificadores** (solo para migraciones de Microsoft Access, Microsoft SQL Server y Sybase Adaptive Server): una cadena que se agregará al comienzo del nombre de los objetos migrados. Por ejemplo, si especifica la cadena como XYZ\_, y si una tabla fuente se llama EMPLEADOS, la tabla migrada se llamará XYZ\_EMPLOYEES. (Tenga en cuenta las restricciones de longitud del nombre del objeto si utiliza esta opción).

**Identificador de** comillas activadas (solo para migraciones de Microsoft SQL Server y Sybase Adaptive Server): si esta opción está habilitada, se pueden usar comillas (comillas dobles) para referirse a los identificadores (por ejemplo, SELECCIONE "Col 1" de "Tabla 1" ); Si esta opción no está habilitada, las comillas identifican literales de cadena. **Importante** : la configuración de esta opción debe coincidir con la configuración en la base de datos de origen que se va a migrar, como se explica en [Antes de migrar desde Microsoft SQL Server o Sybase Adaptive Server](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005806) .

**Migración: Opciones de traducción**

El panel de Opciones de traducción contiene opciones que se relacionan con la conversión de procedimientos almacenados y funciones de su formato de base de datos de origen al formato de Oracle.

**Formato de fecha de origen predeterminado** : Máscara de formato de datos predeterminado para fechas en los datos de origen.

**Visor de** diferencias de traducción: hay varias opciones que afectan la visualización cuando utiliza la función del visor de diferencias de traducción.

### Versiones

Las preferencias de versión afectan el comportamiento de los sistemas de control de versión y administración que puede usar con el Desarrollador SQL. Puede especificar preferencias para CVS y Subversion. Para obtener información sobre el uso del control de versiones con SQL Developer, consulte [Uso del control de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) .

**Versiones: CVS**

El panel CVS especifica opciones para usar con CVS (Sistema de versiones concurrentes).

**Cliente CVS** : **interno para Oracle SQL Developer** (instalado con SQL Developer) o **ejecutable externo** (cliente CVS instalado por separado, para el cual debe especificar el nombre o la ruta).

* **Nombre en la ruta del sistema** : Nombre del ejecutable del servidor CVS. El valor predeterminado (cvs) es correcto para la mayoría de las instalaciones. Esta opción asume que el nombre del ejecutable del servidor CVS está en la ruta del sistema.
* **Ruta desde el entorno** : ubicación del ejecutable del servidor CVS, especialmente si hay más de uno en la ruta del sistema. El área de selección mostrará una lista de todas las instancias del ejecutable del servidor CVS conocido por el sistema local. Es posible que tenga más de una versión de CVS instalada: esta opción le permite especificar cuál de ellas usar con el Desarrollador SQL.
* **Otra ruta** : Ubicación del ejecutable del servidor CVS, si no está en la ruta del sistema.

**Ejecute CVS en el modo Edición / Vigilancia** : si esta opción está habilitada, usted coordina el acceso a los archivos declarando un editor para ellos a través de CVS, después de lo cual pueden ser modificados. Solo se verán afectados los archivos que extraiga después de cambiar esta preferencia. Si esta opción está deshabilitada, los comandos de edición y visualización en el menú Versiones están deshabilitados.

**Esquema de superposición del estado** : Esquema para los iconos que se muestran junto a los nombres de carpetas y archivos en los navegadores para indicar su estado de versiones.

**Versiones: CVS: Comandos**

El panel CVS: Comandos establece las opciones para el control de fuente CVS. Algunas opciones no están disponibles cuando se utiliza el cliente CVS interno.

**Habilitar controles avanzados** : si esta opción está habilitada, los controles CVS avanzados se muestran en los cuadros de diálogo. Si descubre que solo utiliza las funciones básicas de CVS, es posible que desee usar SQL Developer sin controles avanzados, para reducir la complejidad y ahorrar espacio en la pantalla.

**Opciones globales: Ejecutar en silencio** : si esta opción está habilitada, los mensajes informativos se suprimen.

**Opciones globales: No registrar comandos** : si esta opción está habilitada, los comandos CVS no se registran en el historial de comandos del repositorio.

**Opciones globales: Encriptar** : si esta opción está habilitada, todas las comunicaciones entre el cliente y el servidor están encriptadas. El soporte de cifrado no está disponible en CVS de forma predeterminada; debe habilitarse usando una opción de configuración especial cuando construya CVS.

**Establecer nivel de compresión (z)** : si esta opción está habilitada, puede establecer el nivel de compresión para los archivos enviados entre el cliente y el servidor. El nivel se puede configurar desde Mínimo (alta velocidad, baja compresión) hasta Máximo (baja velocidad, alta compresión).

**Modo de sustitución de palabras clave** : CVS utiliza los modos de sustitución de palabras clave para insertar información de revisión en los archivos cuando se extraen o actualizan. Esta opción controla el modo de reemplazo para la sustitución de palabras clave en archivos versionados:

* **Automático** : la opción por defecto, recomendada.
* **Modo de** solo palabra clave: Genera solo nombres de palabras clave en cadenas de palabras clave y omite sus valores. Esta opción es útil para ignorar las diferencias debidas a la sustitución de palabras clave cuando se comparan diferentes revisiones de un archivo.
* **Modo de valor de** palabra clave: Genera cadenas de palabras clave utilizando el formulario predeterminado.
* **Modo de** bloqueo de valor de palabra clave: como el modo de valor de palabra clave, excepto que el nombre del bloqueo siempre se inserta si la revisión dada está actualmente bloqueada.
* **Modo de contenido antiguo** : Genera la cadena de palabra clave anterior, presente en el archivo de trabajo justo antes de que se registrara.
* **Modo de** solo valor: Genera solo valores de palabras clave para cadenas de palabras clave. Esto puede ayudar a generar archivos en lenguajes de programación donde es difícil eliminar los delimitadores de palabras clave de una cadena. Sin embargo, una mayor sustitución de palabras clave no se puede realizar una vez que se eliminan los nombres de palabras clave, por lo que esta opción debe usarse con cuidado.

**Activar: Usar plantillas de comentarios** : si esta opción está habilitada, sus comentarios de confirmación se ingresarán a través de formularios de plantilla. Los formularios son configurados por el administrador del sistema CVS. Puede haber diferentes formas para diferentes circunstancias e instalaciones, y puede ser que ninguna de ellas sea adecuada para sus comentarios de confirmación. En este caso, esta preferencia le permite desactivar el uso de todos los formularios.

**En Confirmar: Agregar archivos automáticamente** : si esta opción está habilitada, los archivos locales se agregan al repositorio de CVS cada vez que realiza una acción de confirmación.

**Crear archivos de copia de seguridad al eliminar** : si esta opción está habilitada, se realizan copias de seguridad de los archivos que se eliminan a través de las acciones del sistema de control de origen.

**Versiones: CVS: General**

El panel CVS: General especifica la configuración del entorno y el tiempo de espera de la operación.

**Usar iconos de superposición de estado del navegador** : si esta opción está habilitada, se utilizan los iconos de superposición de estado. Los iconos de superposición de estado son pequeños símbolos asociados con nombres de objetos en los navegadores. Indican el estado de los archivos controlados por versión (por ejemplo, "actualizado").

**Utilizar etiquetas de superposición de estado de Navigator** : si esta opción está habilitada, se utilizan etiquetas de superposición de estado. Las etiquetas de superposición de estado son información sobre herramientas asociadas con nombres de objetos en los navegadores.

**Hacer que los archivos sean editables automáticamente** : si esta opción está habilitada, un editor se usa automáticamente en un archivo de datos cuando comienza a cambiarlo. (Si edita un archivo de forma no intencionada, use inmediatamente el control de versiones y luego vuelva a deseleccionar).

**Tiempo de espera de la operación** : tiempo máximo permitido para que se completen las operaciones de CVS.

**Versiones: CVS: Navigator Labels**

El panel CVS: Etiquetas de navegador especifica el formato para la información de CVS que aparece en los nodos de navegación y en las sugerencias de herramientas. Para obtener una explicación completa de los modos de sustitución de palabras clave, consulte la documentación de CVS.

**Versiones: CVS: Herramientas de versión**

El panel CVS: Herramientas de la versión especifica las opciones para la ventana de cambios pendientes y el editor de combinación.

**Cuadro de diálogo Utilizar cambios salientes** : le permite hacer un uso óptimo del espacio limitado de la pantalla cuando la ventana Cambios pendientes está abierta. Puede ahorrar espacio en la pantalla al no mostrar el área de Comentarios de la ventana Cambios pendientes, pero es posible que desee agregar comentarios antes de una acción de confirmación. Puede elegir las circunstancias en las que se abre el cuadro de diálogo Confirmar: siempre, solo cuando el área Comentarios de la ventana Cambios pendientes está oculta, o nunca.

**Intervalo del temporizador de cambios entrantes** : la frecuencia con la que se verifica el estado de cambio de los archivos.

**Editor de combinación** : especifica si los archivos se fusionan localmente o en el servidor.

**Versiones: Subversion**

El panel de Subversion especifica el cliente de Subversion para usar con SQL Developer.

**Versiones: Subversion: General**

El panel Subversión: General especifica la configuración del entorno y el tiempo de espera de la operación.

**Usar iconos de superposición de estado del navegador** : si esta opción está habilitada, se utilizan los iconos de superposición de estado. Los iconos de superposición de estado son pequeños símbolos asociados con nombres de objetos en los navegadores. Indican el estado de los archivos controlados por versión (por ejemplo, "actualizado").

**Utilizar etiquetas de superposición de estado de Navigator** : si esta opción está habilitada, se utilizan etiquetas de superposición de estado. Las etiquetas de superposición de estado son información sobre herramientas asociadas con nombres de objetos en los navegadores.

**Hacer que los archivos sean editables automáticamente** : si esta opción está habilitada, un editor se usa automáticamente en un archivo de datos cuando comienza a cambiarlo. (Si edita un archivo de forma no intencionada, use inmediatamente el control de versiones y luego vuelva a deseleccionar).

**Tiempo de espera de la operación** : tiempo máximo permitido para que se completen las operaciones de Subversion.

**Versiones: Subversion: Herramientas de versión**

El panel Subversion: Herramientas de versión especifica las opciones para la ventana de cambios pendientes y el editor de combinación.

**Cuadro de diálogo Utilizar cambios salientes** : le permite hacer un uso óptimo del espacio limitado de la pantalla cuando la ventana Cambios pendientes está abierta. Puede ahorrar espacio en la pantalla al no mostrar el área de Comentarios de la ventana Cambios pendientes, pero es posible que desee agregar comentarios antes de una acción de confirmación. Puede elegir las circunstancias en las que se abre el cuadro de diálogo Confirmar: siempre, solo cuando el área Comentarios de la ventana Cambios pendientes está oculta, o nunca.

**Intervalo del temporizador de cambios entrantes** : la frecuencia con la que se verifica el estado de cambio de los archivos.

**Editor de combinación** : especifica si los archivos se fusionan localmente o en el servidor.

### Navegador web y proxy

La configuración del panel Proxy y navegador web son relevantes solo cuando usa la función Buscar actualizaciones (haga clic en **Ayuda** , luego **Buscar actualizaciones** ), y solo si su sistema está detrás de un firewall.

**Línea de comandos del navegador** : para especificar un navegador web que no sea su navegador predeterminado, especifique el archivo ejecutable para iniciar ese navegador. Para usar su navegador predeterminado, deje este campo en blanco.

**Use HTTP Proxy Server** : revise las opciones o preferencias de su navegador web para los valores apropiados para estos campos.

### Ubicación de la información relacionada con el usuario

SQL Developer almacena información relacionada con el usuario en varios lugares, con la ubicación específica dependiendo del sistema operativo y ciertas especificaciones del entorno. La información relacionada con el usuario incluye informes definidos por el usuario, fragmentos definidos por el usuario, historial de la hoja de trabajo SQL, plantillas de código y preferencias de usuario del desarrollador SQL. En la mayoría de los casos, su información relacionada con el usuario se almacena fuera de la jerarquía del directorio de instalación de SQL Developer, de modo que se conserva si elimina ese directorio e instala una nueva versión.

La información relacionada con el usuario se almacena en o debajo de la siguiente ubicación:

* En sistemas Windows: la HOMEubicación de la variable de entorno, si está definida; de lo contrario, la SQLDEVELOPER\_USER\_DIRubicación, si se define; De lo contrario como se indica en la siguiente tabla.
* En sistemas Linux y Mac OS X: la SQLDEVELOPER\_USER\_DIRubicación, si está definida; De lo contrario como se indica en la siguiente tabla.

La siguiente tabla muestra las ubicaciones predeterminadas típicas (en un directorio o en un archivo) para tipos específicos de recursos en diferentes sistemas operativos. (Tenga en cuenta el período en el nombre de cualquier directorio nombrado .sqldeveloper.)

***Tabla 1-1 Ubicaciones predeterminadas para información relacionada con el usuario***

| **Tipo de recurso** | **Sistema (Windows, Linux o Mac OS X)** |
| --- | --- |
| Informes definidos por el usuario | Windows: C: \ Documents and Settings \ *<nombre de usuario>* \ Application Data \ SQL Developer \ UserReports.xml  Linux o Mac OS X: ~ / .sqldeveloper / UserReports.xml |
| Fragmentos definidos por el usuario | Windows: C: \ Documents and Settings \ <nombre de usuario> \ Application Data \ SQL Developer \ UserSnippets.xml  Linux: ~ / .sqldeveloper / UserSnippets.xml  Mac OS X: / Users / <Your user> / Library / Application Support / SQLDeveloper / UserSnippets.xml |
| Historial de SQL | Windows: C: \ Documents and Settings \ *<nombre de usuario>* \ Application Data \ SQL Developer \ SqlHistory.xml  Linux: ~ / .sqldeveloper / SqlHistory.xml  Mac OS X: / Users / <Your user> / Library / Application Support / SQLDeveloper / SqlHistory.xml |
| Plantillas de código | Windows: C: \ Documents and Settings \ <nombre de usuario> \ Application Data \ SQL Developer \ CodeTemplate.xml  Linux: ~ / .sqldeveloper / CodeTemplate.xml  Mac OS X: / Users / <Your user> / Library / Application Support / SQLDeveloper / CodeTemplate.xml |
| Preferencias de usuario de desarrollador de SQL | Windows: C: \ Documents and Settings \ <nombre de usuario> \ Application Data \ SQL Developer \ system *n.nnnn*  Linux o Mac OS X: ~ / .sqldeveloper / system *n.nnnn* |

Para especificar una SQLDEVELOPER\_USER\_DIRubicación no predeterminada , realice una de las siguientes acciones :

* Establezca la SQLDEVELOPER\_USER\_DIRvariable de entorno para especificar otra ruta de directorio.
* Edite el *<sqldeveloper\_install>*\sqldeveloper\sqldeveloper\bin\sqldeveloper.confarchivo y sustituya la ruta del directorio que desee SQLDEVELOPER\_USER\_DIRen la siguiente línea:
* SetUserHomeVariable SQLDEVELOPER\_USER\_DIR

Si desea evitar que otros usuarios accedan a su información de desarrollador de SQL específica del usuario, debe asegurarse de que los permisos adecuados estén configurados en el directorio donde se almacena esa información o en un directorio que se encuentra arriba en la jerarquía de rutas. Por ejemplo, en un sistema Windows es posible que desee asegurarse de que la SQL Developercarpeta y la *\<user-name>\Application Data\SQL Developer*carpeta que se Documents and Settingsencuentra debajo no se puedan compartir; y en un sistema Linux o Mac OS X es posible que desee asegurarse de que el ~/.sqldeveloperdirectorio no sea legible en todo el mundo.

### Soporte de base de datos en memoria de Oracle TimesTen

Cuando se conecta a una base de datos en memoria de Oracle TimesTen, los tipos de objetos disponibles con los que puede trabajar incluyen varios que se aplican a una base de datos de Oracle, y los siguientes son específicos de TimesTen:

* Grupos de caché
* Esquemas de replicación

Para crear una conexión a una base de datos de TimesTen, use la [pestaña TimesTen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFABAH) en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) .

Para obtener información de uso y referencia sobre TimesTen, consulte la documentación en línea que se incluye con la instalación de TimesTen. Para información adicional, vaya a:<http://www.oracle.com/technology/products/timesten/>

### Usando la ayuda

SQL Developer proporciona un menú de Ayuda y ayuda sensible al contexto (haga clic en el botón Ayuda o presione la tecla F1 en ciertos contextos). Gran parte del contenido de la ayuda también se encuentra en *la Guía del usuario de Oracle Database SQL Developer* , que se encuentra en la Biblioteca de documentación de SQL Developer.

La ayuda se muestra en la ventana del Centro de ayuda, que tiene un panel de contenido a la izquierda, un cuadro de búsqueda en la parte superior derecha y un panel de visualización del tema de ayuda debajo del cuadro de búsqueda. Puede mover el divisor horizontal para cambiar los tamaños del panel (por ejemplo, para hacer que el panel Contenido sea más estrecho, para permitir más espacio para el contenido del tema de ayuda). También puede cambiar el tamaño y la posición de la ventana del Centro de ayuda.

Para Búsqueda, puede hacer clic en el ícono (binoculares) para ver las opciones de búsqueda: distingue entre mayúsculas y minúsculas (Match case) o no distingue mayúsculas y minúsculas; y si se deben hacer coincidir los temas en función de todas las palabras especificadas, palabras especificadas o una expresión booleana.

El botón **Mantener en la parte superior** cambia si la ventana del Centro de ayuda se mantiene en la parte superior de la pantalla cuando cambia el enfoque (clic) nuevamente en la ventana Desarrollador de SQL.

Para imprimir un tema de ayuda, muéstrelo en el panel de visualización del tema y haga clic en el icono Imprimir en la parte superior del panel.

Para aumentar o disminuir el tamaño de la fuente en el visor de temas de ayuda, haga clic en el icono Cambiar tamaño de fuente (A) en la barra de herramientas del área de visualización del tema del Centro de ayuda, luego seleccione Aumentar tamaño de fuente para reducir el tamaño de fuente. Esta configuración se conserva solo durante la duración del panel o ventana de ayuda actual; por lo tanto, es posible que desee mantener abierta la ventana del Centro de ayuda después de configurar la fuente del texto de ayuda a su tamaño preferido.

## 2 Migración de bases de datos de terceros

**Nota:**

Las capacidades de migración en SQL Developer representan una evolución del producto Oracle Migration Workbench.

**La migración** es el proceso de copiar los objetos de esquema y los datos de una base de datos de terceros (no Oracle), como MySQL, Microsoft SQL Server, Sybase Adaptive Server o Microsoft Access, a una base de datos de Oracle. Puede realizar la migración de una manera eficiente, en gran medida automatizada.

Por lo tanto, tiene dos opciones para trabajar con bases de datos de terceros en SQL Developer:

* Creación de conexiones de base de datos para que pueda ver los objetos de esquema y los datos en estas bases de datos
* Migrando estas bases de datos a Oracle, para aprovechar toda la gama de funciones y capacidades de la base de datos Oracle

Este capítulo contiene las siguientes secciones principales:

[Inicio rápido de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDIJEBC)

[Visión general de la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFAEFH)

[Preparando un plan de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDCFGII)

[Antes de comenzar a migrar: Información general](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDICICC)

[Antes de comenzar a migrar: información específica de la fuente](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005774)

[Capturando la base de datos de origen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDIIEEF)

[Creando y personalizando el modelo convertido](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDHFAGE)

[Generando el DDL para los objetos de esquema de Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCJG)

[Migración de los datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJCGGI)

[Hacer que las consultas sean insensibles](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDBIAIE)

[Probando la base de datos Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDCFFJF)

[Despliegue de la base de datos Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDCHHE)

[Uso de informes de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJGAIH)

[Interfaz de usuario de desarrollador SQL para la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCEH)

### Inicio rápido de migración

Para migrar una base de datos de terceros a Oracle, las acciones básicas son: prepararse para la migración, crear o seleccionar asociar un repositorio de migración, capturar la base de datos de origen, convertir la base de datos capturada, generar y ejecutar DDL para los nuevos objetos de esquema de Oracle, y opcionalmente mueva los datos de la base de datos de origen a la nueva base de datos.

Hay dos mecanismos para migrar bases de datos de terceros a Oracle: migración estándar y migración rápida.

### Migración estándar

La migración estándar implica capturar, convertir, generar la base de datos y realizar el movimiento de datos en varios pasos distintos. Este es el enfoque recomendado al realizar una migración. Cualquier problema durante estas fases puede resolverse manualmente y todos los objetos pueden ser inspeccionados o modificados para satisfacer sus necesidades.

**Migración estándar: prepararse para la migración**

1. Prepárese para la migración leyendo los temas relacionados correspondientes en el [Capítulo 2, "Migración de bases de datos de terceros"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDECEIA) .
2. Cree un repositorio de migración en una conexión Oracle nueva o existente. Puede resultarle sencillo y conveniente crear un usuario de base de datos Oracle y una conexión para el trabajo de migración por separado. Luego, seleccione la conexión y cree el repositorio. Por ejemplo:
   1. Cree un usuario de Oracle llamado MIGRATIONS con el espacio de tabla predeterminado USER y el espacio de tabla temporal TEMP; y concédale al menos los privilegios RECURSOS, CREAR SESIÓN y CREAR VISTA. (Para las migraciones multisquemáticas, debe otorgar el rol de RECURSO con la opción ADMINISTRADOR; y también debe otorgarle a este usuario los privilegios CREAR PAPEL, CREAR USUARIO y ALTER ANY TRIGGER, todo con la opción ADMIN.)
   2. Cree una conexión de base de datos llamada Migration\_Repository que se conecte al usuario de MIGRATIONS.
   3. Haga clic con el botón derecho en la conexión Migration\_Repository, y seleccione **Migration Repository** , luego **Associate Migration Repository** para crear el repositorio.
3. Cree y abra una conexión de base de datos para la base de datos de terceros. (Para migraciones que no sean de Microsoft Access, debe configurar la preferencia del controlador JDBC de terceros antes de crear la conexión).

Por ejemplo, cree una conexión de base de datos llamada Sales\_Access a la base de datos de Microsoft Access llamada sales.mdb, y conéctese a ella.

**Migración estándar: objetos de esquema de origen de captura**

Hay dos formas de capturar objetos de esquema de origen: en línea y fuera de línea. Captura en línea que es adecuada en la mayoría de los casos, por lo que se describe aquí.

Para realizar la captura en línea, haga clic con el botón derecho en el nombre de la conexión en el navegador de Connections y seleccione **Capture***database-type* (por ejemplo, Capture MySQL, Capture Microsoft Access, Capture Microsoft SQL Server o Capture Sybase Adaptive Server).

Al seleccionar **Capturar, Microsoft Access** invoca automáticamente la herramienta exportadora de Microsoft Access para crear archivos XML para migrar el esquema y los datos de la tabla. Sin embargo, si desea ejecutar la herramienta de exportación manualmente (por ejemplo, para controlar ciertas opciones), haga clic en **Migración** , luego en **Microsoft Access Exporter** y luego en el elemento de su versión de Microsoft Access. Siga los pasos para la herramienta de exportación, que tiene su propia ayuda en línea.

Después de la captura, el navegador de Modelos capturados muestra un nodo expandible para los objetos capturados (por ejemplo, ventas (Acceso) para los objetos sales.mdb capturados, como se muestra en la figura en la [Interfaz de usuario de SQL Developer para la migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFHCEH) ).

**Migración estándar: convertir objetos capturados**

Para convertir los objetos capturados en objetos de formato Oracle, haga clic con el botón derecho en el nodo apropiado en el navegador de Objetos capturados y seleccione **Convertir a modelo Oracle** , y acepte los valores predeterminados para las asignaciones de datos (o especifique las asignaciones seleccionadas si es necesario).

Después de la conversión, el navegador de modelos convertidos muestra un nodo expandible para los objetos convertidos (por ejemplo, Ventas convertidas (acceso)).

**Migración estándar: generar objetos de base de datos Oracle**

1. Genere un script SQL \* Plus que cree las declaraciones DDL para crear los objetos de la base de datos de Oracle que corresponden a los objetos de la base de datos de origen: haga clic con el botón derecho en el nodo apropiado en el navegador de Modelos capturados y seleccione **Generar** . Se abre una ventana de hoja de trabajo SQL que contiene las sentencias de SQL \* Plus.
2. En la ventana de la hoja de trabajo SQL que se acaba de abrir, seleccione (en la lista desplegable a la derecha) una conexión de base de datos de Oracle en la que ejecutar el script (próximo paso).
3. Examine las sentencias de SQL \* Plus generadas y, opcionalmente, realice los cambios. Por ejemplo, si el usuario de la base de datos que posee los objetos generados ya existe (como lo hará si está siguiendo estas instrucciones de paso rápido), elimine o modifique el USUARIO CREADO y las declaraciones relacionadas.
4. Haga clic en el botón Ejecutar secuencia de comandos en la ventana de la hoja de trabajo SQL para ejecutar la secuencia de comandos.
5. En el navegador de conexiones, cree una conexión con el usuario que se acaba de crear.

En el navegador de Connections, ahora debería ver los nuevos objetos de base de datos correspondientes a los objetos en la base de datos de terceros que migró.

**Migración estándar: mover datos a la base de datos Oracle**

Si lo desea, puede migrar (mover) cualquier dato existente de la base de datos de origen a la base de datos de Oracle. Tiene dos opciones para la migración de datos: en línea o fuera de línea.

* Movimiento de datos en línea: haga clic en **Migración** , luego en **Migrar datos** . En el cuadro de diálogo, especifique la Conexión de origen, la Conexión de destino y el Modelo convertido. Este método utiliza JDBC y, por lo tanto, está restringido por las implementaciones de terceros. Este método es adecuado para mover pequeños conjuntos de datos.
* Movimiento de datos sin conexión: haga clic en **Migración** , luego en **Generación de secuencias de comandos** , luego en **Generar secuencias de datos de movimiento de datos** ; especifique el modelo convertido y un directorio en el que generar los archivos que utilizará para descargar los datos de la base de datos de origen y para importarlos a Oracle utilizando SQL \* Loader. Este método está diseñado para mover grandes volúmenes de datos.

### Migración rápida

La migración rápida es un enfoque simplificado que utiliza un asistente. Proporciona una solución rápida al migrar una base de datos simple; sin embargo, para un mayor control del proceso de migración, debe usar la [migración estándar](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDIGIJ) .

**Migración rápida: prepararse para la migración**

1. Prepárese para la migración leyendo los temas relacionados correspondientes en el [Capítulo 2, "Migración de bases de datos de terceros"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDECEIA) .
2. Cree un repositorio de migración en una conexión Oracle nueva o existente. Puede resultarle sencillo y conveniente crear un usuario de base de datos Oracle y una conexión para el trabajo de migración por separado. Luego, seleccione la conexión y cree el repositorio. Por ejemplo:
   1. Cree un usuario de Oracle llamado MIGRATIONS con el espacio de tabla predeterminado USER y el espacio de tabla temporal TEMP; y concédale al menos los privilegios RECURSOS, CREAR SESIÓN y CREAR VISTA. (Para las migraciones multisquemáticas, debe otorgar el rol de RECURSO con la opción ADMINISTRADOR; y también debe otorgarle a este usuario los privilegios CREAR PAPEL, CREAR USUARIO y ALTER ANY TRIGGER, todo con la opción ADMIN.)
   2. Cree una conexión de base de datos llamada Migration\_Repository que se conecte al usuario de MIGRATIONS.
   3. Haga clic con el botón derecho en la conexión Migration\_Repository, y seleccione **Migration Repository** , luego **Associate Migration Repository** para crear el repositorio.
3. Cree un usuario de Oracle cuyo esquema se utilizará como destino de los objetos que se migrarán, o use un usuario y un esquema de Oracle existentes. Otorgue privilegios suficientes a este usuario.

Por ejemplo, si planea migrar una base de datos de Microsoft Access llamada sales.mdb, puede crear un usuario de Oracle llamado VENTAS, en cuyo esquema se generarán los objetos de la base de datos de Oracle.

1. Cree y abra una conexión de Oracle para el esquema que creó o seleccionó en el paso anterior.

Por ejemplo, cree una conexión de Oracle llamada Sales\_Oracle al esquema asociado con las VENTAS del usuario y conéctese a él.

1. Cree y abra una conexión de base de datos para la base de datos de terceros. (Para migraciones que no sean de Microsoft Access, debe configurar la preferencia del controlador JDBC de terceros antes de crear la conexión).

Por ejemplo, cree una conexión de base de datos llamada Sales\_Access a la base de datos de Microsoft Access llamada sales.mdb, y conéctese a ella.

**Migración rápida: migrar usando el asistente**

1. Haga clic en **Migración** , luego en **Migración rápida** .
2. Para **Source Connection** , seleccione la conexión para la base de datos de terceros que se va a migrar. Por ejemplo: Sales\_Access
3. Para la **conexión de destino** , seleccione la conexión para el esquema de la base de datos Oracle a la que se migrará la base de datos de terceros. Por ejemplo: Sales\_Oracle
4. Para el **Repositorio** , use el repositorio existente seleccionado; o si no existe ningún repositorio, permita que SQL Developer cree un repositorio de migración en el esquema de la conexión de destino.
5. Haga clic en **Verificar** para iniciar la comprobación previa a la migración.
6. Una vez que la comprobación previa a la migración se haya completado satisfactoriamente, especifique el **Tipo de migración** : Migrar tablas, Migrar tablas y datos, o Migrar todo (todos los objetos).
7. Haga clic en **Finalizar** en el panel Resumen para realizar la migración.

Las operaciones específicas realizadas dependen del tipo de migración y del tipo de base de datos de terceros que se está migrando. Por ejemplo, para una base de datos de Microsoft Access, la herramienta Exportador para Microsoft Access se invoca automáticamente. *No interrumpir ninguna de las operaciones de migración.*

Si surge algún problema durante la migración, la migración rápida se detendrá. Para continuar con la migración, siga el enfoque de [migración estándar](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDIGIJ) , que lo ayudará a identificar los problemas y le permitirá modificar los objetos apropiados.

### Visión general de la migración

Una base de datos Oracle le brinda mejor escalabilidad, confiabilidad, mayor rendimiento y mejor seguridad que las bases de datos de terceros. Por este motivo, las organizaciones migran de su base de datos actual, como Microsoft SQL Server, Sybase Adaptive Server o Microsoft Access, a una base de datos Oracle. Aunque la migración de la base de datos puede ser complicada, SQL Developer le permite simplificar el proceso de migración de una base de datos de terceros a una base de datos de Oracle.

SQL Developer captura información de la base de datos de origen y la muestra en el **modelo capturado** , que es una representación de la estructura de la base de datos de origen. Esta representación se almacena en un **repositorio de migración** , que es una colección de objetos de esquema que SQL Developer utiliza para almacenar información de migración.

La información en el repositorio se utiliza para generar el **modelo convertido** , que es una representación de la estructura de la base de datos de destino tal como se implementará en la base de datos de Oracle. Luego puede usar la información del modelo capturado y el modelo convertido para comparar objetos de la base de datos, identificar conflictos con las palabras reservadas de Oracle y administrar el progreso de la migración. Cuando esté listo para migrar, genere los objetos de esquema de Oracle y luego migre los datos.

SQL Developer contiene lógica para extraer datos del diccionario de datos de la base de datos de origen, crear el modelo capturado y convertir el modelo capturado al modelo convertido.

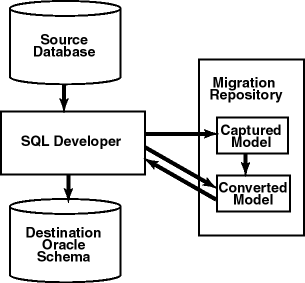
El uso de SQL Developer para migrar una base de datos de terceros a una base de datos de Oracle proporciona los siguientes beneficios:

* Reduce el esfuerzo y los riesgos involucrados en un proyecto de migración.
* Le permite migrar una base de datos de terceros completa, incluidos desencadenantes y procedimientos almacenados
* Le permite ver y comparar el modelo capturado y el modelo convertido y personalizar cada uno si lo desea, para que pueda controlar cuánta automatización hay en el proceso de migración.
* Proporciona comentarios sobre la migración a través de informes.

### Cómo funciona la migración

Los componentes de SQL Developer trabajan juntos para migrar una base de datos de terceros a una base de datos Oracle. [La Figura 2-1, "Arquitectura de migración para desarrolladores de SQL"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005726) muestra cómo SQL Developer lee la información de la base de datos de origen y crea los objetos de esquema de la base de datos de Oracle. SQL Developer utiliza la información almacenada en el repositorio de migración para migrar al esquema de Oracle. Puede realizar cambios en el modelo capturado o el modelo convertido, o ambos, antes de migrar. La información en el modelo convertido se usa para completar la migración, es decir, para generar los objetos de base de datos en el esquema de destino de Oracle.

***Figura 2-1 Arquitectura de migración del desarrollador de SQL***



### Migración implementada como extensiones de desarrollador de SQL

El soporte de migración se implementa en SQL Developer como un conjunto de extensiones. Si lo desea, puede deshabilitar el soporte de migración o el soporte para migrar bases de datos de terceros individuales.

Para ver las extensiones instaladas, y para habilitar o deshabilitar extensiones individuales, haga clic en **Herramientas** , luego en **Preferencias** , luego en **Extensiones** . Tenga en cuenta que SQL Developer incluye todas las extensiones y bases de datos de terceros "complementos" disponibles en el momento del lanzamiento, por lo que para comenzar otras migraciones que no sean para Microsoft Access, solo se deben instalar los controladores de terceros.

### Preparando un plan de migración

Este tema describe el proceso de cómo crear un plan de proyecto de migración. Identifica las secciones a incluir en el plan de migración, describe cómo determinar qué incluir en cada sección y cómo evitar los riesgos que implica un proyecto de migración. Esta información incluye:

* [Tarea 1: Determinación de los requisitos del proyecto de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJIGHA)
* [Tarea 2: Estimar la carga de trabajo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJECFF)
* [Tarea 3: Análisis de los requisitos operacionales](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDCGGDG)
* [Tarea 4: Analizar la aplicación](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDBHIAF)
* [Tarea 5: Planificación del proyecto de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDBICGE)

**Tarea 1: Determinación de los requisitos del proyecto de migración**

En esta tarea, usted identifica qué bases de datos desea migrar y las aplicaciones que acceden a esa base de datos. También evalúa los requisitos del negocio y define los criterios de prueba.

Para determinar los requisitos del proyecto de migración:

1. Definir el alcance del proyecto.

Hay varias opciones que debe hacer sobre la base de datos de terceros y las aplicaciones que acceden a esa base de datos para definir el alcance del proyecto de migración. Para obtener una lista de problemas y dependencias de migración, debe considerar lo siguiente

* + ¿Qué bases de datos de terceros están migrando?
    - ¿Cuál es la versión de la base de datos de terceros?
    - ¿Cuál es el conjunto de caracteres de la base de datos de terceros?
  + ¿Qué aplicaciones de origen se ven afectadas al migrar la base de datos de terceros a una base de datos Oracle?
    - ¿Cuál es el idioma de la aplicación de terceros?
    - ¿Qué versión del lenguaje de la aplicación está utilizando?

En el ámbito del proyecto, debe haber identificado las aplicaciones que debe migrar. Asegúrese de haber incluido todas las aplicaciones necesarias que se ven afectadas por la migración de la base de datos

* + ¿Qué tipos de problemas de conectividad están involucrados en la migración a una base de datos Oracle?
    - ¿Utiliza software de conectividad para conectar las aplicaciones a la base de datos de terceros? ¿Necesita modificar el software de conectividad para conectar las aplicaciones a la base de datos de Oracle?
    - ¿Qué versión del software de conectividad usas? ¿Puede usar esta misma versión para conectarse a la base de datos Oracle?
  + ¿Planea volver a escribir las aplicaciones o modificar las aplicaciones para que funcionen con una base de datos Oracle?

1. Utilice la [Tabla 2-1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CACBAAJI) para determinar si tiene un entorno de base de datos fuente complejo o simple. Identificar los requisitos basados ​​en el escenario específico.

Si el proyecto de migración es un escenario simple, es posible que no tenga que completar todas las tareas enumeradas en esta guía. Usted toma decisiones basadas en su entorno específico. Por ejemplo, si tiene un escenario complejo, puede requerir pruebas adicionales según la complejidad de la aplicación que accede a la base de datos.

***Tabla 2-1 Escenarios complejos y simples***

| **Escenario complejo** | **Escenario simple** |
| --- | --- |
| Más de uno de los siguientes:   * + Base de datos grande (más de 25 GB)   + Almacén de datos   + Aplicaciones grandes (más de 100 formularios, informes y trabajos por lotes)   + La base de datos es utilizada por múltiples líneas de negocio.   + Despliegue distribuido   + Base de usuarios grande (más de 100)   + Requisito de alta disponibilidad (como un entorno de 24 X 7 X 365) | Contiene lo siguiente:   * + Pequeña base de datos (menos de 25 GB)   + Procesamiento simple de transacciones en línea (OLTP)   + Aplicación pequeña (menos de 100 formularios, informes y trabajos por lotes)   + La base de datos es utilizada por un  departamento   + Despliegue centralizado   + Pequeña base de usuarios (menos de 100)   + Disponibilidad media (horario comercial) |

1. Determine si la base de datos de destino requiere hardware adicional y la reescritura de los cronogramas de respaldo.
2. Definir criterios de prueba y aceptación.

Definir pruebas para medir la precisión de la migración. A continuación, utiliza los criterios de aceptación para determinar si la migración se realizó correctamente. Las pruebas que desarrolle a partir de los requisitos también deben medir la estabilidad, evaluar el rendimiento y probar las aplicaciones. Debe decidir la cantidad de pruebas necesarias para poder implementar la base de datos y las aplicaciones de Oracle en un entorno de producción.

1. Cree un documento de requisitos con una lista de requisitos para el proyecto de migración.

El documento de requisitos debe tener tareas claramente definidas y enumerar cada requisito específico, dividiéndolos en sub-requisitos cuando sea necesario.

**Tarea 2: Estimar la carga de trabajo**

En esta tarea, utiliza SQL Developer para tomar decisiones calculadas sobre la cantidad de trabajo que se puede automatizar y cuánto es manual.

Para estimar la carga de trabajo:

1. Capture el modelo capturado, cree el modelo convertido y migre a la base de datos de destino.

Puede analizar la base de datos de origen a través del modelo capturado y una vista previa de la base de datos de destino a través del modelo convertido. Una vez que haya capturado la base de datos de origen, analice los datos capturados contenidos en el modelo capturado y el modelo convertido. Asegúrese de que el contenido y la estructura del repositorio de migración sean correctos y determine cuánto tiempo lleva todo el proceso.

1. Use el panel Registro de migración para evaluar el proceso de captura y migración, categorizar la cantidad total de objetos de la base de datos e identificar la cantidad de objetos que se pueden convertir y migrar automáticamente.

El registro de migración proporciona información sobre las acciones que se han producido y registra las advertencias y los errores. Identifican los cambios que se han realizado en el modelo convertido para que pueda evaluar si debe realizar cambios en las aplicaciones que acceden a la base de datos de destino.

1. Evaluar y categorizar los problemas que ocurrieron. El registro de migración puede ayudar proporcionando información sobre:
   * Tablas que no se cargaron cuando capturó la base de datos de origen
   * Procedimientos almacenados, vistas y activadores que no se analizaron cuando creó el modelo convertido
   * Sintaxis que requiere intervención manual.
   * Objetos de la base de datos que no se crearon correctamente cuando migró la base de datos de destino
   * Datos que no se migraron correctamente cuando migró la base de datos de destino
2. Para cada error o advertencia en el registro de migración, evalúe lo siguiente:
   * Número de veces que ocurrió un problema
   * Tiempo requerido para solucionar los problemas, en horas-persona.
   * Número de recursos necesarios para solucionar el problema

Una vez que haya resuelto un problema complejo, debería ser más fácil y rápido resolverlo la próxima vez que tenga el mismo problema.

**Tarea 3: Análisis de los requisitos operacionales**

En esta tarea, usted analiza los requisitos operativos, de la siguiente manera:

1. Evalúe las consideraciones operativas al migrar la base de datos de origen a una base de datos de destino. Considere las siguientes preguntas:

**Nota:**

Si el alcance del proyecto de migración es un escenario complejo como se define en la [Tabla 2-1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CACBAAJI) , Oracle recomienda que responda todas estas preguntas. Si tiene un escenario simple, determine las respuestas a las preguntas más apropiadas.

* + ¿Qué cambios de copia de seguridad y recuperación necesita?
  + ¿Qué tiempo de inactividad se requiere durante la migración?
  + ¿Has cumplido con los requisitos de rendimiento?
  + ¿Está cambiando la ventana de tiempo operacional?
  + ¿Qué efecto tiene el tiempo de inactividad en el negocio?
  + ¿Qué requisitos de capacitación o consideraciones adicionales del personal se requieren?
  + ¿Es necesario que el tercero y la base de datos Oracle se ejecuten simultáneamente?

1. Para cada tarea, determine los recursos y el tiempo necesarios para completar.
2. Crear un plan de proyecto inicial.

Utilice la información que ha reunido durante los requisitos y la etapa de planificación para desarrollar un plan inicial del proyecto.

**Tarea 4: Analizar la aplicación**

En esta tarea, usted identifica a los usuarios de las aplicaciones que se ejecutan en la base de datos de origen, qué hardware requiere, qué hace la aplicación y cómo interactúa con la base de datos de origen. También analiza el método que utiliza la aplicación para conectarse a la base de datos e identifica las modificaciones necesarias.

**Nota:**

Si el proyecto de migración es un escenario complejo como se define en la [Tabla 2-1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CACBAAJI) , Oracle recomienda que responda todas estas preguntas. Si tiene un escenario simple, determine las respuestas a las preguntas más apropiadas.

Para analizar la aplicación:

1. Determine si se requieren cambios en la aplicación para que se ejecuten de manera efectiva en la base de datos de destino.
2. Si se requieren cambios en la aplicación, determine si es más eficiente reescribir o modificar las aplicaciones.

Si está reescribiendo la aplicación para usar la base de datos Oracle, considere lo siguiente:

* 1. Cree la documentación de proyecto necesaria para reescribir la aplicación. Por ejemplo, necesita una especificación de diseño y documentación de requisitos.
  2. Reescriba la aplicación de acuerdo a la especificación.
  3. Probar la aplicación funciona contra la base de datos Oracle.

Si está modificando la aplicación para usar la base de datos Oracle, considere lo siguiente:

* 1. Identifique el número de conexiones a la base de datos que están en la aplicación y modifique estas conexiones para usar la base de datos Oracle.

Es posible que deba cambiar la información de conexión para utilizar una conexión ODBC o JDBC.

* 1. Identifique las sentencias de SQL incorporado que necesita cambiar en la aplicación antes de poder probarlas en la base de datos de Oracle.
  2. Probar la aplicación utilizando la base de datos Oracle.

1. Asigne tiempo y recursos para abordar cada problema asociado con la reescritura o modificación de la aplicación.
2. Actualice el documento de requisitos generales para el proyecto que creó en la Tarea 1.

**Tarea 5: Planificación del proyecto de migración**

En esta tarea, evalúa las variables desconocidas que puede contener el proyecto de migración, como la diferencia en las tecnologías de la base de datos de origen y la base de datos de destino. Durante la etapa de planificación, usted:

* Estimar las restricciones presupuestarias del proyecto.
* Recolectar información para producir un plan de migración.
* Estime cuánto tiempo debe tomar el proyecto de migración
* Calcule cuántos recursos se requieren para completar y probar la migración

Para planificar un proyecto de migración:

1. Defina una lista de tareas necesarias para completar con éxito los requisitos del proyecto de migración de la Tarea 1.
2. Clasifique la lista de tareas necesarias para completar el proyecto de migración.

Debes agrupar estas tareas de acuerdo a tu negocio. Esto le permite programar y asignar recursos con mayor precisión.

1. Actualice y finalice el plan del proyecto de migración en función de la información que haya obtenido de las Tareas 3 y 4.
2. Asegúrese de que el plan del proyecto de migración cumpla con los requisitos del proyecto de migración.

El plan de migración debe incluir una descripción del proyecto, recursos asignados, requisitos de capacitación, entregables de la migración, requisitos generales, análisis del entorno, análisis de riesgos, evaluación de la aplicación y cronograma del proyecto.

**Antes de comenzar a migrar: Información general**

Es posible que deba realizar ciertas tareas antes de comenzar a migrar una base de datos de terceros a una base de datos de Oracle. Vea lo siguiente para más información:

* [Creación de un usuario de base de datos para el repositorio de migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005739)
* [Requisitos para crear los objetos de Oracle de destino](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJJDEB)

Vea también cualquier información específica de la base de datos de origen que migrará, como se explica en [Antes de comenzar a migrar:](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005774)Información específica de la fuente .

**Nota:**

Oracle recomienda que realice una copia de seguridad completa de la base de datos de origen antes de iniciar la migración. Para obtener más información sobre la copia de seguridad de la base de datos de origen, consulte la documentación para ese tipo de base de datos.

Si es posible, comience la migración utilizando un entorno de desarrollo o prueba, no una base de datos de producción.

### Creación de un usuario de base de datos para el repositorio de migración

SQL Developer requiere un repositorio de migración para migrar una base de datos de terceros a una base de datos Oracle. Para usar una base de datos Oracle para el repositorio de migración, debe tener acceso a esa base de datos utilizando una cuenta de usuario de base de datos. Oracle recomienda que use una cuenta de usuario específica para migraciones. Por ejemplo, puede crear un usuario llamado MIGRATIONS, crear una conexión de base de datos para ese usuario y usar esa conexión para el repositorio de migración; y si lo desea, más tarde puede eliminar al usuario de MIGRACIONES para eliminar todos los rastros de la migración de la base de datos.

Cuando cree un usuario para las migraciones, especifique la información del espacio de tablas como se muestra en el siguiente ejemplo, en lugar de usar los valores predeterminados para los espacios de tablas:

CREAR migraciones de *usuario* identificadas por *contraseña*

Usuarios de TABLESAPACE POR DEFECTO TEMPORAL TABLESPACE temp,

No utilice una cuenta estándar (por ejemplo, SISTEMA) para la migración.

Cuando SQL Developer crea un repositorio de migración, crea muchos objetos de esquema que están destinados solo para su propio uso. Por ejemplo, crea tablas, vistas, índices, paquetes y activadores, muchos de ellos con nombres que comienzan con MD\_y MIGR. No debe modificar directamente estos objetos ni los datos almacenados en ellos.

### Requisitos para crear los objetos de Oracle de destino

El usuario asociado con la conexión de base de datos de Oracle utilizada para realizar la migración (es decir, para ejecutar el script que contiene las declaraciones DDL generadas) debe tener los siguientes roles y privilegios:

**Nota:**

Debe otorgar estos privilegios directamente a una cuenta de usuario. No es suficiente otorgar los privilegios a un rol, que posteriormente se otorga a una cuenta de usuario. No se puede migrar una base de datos como usuario SYS.

**Roles**

CONECTARSE CON LA OPCION DE ADMINISTRACION

RECURSO CON OPCION ADMINISTRATIVA

**Privilegios**

Alterar cualquier rol

Alterar cualquier secuencia

Alterar cualquier tabla

ALTER TABLESPACE

Alterar cualquier gatillo

COMENTAR CUALQUIER TABLA

Crear cualquier secuencia

Crear cualquier tabla

Crear cualquier gatillo

CREAR VISTA CON OPCIÓN DE ADMINISTRACIÓN

CREAR SINÓNIMO PÚBLICO CON OPCIÓN DE ADMINISTRACIÓN

Crear un rol

CREAR USUARIO

CAER CUALQUIER SECUENCIA

GOTA CUALQUIER MESA

Deje caer cualquier gatillo

DROP USUARIO

GOTA CUALQUIER PAPEL

OTORGAR CUALQUIER PAPEL

Inserte cualquier tabla

Selecciona cualquier tabla

Actualizar cualquier tabla

Por ejemplo, puede crear un usuario llamado migrationscon los privilegios mínimos requeridos para migrar una base de datos usando los siguientes comandos:

CREAR migraciones de *usuario* identificadas por *contraseña*

Usuarios de TABLESAPACE POR DEFECTO TEMPORAL TABLESPACE temp;

CONCESIÓN DE SUBVENCIONES, RECURSOS, CREAR VISTA, CREAR SINÓNIMO PÚBLICO PARA

migraciones con opcion de administrador;

OTORGAR ALTERA CUALQUIER PAPEL, ALTERAR CUALQUIER SECUENCIA, ALTERAR CUALQUIER MESA, ALTERAR HABITACIÓN,

ALTERA CUALQUIER DISPOSITIVO, COMENTÉ CUALQUIER MESA, CREA CUALQUIER SECUENCIA, CREA CUALQUIER MESA,

CREAR CUALQUIER DISPARADOR, CREAR UN PAPEL, CREAR TABLESPACE, CREAR USUARIO, COLOCAR CUALQUIER

SECUENCIA, DEJAR CUALQUIER TABLA, DEJAR CUALQUIER GATILLO, DEPÓSITO DE LA GOTA, UTILIZAR EL USUARIO DE LA GOTA, DEJAR CUALQUIERA

FUNCIÓN, OTORGAR CUALQUIER FUNCIÓN, INSERTE CUALQUIER MESA, SELECCIONE UNA MESA, ACTUALICE CUALQUIER MESA PARA

migraciones;

Una vez que haya creado el modelo convertido, se crea y realiza la primera generación de DDL para la nueva base de datos, los scripts indicarán qué privilegios serán necesarios para su situación.

### Antes de comenzar a migrar: información específica de la fuente

Dependiendo de la base de datos de terceros que esté migrando a una base de datos Oracle, es posible que deba configurar la información de conexión e instalar los controladores. Para obtener más información sobre los requisitos específicos de la base de datos de terceros, consulte lo siguiente:

* [Antes de migrar desde Microsoft SQL Server o Sybase Adaptive Server](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005806)
* [Antes de migrar desde Microsoft Access](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005890)
* [Antes de migrar desde MySQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1006073)

### Antes de migrar desde Microsoft SQL Server o Sybase Adaptive Server

Para configurar una base de datos de Microsoft SQL Server o Sybase Adaptive Server para la migración:

1. Asegúrese de que la base de datos de origen sea accesible por el usuario de Microsoft SQL Server o Sybase Adaptive Server que utiliza SQL Developer para la conexión de origen. Este usuario debe poder ver cualquier objeto que se capture en la base de datos de Microsoft SQL Server o Sybase Adaptive Server; Los objetos que el usuario no puede ver no son capturados. Por ejemplo, si el usuario puede ejecutar un procedimiento almacenado pero no tiene suficientes privilegios para ver el código fuente, no se puede capturar el procedimiento almacenado.
2. Asegúrese de que puede conectarse a la base de datos Microsoft SQL Server o Sybase Adaptive Server desde el sistema donde ha instalado SQL Developer.
3. Asegúrese de haber descargado el controlador JTBC JTDS desde <http://jtds.sourceforge.net/>.
4. En SQL Developer, si aún no ha instalado el controlador JTDS usando Buscar actualizaciones (en el menú Ayuda), haga lo siguiente:
   1. Haga clic en **Herramientas** , luego en **Preferencias** , luego en **Base de datos** , luego en **Controladores JDBC de terceros** .
   2. Haga clic en **Agregar entrada** .
   3. Seleccione el archivo jar para el controlador JTDS que descargó <http://jtds.sourceforge.net/>.
   4. Haga clic en **Aceptar** .
5. En SQL Developer, haga clic en **Herramientas** , luego en **Preferencias** , luego en [**Migración: Opciones de identificador**](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDCBDAC) , y asegúrese de que la configuración sea correcta para la opción **El identificador entre comillas está activado** (es decir, la configuración refleja la base de datos que se va a migrar).

Si esta opción está habilitada, las comillas (comillas dobles) se pueden usar para referirse a los identificadores; Si esta opción no está habilitada, las comillas identifican literales de cadena. Como ejemplo de la diferencia de comportamiento, considere el siguiente código T-SQL:

seleccione col1, "col 2" "column\_alias"

de tablex "table\_alias"

Si la opción del identificador de cotización activada está habilitada (marcada), se genera el siguiente código PL / SQL:

SELECCIONAR col1, **col\_2** "column\_alias"

FROM tablex "table\_alias";

Si la opción Is Quoted Identifier On está desactivada (no seleccionada), se genera el siguiente código PL / SQL:

SELECCIONAR col1, **'col 2'** "column\_alias"

FROM tablex "table\_alias";

### Antes de migrar desde Microsoft Access

Para configurar una base de datos de Microsoft Access para la migración:

1. Haga copias de seguridad del archivo o archivos de la base de datos.
2. Asegúrese de que el software necesario (Microsoft Access, quizás otros componentes) esté instalado en el mismo sistema que SQL Developer.
3. Asegúrese de que el usuario administrador tenga al menos permisos de Leer diseño y Leer datos en las tablas del sistema MSysObjects, MSysQueries y MSysRelationships, como se explica en la información sobre la [pestaña Acceso](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJJDIA) en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) .
4. Si la seguridad está habilitada, debes desactivarla copiando el contenido de la base de datos segura en una nueva base de datos, de la siguiente manera:

SQL Developer no admite la migración de bases de datos de Microsoft Access que tienen habilitada la seguridad. De forma predeterminada, SQL Developer usa el nombre del archivo MDB de Microsoft Access como el nombre de usuario para el usuario de destino de Oracle. Si creas un usuario de Oracle de esta manera, la contraseña es ORACLE.

* 1. En el menú **Archivo** en Microsoft Access, seleccione **Nueva base de datos** .
  2. Seleccione el icono **Base de datos en blanco** , luego haga **clic** en **Aceptar** .
  3. En la opción File New Database, escriba un nombre para la base de datos, luego haga clic en **Crear** .
  4. En el menú **Archivo** dentro de la nueva base de datos, seleccione **Obtener datos externos** , luego seleccione **Importar** .
  5. Seleccione la base de datos segura de Microsoft Access que desea importar, luego haga clic en **Importar** .
  6. En el cuadro de diálogo Importar objetos, haga clic en **Opciones** .
  7. Seleccione las opciones Relaciones y Definición y Datos.
  8. En la pestaña Tablas, **seleccione Seleccionar todo** .
  9. Haga clic en **Aceptar** .

Todos los objetos de Microsoft Access se copian a la nueva base de datos de Microsoft Access, excepto la configuración de seguridad.

1. Si la aplicación contiene tablas vinculadas a otras bases de datos de Microsoft Access, actualice estos enlaces abriendo la aplicación en Microsoft Access y realice lo siguiente:

En el menú **Herramientas** en Microsoft Access 97, seleccione **Agregar complementos** , luego seleccione **Administrador de tablas vinculadas** .

En el menú Herramientas en Microsoft Access 2000, seleccione **Utilidades de base de datos** , luego seleccione **Administrador de tablas vinculadas** .

1. Asegúrese de que la base de datos de Microsoft Access no sea una base de datos de réplica, sino una base de datos maestra.

Cuando utiliza el Exportador para Microsoft Access para exportar, se muestra un mensaje de error si la base de datos es una réplica. SQL Developer no admite la migración de una base de datos de réplica.

1. En el menú **Herramientas** dentro de Microsoft Access, seleccione **Base de datos** , luego seleccione **Base** de **datos** compacta para compactar los archivos de la base de datos de Microsoft Access.
2. Asegúrese de que se pueda acceder al archivo MDB de Microsoft Access desde el sistema donde instaló SQL Developer.
3. Utilice Oracle Universal Installer para verificar que tiene instalado el controlador ODBC de Oracle. Si necesita instalar el controlador, está disponible en el servidor de la base de datos Oracle o en el CD del cliente de la base de datos. También puede descargar el controlador ODBC de Oracle desde el sitio web de Oracle Technology Network (OTN):
4. <http://www.oracle.com/technology/software/tech/windows/odbc/>

Instale el controlador ODBC de Oracle en un directorio de inicio de Oracle que contenga los Servicios Net de Oracle. Puede obtener los servicios de red de Oracle del cliente de Oracle o del CD de la base de datos de Oracle. Instala Oracle Net Services para obtener Net Configuration Assistant y Net Manager. Estos le permiten crear una configuración de red en el tnsnames.oraarchivo.

**Nota:**

Para obtener más información sobre la instalación de los productos de red necesarios para conectarse a una base de datos Oracle, consulte la guía de instalación de su versión de la base de datos Oracle.

Creando Archivos XML de Microsoft Access

Para prepararse para capturar una base de datos de Microsoft Access, la herramienta Exporter for Microsoft Access se debe ejecutar, ya sea de forma automática o manual, como se explica en [Captura de la base de datos de origen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDIIEEF) . Esta herramienta está empaquetada como un archivo MDE de Microsoft Access y le permite exportar el archivo MDB de Microsoft Access a un archivo XML.

**Nota:**

No modifique ninguno de los archivos creados por la herramienta Exportador.

Cada base de datos de Microsoft Access que seleccionó se exporta a un archivo XML. La herramienta de exportación actualmente no admite la creación de archivos XML desde bases de datos seguras o de réplica.

### Antes de migrar desde MySQL

Para configurar una base de datos MySQL para la migración, instale MySQLConnector / J versión 3.1.12 o 5.0.4 en el sistema en el que haya instalado SQL Developer y configure las preferencias de SQL Developer adecuadas. Sigue estos pasos:

1. Asegúrese de que puede conectarse a la base de datos MySQL desde el sistema donde ha instalado SQL Developer.
2. Asegúrese de haber descargado la API MySQLConnector / J del sitio web de MySQL en <http://www.mysql.com/>.
3. En SQL Developer, si aún no ha instalado el controlador JDBC de MySQL usando Buscar actualizaciones (en el menú Ayuda), haga lo siguiente:
   1. Haga clic en **Herramientas** , luego en **Preferencias** , luego en **Base de datos** , luego en **Controladores JDBC de terceros** .
   2. Haga clic en **Agregar entrada** .
   3. Seleccione el archivo jar para el controlador MySQL que descargó <http://www.mysql.com/>.
   4. Haga clic en **Aceptar** .
4. Asegúrese de que la base de datos de origen sea accesible para el usuario de MySQL que utiliza SQL Developer para la conexión de origen. Este usuario debe poder ver cualquier objeto que se capture en la base de datos MySQL; Los objetos que el usuario no puede ver no son capturados. Por ejemplo, si el usuario puede ejecutar un procedimiento almacenado pero no tiene suficientes privilegios para ver el código fuente, no se puede capturar el procedimiento almacenado.

### Capturando la base de datos de origen

Antes de migrar una base de datos de terceros, debe extraer información de la base de datos. Esta información es una representación de la estructura de la base de datos de origen y se denomina modelo capturado. El proceso de extracción de la información de la base de datos se denomina captura de la base de datos de origen.

La captura se puede hacer en línea o fuera de línea:

* **La captura en línea** se realiza en una secuencia guiada conveniente dentro de la interfaz del desarrollador de SQL, como se explica en [Captura](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDCCFE) en [línea](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDCCFE) . Puede utilizar la captura en línea con todas las bases de datos de terceros compatibles.
* **La captura sin conexión** implica la creación de un script que ejecuta más adelante, como se explica en [Captura sin conexión](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDHFDFH) . Puede usar la captura sin conexión con MySQL, las bases de datos de Microsoft SQL Server y Sybase Adaptive Server.

Después de capturar la base de datos de origen, puede ver la información de la base de datos de origen en el modelo capturado en SQL Developer. Si es necesario, puede modificar el modelo capturado y cambiar las asignaciones de tipos de datos.

**Nota:**

Oracle recomienda que no cambie las asignaciones de tipo de datos predeterminadas a menos que sea un administrador de bases de datos de Oracle con experiencia.

### Captura en línea

Para realizar una captura en línea de la base de datos de origen, puede hacer que la captura se realice automáticamente como parte de la opción Migración rápida, o puede realizarla como una operación separada haciendo clic derecho en el nombre de la conexión en el navegador de Conexiones y seleccionando **Capturar *producto -nombre*** (por ejemplo, Capture MySQL, Capture Microsoft Access, Capture Microsoft SQL Server o Capture Sybase Adaptive Server).

Para una base de datos de Microsoft Access, al seleccionar **Capturar *nombre de producto*** se invoca automáticamente la herramienta exportadora de Microsoft Access para crear archivos XML para migrar el esquema y los datos de la tabla. Sin embargo, si desea ejecutar la herramienta de exportación manualmente (por ejemplo, para controlar ciertas opciones), haga clic en **Migración** , luego en **Microsoft Access Exporter** y luego en el elemento de su versión de Microsoft Access. Siga los pasos para la herramienta de exportación, que tiene su propia ayuda en línea.

### Captura fuera de línea

Para realizar una captura sin conexión de una base de datos MySQL, Microsoft SQL Server o Sybase Adaptive Server, crea un conjunto de secuencias de comandos de captura sin conexión, ejecute estas secuencias de comandos fuera de SQL Developer para crear la salida de la secuencia de comandos (un volcado de las tablas de metadatos de terceros), y cargue la salida del script (el archivo .ocp que contiene el modelo convertido) utilizando el Desarrollador SQL.

* Para crear el archivo de script (un archivo .bat de Windows o un archivo .sh de Linux o UNIX) y los archivos relacionados, haga clic en **Migración** , luego en **MySQL, SQL Server y Sybase Offline Capture** , luego en **Crear scripts de captura de base de datos** .

Cuando se completa esta operación, se le notifica que se han creado varios archivos (.bat, .sql, .ocp), uno de los cuales es el script de control. Debe ejecutar la secuencia de comandos de control (fuera de SQL Developer) para completar el archivo de propiedades de captura de objeto (.ocp) con información sobre el modelo convertido.

* Para cargar el modelo convertido desde el archivo de propiedades de captura de objeto (.ocp) generado por el script de control de captura sin conexión, haga clic en **Migración** , luego en **MySQL, SQL Server y Sybase Offline Capture** , luego en **Cargar base de datos de salida de script de captura** .

### Creando y personalizando el modelo convertido

Después de capturar una base de datos de terceros, el siguiente paso es convertirla, creando el modelo convertido. El modelo convertido es una representación de la estructura de la base de datos de destino. SQL Developer crea el modelo convertido utilizando la información del modelo capturado.

De forma predeterminada, todos los procedimientos, funciones, activadores y vistas se copian al modelo convertido durante la traducción y se traducen a Oracle PL / SQL. Sin embargo, si la traducción falla para cualquiera de los objetos, esos objetos aparecen en el modelo convertido pero su código SQL original permanece sin cambios. Los objetos que permanecen en su código SQL original no se utilizarán cuando se creen los scripts de generación. Por lo tanto, para migrar cualquiera de estos objetos, debe corregir el problema en el código SQL original antes de generar el script o editar el script generado para reemplazar el código SQL original con un código PL / SQL válido.

Para convertir un modelo capturado en un modelo convertido, haga clic con el botón derecho en el nodo apropiado en el navegador de Modelos capturados y seleccione **Convertir a Oracle** , y especifique o acepte los valores predeterminados para las asignaciones de datos.

El siguiente tema describe cómo modificar el modelo convertido, si es necesario:

* [Corrección de errores en el modelo convertido](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJGEHC)

### Corrección de errores en el modelo convertido

Si los mensajes de error con el prefijo Parse Exceptionse enumeran en el registro de migración, se requiere la intervención manual para resolver los problemas. Para completar el modelo convertido:

1. Tenga en cuenta el objeto de esquema de modelo convertido que falló.
2. Seleccione ese objeto de esquema en el modelo convertido.
3. Copie los objetos de esquema DDL y péguelos en el editor de scratch de traducción (que se muestra al hacer clic en Migration, luego en Translation Scratch Editor).
4. Inspeccione las propiedades en el objeto de esquema en el editor de scratch de traducción para las posibles causas del error.
5. Modifique una propiedad del objeto de esquema en el editor de scratch de traducción.

Por ejemplo, puede comentar una línea de un procedimiento almacenado.

1. Traducir utilizando el traductor adecuado.
2. Si vuelve a aparecer el error, repita los pasos [2](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005778) a 6.
3. Si el error no se puede resolver de esta manera, es mejor modificar el objeto manualmente en el modelo convertido.

### Generando el DDL para los objetos de esquema de Oracle

Para generar las declaraciones DDL para crear los objetos de esquema de Oracle, ya debe haber capturado el modelo capturado y creado el modelo convertido. Después de generar el DDL, ejecute las declaraciones DDL para hacer que los objetos se creen en la base de datos de Oracle. En este punto, el esquema de la base de datos se migra a Oracle.

Después de generar y ejecutar las declaraciones DDL para migrar los objetos del esquema, puede migrar los datos de la base de datos de origen original, como se explica en [Migración de los datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJCGGI) .

### Migración de los datos

Una vez que haya generado y ejecutado sentencias DDL para crear los objetos de esquema de Oracle para la base de datos migrada, puede migrar (mover) cualquier dato existente de la base de datos de origen a la base de datos de Oracle. Tiene dos opciones para la migración de datos: en línea o fuera de línea.

* **Movimiento de** datos en **línea** : haga clic en **Migración** , luego en **Migrar datos** . En el cuadro de diálogo, especifique la Conexión de origen, la Conexión de destino y el Modelo convertido. Este método utiliza JDBC y, por lo tanto, está restringido por las implementaciones de terceros. Los movimientos de datos en línea son adecuados para pequeños conjuntos de datos.
* **Desconectado** mover datos: Haga clic en **Migración** , a continuación, **generar Mover línea de datos de secuencias de comandos** ; especifique el modelo convertido y un directorio en el que generar los archivos que utilizará para descargar los datos de la base de datos de origen y para importarlos a Oracle utilizando SQL \* Loader. El enfoque de movimiento de datos fuera de línea está diseñado para mover grandes volúmenes de datos.

### Transferencia de los datos fuera de línea

Para transferir los datos fuera de línea, genere y use scripts para copiar datos de la base de datos de origen a la base de datos de destino. Durante este proceso usted debe:

* Utilice SQL Developer para generar los scripts de descarga de datos para la base de datos de origen y los scripts de carga de datos correspondientes para la base de datos de destino.
* Ejecute los scripts de descarga de datos para crear archivos de datos desde la base de datos de origen utilizando el procedimiento adecuado para su base de datos de origen:
  + [Creación de archivos de datos desde Microsoft SQL Server o Sybase Adaptive Server](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDCIHHA)
  + [Creando archivos de datos desde Microsoft Access](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDIBDBI)
  + [Creando archivos de datos desde MySQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDDCBC)
* Ejecute los scripts de carga de datos utilizando SQL \* Loader para completar la base de datos de destino con los datos de estos archivos de datos, tal como se describe en Cómo llenar la base de datos de destino con los [archivos de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDEAJCE) .

Creando archivos de datos desde Microsoft SQL Server o Sybase Adaptive Server

Para crear archivos de datos desde una base de datos de Microsoft SQL Server o Sybase Adaptive Server:

1. Copie el contenido del directorio donde SQL Developer generó los scripts de descarga de datos en la computadora donde está instalada la base de datos de origen.
2. Edite el script de extracción BCP para incluir el nombre del servidor de base de datos de origen.
   * En Windows, edite el unload\_script.batscript para alterar las líneas de bcp para incluir las variables apropiadas.

A continuación se muestra una línea de un unload\_script.batscript de ejemplo :

bcp "AdventureWorks.dbo.AWBuildVersion" out "[AdventureWorks]. [dbo]. [AWBuildVersion] .dat" -q -c -t "<EOFD>" -r "<EORD>" -U <Username> -P < Contraseña> -S <NombreServidor>

1. Ejecute el script de extracción BCP.
   * En Windows, introduzca:
   * solicitud> unload\_script.bat

Este script crea los archivos de datos en el directorio actual.

1. Copie los archivos de datos y los scripts, si es necesario, en el sistema de base de datos de Oracle de destino o en un sistema que tenga acceso a la base de datos de Oracle de destino y que tenga instalado SQL \* Loader (Cliente de Oracle).

Creando archivos de datos desde acceso Microsoft

Para crear archivos de datos desde una base de datos de Microsoft Access, use la herramienta Exporter para Microsoft Access.

**Nota:**

Para obtener información sobre cómo crear archivos de datos desde una base de datos de Microsoft Access, consulte la ayuda en línea de la herramienta para exportadores.

Creando archivos de datos desde MySQL

Para crear archivos de datos desde una base de datos MySQL:

1. Copie el contenido del directorio donde SQL Developer generó los scripts de descarga de datos, si es necesario, en el sistema donde está instalada la base de datos de origen o en un sistema que tenga acceso a la base de datos de origen y tenga instalada la herramienta mysqldump.
2. Edite el unload\_scriptscript para incluir el host correcto, el nombre de usuario, la contraseña y el directorio de destino para los archivos de datos.
   * En Windows, edite el unload\_script.batscript.
   * En Linux o UNIX, edite el unload\_script.shscript.

A continuación se muestra una línea de un unload\_script.batscript de ejemplo :

mysqldump -h localhost -u <USERNAME> -p <PASSWORD> -T <DESTINATION\_PATH>

--fields-terminated-by = "<EOFD>" --fields-escaped-by = ""

--lines-terminated-by = "<EORD>" "CarrierDb" "CarrierPlanTb"

Edite esta línea para incluir los valores correctos para *USERNAME, PASSWORD* y *DESTINATION PATH* . No incluya los corchetes angulares en la versión editada de este archivo.

En esta línea de comando, localhostindica una conexión de bucle invertido, que es requerida por la opción -T. (Consulte la documentación de mysqldump para obtener más información).

1. Ejecutar el script.
   * En Windows, introduzca:
   * solicitud> unload\_script.bat
   * En Linux o UNIX, ingrese:
   * solicitud> chmod 755 unload\_script.sh
   * solicitud> sh ./unload\_script.sh

Este script crea los archivos de datos en el directorio actual.

1. Copie los archivos de datos y los scripts, si es necesario, en el sistema de base de datos de Oracle de destino o en un sistema que tenga acceso a la base de datos de Oracle de destino y que tenga instalado SQL \* Loader (Cliente de Oracle).

Poblando la base de datos de destino usando los archivos de datos

Para llenar la base de datos de destino con los archivos de datos, ejecute los scripts de carga de datos con SQL \* Loader:

1. Navegue hasta el directorio donde creó los scripts de descarga de datos.
2. Edite el oracle\_ctl.batarchivo (sistemas Windows) o oractl\_ctl.sh(sistemas Linux o UNIX) para proporcionar el nombre de usuario y las cadenas de contraseña adecuados.
3. Ejecute el script de carga de SQL.
   * En Windows, introduzca:
   * indicador> oracle\_ctl.bat
   * En Linux o UNIX, ingrese:
   * solicitud> ./oracle\_ctl.sh

Para las migraciones de Microsoft SQL Server y Sybase, si está insertando en los campos BLOB con SQL \* Loader, recibirá el siguiente error:

SQL \* Loader-309: No se permiten cadenas SQL como parte de la especificación del campo LARGEOBJECT

Para manejar las situaciones indicadas por este error, puede usar una de las siguientes opciones:

* Habilite la preferencia Generar procedimiento almacenado para migrar blobs sin conexión al desarrollador de SQL (vea [Migración: Opciones de generación](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGAIBA) ).
* Utilice la siguiente [solución](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDBDBGA) .

**Solución**

La solución es cargar los datos (que están en formato hexadecimal) en un campo CLOB adicional y luego convertir el CLOB en un BLOB a través de un procedimiento PL / SQL.

La única forma de exportar datos binarios correctamente a través de Microsoft SQL Server o Sybase Adaptive Server BCP es exportarlos en un formato hexadecimal (hex); sin embargo, para obtener los valores hexadecimales en Oracle, guárdelos en una columna CLOB (contiene texto) y luego convierta los valores hexadecimales en valores binarios e insértelos en la columna BLOB. El problema aquí es que la función HEXTORAW en Oracle solo convierte un máximo de 2000 pares de hex. En consecuencia, escriba su propio procedimiento que convertirá (pieza por pieza) sus datos hexadecimales en binarios. (En los siguientes pasos y ejemplos, modifique START.SQL y FINISH.SQL para reflejar su entorno.

A continuación se muestra el código para dos scripts start.sqly finish.sql, que implementan esta solución. Lea los comentarios en el código y modifique cualquier declaración SQL según sea necesario para reflejar su entorno y sus necesidades.

**Nota:**

Después de ejecutar start.sqly antes de ejecutar finish.sql, ejecute BCP; y antes de ejecutar BCP, cambie la línea relevante en el .ctlarchivo desde:

<blob\_column> CHAR (2000000) "HEXTORAW (: <blob\_column>)"

a:

<blob\_column> \_CLOB CHAR (2000000)

- START.SQL

- Modifica esto para tu entorno.

- Esto debe ejecutarse en el esquema de usuario en Oracle que contiene la tabla.

- DESCRIPCIÓN:

- ALTERA LA TABLA OFENSA PARA QUE SE PUEDA EJECUTAR EL MOVIMIENTO DE DATOS

- DESHABILITAR LOS GATILLAS, ÍNDICES Y SECUENCIAS EN LA TABLA OFENSA

- 1) Agrega una columna extra para mantener la cadena hexadecimal;

modifique la tabla <tablename> add (<blob\_column> \_CLOB CLOB);

- 2) Permitir que la columna BLOB acepte NULLS

alterar tabla <nombre de archivo> MODIFICAR <blob\_column> NULL;

- 3) Deshabilitar los disparadores y las secuencias en <tablename>

alterar desencadenante <triggername> inhabilitar;

alterar tabla <nombre de archivo> soltar cascada de clave principal;

índice de caída <indexname>;

- 4) Permitir que la tabla use el tablespace

alterar tabla <nombre de archivo> mover lob (<blob\_column>) almacén como (espacio de tabla lob\_tablespace);

alterar tabla <nombre de archivo> mover lob (<blob\_column> \_clob) almacén como (espacio de tabla lob\_tablespace);

COMETER;

-- FIN DEL DOCUMENTO

- Acabado.SQL

- Modifica esto para tu entorno.

- Esto debe ejecutarse en el esquema de tabla en Oracle.

- DESCRIPCIÓN:

- MUEVE LOS DATOS DE CLOB A BLOB

- MODIFICA LA MESA DE VUELTA A SU ESPEC. ORIGINAL (sin un clob)

- ENTONCES HABILITAN LAS SECUENCIAS, DISPAROS Y ÍNDICES DE NUEVO

- Actualmente tenemos los valores hexadecimales guardados como

- texto en la columna \_CLOB <blob\_column>

- Y tenemos NULL en todas las filas para la columna <blob\_column>.

- Tenemos que obtener localizadores BLOB para cada fila en la columna BLOB

- Poner manchas vacías en la columna de burbujas

UPDATE <tablename> SET <blob\_column> = EMPTY\_BLOB ();

COMETER;

- crea el siguiente procedimiento en el esquema de tu tabla

CREAR O REEMPLAZAR EL PROCEDIMIENTO DE CLOBTOBLOB

COMO

inputLength NUMBER; - Tamaño de entrada CLOB

offSet NUMBER: = 1;

pieceMaxSize NUMBER: = 2000; - El tamaño máximo de cada pieza.

pieza VARCHAR2 (2000); - Estas piezas conformarán todo el RELOJ.

currentPlace NUMBER: = 1; - Aquí es donde se metieron en el CLOB.

blobloc BLOB; - localizador de burbujas en la tabla

clobLoc CLOB; - puntos del localizador de clob: este es el valor del archivo dat

- ESTO TIENE QUE SER CAMBIADO POR TABLA DE CLIENTES ESPECÍFICOS

- Y NOMBRES DE COLUMNA

CURSOR cur IS SELECT <blob\_column> \_clob clob\_column, <blob\_column> blob\_column FROM / \* table \* / <tablename> PARA ACTUALIZACIÓN;

cur\_rec cur% ROWTYPE;

EMPEZAR

ABIERTO cur;

FETCH cur INTO cur\_rec;

Mientras que cur% se encontró

LAZO

--Resolver el clobLoc y blobLoc

clobLoc: = cur\_rec.clob\_column;

blobLoc: = cur\_rec.blob\_column;

currentPlace: = 1; - restablecer evertime

- encontrar la longitud del clob

inputLength: = DBMS\_LOB.getLength (clobLoc);

- bucle a través de cada pieza

LAZO

- Consigue la siguiente pieza y agrégala al clob.

piece: = DBMS\_LOB.subStr (clobLoc, pieceMaxSize, currentPlace);

- Añadir esta pieza al BLOB.

DBMS\_LOB.WRITEAPPEND (blobLoc, LENGTH (pieza) / 2, HEXTORAW (pieza));

currentPlace: = currentPlace + pieceMaxSize;

SALIR CUANDO inputLength <currentplace;

Bucle final;

FETCH cur INTO cur\_rec;

Bucle final;

END CLOBtoBLOB;

/

- ahora ejecuta el procedimiento

- Se actualizará la columna blob con la representación binaria correcta.

- de la columna clob

EXEC CLOBtoBLOB;

- suelta la cloumn extra clob

alterar tabla <tablename> soltar columna <blob\_column> \_clob;

- 2) aplicar la restricción que eliminamos durante la carga de datos

alter table <tablename> MODIFY FILEBINARY NOT NULL;

- Ahora vuelve a habilitar los disparadores, índices y claves primarias.

alterar activador <triggername> habilitar;

ALTER TABLE <tablename> ADD (CONSTRAINT <pkname> KEY PRINCIPAL (<column>));

CREATE INDEX <index\_name> ON <tablename> (<column>);

COMETER;

-- FIN DEL DOCUMENTO

### Hacer que las consultas sean insensibles

Con varias bases de datos de terceros, es común que las consultas no distingan entre mayúsculas y minúsculas. Por ejemplo, en tales casos, las siguientes consultas devuelven los mismos resultados:

SELECCIONA \* DE ÓRDENES DONDE sales\_rep = 'Oracle';

SELECCIONA \* DE ÓRDENES DONDE sales\_rep = 'oracle';

SELECCIONE \* DE ÓRDENES DONDE sales\_rep = 'OrAcLe';

Si desea que las consultas no distingan entre mayúsculas y minúsculas para un usuario en la base de datos de Oracle, puede crear un desencadenador DESPUÉS DE INICIAR SESIÓN EN LA BASE DE DATOS, en el que estableció, para ese usuario de la base de datos, el parámetro de sesión NLS\_SORT a un nombre de clasificación de Oracle con \_CI(para "el caso" insensible ") adjunto.

El siguiente ejemplo hace que las consultas del usuario SMITH utilicen el orden de clasificación alemán y que no distingan entre mayúsculas y minúsculas:

CREAR O REEMPLAZAR EL GATILLO set\_sort\_order DESPUÉS DE INICIAR SESIÓN EN LA BASE DE DATOS

DECLARAR

nombre de usuario VARCHAR2 (30);

EMPEZAR

nombre de usuario: = SYS\_CONTEXT ('USERENV', 'SESSION\_USER');

SI el nombre de usuario ME GUSTA 'SMITH' entonces

ejecute inmediatamente 'alter sesion set NLS\_COMP = LINGUISTIC';

ejecute inmediatamente 'alter session set **NLS\_SORT = GERMAN\_CI** ';

TERMINARA SI;

FIN;

### Probando la base de datos Oracle

Durante la fase de prueba, prueba la aplicación y la base de datos Oracle para asegurarse de que:

* Los datos migrados son completos y precisos.
* Las aplicaciones funcionan de la misma manera que la base de datos de origen.
* Base de datos Oracle produciendo los mismos resultados que la base de datos fuente.
* Las aplicaciones y la base de datos Oracle cumplen con los requisitos operativos y de rendimiento.

Es posible que ya tenga una colección de pruebas unitarias y de sistema de la aplicación original que puede usar para probar la base de datos Oracle. Debe ejecutar estas pruebas de la misma manera que las pruebas en la base de datos de origen. Sin embargo, independientemente de las características agregadas, debe asegurarse de que la aplicación se conecte a la base de datos Oracle y que las declaraciones SQL que emite produzcan los resultados correctos.

**Nota:**

Las pruebas que ejecute contra la aplicación varían según el alcance de la aplicación. Oracle recomienda que pruebe exhaustivamente cada sentencia de SQL que se cambia en la aplicación. También debe probar el sistema para asegurarse de que la aplicación funciona de la misma manera que en la base de datos de terceros.

Vea también lo siguiente:

* [Metodología de Pruebas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDIDABH)
* [Probando la base de datos Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDGFHCF)

### Metodología de Pruebas

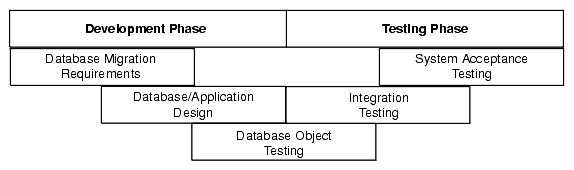
Muchas restricciones dan forma al estilo y la cantidad de pruebas que realiza en una base de datos. Las pruebas pueden contener uno o todos los siguientes:

* Validación de datos simple
* Ciclo de vida completo de pruebas dirigidas a pruebas de unidades individuales
* Sistema y pruebas de aceptación.

Debe seguir una estrategia de prueba que se adapte a su organización y circunstancias. Su estrategia debe definir el proceso mediante el cual prueba la aplicación migrada y la base de datos Oracle. Un método de prueba típico es el modelo V, que es un enfoque por etapas en el que cada característica de la creación de la base de datos se refleja en una fase de prueba.

[La Figura 2-2, "Modelo V con una migración de base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#BHAFDAIH) muestra un ejemplo del modelo V con un escenario de migración de base de datos:

***Figura 2-2 Modelo V con una migración de base de datos***

  
[Descripción de la "Figura 2-2 modelo V con una migración de base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/img_text/vmodel.htm)

Hay varios tipos de pruebas que utiliza durante el proceso de migración. Durante la etapa de prueba, realiza varios ciclos de prueba para mejorar la calidad de la base de datos. Los casos de prueba que use deben asegurarse de que los problemas encontrados en una versión anterior de la base de datos Oracle no se vuelvan a introducir.

Por ejemplo, si tiene que realizar cambios en el esquema migrado en función de los resultados de las pruebas, es posible que deba crear una nueva versión del esquema de la base de datos Oracle. En la práctica, usa SQL Developer para crear un esquema de línea base de Oracle al inicio de la prueba y luego edita este esquema a medida que avanza en la prueba.

**Nota:**

Oracle recomienda que realice un seguimiento de los problemas que encuentre durante un ciclo de prueba en un sistema de seguimiento de problemas. Realice un seguimiento de estos problemas con la versión de la base de datos o la aplicación que está probando.

### Probando la base de datos Oracle

Use los casos de prueba para verificar que la base de datos de Oracle proporcione los mismos resultados de lógica de negocios que la base de datos de origen.

**Nota:**

Oracle recomienda que defina los criterios de finalización para que pueda determinar el éxito de la migración.

Este procedimiento explica una forma de probar la base de datos migrada. Otros métodos están disponibles y pueden ser más apropiados para los requisitos de su negocio.

Para probar la base de datos Oracle:

1. Crear una versión controlada de la base de datos migrada.

Oracle recomienda que guarde los scripts de migración de base de datos en un sistema de control de origen.

1. Diseñe un conjunto de casos de prueba que puede usar para probar la base de datos Oracle desde la unidad hasta el nivel del sistema. Los casos de prueba deben:
   1. Asegúrese de lo siguiente:
      * Todos los usuarios en la base de datos de origen han migrado exitosamente
      * Los privilegios y subvenciones para los usuarios son correctos.
      * Las tablas tienen la estructura correcta, los valores predeterminados funcionan correctamente y no se produjeron errores durante el mapeo o la generación
   2. Valide que los datos se hayan migrado correctamente haciendo lo siguiente:
      * Comparación del número de filas en la base de datos de Oracle con las de la base de datos de origen
      * Calcular la suma de columnas numéricas en la base de datos de Oracle y compararlas con las de la base de datos de origen
   3. Asegúrese de que lo siguiente se aplica a las restricciones:
      * No puede ingresar claves primarias duplicadas
      * Las claves externas le impiden ingresar datos inconsistentes
      * Las restricciones de verificación le impiden ingresar datos no válidos
   4. Compruebe que los índices y las secuencias se crean correctamente.
   5. Asegúrese de que las vistas se hayan migrado correctamente haciendo lo siguiente:
      * Comparación del número de filas en la base de datos de Oracle con las de la base de datos de origen
      * Calcular la suma de columnas numéricas en la base de datos de Oracle y compararlas con las de la base de datos de origen
   6. Asegúrese de que los activadores, procedimientos y funciones se migran correctamente. Compruebe que se devuelven los valores correctos para los activadores y las funciones.
2. Ejecute los casos de prueba contra la base de datos migrada.
3. Cree un informe que evalúe los resultados del caso de prueba.

Estos informes le permiten evaluar los datos para calificar los errores, presentar informes de problemas y proporcionar a un cliente una versión controlada de la base de datos.

1. Si las pruebas pasan, vaya al paso [7](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#i1005807) .

Si todas las pruebas en los casos de prueba pasan o contienen errores aceptables, la prueba pasa. Si se producen errores aceptables, documéntelos en un informe de errores que pueda utilizar para fines de auditoría.

1. Si los casos de prueba fallan:
   1. Identificar la causa del error.
   2. Identificar los casos de prueba necesarios para comprobar los errores.
   3. Registre un problema en la versión controlada del código de base de datos migrado en el informe de problemas.
   4. Agregue el caso de prueba y una descripción del problema al sistema de seguimiento de incidentes de su organización, que podría ser una hoja de cálculo o un sistema de informe de errores. Aparte del caso de prueba, el registro de incidentes debe incluir lo siguiente:
      * Proporcione una descripción clara y concisa del incidente encontrado
      * Proporcione una descripción completa del entorno, como la plataforma y la versión de control de fuente.
      * Adjuntar la salida de la prueba, si es útil.
      * Indicar la frecuencia y previsibilidad del incidente.
      * Proporcionar una secuencia de eventos que conduzcan al incidente.
      * Describa el efecto en la prueba actual, los pasos de diagnóstico tomados y los resultados anotados
      * Describa el efecto posterior persistente, si lo hubiera
   5. Intenta arreglar los errores.
   6. Vuelva al paso [1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#BHAFEJBC) .
2. Identifique las pruebas de aceptación que puede usar para asegurarse de que la base de datos Oracle tenga un nivel de calidad aceptable.

### Pautas para crear pruebas

Es posible que ya tenga una colección de pruebas unitarias y de sistema de la aplicación original que puede usar para probar la base de datos Oracle. Sin embargo, si no tiene ninguna prueba de unidad o sistema, necesita crearlas. Al crear casos de prueba, utilice las siguientes pautas:

* Planifique, especifique y ejecute los casos de prueba, registrando los resultados de las pruebas.

La cantidad de pruebas que realice es proporcional al tiempo y los recursos disponibles para el proyecto de migración. Normalmente, la fase de prueba en un proyecto de migración puede llevar de un 40% a un 60% del esfuerzo de todo el proyecto.

* Identifique los componentes que está probando, el enfoque del diseño de la prueba y los criterios de finalización de la prueba.
* Define cada caso de prueba para que sea reproducible.

Una prueba que no sea reproducible no es aceptable para el seguimiento de problemas o para un proceso de auditoría.

* Divida la base de datos de origen en funciones y procedimientos y cree un caso de prueba para cada función o procedimiento. En el caso de prueba, indique lo que va a probar, defina los criterios de prueba y describa los resultados esperados.
* Registre el resultado esperado de cada caso de prueba.
* Verifique que los resultados reales cumplan con los resultados esperados para cada prueba.
* Defina los casos de prueba que producen resultados negativos, así como aquellos que espera un resultado positivo.

**Ejemplo de un caso de prueba de unidad**

A continuación se muestra un ejemplo de plan de prueba de unidad para Windows:

Nombre Jane Harrison

Módulo Tabla Prueba Emp

Prueba de fecha completada el 23 de mayo de 2007

Ubicación del archivo de registro de cobertura mwb \ database \ TableTestEmp

Descripción Esta prueba de unidad prueba que la tabla emp se migró correctamente.

Revisado por John Smith

| **ID de tarea** | **Descripción de la tarea** | **Resultado Esperado** | **Verificado (Sí / No)** |
| --- | --- | --- | --- |
| 1 | Ejecute lo siguiente en la base de datos de origen para cada tabla:  seleccione cuenta (\*) de emp  Ejecute lo siguiente en la base de datos de destino para cada tabla:  seleccione cuenta (\*) de emp | En la base de datos de origen, la cuenta (\*) produce un número. En este caso, el número es el número de filas en cada tabla.  En la base de datos de destino, el número de conteo (\*) corresponde al número de filas en la nueva tabla de Oracle. | Sí  El número de filas en cada tabla es el mismo en las bases de datos de origen y destino. |
| 2 | Ejecute lo siguiente en la base de datos de origen para cada tabla:  seleccione suma (salario) de emp  Ejecute lo siguiente en la base de datos de destino para cada tabla:  seleccione suma (salario) de emp | En la base de datos de origen, la suma (salario) produce una suma de verificación para la suma de los datos en cada tabla.  En la base de datos de destino, la suma (salario) corresponde a la suma del salario en la tabla emp. | Sí  La suma para cada tabla es la misma en las bases de datos de origen y destino. |

### Despliegue de la base de datos Oracle

La implementación de la base de datos Oracle migrada y probada en un entorno empresarial puede ser difícil. Por lo tanto, es posible que deba considerar diferentes estrategias de implementación según su entorno. Se identifican varias estrategias de implementación para usted, pero puede usar otro enfoque si así lo recomienda su organización.

Durante la fase de implementación, mueve la base de datos de destino de un desarrollo a un entorno de producción. Un grupo separado del equipo de migración y pruebas puede realizar la fase de implementación, como el departamento de TI interno.

El despliegue implica lo siguiente:

* [Elegir una estrategia de lanzamiento](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDBGEEF)
* [Despliegue de la base de datos de destino](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDFIAJB)

### Elegir una estrategia de lanzamiento

La estrategia que utilice para migrar una base de datos de terceros a una base de datos de Oracle debe tener en cuenta los usuarios y el tipo de negocio que puede verse afectado durante el período de transición. Por ejemplo, puede utilizar el enfoque de Big Bang porque no tiene suficientes sistemas para ejecutar la base de datos de origen y la base de datos de Oracle simultáneamente. De lo contrario, es posible que desee utilizar el enfoque por fases para asegurarse de que el sistema esté funcionando correctamente en el entorno del usuario antes de que se publique en la población general de usuarios. Puede utilizar uno de los siguientes enfoques.

**Enfoque por fases**

Usando el enfoque por fases, migra grupos de usuarios en diferentes momentos. Puede decidir migrar un departamento o un subconjunto de la base de usuarios completa. Los usuarios que seleccione deben representar una sección transversal de la base de usuarios completa. Este enfoque le permite crear perfiles de usuarios a medida que los introduce en la base de datos Oracle. Puede reconfigurar el sistema para que solo los usuarios seleccionados se vean afectados por la migración y las interrupciones no programadas solo afecten a un pequeño porcentaje de la población de usuarios. Este enfoque puede afectar el trabajo de los usuarios que migró. Sin embargo, debido a que el número de usuarios es limitado, los servicios de soporte no están sobrecargados de problemas.

El enfoque por fases le permite depurar problemas de escalabilidad a medida que aumenta el número de usuarios migrados. Sin embargo, el uso de este enfoque puede significar que debe migrar datos hacia y desde sistemas heredados durante el proceso de migración. La arquitectura de la aplicación debe soportar un enfoque por fases.

**Enfoque del Big Bang**

Usando el enfoque de Big Bang, migras a todos los usuarios al mismo tiempo. Este enfoque puede ocasionar interrupciones en la programación durante el tiempo en que elimina el sistema anterior, migra los datos, implementa el sistema Oracle y prueba que el sistema está funcionando correctamente. Este enfoque se basa en que usted prueba la base de datos en la misma escala que la base de datos original. Tiene la ventaja de la mínima conversión y sincronización de datos con la base de datos original porque esa base de datos está desactivada. La desventaja es que este enfoque puede requerir mucha mano de obra e interrumpir las actividades comerciales debido al cambio en el período necesario para instalar la base de datos Oracle y realizar las demás tareas del proyecto de migración.

**Enfoque paralelo**

Usando el enfoque Paralelo, usted mantiene simultáneamente la base de datos de origen y la base de datos de destino de Oracle. Para asegurarse de que la aplicación se comporte de la misma manera en el entorno de producción para la base de datos de origen y la base de datos de destino, ingrese los datos en ambas bases de datos y analice los resultados de los datos. La ventaja de este enfoque es que si se producen problemas en la base de datos de destino, los usuarios pueden continuar utilizando la base de datos de origen. La desventaja del enfoque paralelo es que la ejecución y el mantenimiento de la base de datos de origen y de destino pueden requerir más recursos y hardware que otros enfoques.

### Despliegue de la base de datos de destino

Hay varias formas de implementar la base de datos de destino. La siguiente tarea es un ejemplo que debe usar como guía para implementar la base de datos de destino.

**Nota:**

Si tiene un escenario complejo como se define en la [Tabla 2-1](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CACBAAJI) , Oracle recomienda que complete todas las tareas de implementación. Sin embargo, si tiene un escenario simple, debe elegir las tareas de implementación adecuadas para su organización.

1. Configure el hardware, si es necesario.

En un entorno complejo o de gran escala, debe diseñar la distribución del disco para que se corresponda con el diseño de la base de datos. Si utiliza discos redundantes, alinéelos en bandas que puede aumentar a medida que evoluciona la base de datos de destino. Debe instalar y configurar los discos necesarios, verificar la memoria y configurar el sistema.

1. Asegúrese de que el sistema operativo cumpla con los parámetros de la configuración de Oracle.

Antes de instalar cualquier software de Oracle, asegúrese de haber modificado todos los parámetros del sistema. Para obtener más información sobre la modificación de los parámetros del sistema, consulte la guía de instalación correspondiente a su plataforma, como el sistema operativo Solaris.

1. Instale el software de Oracle.

Aparte del software de Oracle que le permite crear una base de datos Oracle, es posible que deba instalar un software auxiliar para admitir la aplicación, como el software de extracción de transformación y carga (ETL) para el almacenamiento de datos.

1. Cree la base de datos de destino a partir de la base de datos de origen y migre los datos a la base de datos de Oracle.

Hay varias formas de poner en producción la base de datos de destino después de probarla, como por ejemplo:

* + Coloque la base de datos probada con éxito en producción. El sistema de prueba es ahora el sistema de producción.
  + Use Oracle Export para extraer la base de datos de destino de la base de datos probada con éxito y use Oracle Import para crear esa base de datos dentro del entorno de producción.
  + Utilice los scripts de migración probados para crear la base de datos Oracle y rellénela con datos usando SQL \* Loader.

1. Realice las comprobaciones finales en la base de datos de destino y las aplicaciones.
2. Coloque la base de datos de destino en producción utilizando una de las estrategias de implementación.
3. Realice una auditoría final haciendo lo siguiente:
   * Auditar la integridad de los datos.
   * Auditar la validez de los procesos, como el respaldo y la recuperación.
   * Obtener la aprobación del proyecto, si es necesario.

### Uso de informes de migración

Varios informes de SQL Developer proporcionan información sobre los objetos que se han capturado, convertido y generado durante las operaciones diseñadas para migrar bases de datos de terceros a Oracle. Cada informe utiliza información de un proyecto de migración seleccionado. Estos informes se enumeran en el navegador de informes: haga clic en **Informes de migración** .

**Cambios automáticos de nombre** : enumera los cambios de nombre que se realizaron automáticamente cuando se generó el modelo convertido. Algunos nombres se cambian automáticamente durante la conversión, de modo que son compatibles con Oracle y no existen colisiones.

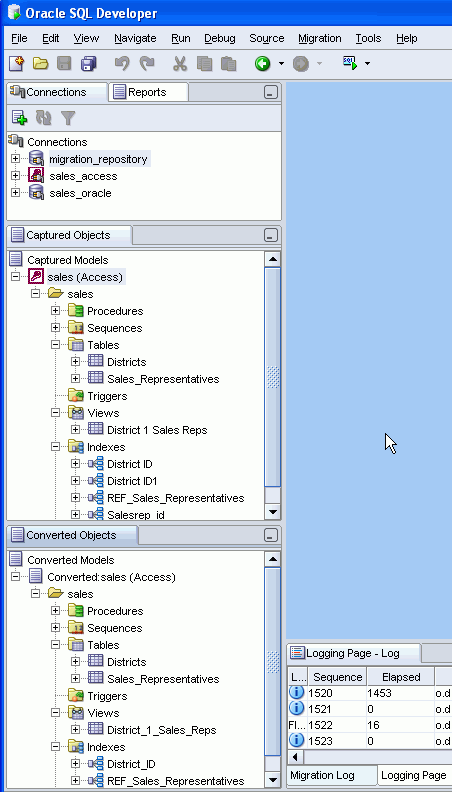
**Detalles de la migración** : enumera todos los objetos y su estado a través de cada fase de la migración (captura, conversión, generación).

**Resumen de migración** : proporciona un resumen de la migración. Incluye el número total de procedimientos, activadores y vistas que se capturaron y convirtieron. y generado con éxito.

### Interfaz de usuario de desarrollador SQL para la migración

Si está realizando una migración de la base de datos, necesita usar algunas características específicas de la migración además de las descritas en la [Interfaz de usuario del desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAHIFGB) . La interfaz de usuario incluye algunas pestañas y paneles adicionales (Modelos capturados y Modelos convertidos) y un menú de Migración, y muchos cambios más pequeños en toda la interfaz. [La Figura 2-3, "Ventana principal para una migración de base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDDJCGI) muestra la ventana principal de SQL Developer con objetos que reflejan la migración de una aplicación de Microsoft Access llamada sales.mdb.

***Figura 2-3 Ventana principal para una migración de base de datos***



En esta figura:

* El navegador de conexiones muestra tres conexiones de base de datos: migration\_repositorypara una conexión (a un usuario llamado MIGRATION) utilizada para el repositorio de migración, sales\_accessconectada a una base de datos de Microsoft Access llamada sales.mdb, y sales\_oracleconectada a un usuario de Oracle llamado SALES cuyo esquema es propietario de los objetos del esquema migrado .
* El navegador de **modelos capturados** muestra un modelo capturado, que se creó utilizando un archivo XML creado por la herramienta exportadora para aplicaciones de Access. (Si la base de datos de origen es un tipo diferente a Microsoft Access, el procedimiento para crear el modelo capturado es diferente: puede generarlo directamente desde la conexión de la base de datos de origen).
* El navegador de **modelos convertidos** muestra un modelo convertido, que es una representación de Oracle de la base de datos de origen. El modelo convertido se crea a partir del modelo capturado y el modelo convertido se utiliza para generar los objetos de esquema que puede ver usando una conexión de base de datos Oracle (sales\_oracle en esta figura).

### Menú de migración

El menú Migración contiene opciones relacionadas con la migración de bases de datos de terceros a Oracle.

**Migración rápida** : Muestra un cuadro de diálogo para realizar una migración rápida usando muchos valores por defecto.

**Gestión del repositorio** : le permite crear, eliminar o truncar (eliminar todos los datos) de un repositorio de migración; seleccione el repositorio de migración actual; y desconectarse del repositorio de migración actual (que desactiva el repositorio actual pero no se desconecta de la base de datos).

**Microsoft Access Exporter** : contiene elementos de submenú desde los cuales se especifica la versión de la herramienta de exportador que se usará para crear un archivo XML que se usará para crear el modelo capturado. También puede utilizar la herramienta de exportación para exportar datos de tabla. Especifique la versión de la herramienta de exportación para la versión de Access que está en su PC y que se usó para crear el archivo .mdb.

**Migrar datos** : muestra un cuadro de diálogo para realizar una migración en línea de los datos de la tabla desde la base de datos de origen al esquema de Oracle.

**Generación de secuencias de comandos** : **Generar Oracle DDL** muestra las declaraciones DDL (lenguaje de definición de datos) en una ventana de hoja de cálculo SQL, donde puede ejecutar la secuencia de comandos para crear el esquema de Oracle y los objetos de esquema; **Generar datos Mover secuencias de comandos** muestra un cuadro de diálogo para especificar la ubicación en la que se crearán los archivos para realizar una migración sin conexión de los datos de la tabla desde la base de datos de origen al esquema de Oracle.

**Capture el XML de Microsoft Access Exporter** : crea un modelo capturado de una base de datos de Microsoft Access a partir del archivo XML creado por la herramienta del exportador.

**Captura sin conexión de MySQL, SQL Server y Sybase** : **crear secuencias de comandos de captura de base de datos** especifica las opciones para crear un archivo de propiedades de captura sin conexión (.ocp), que luego puede cargar y ejecutar; **La salida del script de captura de la base de datos de carga le** permite seleccionar un script para cargar y ejecutar.

**Generación de scripts** : **Generar Oracle DDL** especifica el modelo convertido para el cual generar DDL de Oracle y genera un archivo de script SQL \* Plus que utiliza para la generación sin conexión (es decir, puede ejecutar el script para crear los objetos adecuados en la base de datos de Oracle); **Generate Data Move Scripts** especifica el modelo convertido y el directorio de destino si está realizando una migración de datos fuera de línea.

**Editor de Scratch de traducción** : muestra el editor de scratch de traducción, que se explica en [Uso del Editor de Scratch de traducción](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDCJDBJ) .

### Otros menús: Elementos de migración

El menú **Ver** tiene los siguientes elementos relacionados con la migración de la base de datos:

* **Modelos capturados** : muestra el navegador de modelos capturados.
* **Modelos convertidos** : muestra el navegador de modelos convertidos.

### Preferencias de migración

La ventana de preferencias de usuario de SQL Developer (que se muestra al hacer clic en **Herramientas** , luego en **Preferencias** ) contiene un panel de **Migración** con varios subpanes relacionados y un panel de **Traducción** con un subpano de Preferencias de traducción.

Para obtener información sobre estas preferencias, haga clic en Ayuda en el panel, o vea [Migración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJJGBH) .

### Paneles de registro de migración

**Registro de migración** : contiene errores, advertencias y mensajes informativos relacionados con las operaciones de migración.

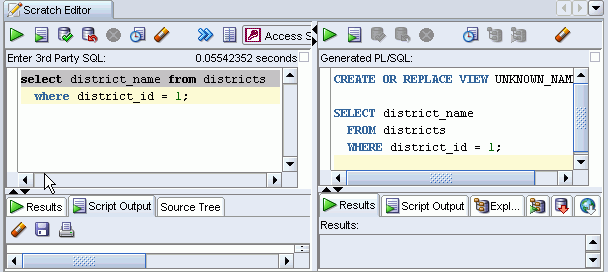
**Página de registro** : contiene una entrada para cada operación relacionada con la migración.

**Registro del editor de datos** : contiene entradas cuando los datos están siendo manipulados por el desarrollador de SQL. Por ejemplo, la salida de una operación de importación de Microsoft Excel se informará aquí como una serie de sentencias INSERT.

### Usando el editor de Scratch de traducción

Puede utilizar el editor de scratch de conversión para ingresar sentencias de SQL de bases de datos de terceros y traducirlas a sentencias de PL / SQL de Oracle. Puede especificar la traducción de Microsoft SQL Server T-SQL a PL / SQL, de Sybase T-SQL a PL / SQL, o de Microsoft Access SQL a PL / SQL.

Puede visualizar el editor de scratch haciendo clic en **Migración** , luego en **Translation Scratch Editor** . El editor de scratch consta de dos ventanas de la hoja de trabajo SQL, una al lado de la otra, como se muestra en la siguiente figura:



Para traducir una declaración a su equivalente de Oracle, seleccione el tipo de traducción, ingrese las declaraciones o declaraciones de SQL de terceros, luego haga clic en el icono **Traducir** (>>) para mostrar las instrucciones o declaraciones PL / SQL generadas.

Las palabras clave de SQL se resaltan automáticamente.

**Nota:**

Para una conexión de Microsoft SQL Server o Sybase Adaptive Server, la hoja de trabajo no admite la ejecución de sentencias T-SQL. Solo admite sentencias SELECT, CREATE, INSERT, UPDATE, DELETE y DROP.

La primera vez que guarde el contenido de cualquiera de las ventanas de la hoja de trabajo en el editor de conversión de la traducción, se le solicitará la ubicación y el nombre del archivo. Si realiza cualquier operación de **Guardar** posterior (independientemente de si ha borrado o cambiado el contenido de la ventana), el contenido se guarda en el mismo archivo. Para guardar el contenido en un archivo diferente, haga clic en **Archivo** , luego **Guardar como** .

## 3 Tutorial: Crear objetos para una pequeña base de datos

En este tutorial, utilizará SQL Developer para crear objetos para una base de datos de biblioteca simplificada, que incluirá tablas para libros, usuarios (personas que tienen tarjetas de biblioteca) y transacciones (sacar un libro, devolver un libro, etc.) .

Las tablas están simplificadas en exceso para este tutorial. No serían adecuados para ninguna biblioteca pública u organizativa real. Por ejemplo, esta biblioteca solo contiene libros (no revistas, revistas u otros formatos de documentos), y no puede contener más de una copia de cualquier libro.

Realizarás los siguientes pasos principales:

1. [Crear una tabla (LIBROS)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHGHGF) .
2. [Crear una tabla (PATRONES)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAGDDHE) .
3. [Crear una tabla (TRANSACCIONES)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAEJHJA) .
4. [Crear una secuencia](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAJGFHF) .
5. [Insertar datos en las tablas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAFGJBF) .
6. [Crear una vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFDCH) .
7. [Crear un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACEAIJ) .
8. [Depurar un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACAEJG) (opcional).
9. [Utilice la hoja de trabajo SQL para consultas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBADBDJF) (opcional).

**Nota:**

Para eliminar los objetos que cree para este tutorial, puede usar las declaraciones DROP al principio del script en [Script para crear y usar los objetos del tutorial de la biblioteca](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFFHI) .

**Temas relacionados**

[Script para crear y usar los objetos del tutorial de la biblioteca](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFFHI)

[Capítulo 1, "Conceptos y uso del desarrollador SQL"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFGBED)

[Interfaz de usuario de desarrollador SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAHIFGB)

[Objetos de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#BHAJFCIJ)

### Crear una tabla (LIBROS)

La tabla BOOKS contiene una fila para cada libro en la biblioteca. Incluye columnas de tipos de caracteres y números, una clave principal, una restricción única y una restricción de verificación. Utilizará el cuadro de diálogo Crear tabla para crear la tabla de forma declarativa; la tabla que cree será esencialmente la misma que si hubiera ingresado la siguiente declaración utilizando la hoja de trabajo de SQL:

Libros de CREATE TABLE (

book\_id VARCHAR2 (20),

título VARCHAR2 (50)

CONSTRAINT title\_not\_null NOT NULL,

author\_last\_name VARCHAR2 (30)

CONSTRAINT last\_name\_not\_null NOT NULL,

author\_first\_name VARCHAR2 (30),

calificación NÚMERO,

CONSTRAINT books\_pk CLAVE PRIMARIA (book\_id),

CONSTRAINT rating\_1\_to\_10 CHECK (la calificación ES NULA O

(calificación> = 1 y calificación <= 10)),

CONSTRAINT author\_title\_unique UNIQUE (author\_last\_name, title));

Para crear la tabla BOOKS, conéctese a la base de datos como usuario en el esquema que desea usar para este tutorial. Haga clic con el botón derecho en el nodo Tablas en la jerarquía de esquemas en el lado izquierdo, seleccione **Nueva tabla** e ingrese la siguiente información. (Si no se menciona una pestaña o campo, no ingrese nada para ella. Asegúrese de que la casilla Avanzada no esté marcada cuando comience a crear la tabla).

Para obtener información detallada sobre el cuadro de diálogo de la tabla y sus pestañas, consulte [Crear tabla (creación rápida)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAEJI) y [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .

**Esquema** : especifique su esquema actual como el esquema en el que se creará la tabla.

**Nombre** : LIBROS

Cree las columnas de la tabla utilizando la siguiente información. Después de crear cada columna, excepto la última (clasificación), haga clic en **Agregar columna** para agregar la siguiente columna. (Si accidentalmente hace clic en Aceptar en lugar de Agregar columna, haga clic con el botón derecho en la tabla LIBROS en la pantalla del navegador de Conexiones, seleccione Editar y continúe agregando columnas).

| **Nombre de columna** | **Tipo** | **tamaño** | **Otra información y notas** |
| --- | --- | --- | --- |
| book\_id | VARCHAR2 | 20 | Clave principal (Verifica automáticamente No Nulo; también se crea un índice en la columna de clave principal. Este es el código Dewey u otro identificador de libro). |
| título | VARCHAR2 | 50 | No nulo |
| author\_last\_name | VARCHAR2 | 30 | No nulo |
| author\_first\_name | VARCHAR2 | 30 |  |
| clasificación | NÚMERO |  | (Clasificación personal del libro por parte del bibliotecario, de 1 (mala) a 10 (excelente)) |

Después de ingresar la última columna (calificación), marque **Avanzado** (junto a Esquema). Esto muestra un panel para más opciones de tabla. Para esta tabla, utilizará los paneles Restricciones únicas y Restricciones de verificación.

**Panel de restricciones únicas**

Haga clic en **Agregar** para agregar una restricción única para la tabla, a saber, que la combinación de author\_last\_name y title debe ser única dentro de la tabla. (Esto está simplificado en exceso deliberadamente, ya que la mayoría de las bibliotecas importantes permitirán más de una copia de un libro en sus existencias. Además, la combinación de apellido y título no siempre es un cheque "a prueba de errores" para la singularidad, pero es suficiente para esto escenario simple.)

**Nombre** : author\_title\_unique

En **Columnas disponibles** , haga doble clic en TITLE y luego en AUTHOR\_LAST\_NAME para moverlas a las Columnas seleccionadas.

**Panel de restricciones de verificación**

Haga clic en **Agregar** para agregar una restricción de verificación para la tabla, es decir, que el valor de la columna de calificación es opcional (puede ser nulo), pero si se especifica un valor, debe ser un número del 1 al 10. Debe ingresar la condición usando Sintaxis SQL que es válida en una cláusula CHECK (pero no incluye la palabra clave CHECK o los paréntesis que encierran todo el texto de la cláusula CHECK).

**Nombre** : rating\_1\_to\_10

**Condición** : la calificación es nula o (calificación> = 1 y calificación <= 10)

Haga **clic** en **Aceptar** para terminar de crear la tabla.

Vaya a [Crear una tabla (PATRONES)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAGDDHE) para crear la siguiente tabla.

### Crear una tabla (PATRONES)

La tabla de PATRONES contiene una fila para cada usuario que puede sacar libros de la biblioteca (es decir, cada persona que tiene una tarjeta de la biblioteca). Incluye una columna de tipo de objeto (MDSYS.SDO\_GEOMETRY). Utilizará el cuadro de diálogo Crear tabla para crear la tabla de forma declarativa; la tabla que cree será esencialmente la misma que si hubiera ingresado la siguiente declaración utilizando la hoja de trabajo de SQL:

CREAR TABLA DE PATRONES (

patron\_id NUMBER,

last\_name VARCHAR2 (30)

CONSTRAINT patron\_last\_not\_null NOT NULL,

nombre primero VARCHAR2 (30),

street\_address VARCHAR2 (50),

city\_state\_zip VARCHAR2 (50),

ubicación MDSYS.SDO\_GEOMETRY,

CONSTRAINT patrons\_pk PRIMARY KEY (patron\_id));

El uso de una sola columna city\_state\_zip para toda esa información no es un buen diseño de base de datos; Se hace aquí simplemente para simplificar su trabajo en el tutorial.

La columna de ubicación (la geometría de Oracle Spatial que representa la dirección geocodificada del usuario) se limita a mostrar el uso de un tipo complejo (objeto).

Para crear la tabla de PATRONES, si aún no está conectado, conéctese a la base de datos como usuario para el esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Tablas en la jerarquía de esquemas en el lado izquierdo, seleccione **Nueva tabla**e ingrese la siguiente información. (Si no se menciona una pestaña o campo, no ingrese nada para ella. Asegúrese de que la casilla Avanzada no esté marcada cuando comience a crear la tabla).

**Esquema** : especifique su esquema actual como el esquema en el que se creará la tabla.

**Nombre** : PATRONES

Cree la mayoría de las columnas de la tabla utilizando la siguiente información. Después de crear cada columna, excepto la columna city\_state\_zip, haga clic en **Agregar columna** para agregar la siguiente columna. (Si accidentalmente hace clic en Aceptar en lugar de Agregar columna, haga clic con el botón derecho en la tabla PATRONES en la pantalla del navegador de Conexiones, seleccione Editar y continúe agregando columnas).

| **Nombre de columna** | **Tipo** | **tamaño** | **Otra información y notas** |
| --- | --- | --- | --- |
| patron\_id | NÚMERO |  | Clave primaria. (Número de identificación de usuario único, con valores que se crearán utilizando una secuencia que creará) |
| apellido | VARCHAR2 | 30 | No nulo |
| nombre de pila | VARCHAR2 | 30 |  |
| Dirección | VARCHAR2 | 30 |  |
| código postal | VARCHAR2 | 30 |  |

La última columna de la tabla (ubicación) requiere un tipo de datos complejo, para el que debe usar la pestaña Columnas con opciones avanzadas. Compruebe **Avanzado** (junto a Esquema). Esto muestra un panel para seleccionar más opciones de tabla.

En el panel Columnas, haga clic en el nombre de la columna city\_state\_zip y haga clic en el icono Agregar columna ( **+** ) para agregar lo siguiente como la última columna de la tabla.

| **Nombre de columna** | **Tipo** | **Otra información y notas** |
| --- | --- | --- |
| ubicación | Tipo complejo  Esquema: MDSYS  Tipo: SDO\_GEOMETRY | (Objeto de geometría de Oracle Spatial que representa la dirección geocodificada del usuario) |

Después de ingresar la última columna (ubicación), haga **clic** en **Aceptar** para terminar de crear la tabla.

Vaya a [Crear una tabla (TRANSACCIONES)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAEJHJA) para crear la tabla siguiente.

### Crear una tabla (TRANSACCIONES)

La tabla TRANSACCIONES contiene una fila para cada transacción que involucra a un usuario y un libro (por ejemplo, alguien que saca un libro o devuelve un libro). Incluye dos columnas de clave externa. Utilizará el cuadro de diálogo Crear tabla para crear la tabla de forma declarativa; la tabla que cree será esencialmente la misma que si hubiera ingresado la siguiente declaración utilizando la hoja de trabajo de SQL:

CREAR tablas de transacciones (

transaction\_id NUMBER,

patron\_id CONSTRAINT for\_key\_patron\_id

REFERENCIAS de los clientes (patron\_id),

book\_id CONSTRAINT for\_key\_book\_id

REFERENCIAS libros (book\_id),

fecha de transacción fecha

CONSTRAINT tran\_date\_not\_null NOT NULL,

tipo de transacción NÚMERO

CONSTRAINT tran\_type\_not\_null NOT NULL,

CONSTRAINT transaction\_pk PRIMARY KEY (transaction\_id));

Para crear la tabla TRANSACCIONES, si aún no está conectado, conéctese a la base de datos como usuario del esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Tablas en la jerarquía de esquemas en el lado izquierdo, seleccione **Nueva tabla**e ingrese la siguiente información. (Si no se menciona una pestaña o campo, no ingrese nada para ella. Asegúrese de que la casilla Avanzada no esté marcada cuando comience a crear la tabla).

**Esquema** : especifique su esquema actual como el esquema en el que se creará la tabla.

**Nombre** : TRANSACCIONES

Cree las columnas de la tabla utilizando la siguiente información. Después de crear cada columna, excepto la última (tipo de transacción), haga clic en **Agregar columna** para agregar la siguiente columna. (Si accidentalmente hace clic en Aceptar en lugar de Agregar columna, haga clic con el botón derecho en la tabla TRANSACCIONES en la pantalla del navegador de Conexiones, seleccione Editar y continúe agregando columnas).

| **Nombre de columna** | **Tipo** | **tamaño** | **Otra información y notas** |
| --- | --- | --- | --- |
| ID de transacción | NÚMERO |  | Clave primaria. (Número de ID de transacción único, con valores que se crearán utilizando un disparador y una secuencia que se creará automáticamente) |
| patron\_id | NÚMERO |  | (Clave externa; debe coincidir con un valor de patron\_id en la tabla PATRONES) |
| book\_id | VARCHAR2 | 20 | (Clave externa; debe coincidir con un valor de book\_id en la tabla BOOKS) |
| Fecha de Transacción | FECHA |  | (Fecha y hora de la transacción) |
| tipo de transacción | NÚMERO |  | (Código numérico que indica el tipo de transacción, como 1 para sacar un libro) |

Después de ingresar la última columna (tipo de transacción), marque **Avanzado** (junto a Esquema). Esto muestra un panel para seleccionar más opciones de tabla. Para esta tabla, utilizará los paneles Secuencias de columnas y Claves externas.

**Panel de secuencias de columnas**

Ya ha especificado TRANSACTION\_ID como la clave principal, y utilizará este panel solo para especificar que los valores de la columna de la clave principal se completarán automáticamente. Este enfoque conveniente utiliza un disparador y una secuencia (ambos creados automáticamente por el Desarrollador de SQL) y garantiza que cada valor de ID de transacción sea único.

**Columna** : TRANSACTION\_ID

**Secuencia** : Nueva secuencia

**Activación** : TRANSACTIONS\_TRG (el valor por defecto; se creará automáticamente una activación antes de insertar con este nombre).

**Pestaña de claves extranjeras**

1. Haga clic en **Agregar** para crear la primera de las dos claves externas para la tabla TRANSACCIONES.

**Nombre** : for\_key\_patron\_id

**Esquema referenciado** : nombre del esquema que contiene la tabla con la clave principal o la restricción única a la que se refiere esta clave externa. Use el esquema que ha estado usando para este tutorial.

**Tabla de referencia** : PATRONES

**Restricción referenciada** : PATRONS\_PK (el nombre de la restricción de clave principal para la tabla PATRONS. Asegúrese de que la **columna referenciada en el** valor que se muestra de PATRONES sea PATRON\_ID).

**Asociaciones: Columna Local** : PATRON\_ID

**Asociaciones: Columna de referencia en** PATRONES: PATRON\_ID

2. Haga clic en **Agregar** para crear la segunda de las dos claves externas para la tabla TRANSACCIONES.

**Nombre** : for\_key\_book\_id

**Esquema referenciado** : nombre del esquema que contiene la tabla con la clave principal o la restricción única a la que se refiere esta clave externa. Use el esquema que ha estado usando para este tutorial.

**Tabla de referencia** : LIBROS

**Restricción referenciada** : BOOKS\_PK (el nombre de la restricción de clave principal para la tabla BOOKS. Asegúrese de que la **columna referenciada en el** valor mostrado de **BOOKS** sea ​​BOOK\_ID).

**Asociaciones: Columna local** : BOOK\_ID

**Asociaciones: Columna referenciada en LIBROS** : BOOK\_ID

3. Haga **clic** en **Aceptar** para terminar de crear la tabla.

Has terminado de crear todas las tablas. Para crear una secuencia para su uso en la generación de valores de clave primaria únicos para la tabla PATRONES, vaya a [Crear una secuencia](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAJGFHF) .

### Crear una secuencia

Cree un objeto de secuencia, que se utilizará en las instrucciones INSERT para generar valores de clave primaria únicos en la tabla PATRONES. (No es necesario crear una secuencia para la clave principal en la tabla TRANSACCIONES, ya que usó la función Desarrollador de SQL que habilita el llenado automático de los valores de la clave primaria para esa tabla). Utilizará el cuadro de diálogo Crear secuencia para crear la secuencia declarativamente la secuencia que cree será esencialmente la misma que si hubiera ingresado las siguientes declaraciones utilizando la hoja de trabajo de SQL:

CREAR SECUENCIA patron\_id\_seq

COMENZAR CON 100

INCREMENTO POR 1;

Después de crear la secuencia, puede usarla en las instrucciones INSERT para generar valores numéricos únicos. El siguiente ejemplo usa la secuencia patron\_id\_seq para crear una fila para un nuevo usuario (usuario de la biblioteca), asignándole un ID de usuario que es el siguiente valor disponible de la secuencia patron\_id\_seq:

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Smith', 'Jane', '123 Main Street', 'Mytown, MA 01234', null);

Para crear la secuencia, si aún no está conectado, conéctese a la base de datos como usuario del esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Secuencias en la jerarquía de esquemas en el lado izquierdo, seleccione **Nueva secuencia**e ingrese información usando el cuadro de diálogo Crear secuencia.

**Esquema** : especifique su esquema actual como el esquema en el que se creará la secuencia.

**Nombre** : patron\_id\_seq

**Incremento** : 1

**Comience con h** : 100

**Valor mínimo** : 100

Haga **clic** en **Aceptar** para terminar de crear la secuencia.

Para insertar datos de muestra en las tablas, vaya a [Insertar datos en las tablas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAFGJBF) .

### Insertar datos en las tablas

Para su comodidad al usar la vista y el procedimiento PL / SQL que creará, agregue algunos datos de muestra a las tablas LIBROS, PATRONES y TRANSACCIONES. (Si no agrega datos de muestra, aún puede crear los objetos restantes en este tutorial, pero la vista y el procedimiento no devolverán ningún resultado).

Vaya a la ventana de la hoja de trabajo SQL asociada con la conexión de base de datos que ha estado usando. (Para obtener información sobre el uso de la hoja de cálculo SQL, consulte [Uso de la hoja de cálculo SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) ). Copie y pegue las siguientes instrucciones INSERT en el cuadro **Introducir declaración SQL** :

Insertar en los libros VALORES ('A1111', 'Moby Dick', 'Melville', 'Herman', 10);

INSERTE EN LOS VALORES de los libros ('A2222', 'Get Rich Really Fast', 'Scammer', 'Ima', 1);

Insertar en los libros VALORES ('A3333', 'Buscando la paz interior', 'Blissford', 'Serenidad', nula);

Insertar en los libros VALORES ('A4444', 'Grandes historias de misterio', 'Whodunit', 'Rodney', 5);

INSERTE EN LOS libros VALORES ('A5555', 'Software Wizardry', 'Abugov', 'D.', 10);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Smith', 'Jane', '123 Main Street', 'Mytown, MA 01234', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Chen', 'William', '16 S. Maple Road ',' Mytown, MA 01234 ', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Fernandez', 'Maria', '502 Harrison Blvd.,' Sometown, NH 03078 ', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Murphy', 'Sam', '57 Main Street ',' Mytown, MA 01234 ', null);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A1111', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A2222', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A3333', SYSDATE, 3);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A2222', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A3333', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (103, 'A4444', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A4444', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A2222', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A5555', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A2222', SYSDATE, 1);

Haga clic en el icono Ejecutar secuencia de comandos o presione la tecla F5.

Para crear una vista, vaya a [Crear una vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFDCH) .

### Crear una vista

Cree una vista que devuelva información sobre los usuarios y sus transacciones. Esta vista consulta las tablas PATRONES y TRANSACCIONES, y devuelve filas que contienen el ID, el apellido y el nombre de un usuario, junto con una transacción y el tipo de transacción. Las filas se ordenan por ID de usuario y por tipo de transacción dentro de ID de usuario.

Para crear la vista patrons\_trans\_view, si aún no está conectado, conéctese a la base de datos como usuario del esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Vistas en la jerarquía de esquemas en el lado izquierdo, seleccione **Nueva vista**e ingrese la siguiente información. (Si no se menciona una pestaña o campo, no ingrese nada para ella).

**Esquema** : especifique su esquema actual como el esquema en el que se creará la vista.

**Nombre** : patrons\_trans\_view

**Pestaña de consulta SQL**

En el cuadro de Consulta SQL, ingrese (o copie y pegue) la siguiente declaración:

SELECCIONE p.patron\_id,

p.last\_name,

p.first\_name,

t.transaction\_type,

t.transaction\_date

De los clientes p, transacciones t

DONDE p.patron\_id = t.patron\_id

ORDEN POR p.patron\_id, t.transaction\_type

Luego haga clic en **Probar sintaxis** y asegúrese de que no haya cometido ningún error de sintaxis. Si cometió algún error, corríjalo y vuelva a hacer clic en Sintaxis de prueba.

**DDL**

Revise la declaración SQL que el desarrollador de SQL usará para crear la vista. Si desea realizar algún cambio, vuelva a la pestaña Consulta de SQL y realice los cambios allí.

Si desea guardar la sentencia CREATE VIEW en un archivo de script SQL, haga clic en **Guardar** y especifique la ubicación y el nombre del archivo.

Cuando haya terminado, haga **clic** en **Aceptar** .

Has terminado de crear la vista. Si insertó datos en las tablas subyacentes, como se describe en [Insertar datos en las tablas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAFGJBF) , puede ver los datos devueltos por esta vista de la siguiente manera: en el navegador de Conexiones, expanda Vistas y seleccione PATRONS\_TRANS\_VIEW, luego haga clic en la pestaña Datos.

Para crear un procedimiento que enumere todos los libros con una calificación específica, vaya a [Crear un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACEAIJ) .

### Crear un procedimiento PL / SQL

Cree un procedimiento que enumere todos los libros con una calificación específica. Luego, puede llamar a este procedimiento con un parámetro de entrada (un número del 1 al 10), y la salida será todos los títulos de todos los libros con esa calificación.

Para crear el procedimiento, si aún no está conectado, conéctese a la base de datos como usuario del esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Procedimientos en la jerarquía de esquema en el lado izquierdo, seleccione **Nuevo procedimiento** e ingrese la siguiente información usando el cuadro de diálogo Crear procedimiento PL / SQL.

**Nombre del objeto** : list\_a\_rating

Haga clic en **Aceptar** . Se abre una ventana de origen para el nuevo procedimiento. Ingrese (o copie y pegue) el siguiente texto de procedimiento, reemplazando cualquier texto existente:

CREAR O REEMPLAZAR

PROCEDIMIENTO list\_a\_rating (in\_rating IN NUMBER) AS

match\_title VARCHAR2 (50);

TYPE my\_cursor IS REF CURSOR;

the\_cursor my\_cursor;

EMPEZAR

ABIERTO the\_cursor

POR 'SELECCIONAR TITULO

De los libros

DONDE calificación =: in\_rating '

UTILIZANDO in\_rating;

DBMS\_OUTPUT.PUT\_LINE ('Todos los libros con una calificación de' || in\_rating || ':');

LAZO

FETCH the\_cursor INTO matching\_title;

SALIR CUANDO the\_cursor% NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (matching\_title);

Bucle final;

CIERRE the\_cursor;

END list\_a\_rating;

Este procedimiento utiliza un cursor (denominado the\_cursor) para devolver solo las filas donde el libro tiene la calificación especificada (parámetro in\_rating), y usa un bucle para generar el título de cada libro con esa calificación.

Haga clic en el icono Guardar para guardar el procedimiento.

Como ejemplo de uso, después de crear el procedimiento llamado LIST\_A\_RATING, si ha insertado datos en la tabla BOOKS (por ejemplo, utilizando las instrucciones INSERT en [Insertar datos en las tablas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAFGJBF) ), puede usar la siguiente declaración para devolver todos los libros con una calificación de 10:

CALL list\_a\_rating (10);

Para ejecutar este procedimiento dentro de SQL Developer, haga clic con el botón derecho en LIST\_A\_RATING en la pantalla de la jerarquía del navegador de Connections y seleccione **Ejecutar** . En **Bloque** PL / SQL en el cuadro de diálogo Ejecutar PL / SQL, cambie IN\_RATING => IN\_RATING a IN\_RATING => **10** , y haga **clic** en **Aceptar** . La visualización de la ventana de registro ahora incluirá el siguiente resultado:

Todos los libros con una calificación de 10:

Moby Dick

Software Wizardry

### Depurar un procedimiento PL / SQL

Si desea practicar la depuración de un procedimiento PL / SQL con SQL Developer, cree un procedimiento similar al procedimiento list\_a\_rating que creó en [Crear un procedimiento PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACEAIJ) , pero con un error lógico. (La codificación también es deliberadamente ineficiente, para permitir la visualización de la calificación en una variable).

Antes de que pueda depurar el procedimiento, debe asegurarse de que el usuario asociado con la conexión de la base de datos tenga los privilegios DEBUG CONNECT SESSION y DEBUG ANY PROCEDURE.

Para crear este procedimiento, si aún no está conectado, conéctese a la base de datos como usuario del esquema que está utilizando para este tutorial. Haga clic con el botón derecho en el nodo Procedimientos en la jerarquía de esquema en el lado izquierdo, seleccione **Nuevo procedimiento** e ingrese la siguiente información usando el cuadro de diálogo Crear procedimiento PL / SQL.

**Nombre del objeto** : list\_a\_rating2

Haga clic en **Aceptar** . Se abre una ventana de origen para el nuevo procedimiento. Ingrese (o copie y pegue) el siguiente texto de procedimiento, reemplazando cualquier texto existente:

CREAR O REEMPLAZAR

PROCEDIMIENTO list\_a\_rating2 (in\_rating IN NUMBER) AS

match\_title VARCHAR2 (50);

NUMERO;

TYPE my\_cursor IS REF CURSOR;

the\_cursor my\_cursor;

rating\_cursor my\_cursor;

EMPEZAR

ABIERTO the\_cursor

POR 'SELECCIONAR TITULO

De los libros

DONDE calificación <=: en\_rating '

UTILIZANDO in\_rating;

OPEN rating\_cursor PARA 'SELECCIONE la calificación DE los libros DONDE

rating <=: in\_rating 'USANDO in\_rating;

DBMS\_OUTPUT.PUT\_LINE ('Todos los libros con una calificación de' || in\_rating || ':');

LAZO

FETCH the\_cursor INTO matching\_title;

FETCH rating\_cursor INTO matching\_rating;

SALIR CUANDO the\_cursor% NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (matching\_title);

Bucle final;

CIERRE the\_cursor;

CERRAR rating\_cursor;

END list\_a\_rating2;

Este procedimiento contiene un error lógico en la definición de the\_cursor: selecciona títulos donde la calificación es menor o igual que una calificación específica, mientras que debería seleccionar títulos solo donde la calificación es igual a la calificación especificada.

Haga clic en el icono Guardar para guardar el procedimiento.

Suponga que desea ejecutar este procedimiento y enumere todos los libros con una calificación de 10. Haga clic con el botón derecho en LIST\_A\_RATING2 en la pantalla de la jerarquía del navegador de Conexiones y seleccione **Ejecutar** . En **Bloque** PL / SQL en el cuadro de diálogo Ejecutar PL / SQL, cambie, cambie IN\_RATING => IN\_RATING a IN\_RATING => **10** , y haga **clic** en **Aceptar** . Sin embargo, en la ventana de Registro, se ven resultados inesperados: se enumeran muchos títulos, incluidos algunos con calificaciones distintas a 10. Por lo tanto, decide depurar el procedimiento.

Para depurar el procedimiento, siga estos pasos:

1. Haga clic en el icono Compilar para depurar en la barra de herramientas debajo de la pestaña LIST\_A\_RATING2.
2. Establezca dos puntos de interrupción haciendo clic en el margen izquierdo (a la izquierda de la línea vertical delgada) al lado de cada una de estas dos líneas:
3. FETCH the\_cursor INTO matching\_title;
4. FETCH rating\_cursor INTO matching\_rating;

Al hacer clic en el margen izquierdo se alterna la configuración y la eliminación de los puntos de interrupción. Al hacer clic al lado de estas dos líneas, podrá ver los valores de las variables matching\_title y matching\_rating a medida que avanza la ejecución en el modo de depuración.

1. En el menú **Depurar** , seleccione **Depurar LIST\_A\_RATING2** . Asegúrese de que la línea IN\_RATING => IN\_RATING se haya cambiado a IN\_RATING => **10** , y haga **clic** en **Aceptar** .
2. Haga clic en **Ver** , luego en **Depurador** , luego en **Datos** para mostrar el panel Datos. (Sugerencia: expanda el ancho de la columna Nombre para que pueda ver MATCHING\_RATING).
3. Presione la tecla **F9** (o haga clic en **Depurar** , luego **Reanudar** ) para que la ejecución continúe, deteniéndose en el siguiente punto de interrupción.
4. Presione repetidamente la tecla **F9** (o haga clic en **Depurar** , luego en **Reanudar** ), observando especialmente el valor de MATCHING\_RATING a medida que se procesa cada fila. Notará el primer resultado incorrecto cuando vea que se incluye el título *Get Rich Really Fast* , a pesar de que su calificación es solo de 1 (obviamente menos de 10). (Consulte la ilustración de la pantalla con información de depuración en [Ejecución y depuración de funciones y procedimientos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIGHA) ).
5. Cuando tenga suficiente información para solucionar el problema, puede hacer clic en la pestaña Depuración - Registro y en el icono Terminar en la barra de herramientas de depuración.

Desde esta sesión de depuración, usted sabe que para corregir el error lógico, debe cambiar rating <= :in\_ratinga rating = :in\_ratingla definición de the\_cursor.

### Usa la hoja de trabajo SQL para consultas

Puede usar la hoja de trabajo de SQL para probar sentencias de SQL usando una conexión de base de datos. Para mostrar la hoja de trabajo, en el menú **Herramientas** , seleccione **Hoja de trabajo SQL** . En el cuadro de diálogo Seleccionar conexión, seleccione la conexión de base de datos que usó para crear las tablas LIBROS, PATRONES y TRANSACCIONES para el tutorial en el [Capítulo 3, "Tutorial: Crear objetos para una pequeña base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHDFAF) .

La hoja de trabajo SQL tiene la interfaz de usuario que se muestra en [Uso de la hoja de trabajo SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDHAAI) .

En el cuadro **Ingresar instrucción SQL** , ingrese la siguiente declaración (el punto y coma es opcional para la hoja de cálculo SQL):

SELECCIONAR author\_last\_name, título DE los libros;

Observe el resaltado automático de las palabras clave de SQL (SELECCIONE y DESDE en este ejemplo).

Haga clic en el icono Ejecutar instrucción SQL en la barra de herramientas de la hoja de trabajo SQL. Los resultados de la consulta se muestran en la pestaña **Resultados** debajo del área en la que ingresó la declaración SQL.

En el cuadro **Ingresar declaración SQL** , ingrese (o copie y pegue) la siguiente declaración, que es la misma que la instrucción SELECT en la vista que creó en [Crear una vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFDCH) :

SELECCIONE p.patron\_id,

p.last\_name,

p.first\_name,

t.transaction\_type,

t.transaction\_date

De los clientes p, transacciones t

DONDE p.patron\_id = t.patron\_id

ORDEN POR p.patron\_id, t.transaction\_type;

Haga clic en el icono Ejecutar declaración SQL en la barra de herramientas de la hoja de trabajo SQL y vea los resultados de la consulta.

Haga clic en el icono Ejecutar plan de explicación en la barra de herramientas de la hoja de trabajo de SQL para ver el plan de ejecución (que se muestra en la pestaña Explicación) que sigue la base de datos de Oracle para ejecutar la instrucción SQL. La información incluye la estrategia del optimizador y el costo de ejecutar la declaración. (Para obtener información sobre cómo generar e interpretar los planes de ejecución, consulte la *Guía de ajuste de rendimiento de la base de datos de Oracle* ).

### Script para crear y usar los objetos del tutorial de la biblioteca

Las siguientes declaraciones crean y utilizan los objetos de la base de datos que ha creado (o creará) para el tutorial en el [Capítulo 3, "Tutorial: Crear objetos para una pequeña base de datos"](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHDFAF) . Puede ver estos comandos para ayudarlo a comprender los objetos de la base de datos de la biblioteca que se tratan en el tutorial.

- Limpiar de cualquier acción tutorial anterior.

DROP TABLE transacciones;

DROP TABLE libros;

DROP TABLE patrones;

SECUENCIA DE LA CAÍDA patron\_id\_seq;

DROP SEQUENCE transaction\_seq;

DROP TRIGGER transaction\_trg;

DROP VIEW patrons\_trans\_view;

DROP PROCEDURE list\_a\_rating;

PROCEDIMIENTO DE CAÍDA list\_a\_rating2;

establecer serveroutput en

- Crear objetos.

Libros de CREATE TABLE (

book\_id VARCHAR2 (20),

título VARCHAR2 (50)

CONSTRAINT title\_not\_null NOT NULL,

author\_last\_name VARCHAR2 (30)

CONSTRAINT last\_name\_not\_null NOT NULL,

author\_first\_name VARCHAR2 (30),

calificación NÚMERO,

CONSTRAINT books\_pk CLAVE PRIMARIA (book\_id),

CONSTRAINT rating\_1\_to\_10 CHECK (la calificación ES NULA O

(calificación> = 1 y calificación <= 10)),

CONSTRAINT author\_title\_unique UNIQUE (author\_last\_name, title));

CREAR TABLA DE PATRONES (

patron\_id NUMBER,

last\_name VARCHAR2 (30)

CONSTRAINT patron\_last\_not\_null NOT NULL,

nombre primero VARCHAR2 (30),

street\_address VARCHAR2 (50),

city\_state\_zip VARCHAR2 (50),

ubicación MDSYS.SDO\_GEOMETRY,

CONSTRAINT patrons\_pk PRIMARY KEY (patron\_id));

CREAR tablas de transacciones (

transaction\_id NUMBER,

patron\_id CONSTRAINT for\_key\_patron\_id

REFERENCIAS de los clientes (patron\_id),

book\_id CONSTRAINT for\_key\_book\_id

REFERENCIAS libros (book\_id),

fecha de transacción fecha

CONSTRAINT tran\_date\_not\_null NOT NULL,

tipo de transacción NÚMERO

CONSTRAINT tran\_type\_not\_null NOT NULL,

CONSTRAINT transaction\_pk PRIMARY KEY (transaction\_id));

CREAR SECUENCIA patron\_id\_seq

COMENZAR CON 100

INCREMENTO POR 1;

- La secuencia para la transaction\_id

- En el tutorial se crea automáticamente,

- y puede tener el nombre de TRANSACTIONS\_SEQ.

CREAR SECUENCIA transacciones\_seq

COMENZAR CON 1

INCREMENTO POR 1;

- El activador de inserción anterior para valores de ID de transacción

- En el tutorial se crea automáticamente,

- y puede tener el nombre TRANSACTIONS\_TRG.

CREAR O REEMPLAZAR TRIGGER transaction\_trg

ANTES DE INSERTAR EN LAS TRANSACCIONES

POR CADA FILA

EMPEZAR

SELECCIONE TRANSACCIONES\_SEQ.NEXTVAL EN: NEW.TRANSACTION\_ID DESDE DUAL;

FIN;

/

CREAR VISTA patrons\_trans\_view AS

SELECCIONE p.patron\_id,

p.last\_name,

p.first\_name,

t.transaction\_type,

t.transaction\_date

De los clientes p, transacciones t

DONDE p.patron\_id = t.patron\_id

ORDEN POR p.patron\_id, t.transaction\_type;

- Procedimiento: Listar todos los libros que tienen una calificación específica.

CREAR O REEMPLAZAR EL PROCEDIMIENTO list\_a\_rating (in\_rating IN NUMBER) COMO

match\_title VARCHAR2 (50);

TYPE my\_cursor IS REF CURSOR;

the\_cursor my\_cursor;

EMPEZAR

ABIERTO the\_cursor

POR 'SELECCIONAR TITULO

De los libros

DONDE calificación =: in\_rating '

UTILIZANDO in\_rating;

DBMS\_OUTPUT.PUT\_LINE ('Todos los libros con una calificación de' || in\_rating || ':');

LAZO

FETCH the\_cursor INTO matching\_title;

SALIR CUANDO the\_cursor% NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE (matching\_title);

Bucle final;

CIERRE the\_cursor;

FIN;

/

mostrar errores;

- Insertar y consultar datos.

Insertar en los libros VALORES ('A1111', 'Moby Dick', 'Melville', 'Herman', 10);

INSERTE EN LOS VALORES de los libros ('A2222', 'Get Rich Really Fast', 'Scammer', 'Ima', 1);

Insertar en los libros VALORES ('A3333', 'Buscando la paz interior', 'Blissford', 'Serenidad', nula);

Insertar en los libros VALORES ('A4444', 'Grandes historias de misterio', 'Whodunit', 'Rodney', 5);

INSERTE EN LOS libros VALORES ('A5555', 'Software Wizardry', 'Abugov', 'D.', 10);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Smith', 'Jane', '123 Main Street', 'Mytown, MA 01234', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Chen', 'William', '16 S. Maple Road ',' Mytown, MA 01234 ', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Fernandez', 'Maria', '502 Harrison Blvd.,' Sometown, NH 03078 ', null);

INSERTAR EN LOS VALORES de los usuarios (patron\_id\_seq.nextval,

'Murphy', 'Sam', '57 Main Street ',' Mytown, MA 01234 ', null);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A1111', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A2222', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A3333', SYSDATE, 3);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A2222', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A3333', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (103, 'A4444', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (100, 'A4444', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A2222', SYSDATE, 2);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (102, 'A5555', SYSDATE, 1);

INSERT INTO transacciones (patron\_id, book\_id,

transaction\_date, transaction\_type)

VALORES (101, 'A2222', SYSDATE, 1);

- Probar la vista y el procedimiento.

SELECT \* FROM patrons\_trans\_view;

CALL list\_a\_rating (10);

## 4 Cuadros de diálogo para crear / editar objetos

SQL Developer usa cuadros de diálogo para crear y editar conexiones de base de datos y objetos en la base de datos (tablas, vistas, procedimientos, etc.). Los cuadros de diálogo a veces tienen varias pestañas, cada una de las cuales refleja una agrupación lógica de propiedades para ese tipo de objeto.

Para obtener una explicación de cualquier cuadro de diálogo o pestaña, haga clic en el botón **Ayuda** o presione la tecla **F1** .

Los cuadros de diálogo no se presentan aquí en ningún orden riguroso, porque la ayuda para cada cuadro es una información independiente y se ve normalmente cuando hace clic en Ayuda o presiona F1 en ese cuadro.

**Nota:**

Para todos los campos de Nombre, cualquier nombre que escriba se convierte automáticamente y se almacena en los metadatos de la base de datos en mayúsculas, a menos que incluya el nombre entre comillas (""). (Los nombres de los objetos de la base de datos en las sentencias de SQL y PL / SQL no distinguen entre mayúsculas y minúsculas).

A incluya caracteres en minúscula, caracteres especiales o espacios en los nombres de los objetos, escríbalo entre comillas ("") cuando lo escriba. Ejemplo:"My table"

### Añadir extensión

Este cuadro de diálogo se muestra al hacer clic en Agregar en el panel [Tipos de archivo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFCAIH) de [Preferencias de desarrollador](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) de [SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) .

**Extensión** : especifique la extensión del archivo, incluido el período (por ejemplo, .xyz).

Después de hacer clic en Aceptar, puede seleccionar esa extensión y modificar sus detalles, incluido el tipo de archivo, el tipo de contenido y si los archivos tienen la extensión abierta automáticamente por el Desarrollador de SQL.

### Rama / Etiqueta

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en un directorio remoto en el repositorio de Subversion y selecciona Rama / Etiqueta. Cree una rama copiando la copia de trabajo actual o una revisión del repositorio a una ubicación seleccionada en el repositorio.

**Desde** : Ubicación de la copia de trabajo o revisión.

**Copia de trabajo** : hace que se **copie** la copia de trabajo actual.

**Revisión HEAD** : hace que se copie la revisión HEAD (la última revisión en el repositorio).

**Utilizar revisión** : hace que se copie la revisión especificada en el cuadro de texto. Para ver una lista de revisiones entre las que puede elegir, haga clic en **Revisar lista** .

**A** : Ubicación de destino.

**Comentario** : **Comentario** descriptivo opcional.

**Cambiar a nueva rama / etiqueta** : si esta opción está marcada, la copia de trabajo existente se cambia a la nueva rama.

Después de hacer clic en Aceptar, la Consola SVN - Panel de registro se muestra en la parte inferior, con mensajes sobre los comandos que se ejecutaron.

### Buscar actualizaciones

Cuando hace clic en **Ayuda** y luego en **Buscar actualizaciones** , puede buscar y descargar actualizaciones disponibles de SQL Developer. Las siguientes páginas pueden ser mostradas. (Si ha habilitado la preferencia de desarrollador de SQL para buscar actualizaciones automáticamente en el inicio, y si hace clic para ver las actualizaciones disponibles en el inicio, se muestra la página **Actualizaciones** ).

Si no puede buscar actualizaciones porque su sistema está detrás de un firewall, es posible que deba configurar las preferencias de usuario de SQL Developer para [el navegador web y el proxy](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDBFJA) .

1. **Fuente** : seleccione la fuente o las fuentes que se verificarán para ver las actualizaciones disponibles: cualquiera o todos los centros de actualización en línea especificados, o un archivo ZIP local que contenga un paquete de actualización. También puede hacer clic en **Agregar** para agregar un centro de actualización definido por el usuario.
2. **Actualizaciones** : si hay actualizaciones disponibles de la fuente o las fuentes seleccionadas, seleccione las que desea descargar. Las actualizaciones disponibles incluyen ciertos controladores JDBC de terceros, que requieren que acepte los términos de sus licencias.

La opción **Mostrar solo actualizaciones** restringe la visualización a las actualizaciones de los componentes de SQL Developer actualmente instalados. Para habilitar la visualización de todos los componentes nuevos y actualizados, ya estén instalados o no, desmarque esta opción.

Después de hacer clic en Siguiente, es posible que se le solicite ingresar el nombre de usuario y la contraseña de su cuenta web de Oracle. Si no tiene una cuenta, puede hacer clic en el enlace Registrarse.

1. **Contratos de licencia** (que se muestran solo si seleccionó actualizaciones que requieren un acuerdo de licencia): para cada actualización que requiera que acepte los términos de una licencia, revise el texto de la licencia y haga clic en **Acepto** . Debe hacer esto para cada licencia aplicable.
2. **Descargar** : Si seleccionó alguna actualización para descargar, esta página muestra el progreso de la operación de descarga.
3. **Resumen** : muestra información sobre las actualizaciones que se descargaron. Después de hacer clic en Finalizar, se le preguntará si desea instalar las actualizaciones ahora y reiniciar SQL Developer.

### Salida de CVS

Use este cuadro de diálogo para retirar los módulos de un repositorio de CVS.

**Nombre de conexión** : Nombre de la conexión con el repositorio

**Nombre del módulo** : Nombre del módulo a verificar.

**Ruta** : Ruta al módulo.

**Obtener / Actualizar** lista de módulos: muestra la lista de módulos o actualiza la pantalla actual.

**Carpeta de destino** : carpeta en la que colocar los archivos desprotegidos.

**Usar revisión o etiqueta** : si esta opción está marcada, se utiliza la revisión o etiqueta que especifique en el cuadro de texto. Para ver las etiquetas disponibles, haga clic en el ícono de binoculares.

**Eliminar carpetas vacías** : si esta opción está marcada, las carpetas vacías se eliminan del directorio de trabajo.

### Salida de Subversion

Use este cuadro de diálogo para retirar los módulos de un repositorio de Subversion.

**Conexión del repositorio** : Nombre de la conexión al repositorio.

**Ruta** : Ruta al módulo que se va a comprobar.

**Destino** : Directorio o carpeta en la que colocar los archivos desprotegidos.

**Utilizar revisión** : si esta opción está marcada, se utiliza la revisión que especifique en el cuadro de texto. Para ver las revisiones disponibles, haga clic en el ícono de binoculares.

**Recursivo** : si esta opción está marcada, la jerarquía de carpetas se busca de forma recursiva desde arriba hacia abajo. Si esta opción no está marcada, solo se busca en la carpeta superior.

### Elegir directorio

Este es un cuadro estándar para elegir un directorio en el que colocar los archivos: use **Ubicación** para navegar (haciendo doble clic) la carpeta en la que desea guardar los archivos, o ingrese un nombre de directorio. Si el directorio no existe, se crea.

### Confirmar solicitud de caída

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en una aplicación de Application Express y selecciona Descartar. Para descartar la aplicación, haga clic en **Sí** ; mantener (no dejar caer) la aplicación, haga clic en **No** .

Si la aplicación contiene una secuencia de comandos de desinstalación, esa secuencia de comandos se ejecuta antes de que la aplicación se descarte.

### Confirmar ejecutar SQL

Este cuadro de diálogo se muestra en ciertas situaciones cuando SQL Developer necesita ejecutar un script de configuración en el servidor. El script se muestra en un cuadro de texto, donde puede ver o editar los contenidos. Para permitir que el script se ejecute, haga clic en **Sí** ; a prevent.the guión se ejecute, haga clic en **No** .

### La conexión tiene cambios no confirmados

Este cuadro de diálogo se muestra si intenta finalizar la sesión de la base de datos activa mientras haya transacciones por confirmar. Seleccione la opción apropiada y haga **clic** en **Aceptar** .

Para confirmar los cambios y finalizar la sesión, seleccione **Confirmar cambios** . Para revertir los cambios y finalizar la sesión, seleccione **Revertir cambios** . Para cancelar el intento de finalizar la sesión, seleccione **Abort Connection Disconnect** . (Seleccionar Abortar la desconexión de la conexión y hacer clic en Aceptar tiene el mismo efecto que hacer clic en Cancelar).

### Crear / Editar Nuevo Objeto (Nueva Galería)

Especifique el tipo de objeto a crear. Después de hacer clic en Aceptar, se muestra el cuadro de diálogo para crear ese tipo de objeto.

**Buscar** : le permite restringir los tipos de objetos según una cadena que especifique. Por ejemplo, para mostrar solo el elemento para crear una tabla, especifique table.

**Categorías** : Una visualización jerárquica de tipos de objetos. Para ver todos los tipos de objetos que puede crear, seleccione Todos los elementos.

**Elementos** : tipos de objetos que puede crear dentro de la categoría seleccionada y según lo limitado por cualquier cadena de búsqueda que haya especificado.

**Mostrar todas las descripciones** : si esta opción está marcada, se muestran descripciones completas de todos los elementos mostrados.

### Crear / Editar conexión CVS

Esta información se aplica a la creación o edición de una conexión CVS (Sistema de versiones concurrentes). Para obtener información sobre la compatibilidad de SQL Developer para la creación de versiones y CVS, consulte [Uso de la](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) versión .

**Conexión**

**Método de acceso** : el método por el cual el cliente obtendrá acceso y se autentificará en el servidor. Los métodos disponibles dependen de las preferencias de CVS que haya establecido; los métodos disponibles pueden incluir External, Password Server, Secure Shell a través de SSH2 y [Other].

La mayoría de los campos de conexión restantes se aplican solo a valores específicos del método de acceso.

**Nombre de usuario** : un nombre de usuario de CVS conocido por el repositorio.

**Nombre de host** : **Nombre de host** calificado o dirección IP del sistema del servidor CVS.

**Puerto** : número de puerto TCP / IP en el que está escuchando el repositorio.

**Ruta del repositorio** : la ubicación del software de repositorio CVS. Se puede sobrescribir el valor seeded / con una ruta en el formato adecuado para su sistema operativo, por ejemplo, c: \ cvs. Un error de formato simple, como una barra inclinada en lugar de una barra diagonal inversa, dará como resultado un mensaje que le pedirá que ingrese una ruta de depósito válida.

**SSH2 Key** : ruta y nombre de archivo para el archivo de clave privada SSH2 para esta conexión. Puede generar un archivo de clave privada SSH2 utilizando Generar par de claves SSH2.

**Generar par de claves SSH2** : muestra un cuadro de diálogo para generar un par de claves SSH2 (es decir, un archivo de clave privada y un archivo de clave pública). Usted especifica el archivo de clave privada en el cuadro de clave SSH2. Usted agrega los detalles del archivo de clave pública a la lista de claves públicas en el sistema del servidor CVS

**Use HTTP Proxy Settings** : marque (habilite) esta opción si está detrás de un firewall y necesita usar HTTP para acceder al servidor CVS.

**Configuración del localizador externo** : muestra el cuadro de diálogo [Configuración del localizador externo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEIAAF) , en el que puede editar los detalles del cliente de shell remoto y el programa del servidor remoto.

**Raíz**

**Valor de CVSROOT** : la variable raíz de CVS está formada por la información que ya ha proporcionado. Esta variable proporciona al cliente detalles de acceso cuando se comunica con el servidor. El formato de la variable sembrada es:: accessmethod: username @ serverlocation: repositorypath

Normalmente no necesitarías cambiar este valor. Una instancia en la que cambiaría este valor es cuando intenta conectarse a un servidor CVSNT a través de un firewall. En este caso, debe agregar información de proxy al comienzo de la parte del nombre de usuario, de modo que la variable raíz CVS tome la siguiente forma:: accessmethod: proxy = proxyname; proxyport = portnumber: username @ serverlocation: repositorypath

**Prueba**

**Probar conexión** : Intenta establecer una conexión con el repositorio de CVS.

**Estado** : muestra el resultado de la prueba (éxito o un mensaje de error).

**Nombre**

**Nombre de la conexión** : Nombre para identificar la conexión al repositorio de CVS. El nombre predeterminado es el mismo que el valor CVSROOT.

**Resumen**

Muestra la información de conexión que ha especificado. Para hacer cualquier corrección, haga clic en **Atrás** según sea necesario y modifique la información. Para crear la conexión, haga clic en **Finalizar** .

### Crear / Editar / Seleccionar conexión de base de datos

El cuadro de diálogo de conexión de la base de datos muestra las conexiones existentes. Según el contexto, puede seleccionar una conexión para conectarse a la base de datos, editar la información sobre las conexiones existentes o especificar la información mientras crea una nueva conexión. (Consulte [Creación y edición de conexiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFJBBD) .)

**Nombre de conexión** : un alias para una conexión a la base de datos utilizando la información que ingresa. (El nombre de la conexión no se almacena en la base de datos y la conexión no es un objeto de la base de datos). Sugerencia: incluya el nombre de la base de datos (SID) y el nombre de usuario en el nombre de la conexión. Ejemplo: personnel\_herman para conectarse a la base de datos de personal como usuario Herman.

**Nombre de usuario** : nombre del usuario de la base de datos para la conexión. Este usuario debe tener privilegios suficientes para realizar las tareas que desea realizar mientras está conectado a la base de datos, como crear, editar y eliminar tablas, vistas y otros objetos.

**Contraseña** : contraseña asociada con el usuario de la base de datos especificada.

**Guardar contraseña** : si esta opción está marcada, la contraseña se guarda con la información de conexión, y no se le solicitará la contraseña en los intentos posteriores de conexión utilizando esta conexión.

**Pestaña de oráculo**

La siguiente información se aplica a una conexión a una base de datos Oracle.

**Rol** : El conjunto de privilegios que se asociarán con la conexión. Para un usuario al que se le ha otorgado el privilegio del sistema SYSDBA, puede especificar una conexión que incluya el privilegio.

**Tipo de conexión** : seleccione Básico, TNS, LDAP (Protocolo ligero de acceso a directorios) o Avanzado. (La visualización de los campos cambia para reflejar cualquier cambio en el tipo de conexión).

**Autenticación del SO** : si esta opción está marcada, el control de la autenticación del usuario se pasa al sistema operativo (OS). Esto permite al usuario especificado conectarse a la base de datos al autenticar el nombre de usuario del sistema operativo de ese usuario en la base de datos. No se asocia ninguna contraseña a la conexión, ya que se supone que la autenticación del sistema operativo es suficiente. Para obtener información sobre el uso de la autenticación del sistema operativo, consulte *la Guía y referencia para desarrolladores de JDBC de la base de datos Oracle* .

**Autenticación Kerberos** : si esta opción está marcada, las credenciales se pueden compartir en muchas aplicaciones habilitadas para Kerberos (por ejemplo, para tener el mismo nombre de usuario y contraseña tanto para el sistema operativo como para la base de datos Oracle). La configuración del controlador grueso se realiza a través de sqlnet.ora ( sqlnet.authentication\_services=(KERBEROS)y los parámetros relacionados), por lo que no se necesitan nombre de usuario ni contraseña. La configuración del controlador ligero utiliza el archivo de configuración (.conf) y el caché de credenciales, y utiliza un principal de servicio y una contraseña. Para obtener más información sobre las opciones de autenticación de Kerberos, consulte [Base de datos: Parámetros avanzados](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFGCAC) . Para obtener información sobre la configuración de la autenticación Kerberos, consulte [*la Guía del administrador de seguridad avanzada de la base de datos Oracle*](https://docs.oracle.com/cd/E11882_01/network.112/e40393/asokerb.htm#ASOAG060) .

**Conexión de proxy** : si se marca esta opción, se utilizará la autenticación de proxy, como se explica en [Conexiones con autenticación de proxy](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDECIII) . Muestra el cuadro de diálogo [Autenticación de Proxy Oracle](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFCJGC) .

**Tipo de conexión básica**

**Nombre de host** : sistema de host para la base de datos Oracle.

**Puerto** : puerto de escucha.

**SID** : nombre de la base de datos.

**Nombre del servicio** : nombre del servicio de red de la base de datos (para una conexión de base de datos remota a través de una conexión segura).

**Tipo de conexión TNS**

**Alias ​​de red** : **alias de** Oracle Net para la base de datos. (La lista para seleccionar un alias de red se llena inicialmente desde el archivo tnsnames.ora en su sistema, si ese archivo existe).

**Identificador de conexión** : **Identificador de** conexión de red de Oracle.

**Tipo de conexión LDAP**

Los usuarios empresariales se autentican con el servidor del Protocolo ligero de acceso a directorios (LDAP). La información de inicio de sesión del usuario debe configurarse en el servidor LDAP y asignarse a un esquema en la base de datos. La compatibilidad con servidores de directorio compatibles con LDAP proporciona un vehículo centralizado para administrar y configurar una red distribuida de Oracle. El servidor de directorio puede reemplazar los archivos tnsnames.ora localizados del lado del cliente y del lado del servidor.

**Servidor LDAP** : seleccione de la lista (de <DIRECTORY\_SERVER> entradas en el archivo ldap.ora, o ingrese un nuevo servidor de directorio.

Contexto: contexto administrativo LDAP. Se enumeran los contextos disponibles en el servidor seleccionado.

**Servicio DB** : Información de conexión a la base de datos. Si una conexión utiliza el controlador OCI / Thick (consulte la preferencia **Usar OCI / Thick** en [Base de datos: Parámetros avanzados](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFGCAC) ), el sistema en el que se está ejecutando SQL Developer debe tener una instalación del Cliente Oracle que contenga las bibliotecas JDBC y orai18n. ruta, y es la versión 10.2 o posterior. Para cargar la información, haga clic en **Cargar** .

**Tipo de conexión avanzada**

**URL de JDBC personalizada** : URL para conectarse directamente desde Java a la base de datos; anula cualquier otra especificación de tipo de conexión. Si está utilizando TNS o un servicio de nombres con el controlador OCI, debe especificar esta información: Ejemplo:

jdbc: oracle: thin: scott / @ localhost: 1521: orcl

Tenga en cuenta que en este ejemplo, se requiere "/" y se le solicitará al usuario que ingrese la contraseña.

Para usar una URL de JDBC personalizada, el sistema en el que se está ejecutando SQL Developer debe tener una instalación del Cliente de Oracle que contenga las bibliotecas JDBC y orai18n, esté presente en la ruta y sea la versión 10.2 o posterior.

**Pestaña TimesTen**

La siguiente información se aplica a una conexión a una base de datos en memoria de Oracle TimesTen.

Para **Nombre de usuario** y **Contraseña** , especifique el nombre de usuario y la contraseña de la cuenta de usuario en la base de datos de TimesTen.

**DSN** : Nombre de la fuente de datos. Seleccione un DSN existente (si se muestra alguno), o *Especificado por el usuario* para crear un nuevo DSN. Un DSN es una cadena de caracteres que identifica una base de datos TimesTen e incluye atributos de conexión que se utilizarán al conectarse a la base de datos. Un DSN tiene las siguientes características: su longitud máxima es de 32 caracteres; no puede contener espacios; y consta de caracteres ASCII, excepto por lo siguiente: [] {},;? \* =! @ \

**Tipo de conexión** (si DNS es especificado por el usuario): *C / S* para el modo cliente-servidor o *Directo* para el modo directo

**Cadena de conexión** : atributos de conexión que incluyen atributos de la base de datos, primeros atributos de conexión, atributos generales de conexión, atributos NLS y atributos de Cache Connect. (Consulte la *Guía de referencia de la API de la base de datos en memoria de Oracle TimesTen* para obtener información sobre los atributos).

**Contraseña de Oracle (para caché)** : la contraseña de la cuenta de usuario de TimesTen en la base de datos Oracle. (Para obtener más información, consulte la *Guía de conexión de caché de la base de datos de nivel de aplicación Oracle TimesTen* ).

Para obtener más información sobre el soporte de SQL Developer para TimesTen, consulte [Soporte de base de datos en memoria de Oracle TimesTen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJFIIH) .

Para obtener información detallada sobre el uso y la referencia sobre Oracle TimesTen, consulte la documentación en línea que se incluye en la instalación de TimesTen. Para información adicional, vaya a:<http://www.oracle.com/technology/products/timesten/>

**Pestaña de acceso**

Para una conexión a una base de datos de Microsoft Access, haga clic en Examinar y busque el archivo de la base de datos (.mdb). Sin embargo, para poder usar la conexión, primero debe asegurarse de que las tablas del sistema en el archivo de la base de datos sean legibles por el Desarrollador de SQL, de la siguiente manera:

1. Abra el archivo de base de datos (.mdb) en Microsoft Access.
2. Haga clic en **Herramientas** , luego en **Opciones** , y en la pestaña **Ver** asegúrese de que se muestran los **objetos del sistema** .
3. Haga clic en **Herramientas** , luego en **Seguridad** y, si es necesario, modifique los permisos de usuario y grupo para las tablas MSysObjects, MsysQueries y MSysRelationships de la siguiente manera: seleccione la tabla y otorgue al usuario Admin el permiso Leer diseño y Leer datos en la tabla.
4. Guarde los cambios y cierre el archivo de base de datos de Access.
5. Crea y prueba la conexión en SQL Developer.

**Pestaña MySQL**

La siguiente información se aplica a una conexión a una base de datos MySQL.

Tenga en cuenta que para conectarse a una base de datos MySQL, primero debe descargar el controlador de conexión MySQL apropiado y luego hacer clic en **Herramientas** , luego en **Preferencias** , y usar el panel de preferencias de usuario del Desarrollador de SQL para [Base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) de [datos: Controladores JDBC de terceros](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) para agregar el controlador.

**Nombre de host** : sistema de host para la base de datos MySQL.

**Puerto** : **puerto** TCP / IP en el que el servidor MySQL escuchará.

**Elegir base de datos** : Nombre de la base de datos MySQL.

**Manejo de fecha cero** : debido a que el controlador JDBC de MySQL no puede manejar la fecha de inicio predeterminada, especifique una de las siguientes opciones para manejar esta fecha: **Establezca en NULL** para establecerlo en un valor nulo, o **Redondee a 0001-01-01** para configurarlo en 0001-01-01.

**SQL Server y Pestañas Sybase**

La siguiente información se aplica a una conexión a una base de datos de Microsoft SQL Server o Sybase Adaptive Server.

Tenga en cuenta que para conectarse a una base de datos de Microsoft SQL Server o Sybase Adaptive Server, primero debe descargar el controlador de conexión adecuado y luego hacer clic en **Herramientas** , luego en **Preferencias** , y usar el panel de preferencias de usuario de SQL Developer para [Base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) de [datos: Controladores JDBC de terceros](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIEGDD) para agregar conductor.

**Nombre del host** : sistema host para la base de datos Microsoft SQL Server o Sybase Adaptive Server.

**Puerto** : **Puerto** TCP / IP en el que escuchará Microsoft SQL Server o Sybase Adaptive Server.

**Recuperar base de datos** : Nombre de la **base de datos** de Microsoft SQL Server o Sybase Adaptive Server.

**Creación y edición de conexiones**

*Para crear una nueva conexión cuando no exista conexión* , ingrese la información de conexión y haga clic en **Conectar** . Para probar la conexión antes de crearla, haga clic en **Probar** .

*Para crear una nueva conexión cuando una o más conexiones ya existen* , haga clic para seleccionar una conexión existente, cambie el nombre de la conexión al nombre deseado, edite otra información de conexión según sea necesario y haga clic en **Guardar** o **Conectar** para crear la nueva conexión. Para probar la conexión antes de crearla, haga clic en **Probar** .

*Para editar una conexión existente* , haga clic en su entrada en la columna Nombre de la conexión, cambie cualquier información de conexión, excepto el nombre de la conexión, y haga clic en **Guardar** o **Conectar** . Para probar la conexión antes de guardar los cambios, haga clic en **Probar** .

### Renombrar Modelo (Migración)

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en un modelo capturado o convertido y selecciona Cambiar nombre de modelo. Para cambiar el nombre del modelo, cambie el nombre y haga **clic** en **Aceptar** .

### Renombrar elemento de base de datos (migración)

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en un objeto de base de datos bajo un modelo capturado o convertido y selecciona Cambiar nombre. Para cambiar el nombre del objeto, cambie el nombre y haga **clic** en **Aceptar** .

### Seleccione Conexión

Use este cuadro de diálogo para seleccionar una conexión de base de datos para usar con una característica específica del Desarrollador de SQL (por ejemplo, la hoja de cálculo de SQL o el navegador de Informes). Después de hacer **clic** en **Aceptar** , se muestra la interfaz del componente, con el usuario actual igual al que se especificó en la conexión.

Para crear una nueva conexión de base de datos, haga clic en el icono más ( **+** ); para editar la conexión de la base de datos seleccionada, haga clic en el icono de lápiz. En ambos casos, se muestra un cuadro de diálogo para especificar la información de conexión (vea [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) ).

### Información de conexión

Use este cuadro de diálogo para especificar el nombre de usuario y la contraseña para la conexión de la base de datos seleccionada.

Si el nombre de usuario especificado no existe en la base de datos asociada con la conexión, o si la contraseña especificada no es la correcta para ese usuario, se rechaza la conexión.

### No se ha encontrado conexión

Este cuadro de diálogo se muestra cuando intenta realizar una operación que requiere una conexión de base de datos, pero actualmente no existe ninguna conexión para esa operación. Por ejemplo, es posible que haya abierto un archivo SQL pero no haya seleccionado una conexión, o que la conexión se haya desconectado; o es posible que haya intentado realizar una operación de copia de esquema sin especificar las conexiones Desde esquema y Al esquema.

Para seleccionar una conexión en la hoja de trabajo de SQL, haga clic en Aceptar para cerrar este cuadro de diálogo, luego seleccione una conexión de la lista desplegable en la barra de iconos de la hoja de trabajo de SQL.

### Error de cambio de nombre de conexión

Este cuadro de diálogo se muestra cuando intenta cambiar el nombre de una conexión de base de datos a un nombre que ya se utiliza para otra conexión. Por ejemplo, es posible que haya olvidado ingresar un nombre nuevo para la conexión cuyo nombre desea cambiar.

Para cambiar el nombre de la conexión, haga clic en Aceptar para cerrar este cuadro de diálogo, luego especifique un nombre de conexión único.

### Nueva carpeta (Conexiones)

Este cuadro de diálogo le permite crear o cambiar el nombre de una carpeta para organizar las conexiones de la base de datos. Si está creando una carpeta, ingrese el nombre de la nueva carpeta; si está cambiando el nombre de una carpeta, reemplace el nombre existente con el nuevo nombre deseado. Para obtener información sobre el uso de carpetas, consulte [Uso de carpetas para agrupar conexiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDGIJ) .

### Continuar después de una pausa

Este cuadro de diálogo se muestra cuando se encuentra una instrucción de pausa en una secuencia de comandos que está ejecutando en la hoja de cálculo de SQL.

Para continuar con la ejecución en la instrucción después de la instrucción PAUSE, haga clic en **Sí** . Para detener la ejecución y no continuar con la instrucción después de la instrucción PAUSA, haga clic en **No** .

### Seleccionar biblioteca

Este cuadro de diálogo se muestra cuando hace clic en Examinar en el panel [Base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) al configurar [las Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) . Use este cuadro para seleccionar la biblioteca para la clase de controlador JDBC especificada.

### Crear Biblioteca

Este cuadro de diálogo se muestra cuando hace clic en Nuevo en el cuadro de diálogo Seleccionar biblioteca, que se muestra cuando hace clic en Examinar en el panel [Base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) al configurar [las Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) . Use este cuadro para crear la biblioteca para la clase de controlador JDBC especificada.

### Datos de importacion

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en el nodo Tablas o el nombre de una tabla en el navegador de Conexiones, selecciona Importar datos y especifica el archivo .xls o .csv desde el cual se importan los datos. Le permite crear una tabla e importar datos a ella desde un archivo de Microsoft Excel, o importar datos de Microsoft Excel a una tabla existente.

**Vista previa de datos**

**Hoja de trabajo** : nombre de una hoja de trabajo en el archivo de Microsoft Excel.

**Fila de encabezado?**: Si esta opción está marcada, la primera fila de la hoja de cálculo de Microsoft Excel seleccionada se considera una fila con texto para los encabezados de columna. Si esta opción no está marcada, se considera que la primera fila contiene datos de la hoja de trabajo.

**Locale** : Idioma para cualquier dato de texto en la hoja de trabajo.

**Elegir columnas**

**Columnas disponibles** : enumera las columnas de la hoja de cálculo de Microsoft Excel desde las que puede seleccionar para importarlas en la tabla. Para seleccionar una o más columnas de la hoja de trabajo, use los botones de flecha para mover columnas de Disponible a Seleccionado.

**Columnas seleccionadas** : enumera las columnas cuyos datos se importarán a las columnas en la tabla de la base de datos. Para cambiar el orden de una columna seleccionada en la lista para la operación de importación, selecciónela y use los botones de flecha hacia arriba y hacia abajo.

**Definición de columna**

Le permite especificar el nombre de la tabla de la base de datos e información sobre las columnas en esa tabla.

**Nombre de la tabla** : Nombre de la tabla de la base de datos en la que se importan los datos de Excel.

**Columnas de datos de origen** y **columnas de la tabla de destino** : puede seleccionar una columna de datos de origen (Excel) para mostrar sus propiedades de columna de destino (Oracle). Para **Tipo de datos** , seleccione uno de los tipos admitidos para esta operación de importación. Para una columna VARCHAR2 o NUMBER, debe especificar un valor de **Tamaño / Precisión** apropiado . Puede especificar si el valor de la columna puede ser nulo ( **¿Nullable?** ) Y puede especificar un valor predeterminado ( **Predeterminado** ).

**Terminar**

**Verificar** : Debes verificar los parámetros de importación. Si cualquier prueba falla, la columna Información contiene una breve explicación y debe regresar y corregir cualquier error antes de hacer clic en Finalizar.

**Enviar a** la hoja de **trabajo** : no realiza la operación de importación de inmediato, sino que abre una hoja de trabajo SQL con declaraciones que se utilizarán después de hacer clic en el icono Ejecutar script en la hoja de trabajo.

Para realizar la operación de importación, o para enviar las declaraciones a una hoja de cálculo SQL si así lo especificó, haga clic en **Finalizar**.

### Exportar / importar descriptores de conexión

El cuadro de diálogo Exportar descriptores de conexión exporta información sobre una o más conexiones de base de datos a un archivo XML. El cuadro de diálogo Importar descriptores de conexión importa las conexiones que se han exportado. Las conexiones que importa se agregan a cualquier conexión que ya exista en SQL Developer.

**Nombre de archivo** : Nombre del archivo XML que contiene información exportada o que contiene información para importar. Utilice el botón Examinar para especificar la ubicación.

**Conexiones** : nombres de las conexiones que puede seleccionar para la operación de exportación o importación.

### Crear / Editar enlace de base de datos

La siguiente información se aplica a un enlace de base de datos, que es un objeto de base de datos en una base de datos que le permite acceder a objetos en otra base de datos, como se explica en [Enlaces de bases](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBCIGJ) de [datos (públicos y privados)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBCIGJ) .

**Público** : si esta opción está marcada, el enlace de la base de datos es público (disponible para todos los usuarios). Si esta opción no está marcada, el enlace de la base de datos es privado y solo está disponible para usted.

**Esquema** : esquema de base de datos en el que se creará el enlace de la base de datos.

**Nombre** : Nombre del enlace de la base de datos. Debe ser único dentro de un esquema.

**Nombre de host:** El nombre de servicio de una base de datos remota. Si solo especifica el nombre de la base de datos, la base de datos Oracle agrega de manera implícita el dominio de la base de datos a la cadena de conexión para crear un nombre de servicio completo. Por lo tanto, si el dominio de la base de datos de la base de datos remota es diferente al de la base de datos actual, debe especificar el nombre completo del servicio.

**Usuario actual** : crea un enlace de base de datos de usuario actual. El usuario actual debe ser un usuario global con una cuenta válida en la base de datos remota. Si el enlace de la base de datos se usa directamente, es decir, no desde un objeto almacenado, entonces el usuario actual es el mismo que el usuario conectado.

**Usuario fijo** : crea un enlace de base de datos de usuario fijo, para el que especifica el nombre de usuario y la contraseña utilizados para conectarse a la base de datos remota.

**Compartido** : si esta opción está marcada, se utiliza una sola conexión de red para crear un enlace de base de datos pública que se puede compartir entre varios usuarios. En este caso, también debe especificar la información de autenticación.

**Autenticación - Nombre de usuario** y **Contraseña** : el nombre de usuario y la contraseña en la instancia de destino. Esta información autentica al usuario en el servidor remoto y se requiere por seguridad. El usuario y la contraseña especificados deben ser un usuario y una contraseña válidos en la instancia remota.

**Pestaña DDL**

Puede revisar y guardar la declaración SQL que SQL Developer usará para crear el enlace de la base de datos.

### Crear / Editar Índice

La siguiente información se aplica a un índice, que es un objeto de base de datos que contiene una entrada para cada valor que aparece en la (s) columna (s) indexada (s) de la tabla o clúster y proporciona acceso directo y rápido a las filas, como se explica en [Índices](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGABDH) .

**Avanzado** : si esta opción está marcada, el cuadro de diálogo cambia para permitirle establecer propiedades avanzadas (seleccione Avanzado en el lado izquierdo).

**Esquema** : esquema de base de datos que posee la tabla asociada con el índice.

**Tabla** : Nombre de la tabla asociada al índice.

**Nombre** : Nombre del índice. Debe ser único dentro de un esquema.

**Tipo de índice: Normal** para un índice de Oracle estándar, en cuyo caso también especifica no único, único o mapa de bits, así como una o más expresiones de índice; o **Texto** para un índice de texto de Oracle (creado con INDEXTYPE IS CTXSYS.CONTEXT), en cuyo caso se especifica la columna a indexar.

**No único** significa que el índice puede contener múltiples valores idénticos; **Único** significa que no se permiten valores duplicados; **El mapa de bits** almacena las filas asociadas a un valor clave como un mapa de bits.

**Índice** : una lista de expresiones de índice, es decir, las columnas de la tabla o expresiones de columna en el índice. Para agregar una expresión de índice, haga clic en el icono Agregar expresión de columna (+); esto agrega un nombre de columna aquí y en Expresión de columna, donde puede editarlo. Para eliminar una expresión de índice, haga clic en el icono Eliminar expresión de columna (X); para mover una expresión de índice hacia arriba o hacia abajo en la lista, haga clic en los iconos Mover columna arriba y Mover columna abajo. Un índice debe tener al menos una expresión de índice.

Por ejemplo, para crear un índice en la columna AUTHOR\_LAST\_NAME de la tabla BOOKS del tutorial (ver [Crear una tabla (BOOKS)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHGHGF) ), haga clic en el icono + y seleccione AUTHOR\_LAST\_NAME en Nombre de columna o Expresión (campo siguiente), que cambia los LIBROS a AUTHOR\_LAST\_NAME en el campo de índice.

**Nombre o expresión de** columna: un nombre de columna o expresión de columna. Una expresión de columna es una expresión creada a partir de columnas, constantes, funciones de SQL y funciones definidas por el usuario. Cuando especifica una expresión de columna, crea un índice basado en funciones.

**Orden** : ASC para un índice ascendente (valores de índice ordenados en orden ascendente); DESC para un índice descendente (valores de índice ordenados en orden descendente).

**Propiedades**

Le permite especificar propiedades de índice como la compresión, el paralelismo y las opciones de almacenamiento.

**Comprimir** : si esta opción está marcada, la compresión clave está habilitada, lo que elimina la repetición de valores de columna clave y puede reducir sustancialmente el almacenamiento. Si esta opción está marcada, puede ingresar un número entero para especificar la longitud del prefijo (número de columnas de prefijo para comprimir).

**Paralelo** : si esta opción está marcada, se habilita la creación paralela del índice. También puede ingresar un número entero en el cuadro de texto para especificar el grado de paralelismo, que es el número de hilos paralelos utilizados en la operación paralela. (Cada subproceso paralelo puede usar uno o dos servidores de ejecución paralelos.) Si especifica Paralelo sin ingresar un número entero, el grado óptimo de paralelismo se calcula automáticamente.

**Opciones de almacenamiento** : le permite especificar opciones de almacenamiento para el índice. Muestra el cuadro de diálogo [Opciones de almacenamiento](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEGGFG) .

### Crear filtro

Este cuadro de diálogo se muestra cuando hace clic en Nuevo para agregar un filtro de exclusión definido por el usuario al importar archivos en un repositorio de CVS.

**Filtro** : patrón de nombre de archivo de shell, que puede contener caracteres normales y meta-caracteres, incluidos comodines. (Consulte la lista de Filtros seleccionados suministrada para ver los patrones típicos). Por ejemplo, para excluir archivos con la extensión *xyz* , ingrese lo siguiente: \* .xyz

Cuando hace clic en Aceptar, el filtro especificado se agrega a la lista Filtros seleccionados.

### Crear / Editar registro de vista materializada

Utilice este cuadro de diálogo para crear o editar un registro de vista materializada, que es una tabla asociada con la tabla maestra de una vista materializada. Para obtener más información, consulte [Registros de vistas materializadas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHHAHE) .

**Esquema** : esquema de base de datos en el que se creará el registro de vista materializada.

**Nombre** : Nombre de la tabla maestra de la vista materializada que se asociará con este registro de vista materializada.

**Pestaña de propiedades**

**Espacio de tabla** : espacio de tabla en el que se creará el registro de vista materializada.

**Registro** : LOGGING o NOLOGGING, para establecer las características de registro para el registro de vista materializada.

**ID de fila** : Sí indica que el rowid de todas las filas cambiadas debe registrarse en el registro de vista materializada; No indica que el rowid de todas las filas modificadas no se debe registrar en el registro de vista materializada.

**Clave principal** : Sí indica que la clave principal de todas las filas modificadas debe registrarse en el registro de vista materializada; No indica que la clave principal de todas las filas modificadas no se debe registrar en el registro de la vista materializada.

**Nuevos valores** : INCLUYENDO guarda los valores antiguos y nuevos para las operaciones de actualización de DML en el registro de vista materializada; EXCLUYENDO deshabilita la grabación de nuevos valores en el registro de vista materializada. Si este registro es para una tabla en la que tiene una vista agregada materializada de una sola tabla, y si desea que la vista materializada sea elegible para una actualización rápida, debe especificar INCLUYENDO.

**Caché** : para los datos a los que se accederá con frecuencia, CACHE especifica que los bloques recuperados para este registro se colocan en el extremo más recientemente utilizado de la lista de los menos recientemente utilizados (LRU) en la memoria caché del búfer cuando se realiza un escaneo completo de la tabla. Este atributo es útil para tablas de búsqueda pequeñas. NOCACHE especifica que los bloques se colocan en el extremo de la lista LRU utilizado menos recientemente.

**Paralela** : si esta opción está marcada, se admitirán operaciones paralelas para el registro de vista materializada.

**ID de objeto** : solo para un registro en una tabla de objetos: Sí indica que el identificador de objeto definido por el usuario o generado por el sistema de cada fila modificada debe registrarse en el registro de la vista materializada; No indica que el identificador de objeto generado por el sistema o definido por el usuario de cada fila modificada no debe registrarse en el registro de vista materializada.

**Secuencia** : Sí indica que un valor de secuencia que proporcione información de pedido adicional debe registrarse en el registro de la vista materializada; No indica que un valor de secuencia que proporciona información de pedido adicional no debe registrarse en el registro de vista materializada. Los números de secuencia (es decir, Sí para esta opción) son necesarios para admitir una actualización rápida después de algunos escenarios de actualización.

**Columnas de filtro disponibles** : columnas adicionales, que son columnas de clave no primaria a las que hacen referencia las vistas materializadas de la subconsulta, que se registrarán en el registro de la vista materializada. Para seleccionar una o más columnas de filtro, use los botones de flecha para mover columnas de Disponible a Seleccionado.

**Pestaña DDL**

Si está editando un registro de vista materializada existente o si solo ha creado parcialmente un registro de vista materializada, esta pestaña contiene una visualización de solo lectura de una declaración SQL que refleja la definición actual del registro de vista materializada.

Para guardar la declaración SQL en un archivo de script, haga clic en **Guardar** y especifique la ubicación y el nombre del archivo.

### Crear paquete PL / SQL

Use este cuadro de diálogo para crear un paquete que contenga subprogramas PL / SQL (funciones o procedimientos, o una combinación).

**Esquema** : esquema de base de datos en el que se crea el paquete PL / SQL.

**Nombre** : Nombre del paquete. Debe ser único dentro de un esquema.

**Agregar nueva fuente en minúsculas** : si esta opción está marcada, el nuevo texto se ingresará en minúsculas, independientemente del caso en el que lo escriba. Esta opción afecta solo a la apariencia del código, porque PL / SQL no distingue entre mayúsculas y minúsculas en su ejecución.

El paquete se crea y se muestra en la ventana Editor, donde puede ingresar los detalles.

### Crear subprograma PL / SQL (Función o Procedimiento)

Use este cuadro de diálogo para crear un subprograma PL / SQL (función o procedimiento). Una función devuelve un valor; Un procedimiento no devuelve un valor.

Especifique la información para el paquete y para cada parámetro, luego haga clic en Aceptar para crear el subprograma y que se muestre en la ventana del Editor, donde puede ingresar los detalles.

**Esquema** : esquema de base de datos en el que se creará el subprograma PL / SQL.

**Nombre** : Nombre del subprograma. Debe ser único dentro de un esquema.

**Agregar nueva fuente en minúsculas** : si esta opción está marcada, el nuevo texto se ingresará en minúsculas, independientemente del caso en el que lo escriba. Esta opción afecta solo a la apariencia del código, porque PL / SQL no distingue entre mayúsculas y minúsculas en su ejecución.

**Pestaña parametros**

Para cada parámetro en el procedimiento que se creará, especifique la siguiente información.

**Nombre** : Nombre del parámetro.

**Tipo** : **Tipo de** datos del parámetro.

**Modo** : IN solo para entrada, OUT solo para salida o IN OUT para entrada y salida (es decir, la salida se almacena en el parámetro que sobrescribe su valor de entrada inicial).

**Valor predeterminado** : Opcionalmente, el valor predeterminado si el parámetro se omite o se especifica como nulo cuando se llama al subprograma.

Para agregar un parámetro, haga clic en el icono Agregar (+); para eliminar un parámetro, haga clic en el icono Eliminar (X); para mover un parámetro hacia arriba o hacia abajo en la lista, haga clic en el icono de flecha hacia arriba o hacia abajo.

**Pestaña DDL**

Esta pestaña contiene una visualización de solo lectura de una declaración SQL que refleja la definición actual del subprograma.

### Crear directorio remoto

Use este cuadro de diálogo para crear un directorio remoto para una conexión en un repositorio de Subversion.

**Nombre del directorio** : nombre del directorio que se asociará con la URL especificada.

**Comentarios** : **Comentario** descriptivo opcional.

### Crear / Editar Secuencia

La siguiente información se aplica a una secuencia, que es un objeto desde el cual varios usuarios pueden generar enteros únicos. Puedes usar secuencias para generar automáticamente valores de clave primaria.

**Esquema** : esquema de base de datos en el que se crea la secuencia.

**Nombre** : Nombre de la secuencia. Debe ser único dentro de un esquema.

**Incremento** : Intervalo entre números sucesivos en una secuencia.

**Comience con h** : Valor inicial de la secuencia.

**Valor mínimo** : el **valor** más bajo posible para la secuencia. El valor predeterminado es 1 para una secuencia ascendente y - (10 ^ 26) para una secuencia descendente.

**Valor máximo** : el **valor** más alto posible para la secuencia. El valor predeterminado es 10 ^ 27 para una secuencia ascendente y -1 para una secuencia descendente.

**Ciclo** : indica si la secuencia se "envuelve" para reutilizar números después de alcanzar su valor máximo (para una secuencia ascendente) o su valor mínimo (para una secuencia descendente). Si el ciclo de valores no está habilitado, la secuencia no puede generar más valores después de alcanzar su valor máximo o mínimo.

**Tamaño de caché** y **caché** : si se marca Caché, los valores de secuencia se preasignan en caché, lo que puede mejorar el rendimiento de la aplicación; Tamaño de caché indica el número de valores de secuencia preasignados en caché. Si no se selecciona la memoria caché, los valores de secuencia no se preasignan en la memoria caché.

**Orden** : indica si los números de secuencia se generan en el orden en que se solicitan. Si no se especifica ningún orden, no se garantiza que los números de secuencia estén en el orden en que se solicitaron.

**Pestaña DDL**

Puede revisar la declaración SQL que el desarrollador de SQL utilizará para crear una nueva secuencia o que refleje cualquier cambio que haya realizado en las propiedades de la secuencia.

### Crear archivo SQL

Use este cuadro de diálogo para crear un archivo de script SQL y para abrir el archivo en una hoja de cálculo SQL para editarlo.

**Nombre del archivo** : Nombre y extensión del archivo que se creará. La extensión predeterminada y recomendada es .sql.

**Nombre del directorio** : Ruta del directorio para el archivo. Para especificar un directorio, puede hacer clic en Examinar. El directorio predeterminado es la [ubicación de la información relacionada con el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CIHFCGCD) .

### Crear / Editar conexión de Subversion

Esta información se aplica a la creación o edición de una conexión de Subversion. Para obtener información sobre la compatibilidad de SQL Developer con las versiones y Subversion, consulte [Uso de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) .

**URL del repositorio** : **URL** completa y válida para la ubicación del repositorio de Subversion. Los siguientes son esquemas de URL y los métodos de acceso a los que se asignan:

* file: /// - Acceso directo al repositorio (en el disco local)
* http: // - Acceso a través del protocolo WebDAV a un servidor Apache compatible con Subversion
* https: // - Igual que http: //, pero con cifrado SSL
* svn: // - Acceso a través de un protocolo personalizado a un servidor svnserve
* svn + ssh: // - Igual que svn: //, pero a través de un túnel SSH

**Nombre de la conexión** : Nombre de esta conexión. Si deja este cuadro en blanco, la conexión recibirá un nombre basado en la URL de la ubicación del repositorio.

**Nombre de usuario** : nombre de usuario conocido por el repositorio, si el repositorio requiere validación de usuario y contraseña.

**Contraseña** : Contraseña para el usuario especificado, o en blanco si no se requiere una contraseña.

**Probar conexión** : Intenta establecer una conexión con el repositorio de Subversion.

**Estado** : muestra el resultado de la prueba (éxito o un mensaje de error).

### Crear Repositorio Subversion

Esta información se aplica a la creación de un repositorio de Subversion. Una conexión al repositorio se creará automáticamente. Para obtener información sobre la compatibilidad de SQL Developer con las versiones y Subversion, consulte [Uso de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) .

**Ruta del repositorio** : Ubicación del nuevo repositorio de Subversion. Puede **navegar** para seleccionar la ubicación.

**Tipo de sistema de archivos** : **Tipo de** sistema de almacenamiento de datos para el repositorio. Para obtener información sobre cómo elegir un sistema, consulte "Control de versiones con Subversion" en <http://svnbook.red-bean.com/>.

* **Nativo** : el tipo de sistema de archivos utilizado por el sistema operativo.
* **Berkeley DB** : hace que una base de datos Berkeley DB se utilice como sistema de almacenamiento de datos.

**Nombre de la conexión** : Nombre de esta conexión. Si deja este cuadro en blanco, la conexión recibirá un nombre basado en la URL de la ubicación del repositorio.

### Crear / Editar Sinónimo

La siguiente información se aplica a un sinónimo, que es un nombre alternativo para una tabla, vista, secuencia, procedimiento, función almacenada, paquete, vista materializada, objeto de base de datos de clase Java, tipo de objeto definido por el usuario u otro sinónimo.

**Público** : si esta opción está marcada, el sinónimo es accesible para todos los usuarios. (Sin embargo, cada usuario debe tener los privilegios adecuados en el objeto subyacente para poder usar el sinónimo.) Si esta opción no está marcada, el sinónimo es un sinónimo privado y solo se puede acceder a él dentro de su esquema.

**Esquema** : esquema de base de datos en el que se creará el sinónimo.

**Nombre** : Nombre del sinónimo. Un sinónimo privado debe ser único dentro de su esquema; un sinónimo público debe ser único dentro de la base de datos.

**Para: esquema de referencia** : esquema que contiene el objeto o nombre al que hace referencia este sinónimo.

**Basado en objeto** : especifique el objeto al que hace referencia este sinónimo.

**Basada en** el nombre: ingrese el nombre del objeto al que hace referencia este sinónimo.

**Pestaña DDL**

Puede revisar la declaración SQL que el Desarrollador de SQL usará para crear un nuevo sinónimo o que refleje cualquier cambio que haya realizado en las propiedades del mismo.

### Crear tabla (creación rápida)

Este cuadro de diálogo (si no marca el cuadro Avanzado) crea una nueva tabla rápidamente especificando columnas y algunas características de uso frecuente. (Si necesita agregar o cambiar características después de crear la tabla, puede editar la tabla haciendo clic en el ícono Modificar mientras ve la tabla o haciendo clic con el botón derecho en su nombre en el navegador de Conexiones y seleccionando Propiedades, que muestra [Crear / Editar Cuadro de](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) diálogo de [tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .

Para crear una nueva tabla, lo único que **debe** hacer es especificar el esquema y el nombre de la tabla, agregar las columnas necesarias y hacer clic en Aceptar. Aunque no es obligatorio, también debe especificar una clave principal.

**Avanzado** : si esta opción está marcada, el cuadro de diálogo cambia para incluir un conjunto extendido de funciones para crear la tabla. Por ejemplo, debe marcar esta opción si desea crear una tabla particionada, una tabla organizada por índice o una tabla externa.

**Esquema** : esquema de base de datos en el que se creará la tabla.

**Nombre** : Nombre de la tabla. Debe ser único dentro de un esquema.

**Pestaña de tabla (creación rápida)**

Especifica propiedades para cada columna en la tabla.

**Columnas** : enumera las columnas actualmente en la tabla.

**Nota:**

Para agregar una columna después de la columna seleccionada actualmente, haga clic en **Agregar columna** ; para eliminar una columna, selecciónela y haga clic en **Eliminar columna** .

**Columna Nombre** : Nombre de la columna. Debe ser único dentro de la mesa. Sugerencia: para una nueva columna, reemplace cualquier nombre predeterminado, como COLUMN1.

**Tipo** : **Tipo de** datos para la columna. La lista desplegable incluye solo los tipos de datos de uso frecuente seleccionados. Para especificar cualquier otro tipo para la columna, debe usar el panel Columnas del cuadro de diálogo [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .

**Tamaño** : para los datos VARCHAR2, el tamaño máximo de los datos de la columna; para NUMBER data, el número máximo de dígitos.

**No nulo** : si esta opción está marcada, la columna debe contener datos; no puede especificar ningún valor o un valor nulo explícito para esta columna cuando inserta una fila. Si esta opción no está marcada, la columna puede contener datos o no datos.

**Clave principal** : si esta opción está marcada, la columna es la clave principal, o parte de la clave principal, para la tabla. La clave principal es la columna, o conjunto de columnas, que identifica de forma única cada fila en la tabla. Una columna de clave principal no puede ser nula.

Si desea que los valores de la clave principal se completen automáticamente con un método conveniente que utilice un disparador de inserción anterior y una secuencia, antes de que termine de crear la tabla, debe marcar la casilla Avanzado y usar la pestaña de la Clave principal, comenzando con la Rellenar campo de columna de clave principal.

Para agregar otra columna, haga clic en **Agregar columna** . Cuando haya terminado de agregar columnas, haga **clic** en **Aceptar** o haga clic en la pestaña DDL para revisar la declaración CREAR TABLA.

**Pestaña DDL (creación rápida)**

Puede revisar y guardar la sentencia CREATE TABLE que SQL Developer usará para crear una nueva tabla o que refleje cualquier cambio que haya realizado en las propiedades de la tabla. Si desea realizar algún cambio, vuelva a la pestaña Tabla y realice los cambios allí.

Cuando haya terminado, haga **clic** en **Aceptar** .

### Crear / Editar tabla (con opciones avanzadas)

El cuadro de diálogo de la tabla se utiliza para crear una nueva tabla o editar una tabla existente. Las propiedades de la tabla se agrupan en varias pestañas.

Para crear una nueva tabla, lo único que **debe** hacer es especificar el esquema y el nombre de la tabla, agregar las columnas necesarias y hacer clic en Aceptar. Aunque no es obligatorio, también debe especificar una clave principal mediante el [panel Clave principal](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCJJJJ) . Para otras características relacionadas con la tabla, use las pestañas apropiadas; El orden en el que visita las pestañas generalmente no importa, aunque puede que le resulte conveniente visitarlas en la secuencia de este tema. Si está editando una tabla existente, puede visitar las pestañas en cualquier orden.

Si hace clic en Aceptar antes de terminar de crear o editar la tabla, haga clic con el botón derecho en el nombre de la tabla en el navegador de Conexiones, seleccione Editar y continúe creando o editando la tabla.

**Esquema** : esquema de base de datos en el que se creará la tabla.

**Nombre** : Nombre de la tabla. Debe ser único dentro de un esquema.

**Tipo** : El tipo de tabla:

* **Normal** : una tabla de base de datos regular. Se puede particiones (ver [panel Partitioning](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDHHBG) , [panel SUBPARTITION plantillas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDGCFG) , y [panel Definiciones de partición](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACFDFHB) ).
* **Externo** : una tabla externa (vea el [panel de propiedades de la tabla externa](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACDHDBC) ).
* **Índice organizado** : una tabla organizada por índice (consulte el [panel Propiedades organizadas por índice](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBDCEG) ).
* **Tabla temporal** : una tabla temporal, que no se almacena permanentemente en la base de datos. La definición de la tabla temporal persiste de la misma manera que la definición de una tabla regular, pero el segmento de la tabla y los datos en la tabla temporal persisten solo durante la transacción ( Opción de **transacción** ) o la sesión ( Opción de **sesión** ).

**Panel de columnas**

Especifica propiedades para cada columna en la tabla.

**Columnas** : enumera las columnas actualmente en la tabla. Para agregar una columna, haga clic en el icono Agregar columna (+); para eliminar una columna, selecciónela y haga clic en el icono Eliminar columna (X); para mover una columna hacia arriba o hacia abajo en la definición de la tabla, selecciónela y use los botones de flecha hacia arriba y hacia abajo.

**Nota:**

Después de agregar una columna, **para agregar otra columna** , haga clic en el icono Agregar columna (+).

**Nombre** : Nombre de la columna. Debe ser único dentro de la mesa. Sugerencia: para una nueva columna, reemplace cualquier nombre predeterminado, como COLUMN1.

**Tipo de datos** : **Simple** indica un tipo de datos simple (sin objeto); **Complejo** indica un tipo de objeto. Para un tipo complejo, debe especificar el esquema y el nombre del tipo (por ejemplo, MDSYS y SDO\_GEOMETRY para el tipo de geometría de Oracle Spatial).

**Tipo** : Nombre del tipo de datos. La mayor parte de la información restante depende del tipo específico.

**Precisión** : para los datos numéricos, la precisión (número total de dígitos significativos que se pueden representar) de los datos de la columna.

**Escala** : para datos numéricos, la escala (número de dígitos después del punto decimal) de los datos de la columna.

**Tamaño** : para datos de caracteres, el tamaño máximo de los datos de columna.

**Unidades** : para datos de caracteres, las unidades representadas por el Tamaño: **BYTE** para bytes o **CHAR** para caracteres. Este atributo es importante si la base de datos puede contener datos en formato Unicode, con múltiples bytes para cada carácter.

**Predeterminado** : para los tipos relevantes, el valor predeterminado insertado en la columna si no se especifica ningún valor cuando se inserta una fila.

**No puede ser NULL** : si esta opción está marcada, la columna debe contener datos; no puede especificar ningún valor o un valor nulo explícito para esta columna cuando inserta una fila. Si esta opción no está marcada, la columna puede contener datos o no datos. Una columna de clave principal (ver [panel Clave principal](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCJJJJ) ) no puede ser nula.

**Comentario** : **Comentario** descriptivo opcional sobre la columna.

Para agregar otra columna, haga clic en el icono Agregar columna (+).

**Panel de clave principal**

Especifica la clave principal de la tabla. La clave principal es la columna, o conjunto de columnas, que identifica de forma única cada fila en la tabla.

Se crea automáticamente un índice en la clave principal.

**Nombre** : nombre de la restricción que se asociará con la definición de clave primaria. Debe ser único dentro de la base de datos.

**Habilitado** : si esta opción está marcada, la restricción de clave principal se aplica: es decir, los datos en la columna de clave principal (o conjunto de columnas) deben ser únicos y no nulos.

**Columnas disponibles** : enumera las columnas que están disponibles para agregarse a la definición de clave principal.

**Columnas seleccionadas** : enumera las columnas que se incluyen en la definición de clave principal.

Para agregar una columna a la definición de clave principal, selecciónela en Columnas disponibles y haga clic en el icono Agregar (>); para eliminar una columna de la definición de clave principal, selecciónela en Columnas seleccionadas y haga clic en el icono Eliminar (<). Para mover todas las columnas de disponibles a seleccionadas (o al revés), use el icono Agregar todo (>>) o Eliminar todo (<<). Para mover una columna hacia arriba o hacia abajo en la definición de clave principal, selecciónela en Columnas seleccionadas y use los botones de flecha.

Los campos restantes (completar la columna de clave principal a través del nombre de activador) aparecen *solo* cuando está creando una tabla. No están disponibles cuando está editando una tabla existente.

**Rellenar columna de clave principal** : cuando está creando una tabla, si desea usar un activador y una secuencia para insertar automáticamente un valor único en la columna de clave principal cuando inserte una nueva fila, especifique la columna de clave principal.

**De** : una secuencia existente que seleccione, o una nueva secuencia cuyo nombre ingrese. (Para una nueva secuencia, SQL Developer crea la secuencia automáticamente con el nombre que ingresa).

**Nombre del activador** : el nombre del activador anterior a la inserción que se creará automáticamente. Este activador utiliza la secuencia para generar un nuevo valor para la clave principal cuando se inserta una fila. Para ver un ejemplo del uso de esta técnica, consulte la sección de tutoriales [Crear una tabla (TRANSACCIONES)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAEJHJA) .

**Panel de restricciones únicas**

Especifica una o más restricciones únicas para la tabla. Una restricción única especifica una columna o conjunto de columnas, cuyos valores de datos deben ser únicos: cada valor de datos no debe ser nulo y no debe ser el mismo que cualquier otro valor en la columna.

Para una restricción única de varias columnas, la combinación de valores debe ser única, y ninguna columna en la definición de restricción puede tener un valor nulo. Por ejemplo, si especifica las columnas office\_name y city para una restricción única, no podría tener dos oficinas de ventas en Chicago, pero podría tener una oficina de ventas en Chicago y una oficina de ventas en Atlanta.

**Restricciones únicas** : enumera las restricciones únicas actualmente definidas en la tabla. Para agregar una restricción única, haga clic en el botón Agregar; para eliminar una restricción única, selecciónela y haga clic en el botón Eliminar.

**Nota:**

Después de agregar una restricción única, **para agregar otra restricción única** , haga clic en el botón **Agregar** .

**Nombre** : Nombre de la restricción única. Debe ser único dentro de la base de datos.

**Habilitado** : si esta opción está marcada, se aplica la restricción única.

**Columnas disponibles** : enumera las columnas que están disponibles para agregarse a la definición de restricción única.

**Columnas seleccionadas** : enumera las columnas que se incluyen en la definición de restricción única.

Para agregar una columna a la definición de restricción única, selecciónela en Columnas disponibles y haga clic en el icono Agregar (>); para eliminar una columna de la definición de restricción única, selecciónela en Columnas seleccionadas y haga clic en el icono Eliminar (<). Para mover todas las columnas de disponibles a seleccionadas (o al revés), use el icono Agregar todo (>>) o Eliminar todo (<<). Para mover una columna hacia arriba o hacia abajo en la definición de restricción única, selecciónela en Columnas seleccionadas y use los botones de flecha.

**Panel de claves externas**

Especifica una o más claves externas para la tabla. Una clave externa especifica una columna ("columna local"), cada uno de cuyos valores de datos debe coincidir con un valor en la clave principal o una restricción única de otra tabla.

**Claves foráneas** : enumera las claves foráneas actualmente definidas en la tabla. Para agregar una clave externa, haga clic en el botón Agregar; para eliminar una clave externa, selecciónela y haga clic en el botón Eliminar.

**Nota:**

Después de agregar una clave externa, **para agregar otra clave externa** , haga clic en el botón **Agregar** .

**Nombre** : Nombre de la definición de clave foránea. Debe ser único dentro de la base de datos.

**Habilitado** : si esta opción está marcada, se aplica la clave foránea.

**Esquema referenciado** : nombre del esquema que contiene la tabla con la clave principal o la restricción única a la que se refiere esta clave externa.

**Tabla de referencia** : Nombre de la tabla con la clave principal o restricción única a la que se refiere esta clave externa.

**Restricción referenciada** : nombre de la clave principal o restricción única a la que se refiere esta clave externa.

**Asociaciones: Columna local** : muestra la columna en la tabla actualmente seleccionada (local) que se incluye en la definición de clave externa. Para cada columna local en la definición de clave externa, seleccione el nombre de una columna en la tabla local.

**Asociaciones: Columna referenciada en [tabla]** : para cada columna local, identifica la columna en la otra tabla (extranjera) que debe tener un valor que coincida con el valor en la columna local.

**Panel de restricciones de verificación**

Especifica una o más restricciones de verificación para la tabla. Una restricción de verificación especifica una condición que debe cumplirse cuando se inserta una fila en la tabla o cuando se modifica una fila existente.

**Restricciones de verificación** : enumera las restricciones de verificación actualmente definidas en la tabla. Para agregar una restricción de verificación, haga clic en el botón Agregar; para eliminar una restricción de verificación, selecciónela y haga clic en el botón Eliminar.

**Nota:**

Después de agregar una restricción de verificación, **para agregar otra restricción de verificación** , haga clic en el botón **Agregar** .

**Nombre** : Nombre de la definición de restricción de verificación. Debe ser único dentro de la base de datos.

**Habilitado** : si esta opción está marcada, se aplica la restricción de verificación.

**Condición** : Condición que debe cumplirse para una fila. Puede ser cualquier cláusula CHECK válida (sin la palabra clave CHECK). Por ejemplo, para indicar que el valor en una columna numérica denominada RATING debe ser de 1 a 10, puede especificar:rating >=1 and rating <= 10

Para agregar otra restricción de verificación, haga clic en el botón **Agregar** .

**Panel de índices**

Especifica propiedades para cada índice en la tabla.

**Índices** : enumera los índices actualmente definidos en la tabla. Para agregar un índice, haga clic en el icono Agregar índice (+); para eliminar un índice, selecciónelo y haga clic en el icono Eliminar índice (X).

**Nota:**

Después de agregar un índice, **para agregar otro índice** , haga clic en el ícono Agregar índice (+).

**Nombre** : Nombre del índice. Debe ser único dentro del esquema.

**Índice** : una lista de expresiones de índice, es decir, las columnas de la tabla o expresiones de columna en el índice. Para agregar una expresión de índice, haga clic en el icono Agregar expresión de columna (+); esto agrega un nombre de columna aquí y en Expresión de columna, donde puede editarlo. Para eliminar una expresión de índice, haga clic en el icono Eliminar expresión de columna (X); para mover una expresión de índice hacia arriba o hacia abajo en la lista, haga clic en los iconos Mover columna arriba y Mover columna abajo. Un índice debe tener al menos una expresión de índice.

Por ejemplo, para crear un índice en la columna AUTHOR\_LAST\_NAME de la tabla BOOKS del tutorial (ver [Crear una tabla (BOOKS)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBAHGHGF) ), haga clic en el icono + y seleccione AUTHOR\_LAST\_NAME en Nombre de columna o Expresión (campo siguiente), que cambia los LIBROS a AUTHOR\_LAST\_NAME en el campo de índice.

**Nombre o expresión de** columna: un nombre de columna o expresión de columna. Una expresión de columna es una expresión creada a partir de columnas, constantes, funciones de SQL y funciones definidas por el usuario. Cuando especifica una expresión de columna, crea un índice basado en funciones.

**Orden** : ASC para un índice ascendente (valores de índice ordenados en orden ascendente); DESC para un índice descendente (valores de índice ordenados en orden descendente).

**Panel de secuencias de columnas**

Le permite especificar secuencias y los activadores de inserción anterior que se utilizarán para completar una columna con valores. Este enfoque es especialmente conveniente para rellenar automáticamente valores de columna de clave primaria con valores únicos.

**Columna** : nombre de la columna para la cual se deben utilizar una secuencia y un activador para insertar valores únicos. El tipo de datos de la columna debe ser numérico.

**Secuencia** : Ninguna hace que no se use ninguna secuencia; La secuencia existente utiliza la secuencia que especifique; Nueva secuencia crea una nueva secuencia con un nombre predeterminado o especificado.

**Activador** : el **activador** de inserción previa que inserta automáticamente el siguiente valor de la secuencia especificada en la columna cuando se inserta una nueva fila.

**Panel de propiedades de la tabla**

Le permite especificar las propiedades de la tabla, como las opciones de compresión, paralelismo y almacenamiento.

**Comprimir** (solo tablas organizadas en heap): si esta opción está marcada, los segmentos de datos se comprimen para reducir el uso del disco. Esta cláusula es especialmente útil en entornos como los almacenes de datos, donde la cantidad de operaciones de inserción y actualización es pequeña, y en entornos OLTP.

**Paralela** : si se marca esta opción, se habilita la creación paralela de la tabla y se establece el grado predeterminado de paralelismo para las consultas y las instrucciones DML INSERT, UPDATE, DELETE y MERGE después de la creación de la tabla. También puede ingresar un número entero en el cuadro de texto para especificar el grado de paralelismo, que es el número de hilos paralelos utilizados en la operación paralela. (Cada subproceso paralelo puede usar uno o dos servidores de ejecución paralelos.) Si especifica Paralelo sin ingresar un número entero, el grado óptimo de paralelismo se calcula automáticamente.

**Opciones de almacenamiento** : le permite especificar opciones de almacenamiento para la tabla. Muestra el cuadro de diálogo [Opciones de almacenamiento](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEGGFG) .

**Panel de parámetros LOB**

Especifica las opciones de almacenamiento para las columnas LOB (objeto grande), lo que le permite anular las opciones de almacenamiento predeterminadas.

**Columna** : Nombre de la columna LOB.

**Parámetros de LOB** : Si esta opción está marcada, se utilizan los valores especificados para el campo restante. Si esta opción no está marcada, se utilizan los valores predeterminados para todos los campos.

**Segmento** : ID de segmento LOB.

**Espacio** de tabla: Nombre del espacio de tabla para los datos LOB.

**Almacenar en fila** : si esta opción está marcada, el valor LOB se almacena en la fila (en línea) si su longitud es menor a aproximadamente 4000 bytes menos la información de control del sistema.

**Caché:** especifica cómo Oracle Database debe almacenar los bloques en el caché del búfer:

* CACHE: para los datos a los que se accede con frecuencia, indica que los bloques recuperados para esta tabla se colocan en el extremo más recientemente utilizado de la lista de los menos recientemente utilizados (LRU) en la memoria caché del búfer cuando se realiza un escaneo completo de la tabla. Este atributo es útil para tablas de búsqueda pequeñas.
* NOCACHE: para los datos a los que no se accede con frecuencia, indica que los bloques recuperados para esta tabla se colocan en el extremo de la lista LRU que se usó menos recientemente en la memoria caché del búfer cuando se realiza un escaneo completo de la tabla. NOCACHE es el valor predeterminado para el almacenamiento de LOB.
* CACHE READS: Los valores LOB se introducen en la memoria caché del búfer solo durante las operaciones de lectura, pero no durante las operaciones de escritura.

**Retención** : si esta opción está marcada, las versiones antiguas de esta columna de LOB se conservarán. Puede especificar esta opción solo si la base de datos se está ejecutando en modo de deshacer automático y si no especifica un valor de Versión de Pct.

**Registro** : <DEFAULT> significa utilizar el valor predeterminado de la base de datos Oracle. ENCENDIDO significa que la creación de la tabla y cualquier operación subsiguiente del cargador directo (SQL \* Loader) y la ruta directa INSERT en la tabla, partición o almacenamiento de LOB se registran en el archivo de registro de rehacer. OFF significa que estas operaciones no se registran en el archivo de registro de rehacer.

**Chunk** : el número de bytes que se asignarán para la manipulación de LOB. Si el valor no es un múltiplo del tamaño de bloque de la base de datos, entonces la base de datos se redondea en bytes al siguiente múltiplo. El valor máximo es 32768 (32K), que es el mayor tamaño de bloque de base de datos de Oracle permitido. El tamaño predeterminado de CHUNK es un bloque de base de datos Oracle.

**Versión Pct** : Especifica el porcentaje máximo del espacio de almacenamiento LOB general utilizado para mantener las versiones antiguas del LOB. El valor predeterminado es 10, lo que significa que las versiones anteriores de los datos LOB no se sobrescriben hasta que consumen el 10% del espacio de almacenamiento LOB general. Puede especificar un valor de Versión de Pct si la base de datos se está ejecutando en modo manual (donde es el valor predeterminado) o en modo automático de deshacer (donde Retención es el valor predeterminado). No puede especificar un valor de Versión de PCT y la opción Retención.

**Piscinas gratuitas** : Especifica el número de grupos de listas libres para el segmento LOB, por lo general el número de instancias en un ambiente Application Clusters real o 1 para una base de datos de instancia única. Puede especificar esta opción solo si la base de datos se está ejecutando en modo de deshacer automático. No puede especificar tanto un valor de Grupos libres como los campos Listas libres.

**Extensiones - Inicial** : Tamaño de la primera extensión de la tabla. Especifique K (kilobytes) o M (megabytes) para la unidad asociada con el número.

**Extensiones - Siguiente** : Tamaño de la siguiente extensión que se asignará a la tabla. Especifique K (kilobytes) o M (megabytes) para la unidad asociada con el número.

**Extents - Min** : Número mínimo de extensiones asignadas cuando se crea la tabla.

**Extents - Max** : Número máximo de extensiones asignadas cuando se crea la tabla. **Ilimitado** (si está marcado) significa que no hay un máximo (y se ignora cualquier máximo especificado).

**Extensiones - Aumento de PCT** : Porcentaje que cada extensión crece sobre la extensión anterior.

**Buffer Pool** : <DEFAULT> significa utilizar el valor predeterminado de la base de datos Oracle. MANTENER significa poner bloques del segmento en la agrupación de almacenamiento intermedio Mantener; el mantenimiento de un grupo de búferes Keep de tamaño adecuado permite a Oracle conservar el objeto de la base de datos en la memoria para evitar las operaciones de E / S. RECICLAR significa poner bloques del segmento en el grupo de reciclaje; un grupo de reciclaje de tamaño adecuado reduce la cantidad de objetos cuyo grupo predeterminado es el grupo de reciclaje para que no ocupe espacio en caché innecesario.

**Free Lists**: Number of free lists for each of the free list groups for the table. The default and minimum value for this parameter is 1, meaning that each free list group contains one free list.

**Free List Groups**: Number of groups of free lists for the table. The default and minimum value for this parameter is 1. Oracle uses the instance number of Real Application Clusters instances to map each instance to one free list group.

**Partitioning pane**

Specifies partitioning options for a **partitioned table**, which is a table that is organized into smaller and more manageable pieces called partitions. SQL queries and DML statements do not need to be modified in order to access partitioned tables; however, after partitions are defined, DDL statements can access and manipulate individuals partitions rather than entire tables or indexes. Also, partitioning is entirely transparent to applications.

**Partición por** : El tipo de partición: **RANGO** divide la tabla en rangos de valores de la lista de columnas (que para una tableta organizada por índice debe ser un subconjunto de las columnas de clave primaria de la tabla); **HASH** particiona la tabla usando el método hash (filas asignadas a particiones usando una función hash en los valores encontrados en las columnas designadas como la clave de partición); **LIST**particiona la tabla en listas de valores literales de la columna (útil para controlar cómo las filas individuales se asignan a particiones específicas).

**Disponible** : enumera las columnas cuyos valores están disponibles para ser usados ​​al asignar filas a las particiones.

**Seleccionado** : enumera la columna cuyos valores se utilizarán al asignar filas a las particiones.

Para agregar una columna a la definición de partición, selecciónela en Columnas disponibles y haga clic en el icono Agregar (>); para eliminar una columna de la definición de partición, selecciónela en Columnas seleccionadas y haga clic en el icono Eliminar (<). Para mover todas las columnas de disponibles a seleccionadas (o al revés), use el icono Agregar todo (>>) o Eliminar todo (<<). Para mover una columna hacia arriba o hacia abajo en la definición de partición, selecciónela en Columnas seleccionadas y use los botones de flecha.

**Subpartición por** : el tipo de partición que se usará para crear subparticiones dentro de cada partición de rango. Utilice las listas de columnas Disponibles y Seleccionados, seleccione y deseleccione una columna para la subpartición.

**Panel de plantillas de subpartición**

Especifica las opciones de subpartición para una tabla particionada. Las opciones dependen del tipo de subparte y pueden incluir lo siguiente.

**Cantidad de hash** : **Cantidad de** partición de hash.

**Espacios de tabla** : **espacios de** tabla disponibles y seleccionados para el almacenamiento de los datos en una subpartición.

**Plantillas de** subpartición: Especificaciones (plantillas de subparte) para controlar la ubicación de las filas en cada subparte. Haga clic en el icono Agregar (+) para agregar una plantilla de subpartición que sea apropiada para el tipo de subpartición.

**Detalles de la subparte** : para cada plantilla de subparte, especifique un nombre y (si corresponde) un valor o conjunto de valores que sea apropiado para el tipo de subparte.

**Almacenamiento** : le permite especificar un espacio de tabla para la subparte.

**Panel Definiciones de partición**

Define cada partición para una tabla particionada. Las opciones dependen del tipo de partición y pueden incluir lo siguiente.

**Particiones** : Especificaciones para controlar la colocación de filas en cada partición. Haga clic en el icono Agregar (+) para agregar una especificación de partición que sea apropiada para el tipo de partición.

**Detalles de la partición** : para cada especificación de la partición, especifique un nombre y (si corresponde) un valor o conjunto de valores que sea apropiado para el tipo de subpartición.

**Almacenamiento** : le permite especificar un espacio de tabla para la partición.

**Subparticiones** : le permite especificar información de subpartición.

**Índice de panel de propiedades organizadas**

Especifica las opciones para una **tabla organizada** por **índice** , que es una tabla en la que las filas, tanto los valores de columna de clave principal como los valores de columna no clave, se mantienen en un índice creado en la clave principal. Las tablas organizadas por índices son las más adecuadas para el acceso y la manipulación basados ​​en clave principal.

**PCTTHRESHOLD** : el porcentaje de espacio reservado en el bloque de índice para una fila de tabla organizada por índice; debe ser lo suficientemente grande como para mantener la clave principal. Todas las columnas finales de una fila, que comienzan con la columna que hace que se supere el umbral especificado, se almacenan en el segmento de desbordamiento. PCTTHRESHOLD debe ser un valor de 1 a 50; el valor predeterminado es 50.

**Compresión de clave** : si se marca esta opción, se habilita la compresión de clave, lo que elimina la repetición de valores de columna de clave principal en tablas organizadas por índice. En el cuadro a la derecha de este campo, puede especificar la longitud del prefijo, que es el número de columnas de prefijo para comprimir. (Este valor puede ser de 1 a la cantidad de columnas de clave primaria menos 1; la longitud del prefijo predeterminado es la cantidad de columnas de clave primaria menos 1).

**Incluir columna** : columna en la que se divide una fila de la tabla organizada por índice en partes de índice y desbordamiento. Las columnas de clave primaria siempre se almacenan en el índice. La columna Incluir puede ser la última columna de clave principal o cualquier columna de clave no primaria. Todas las columnas de clave no primaria que siguen a la columna Incluir se almacenan en el segmento de datos de desbordamiento.

**Tabla de asignación** : si esta opción está marcada, SQL Developer crea una asignación de ROWIDs locales a físicos y los almacena en una tabla organizada en montón. Esta asignación es necesaria para crear un índice de mapa de bits en la tabla organizada por índice. Si la tabla organizada por índice está particionada, entonces la tabla de mapeo también está particionada y sus particiones tienen el mismo nombre y atributos físicos que las particiones de la tabla base.

**Desbordamiento** : Especificaciones para el segmento de desbordamiento. Las opciones son las mismas que en el cuadro de diálogo [Opciones de almacenamiento](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACEGGFG) .

**Panel de propiedades de la tabla externa**

Especifica las opciones para una **tabla externa** , que es una tabla de solo lectura cuyos metadatos se almacenan en la base de datos pero cuyos datos se almacenan fuera de la base de datos. Entre otras capacidades, las tablas externas le permiten consultar datos sin cargarlos primero en la base de datos.

**Controlador de acceso** : El controlador de acceso de la tabla externa. El controlador de acceso es la API que interpreta los datos externos para la base de datos: ORACLE\_LOADER o ORACLE\_DATAPUMP. Debe especificar el controlador de acceso ORACLE\_DATAPUMP si especifica la cláusula de subconsulta AS para descargar datos de una base de datos Oracle y volver a cargarlos en la misma base de datos o en una base de datos Oracle diferente.

**Tipo de acceso** : Tipo de datos que se convertirán automáticamente durante las cargas y descargas: BLOB o CLOB.

**Default Directory**: A default directory object corresponding to a directory on the file system where the external data sources may reside. The default directory can also be used by the access driver to store auxiliary files such as error logs.

**Project Column**: Determines how the access driver validates the rows of an external table in subsequent queries. **ALL** processes all column values, regardless of which columns are selected, and validates only those rows with fully valid column entries. If any column value would raise an error, such as a data type conversion error, the row is rejected even if that column was not referenced in the select list. **REFERENCED**processes only those columns in the select list.

The ALL setting guarantees consistent result sets. The REFERENCED setting can result in different numbers of rows returned, depending on the columns referenced in subsequent queries, but is faster than the ALL setting. If a subsequent query selects all columns of the external table, then the settings behave identically.

**Reject Limit**: The number of conversion errors can occur during a query of the external data before an Oracle Database error is returned and the query is aborted.

**Access Parameters**: Values to the parameters of the specific access driver for this external table.

**Especificaciones de ubicación** : Una o más fuentes de datos externas. Cada uno suele ser un archivo, pero no tiene por qué serlo. Oracle Database no interpreta esta cláusula; Depende del controlador de acceso interpretar esta información en el contexto de los datos externos. Use el icono Agregar (+) para agregar cada especificación de ubicación.

**Panel de comentarios**

Comentario descriptivo opcional sobre la mesa.

**Panel DDL**

Puede revisar y guardar la sentencia CREATE TABLE que SQL Developer usará para crear una nueva tabla o que refleje cualquier cambio que haya realizado en las propiedades de la tabla. Si desea realizar algún cambio, vuelva a las pestañas relevantes y realice los cambios allí.

Para guardar la declaración SQL en un archivo de script, haga clic en **Guardar** y especifique la ubicación y el nombre del archivo.

Cuando haya terminado, haga **clic** en **Aceptar** .

### Opciones de almacenamiento

Este cuadro de diálogo se muestra si hace clic en Opciones de almacenamiento en el panel Propiedades al crear o editar una tabla o un índice. Le permite anular las opciones de almacenamiento predeterminadas.

**Espacio** de tabla: nombre del espacio de tabla para la tabla o índice.

**Pct Free** : Porcentaje de espacio en cada uno de los bloques de datos de la tabla o índice reservado para futuras actualizaciones. Puede introducir un valor de 0 a 99.

**Porcentaje utilizado** : porcentaje mínimo del espacio utilizado que Oracle mantiene para cada bloque de datos. Un bloque se convierte en candidato para inserciones de fila cuando su espacio utilizado cae por debajo del valor Pct Used. Puede introducir un valor de 1 a 99.

**Registro** : <DEFAULT> significa utilizar el valor predeterminado de la base de datos Oracle. ENCENDIDO significa que la creación de la tabla y cualquier operación subsiguiente del cargador directo (SQL \* Loader) y la ruta directa INSERT en la tabla, partición o almacenamiento de LOB se registran en el archivo de registro de rehacer. OFF significa que estas operaciones no se registran en el archivo de registro de rehacer.

**Ini Trans** : Número de entradas de transacciones de actualización para las cuales el espacio se reserva inicialmente en el encabezado del bloque de datos.

**Trans máxima** : Número de entradas de transacciones que podrían usar datos simultáneamente en un bloque de datos. Este parámetro ha sido desaprobado. Oracle Database ahora permite automáticamente hasta 255 transacciones de actualización simultáneas para cualquier bloque de datos, según el espacio disponible en el bloque.

**Extensiones - Inicial** : Tamaño de la primera extensión de la tabla o índice. Especifique K (kilobytes) o M (megabytes) para la unidad asociada con el número.

**Extensiones - Siguiente** : Tamaño de la siguiente extensión que se asignará a la tabla o índice. Especifique K (kilobytes) o M (megabytes) para la unidad asociada con el número.

**Extents - Min** : Número mínimo de extensiones asignadas cuando se crea la tabla o el índice.

**Extents - Max** : Número máximo de extensiones asignadas cuando se crea la tabla o el índice. **Ilimitado** (si está marcado) significa que no hay un máximo (y se ignora cualquier máximo especificado).

**Aumento de PCT** : Porcentaje que cada extensión crece sobre la extensión anterior.

**Buffer Pool** : <DEFAULT> significa utilizar el valor predeterminado de la base de datos Oracle. MANTENER significa poner bloques del segmento en la agrupación de almacenamiento intermedio Mantener; el mantenimiento de un grupo de búferes Keep de tamaño adecuado permite a Oracle conservar el objeto de la base de datos en la memoria para evitar las operaciones de E / S. RECICLAR significa poner bloques del segmento en el grupo de reciclaje; un grupo de reciclaje de tamaño adecuado reduce la cantidad de objetos cuyo grupo predeterminado es el grupo de reciclaje para que no ocupe espacio en caché innecesario.

**Listas gratuitas** : Número de listas libres para cada uno de los grupos de listas libres para la tabla o el índice. El valor predeterminado y mínimo para este parámetro es 1, lo que significa que cada grupo de listas libres contiene una lista libre.

**Grupos** de listas libres: Número de grupos de listas libres para la tabla o índice. El valor predeterminado y mínimo para este parámetro es 1. Oracle utiliza el número de instancia de las instancias de Real Application Clusters para asignar cada instancia a un grupo de lista libre.

### Crear gatillo

La siguiente información se aplica a un activador, que es un bloque PL / SQL almacenado asociado con una tabla, un esquema o la base de datos, o un bloque PL / SQL anónimo o una llamada a un procedimiento implementado en PL / SQL o Java . El disparador se ejecuta automáticamente cuando ocurren las condiciones especificadas.

**Esquema** : esquema de la base de datos en el que se crea el activador.

**Nombre** : Nombre del disparador. Debe ser único dentro de la base de datos.

**Agregar nueva fuente en minúsculas** : si esta opción está marcada, el nuevo texto se ingresará en minúsculas, independientemente del caso en el que lo escriba. Esta opción afecta solo a la apariencia del código, porque PL / SQL no distingue entre mayúsculas y minúsculas en su ejecución.

**Pestaña de disparo**

**Tipo de activador** : el tipo de objeto en el que se creará el activador: TABLE, VIEW, SCHEMA o DATABASE. (Los elementos restantes dependen del tipo de disparador).

**Propietario de tabla** o **Propietario de vista** : para un activador en una tabla o vista, el nombre del propietario de la tabla o la vista.

**Nombre de tabla** o **nombre de vista** : para un activador en una tabla o vista, el nombre de la tabla o la vista.

**Antes** o **después** : para un activador en una tabla, seleccione Antes para que la base de datos active el activador antes de ejecutar el evento de activación, o seleccione Después para que la base de datos active el activador después de ejecutar el evento de activación.

**Nivel de** instrucción o **Nivel de fila** : para un activador en una tabla, el Nivel de instrucción activa el activador una vez antes o después de la instrucción de activación que cumple con la restricción de activación opcional definida en la condición CUANDO; El nivel de *fila* activa el activador una vez *para cada fila* que se ve afectada por la instrucción de activación y que cumple con la restricción de activación opcional definida en la condición CUANDO.

**Insertar** , **actualizar** , **eliminar** : para un activador en una tabla o una vista, Insertar activa el activador cada vez que una instrucción INSERT agrega una fila a una tabla o agrega un elemento a una tabla anidada; Los disparos de actualización activan el disparador cada vez que una instrucción UPDATE cambia un valor en una de las columnas especificadas en Columnas seleccionadas (o en cualquier columna si no se especifican columnas); Eliminar activa el desencadenante siempre que una instrucción DELETE elimina una fila de la tabla o elimina un elemento de una tabla anidada.

**Referencia: antiguo** : para un disparador en una tabla, los nombres de correlación en el bloque PL / SQL y la condición WHEN de un disparador de fila se refieren específicamente al valor antiguo de la fila actual.

**Referencia - Nuevo** : para un disparador en una tabla, los nombres de correlación en el bloque PL / SQL y la condición de CUÁNDO de un disparador de fila se refieren específicamente al nuevo valor de la fila actual.

**Columnas disponibles** : para un activador en una tabla, enumera las columnas que puede seleccionar para usar en una definición de activador de actualización.

**Columnas seleccionadas** : para un activador en una tabla, enumera las columnas utilizadas en una definición de activador de actualización.

**Cuándo** : para un disparador en una tabla, una condición de disparador opcional, que es una condición de SQL que debe cumplirse para que la base de datos active el disparador. Esta condición debe contener nombres de correlación y no puede contener una consulta.

**Esquema** : para un activador en un esquema, el nombre del esquema en el que se creará el activador.

**Eventos disponibles** : para un activador en un esquema o base de datos, enumera los eventos que puede seleccionar para usar en la definición del activador.

**Eventos seleccionados** : para un activador en un esquema o base de datos, enumera los eventos utilizados en la definición del activador.

**Pestaña DDL**

Esta pestaña contiene una visualización de solo lectura de una declaración SQL que refleja la definición actual del activador.

### Crear tipo (definido por el usuario)

Este cuadro de diálogo se muestra al hacer clic con el botón derecho en Tipos en el navegador de Conexiones y seleccionar Crear tipo para crear un tipo definido por el usuario. Después de completar la información en este cuadro de diálogo y hacer clic en Aceptar, se muestra una hoja de trabajo SQL en la que debe especificar la definición adecuada del tipo.

**Esquema** : esquema de base de datos en el que se crea el tipo.

**Name**: Name of the type. Must be unique within its schema.

**Type**: Select the type of data type to be created: array type, object type specification, object type specification and type body, or table type.

For more information about creating a user-defined type, see the CREATE TYPE statement in *Oracle Database SQL Language Reference*.

### Create/Edit User

The user dialog box is used for creating a new database user or editing an existing database user. The user properties are grouped under several tabs.

To create or edit a database user, the user associated with your database connection must have the DBA role. You should also be familiar with the main concepts and techniques documented in *Oracle Database Administrator's Guide*.

**User tab**

Especifica propiedades generales para el usuario de la base de datos.

**Nombre de usuario** : la cadena de nombre de usuario. Para un usuario existente, este campo es de solo lectura; para cambiar el nombre, debe dejar al usuario y crear un nuevo usuario con el nombre deseado.

**Nueva contraseña** : cadena de contraseña para el nuevo usuario o nueva contraseña para un usuario existente. También debe escribir la misma cadena de contraseña para **Confirmar contraseña** .

**Contraseña caducada** : si esta opción está marcada, la contraseña está marcada como caducada y el usuario debe cambiarla antes de que se le permita conectarse a la base de datos.

**Cuenta bloqueada** : si esta opción está marcada, el usuario no podrá conectarse a la base de datos hasta que un usuario de DBA desbloquee la cuenta asociada con este usuario.

**Pestaña de roles**

Especifica los roles que se otorgarán al usuario. Para cada rol, puede marcar **Concedido** para otorgar el rol, **Admin** para permitir que el usuario conceda el rol a otros usuarios y **Predeterminado** para usar la configuración predeterminada para Concedido y Admin.

Para su comodidad, puede hacer clic en los botones para afectar a todas las configuraciones (Otorgar todo, Revocar todo, Admin. Todo, Admin. Ninguno, Predeterminado todo, Predeterminado ninguno); luego, puede especificar otras configuraciones para roles individuales.

**Pestaña Privilegios del sistema**

Especifica los privilegios que se otorgarán al usuario. Para cada privilegio, puede marcar **Concedido** para otorgar el privilegio, y **Opción de administración** para permitir que el usuario conceda el privilegio a otros usuarios.

Para su comodidad, puede hacer clic en los botones para afectar a todas las configuraciones (Otorgar todo, Revocar todo, Administrar todo, Admin. Ninguno); luego, puede especificar otras configuraciones para privilegios individuales.

**Pestaña cuotas**

Especifica los límites de uso del disco en espacios de tabla especificados para el usuario. Si marca Sin límite, no hay límite de uso de disco en el espacio de tabla.

**Pestaña SQL**

Muestra las instrucciones SQL que el desarrollador de SQL utilizará para crear (después de ejecutar una instrucción CREATE USER) un nuevo usuario o para editar un usuario existente. Esta pantalla es de solo lectura; Si desea realizar algún cambio, vuelva a las pestañas relevantes y realice los cambios allí.

### Crear / Editar informe definido por el usuario

La siguiente información se aplica a un informe definido por el usuario. Para obtener información sobre cómo crear un informe definido por el usuario, así como ejemplos de cómo crear dichos informes, consulte [Informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGGBDG) .

**Pestaña de detalles**

**Nombre** : Nombre del informe definido por el usuario.

**Estilo** : **Estilo de** informe: *Tabla* (predeterminado), *Código* (da formato al código en la salida), *Gráfico* ( *gráfico de* barras o circular; consulte el [Ejemplo de informe definido por el usuario: Gráfico](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHBIDD) para un ejemplo), *Indicador* (indicador de estado o dial; consulte [Indicadores: En la hoja de trabajo SQL e informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDJDIBD) ), *plsql-dbms\_output* (HTML dinámico; consulte el [Ejemplo de informe definido por el usuario: HTML dinámico](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEBHHB) para un ejemplo), o *Script* ( *secuencia de comandos* ejecutable).

**Descripción** : **Descripción** opcional del informe.

**Información sobre herramientas** : texto de información **sobre herramientas** opcional que se mostrará cuando el puntero del mouse permanezca brevemente sobre el nombre del informe en la pantalla del navegador de informes.

**Declaración SQL** : la declaración SQL completa para recuperar la información que se mostrará en el informe definido por el usuario. Como ejemplo trivial, la instrucción SELECT user "Current User" de DUAL muestra a User actual como el encabezado y el nombre del usuario asociado a la conexión de base de datos actual.

Sugerencia: mire las sentencias de SQL para varios informes proporcionados por el desarrollador de SQL; verifique el panel Mensajes - Registro debajo de los resultados del informe, o haga clic en el icono de SQL en la pestaña Resultados del informe.

**Agregar hijo** : agrega un informe secundario (subinforme) de este informe.

**Prueba** : prueba la definición del informe ejecutándolo en una ventana separada. Esta función le permite probar el informe antes de crearlo.

**Pestaña de columnas**

**Nombre** : Nombre de la columna.

**Formato** : Formato de la columna. Si se especifica, debe utilizar la MessageFormatsintaxis de formato de Java . Para obtener información detallada y ejemplos, consulte la información de Sun Javadoc para la MessageFormatclase en el java.textpaquete.

**hAlinear** : alineación horizontal: *izquierda* o *derecha*

**vAlinear** : alineación vertical: *abajo* , *centro* o *arriba*

**Añadir columna** : agrega una nueva columna.

**Eliminar columna** : elimina la columna seleccionada.

**Pestaña de enlaces**

**Nombre** : Nombre de la variable de enlace.

**Preguntar** : Cadena que se muestra cuando se le solicita al usuario que ingrese un valor. Ejemplo: *nombre de la tabla*

**Predeterminado** : Valor predeterminado si el usuario no ingresa un valor en la solicitud. Para aceptar el valor SQL de Oracle, especifique NULL\_VALUE.

**Información sobre herramientas** : texto de información **sobre herramientas** opcional que se mostrará cuando el puntero del mouse permanezca brevemente sobre el nombre de la variable de enlace.

**Pestaña Detalles del gráfico**

Disponible si el tipo de informe es Gráfico.

**Tipo de gráfico** : gráfico de barras con barras horizontales o verticales, o gráfico circular.

**Gráfica 3D** : *Verdadero* para una apariencia tridimensional; *Falso* para una apariencia bidimensional.

**Efecto** degradado : *Verdadero* para un efecto degradado; *Falso* por ningún efecto degradado.

**Estilo de gráfico** : nombre temático para la apariencia general del gráfico.

**Mostrar cuadrícula** : *Verdadero* para mostrar las líneas de cuadrícula; *Falso* para ocultar las líneas de la cuadrícula.

**Mostrar leyenda** : *Verdadero* para mostrar la leyenda del gráfico; *Falso* para ocultar la leyenda del gráfico.

**Pestaña Detalles del medidor**

Disponible si el tipo de informe es Indicador.

**Tipo de indicador** : Dial (como un indicador de combustible en un automóvil) o medidor de estado (representación de barra).

**Basado en consultas** : *Verdadero* si los valores mínimo, máximo, bajo y alto se especifican en la consulta SQL; *Falso* para especificar los valores mínimo, máximo, bajo y alto en los campos restantes.

**Min** : valor mínimo mostrado en el indicador.

**Max** : valor máximo mostrado en el indicador.

**Bajo** : valor "bajo"; Usualmente mayor que Min y menor que High.

**Alto** : valor "Alto"; por lo general mayor que Bajo y menor que Máx.

### Crear / Editar carpeta de informes definidos por el usuario

La siguiente información se aplica a una carpeta para organizar informes definidos por el usuario. Cada carpeta puede contener informes y otras carpetas (subcarpetas). Por ejemplo, puede crear una carpeta llamada Ventas y luego, bajo esa carpeta, crear carpetas llamadas Ventas por distrito y Ventas por producto.

Para obtener información sobre cómo crear informes y carpetas definidos por el usuario para estos informes, consulte [Informes definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGGBDG) .

**Nombre** : Nombre de la carpeta.

**Descripción** : **Descripción** opcional de la carpeta.

**Información sobre herramientas** : texto de información **sobre herramientas** opcional que se mostrará cuando el puntero del mouse permanezca brevemente sobre el nombre de la carpeta en la pantalla del navegador de informes.

### Crear / Editar vista

El cuadro de diálogo de vista se utiliza para crear o editar una vista o una vista materializada. Puede usar la pestaña Consulta de SQL o una serie de paneles para especificar la parte de consulta de la definición de vista, y puede usar uno o más paneles (según el tipo de vista) para otras partes de la definición.

Si hace clic en Aceptar antes de terminar de crear o editar la vista, haga clic con el botón derecho en el nombre de la vista en el navegador de Conexiones, seleccione Editar y continúe creando o editando la vista.

**Esquema** : esquema de base de datos en el que se creará la vista.

**Nombre** : Nombre de la vista. Debe ser único dentro de un esquema.

**Avanzado** : si esta opción está marcada, el cuadro de diálogo cambia para incluir un panel que proporciona un conjunto extendido de funciones para crear la vista.

**Pestaña o panel de consulta SQL**

Como una pestaña (si no marcó la casilla Avanzado), contiene el código SQL para la parte de consulta de la definición de vista, usando las palabras clave SELECCIONAR y DESDE y generalmente una cláusula WHERE con la sintaxis necesaria para recuperar la información deseada.

Como un panel (si marcó la casilla Avanzado), presenta opciones para construir partes específicas de la consulta.

Por ejemplo, la siguiente consulta, del tema del tutorial [Crear una vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFDCH) , selecciona columnas de las tablas PATRONES y TRANSACCIONES, ordenándolas primero por valores en la columna PATRON\_ID en la tabla PATRONES y luego por valores en la columna TRANSACTION\_TYPE en la tabla TRANSACCIONES. El resultado es un listado por ID de usuario de todos los usuarios que tuvieron transacciones, y para cada usuario listado la información de transacción enlistada por tipo de transacción

CREAR VISTA patrons\_trans\_view AS

SELECCIONE p.patron\_id, p.last\_name, p.first\_name,

t.transaction\_type, t.transaction\_date

De los clientes p, transacciones t

DONDE p.patron\_id = t.patron\_id

ORDEN POR p.patron\_id, t.transaction\_type;

**Resultados de SQL Parse** : si hace clic en Probar sintaxis, muestra los errores de sintaxis de SQL o muestra un mensaje que indica que no hay errores si no hay errores de sintaxis.

**Revertir** : cancela cualquier edición que haya realizado en el cuadro de Consulta SQL y muestra el contenido del cuadro antes de estas modificaciones.

**Sintaxis de prueba** : comprueba la declaración en el cuadro de consulta SQL para cualquier error de sintaxis SQL.

**Panel de objetos de selección rápida**

Especifica los objetos que puede usar en las cláusulas SELECT, FROM y WHERE de la definición de vista. Identifique las tablas y vistas en las que se basa esta vista, y las columnas en esas tablas y vistas que se utilizan en la definición de esta vista. Para ver los resultados de su especificación de selección rápida, verifique Auto-Consulta o haga clic en Consulta.

**Esquema** : esquema de base de datos que contiene los objetos a seleccionar.

**Tipo de filtro - Tipos de filtro** : le permite limitar la visualización de los objetos disponibles para su selección a ciertos tipos de objetos de base de datos (por ejemplo, para mostrar solo tablas o vistas).

**Filtro de nombre** : le permite limitar la visualización de los objetos disponibles para su selección según una cadena de caracteres en el nombre, con el signo de porcentaje (%) como un carácter comodín. Por ejemplo, para limitar la visualización de las tablas y vistas disponibles a aquellos cuyos nombres comienzan con la cadena EM, especifique el siguiente filtro de nombre:EM%

**Consulta automática** : si esta opción está habilitada, la visualización de los objetos disponibles se actualiza automáticamente cuando especifica o cambia el valor del Filtro de tipo o Filtro de nombre.

**Query**: Refreshes the display of available objects based on the Type Filter and Name Filter values.

**Available**: Lists the objects (typically, tables and views in a hierarchical display) from which you can select objects to use in the SELECT, FROM, and WHERE clauses of the view definition.

**Selected**: Lists the objects (typically, columns) that you can use in the SELECT, FROM, and WHERE clauses of the view definition.

To add an object as selected, select it in Available and click the Add (>) icon; to remove an object as selected, select it in Selected and click the Remove (<) icon. To move all objects from selected to available, use the Remove All (<<) icon. To move an object up or down in the selected list, select it in Selected and use the arrow buttons.

For the example in [DDL tab or pane](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGCJGG), select the DEPTNO and SAL columns from the EMP table.

**FROM Clause pane**

Specifies the tables and views that you can use in the FROM clause of the view definition.

**Available**: Lists the tables and views that are available to be selected for use in the FROM clause of the view definition.

**Selected**: Lists the tables and views that you can use in the FROM clause of the view definition.

To add an object as selected, select it in Available and click the Add (>) icon; to remove an object as selected, select it in Selected and click the Remove (<) icon. To move all objects from available to selected, use the Add All (<<) icon; to move all objects from selected to available, use the Remove All (<<) icon.

**Alias** : Alias ​​para la tabla o vista.

Para el ejemplo en la [pestaña o panel DDL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGCJGG) , seleccione la tabla EMP.

**Panel de cláusula SELECT**

Especifica los objetos que puede utilizar en la cláusula SELECT de la definición de vista.

**Lista SELECT** : enumera los objetos (normalmente, columnas) que puede usar actualmente en la cláusula SELECT. Para agregar un objeto, haga clic en el icono Agregar (+); para eliminar un objeto, selecciónelo y haga clic en el icono Eliminar (X); para mover un objeto hacia arriba o hacia abajo en la definición de la vista, selecciónelo y use los botones de flecha hacia arriba y hacia abajo.

**Nota:**

Después de agregar un objeto, **para agregar otro objeto** , haga clic en el icono Agregar (+).

**Expresión** : nombre de columna o una expresión. Para las expresiones, puede escribirlas o puede usar la **Paleta de expresiones** para agregar nombres de objetos y nombres de funciones.

**Validar** : comprueba la validez de la entrada de expresión.

Para el ejemplo en la [pestaña o panel DDL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGCJGG) , seleccione la columna DEPTNO y las funciones MIN (emp.sal) y MAX (emp.sal).

**Panel de cláusula WHERE**

Especifica la cláusula WHERE de la definición de vista.

**WHERE** : El texto de la cláusula WHERE, sin la palabra clave WHERE. Puede escribir el texto completamente; o puede escribir parte del texto y usar la **Paleta de expresiones** para agregar nombres de objetos, nombres de funciones y operadores.

Ejemplo (del ejercicio Tutorial [Crear una vista](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/tut_library.htm#CBACFDCH) ):p.patron\_id = t.patron\_id

**Panel de cláusula GROUP BY**

Especifica una cláusula que se utilizará para agrupar las filas seleccionadas según el valor de las columnas para cada fila y devolver una fila única de información de resumen para cada grupo. La cláusula GROUP BY agrupa las filas pero no garantiza el orden del conjunto de resultados; para ordenar las agrupaciones, use la cláusula ORDER BY.

**Disponible** : enumera las tablas y vistas, y las columnas en cada una, que están disponibles para ser seleccionadas para su uso en la cláusula GROUP BY de la definición de vista.

**Seleccionado** : enumera las tablas y vistas, y las columnas en cada una, que puede usar en la cláusula GROUP BY de la definición de vista.

Para agregar un objeto como se seleccionó, selecciónelo en Disponible y haga clic en el icono Agregar (>); para eliminar un objeto según lo seleccionado, selecciónelo en Seleccionado y haga clic en el icono Eliminar (<). Para mover todos los objetos de disponibles a seleccionados, use el ícono Agregar todos (<<); para mover todos los objetos seleccionados a disponibles, use el icono Eliminar todo (<<).

**Panel de cláusula que tiene**

Especifica una expresión que debe cumplirse para que las filas sean procesadas por la cláusula GROUP BY. Por ejemplo, HAVING MIN(salary) < 30000hace que la cláusula GROUP BY considere solo las filas en las que el valor mínimo de los valores salariales relevantes sea inferior a 30000.

**TENIENDO** : puede escribir el texto de la expresión completa, o puede usar la **Paleta de expresiones** para agregar nombres de objetos, nombres de funciones y operadores al texto de la expresión.

**Panel de cláusula ORDER BY**

Especifica una o más columnas o expresiones de columna cuyos valores se utilizarán para ordenar los resultados devueltos por la vista. Sin una cláusula ORDER BY, no existe garantía de que la misma consulta ejecutada más de una vez recupere filas en el mismo orden.

**Lista ORDER BY** : enumera los objetos (normalmente, columnas) que puede usar actualmente en la cláusula ORDER BY. Para agregar un objeto, haga clic en el icono Agregar (+); para eliminar un objeto, selecciónelo y haga clic en el icono Eliminar (X); para mover un objeto hacia arriba o hacia abajo en la definición de la vista, selecciónelo y use los botones de flecha hacia arriba y hacia abajo.

**Nota:**

Después de agregar un objeto, **para agregar otro objeto** , haga clic en el icono Agregar (+).

**ORDENAR POR Filtro de expresión** : para cada columna o expresión de columna, puede escribir el texto completamente en el cuadro **Expresión** ; o puede escribir parte del texto y usar la **Paleta de expresiones** para agregar nombres de objetos, nombres de funciones y operadores.

**Validar** : prueba la validez de la sintaxis de la expresión.

**Orden** : ASC para ascendente (valores de expresión ordenados en orden ascendente); DESC para descendente (valores de expresión ordenados en orden descendente).

**Orden de nulos** : NULLS FIRST para que los valores de expresión nula aparezcan antes que los valores no nulos; NULLS LAST tiene valores de expresión nula después de valores no nulos. (Las posiciones "Antes" y "Después" están determinadas por el valor del Pedido).

**Panel de información de vista o propiedades de vista materializada**

*Opciones para una vista estándar* :

**Restringir consulta** : si esta opción está marcada, puedes habilitar una de las siguientes opciones

* **Solo lectura** : evita que la vista se use para agregar, eliminar o cambiar datos en la tabla o tablas subyacentes.
* **Opción de verificación** : si esta opción está marcada, prohíbe cualquier cambio en la tabla o tablas subyacentes que produzcan filas que no están incluidas en esta vista.

**Force on create**: If this option is checked, the view is created even if it has errors in its definition. This option is useful if you want to create the view regardless of any errors, and go back and correct the errors later. If this option is not checked, the view is not created is its definition contains any errors.

*Options for a materialized view*:

**Refresh Options**:

**Method**: The method of refresh operation to be performed:

* **Complete Refresh**: Executes the defining query of the materialized view, even if a fast refresh is possible.
* **Actualización rápida** : utiliza el método de actualización incremental, que realiza la actualización de acuerdo con los cambios que se han producido en las tablas maestras. Los cambios para los cambios DML convencionales se almacenan en el registro de vista materializada asociado con la tabla maestra. Los cambios para las operaciones INSERT de ruta directa se almacenan en el registro del cargador directo.
* **Forzar actualización** : realiza una actualización rápida si es posible; de lo contrario, realiza una actualización completa.
* **Nunca** : No realice operaciones de actualización.

**Cuándo** : El tipo de operación de actualización a realizar:

* **On Demand** : realiza una actualización cuando se llama a uno de los procedimientos de actualización DBMS\_MVIEW.
* **Al confirmar** : realiza una actualización rápida cada vez que la base de datos confirma una transacción que opera en una tabla maestra de la vista materializada. Esto puede aumentar el tiempo necesario para completar la confirmación, ya que la base de datos realiza la operación de actualización como parte del proceso de confirmación.
* **Especificar** : realiza las operaciones de actualización de acuerdo con lo que especifique en los campos *Iniciar* y *Siguiente* .
* **Nunca** : No realiza una operación de actualización.

**Tipo** : **Tipo de** actualización, que determina el tipo de vista materializada:

* **Clave principal** : crea una vista materializada de clave principal, que permite que las tablas maestras de vistas materializadas se reorganicen sin afectar la elegibilidad de la vista materializada para una actualización rápida.
* **Id. De fila** : crea una vista materializada de rowid, que es útil si la vista materializada no incluye todas las columnas de clave principal de las tablas maestras.

**Comenzar en** : fecha y hora de inicio de la primera operación de actualización automática. Debe estar en el futuro.

**Siguiente** : Hora para la siguiente operación de actualización automática. El intervalo entre los tiempos de *Inicio* y *Próximo* establece el intervalo para las siguientes operaciones de actualización automática. Si no especifica un valor, la operación de actualización se realiza solo una vez a la hora especificada para *Inicio* .

**Restricciones** : Si esta opción está marcada, se pueden usar más alternativas de reescritura durante la operación de actualización, lo que resulta en una ejecución de actualización más eficiente. El comportamiento de esta opción se ve afectado por la selección de *Forzado* o *Confianza* .

**Forzado** : hace que solo se usen restricciones impuestas durante la operación de actualización.

**De confianza** : permite el uso de información de dimensiones y restricciones que el administrador de la base de datos ha declarado confiable, pero que la base de datos no ha validado. Si la información de la dimensión y la restricción es válida, el rendimiento puede mejorar. Sin embargo, si esta información no es válida, entonces el procedimiento de actualización puede dañar la vista materializada a pesar de que devuelve un estado de éxito.

**Materialized View Options**:

**Parallel**: If this option is checked, parallel operations will be supported for the materialized view, and you can specify a number for the default degree of parallelism for queries and DML on the materialized view after creation.

**Enable Cache**: If this option is checked, the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This setting is useful for small lookup tables. If this option is not checked, the blocks are placed at the least recently used end of the LRU list.

**Tipo de compilación** : especifica cuándo se debe rellenar la vista materializada. **Inmediato** indica que la vista materializada se debe completar de inmediato. **Aplazado** indica que la vista materializada se completará con la siguiente operación de actualización. Si especifica Aplazado, la primera actualización (aplazada) siempre debe ser una actualización completa; hasta entonces, la vista materializada tiene un valor de inactividad inutilizable, por lo que no se puede utilizar para la reescritura de consultas.

**Habilitar** reescritura de consultas: si esta opción está marcada, la vista materializada se habilita para la reescritura de consultas, una técnica de optimización que transforma una solicitud de usuario escrita en términos de tablas maestras en una solicitud semánticamente equivalente que incluye una o más vistas materializadas.

**Opción predefinida** : si esta opción está marcada, una tabla existente se registra como una vista materializada preinicializada. Esta opción es particularmente útil para registrar grandes vistas materializadas en un entorno de almacenamiento de datos. La tabla debe tener el mismo nombre y estar en el mismo esquema que la vista materializada resultante, y la tabla debe reflejar la materialización de una subconsulta. **La precisión reducida** autoriza la pérdida de precisión que se producirá si la precisión de las columnas de la tabla o de la vista materializada no coincide exactamente con la precisión devuelta por la subconsulta. **Ninguna precisión reducida** requiere que la precisión de la tabla o las columnas de vista materializada coincidan exactamente con la precisión devuelta por la subconsulta, o la operación de creación fallará.

**Opciones de almacenamiento de índice** :

**Usar índice** : si esta opción está marcada, se crea un índice predeterminado y se utiliza para acelerar la actualización incremental (rápida) de la vista materializada. Si esta opción no está marcada, este índice predeterminado no se crea. (Por ejemplo, puede elegir suprimir la creación del índice ahora y crear dicho índice explícitamente más adelante).

**Usar espacio de tabla** : si esta opción está marcada, puede especificar el espacio de tabla en el que se creará la vista materializada. Si esta opción no está marcada, la vista materializada se crea en el espacio de tabla predeterminado del esquema que contiene la vista materializada.

**Pestaña o panel DDL**

Si está editando un objeto existente o si solo ha creado parcialmente un objeto, esta pestaña contiene una visualización de solo lectura de una declaración SQL que refleja la definición actual del objeto.

Para guardar la declaración SQL en un archivo de script, haga clic en **Guardar** y especifique la ubicación y el nombre del archivo.

### Crear esquema XML

Este cuadro de diálogo le permite especificar la URL de un esquema XML que puede asociarse con instancias de documentos XML.

**Esquema** : nombre del esquema en el que se creará el objeto de esquema XML.

**Nombre** : URL del esquema XML.

### Configurar extensión

Este cuadro de diálogo, que se muestra si hace clic en Configurar para la compatibilidad con las versiones en el panel de preferencias de Extensiones, le permite seleccionar entre las extensiones de compatibilidad de versiones disponibles para SQL Developer. Para obtener información sobre el uso del control de versiones con SQL Developer, consulte [Uso del control de versiones](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDIICEH) .

Si cambia alguna configuración existente, deberá reiniciar SQL Developer.

### Configurar asociaciones de tipo de archivo

Este cuadro de diálogo, que se muestra la primera vez que inicia SQL Developer, le permite asociar ciertos tipos de archivos con SQL Developer. Si un tipo de archivo está asociado con SQL Developer, los archivos con la extensión de ese tipo serán abiertos automáticamente por SQL Developer cuando haga doble clic en el nombre del archivo. Se reemplaza cualquier asociación anterior para ese tipo de archivo.

Si no asocia un tipo de archivo con SQL Developer, cualquier asociación existente para ese archivo no se modifica.

Después de cerrar este cuadro, puede cambiar las asociaciones para estos tipos de archivo y muchos otros haciendo clic en **Herramientas** y luego en **Preferencias** , y seleccionando **Tipos de archivo** (consulte [Tipos de archivo](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDFCAIH) ).

### Filtros personalizados

Este cuadro de diálogo se muestra si hace clic con el botón derecho y selecciona Personalizar filtros en la pestaña Historial para un archivo de seguimiento de SQL (.trc). Puede modificar un filtro existente o crear un filtro nuevo.

**Lista de filtros** : Nombres de los filtros disponibles. Para editar un filtro existente, seleccione su nombre; Los detalles de ese filtro se muestran en el cuadro de diálogo. Para eliminar un filtro existente, seleccione su nombre y haga clic en **Eliminar** .

Para crear un nuevo filtro, haga clic en **Agregar** y especifique el nombre del filtro.

Para restaurar los filtros a aquellos en la instalación de SQL Developer, haga clic en **Restaurar valores predeterminados** . Esto eliminó cualquier filtro que se haya agregado desde la instalación.

**Expresión simple** : cree el filtro seleccionando una columna y un operador y especificando un valor. Para seleccionar un valor de una lista, haga clic en **Insertar** .

**Expresión compleja** : crea el filtro ingresando una expresión compleja.

### Copia de base de datos (objetos de esquema)

Este cuadro de diálogo se muestra si hace clic en Herramientas y luego en Copiar base de datos. Especifique el tipo de operación y las conexiones para los esquemas de origen y destino. Todos los objetos de la base de datos se copian del esquema de origen al esquema de destino, sujeto a cualquier restricción según el tipo de operación, lo que determina el comportamiento si existen objetos con el mismo nombre en el esquema de destino.

**Panel de origen / destino**

**Conexión de origen** : conexión de base de datos para el esquema desde el que se copian los objetos.

**Conexión de destino** : conexión de base de datos para el esquema en el que se copiarán los objetos

**Crear objetos** : copia los objetos a objetos nuevos en el esquema de destino solo si un objeto existente de ese tipo con el mismo nombre ya no existe.

**Truncar objetos** : elimina las filas existentes en cualquier tabla existente con el mismo nombre y luego carga las filas de la fuente.

**Eliminar objetos** : elimina cualquier tabla existente con el mismo nombre y luego la crea y carga desde la fuente.

**Panel de resumen de copia**

Puede revisar las sentencias de SQL que se utilizarán para realizar la operación de copia de acuerdo con sus especificaciones.

Para regresar y hacer cualquier cambio, haga clic en **Atrás** .

To perform the copy operation, click **Finish**. After the copy operation completes, a log file is displayed.

### Database Schema Differences

This interface is displayed if you click Tools, then Database Diff. You can find differences between objects of the same type and name (for example, tables named CUSTOMERS) in two different schemas, and optionally update the objects in one schema (destination) to reflect differences in the other schema (source).

Use the [Source/Destination pane](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm" \l "BACFBHHB) to specify the source and destination database connections. Database objects in the schemas associated with these connections will be compared. The schemas for the source and destination connections can be in the same database or different databases.

**Source/Destination pane**

**Source Connection**: Database connection for the source schema (the schema in which selected objects are to be compared with objects in the destination schema).

**Destination Connection**: Database connection for the database that contains the destination schema (the schema containing one or more objects of the same type and name as those selected in the source schema). The selected connection can be the same as, or different from, the connection for the source schema.

**Diff Objects**: Check the types of objects that you want to be compared in the source and destination connections. You can click *Toggle All* to check and uncheck all individual types. You must select at least one object type.

**Continuar con el resumen** : si esta opción está marcada, al hacer clic en Siguiente lo lleva directamente al [panel Resumen de dif](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBDGIF) .

**Panel especificar objetos**

Puede limitar los tipos u objetos, y los objetos dentro de los tipos seleccionados, para la operación de comparación.

**Tipo de objeto** : seleccione Todos para todos los tipos de objeto, o un tipo específico de objeto.

**Ir** : haga clic en Ir para mostrar una lista de objetos que cumplen con los criterios de selección para la conexión seleccionada. Use las teclas de flecha para mover los objetos seleccionados o todos los objetos del cuadro de objetos disponibles al cuadro de objetos seleccionados.

**Diff Summary pane**

Puede revisar la información que se utilizará para comparar las conexiones de origen y destino, de acuerdo con sus especificaciones.

Para volver y realizar cambios, haga clic en **Atrás** según sea necesario.

Para realizar la comparación, haga clic en **Finalizar** . Los resultados se muestran en una ventana de *Diff Report* , donde puede ver las declaraciones DDL para actualizar los objetos en el esquema de destino para reflejar las diferencias en el esquema de origen. Para crear un archivo que contenga estas declaraciones DDL, haga clic en el icono **Generar secuencia de comandos** en esa ventana. Para alternar la visualización entre todos los objetos y solo los objetos con diferencias, haga clic en **Mostrar objetos iguales** .

### Panel DDL para crear o editar un objeto

Puede revisar y guardar la declaración SQL que el Desarrollador de SQL usará para crear o editar el objeto, para reflejar cualquier cambio que haya realizado en las propiedades del objeto. Si desea realizar algún cambio, vuelva a los paneles relevantes y realice los cambios allí.

Para guardar la declaración SQL en un archivo de script, haga clic en **Guardar** y especifique la ubicación y el nombre del archivo.

### Depurador - Adjuntar a JPDA

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en un nombre de conexión de base de datos y selecciona Depuración remota. Use este cuadro de diálogo si está utilizando la Arquitectura de depuración de la plataforma Java (JPDA) de Sun Microsystem y le gustaría que el depurador escuche para que un depurador se pueda adjuntar al depurador. Para obtener más información sobre la depuración remota, consulte [Depuración remota](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDDCIEC) .

**Host** : nombre o dirección IP del host remoto en el que SQL Developer debe escuchar la conexión de la base de datos.

**Puerto** : Número de puerto de escucha en el host remoto. Puede elegir cualquier número de puerto válido que no esté en uso por otro proceso.

**Tiempo de espera** : la cantidad de segundos que SQL Developer esperará a que la base de datos remota realice una conexión de depuración.

**No mostrar el cuadro de diálogo antes de conectarse** : si esta opción está marcada, este cuadro de diálogo no se mostrará antes de futuras conexiones para la depuración remota.

### Implementar o importar aplicación

Utilice este asistente para implementar o importar una aplicación de Application Express en un esquema de destino específico.

**Implementar en Connection o especificar archivo para importar**

**Elija Conexión para implementar la aplicación** : Para una operación de implementación, especifique la conexión de la base de datos para el esquema de destino en el que se implementará la aplicación.

**Especificar archivo para importar** : para una operación de importación, especifique la ubicación y el nombre del archivo SQL que contiene la aplicación exportada (generalmente el resultado de una operación anterior de "aplicación de exportación").

**Elegir opciones de importación**

Especifica las opciones para la aplicación que se va a implementar o importar.

**Área de trabajo** : Nombre del área de **trabajo** de Application Express.

**Analizar como esquema** : esquema contra el cual se analizarán todos los SQL y PL / SQL de la aplicación.

**Nombre de la aplicación** : nombre de la aplicación.

**Alias ​​de aplicación** : Alias ​​para la aplicación. Se recomienda que nunca codifique la ID de la aplicación en su aplicación, sino que utilice el alias de la aplicación o una cadena de sustitución incorporada (como APP\_ID y APP\_ALIAS).

**Estado de compilación** : RUN\_ONLY o RUN\_AND\_BUILD

**Id. De aplicación** : especifique si desea que se asigne automáticamente un Id. De aplicación, para usar un Id. Listado existente o para especificar un nuevo Id. Utilice estas opciones para evitar conflictos de ID de aplicación, como cuando necesita tener dos versiones de la misma aplicación en la misma instancia. Por ejemplo, puede estar migrando una aplicación a una instancia de producción y aún debe mantener la versión de desarrollo.

**ID actualmente utilizado por** y **sobrescribir** : si la ID de la aplicación especificada está siendo utilizada actualmente por otra aplicación, puede habilitar la **sobrescritura** para tener el ID de la aplicación asociado a la aplicación que se está implementando o importando.

**Resumen**

Muestra las opciones seleccionadas para la aplicación que se va a implementar o importar. Para hacer cualquier cambio, haga clic en Atrás. Para realizar la operación, haga clic en Finalizar.

### Describir la ventana de objetos

Esta ventana se muestra cuando selecciona un nombre de objeto de base de datos en la hoja de trabajo de SQL, hace clic con el botón derecho y selecciona **Describir** . La información es de solo lectura y se muestra mediante pestañas adecuadas para el tipo de objeto.

Por ejemplo, si la pantalla es para una tabla, la información que se muestra es similar a la del cuadro de diálogo [Crear / Editar tabla (con opciones avanzadas)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGJDDB) .

### Editar valor (Datos de columna de tabla)

Este cuadro de diálogo le permite editar datos en una celda de la cuadrícula de datos de la tabla (es decir, editar el valor de una sola columna dentro de una fila). Puede cambiar el valor de los datos y luego hacer **clic** en **Aceptar** .

Las opciones específicas disponibles dependen del tipo de datos de la columna asociada con esa celda en la cuadrícula.

Si no tiene permiso para modificar los datos, la pantalla de Valor es de solo lectura.

### Introduzca valores de enlace

Este cuadro de diálogo le permite ingresar valores para cada variable de enlace. Si la opción NULA está marcada, no puede ingresar un valor en este cuadro de diálogo.

### Error al escribir al archivo de exportación

Este cuadro se muestra si ha intentado exportar datos de tabla a un archivo, pero la ruta del directorio o carpeta no existe.

Haga **clic** en **Aceptar** para cerrar el cuadro, luego ingrese una ruta válida en el cuadro de diálogo Exportar datos y haga clic en **Aplicar** .

### Exportar (Objetos de base de datos y datos)

Esta interfaz se muestra cuando hace clic en Herramientas y luego en Exportación de base de datos para exportar objetos de la base de datos y, opcionalmente, datos. Para una conexión de base de datos seleccionada, puede exportar algunos o todos los objetos de uno o más tipos de objetos de base de datos a un archivo que contiene declaraciones de lenguaje de definición de datos SQL (DDL) para crear estos objetos. Para especificar opciones para la operación de exportación, use el [panel Tipos a exportar](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJIDCC) . Para especificar los objetos o tipos de objetos para exportar, use el [panel Especificar objetos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCBJBJ) .

En varios paneles, si selecciona **Continuar al resumen** , al hacer clic en Siguiente lo lleva al [panel Exportar resumen](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJFDIF) .

**Panel de origen / destino**

Especifique el archivo de salida, la conexión de la base de datos y las opciones que afectan el contenido (declaraciones DDL) del archivo de salida.

**Archivo** : especifique el nombre del archivo de script que contenga las declaraciones DDL para crear los objetos que se exportarán y las declaraciones INSERT si también va a exportar datos (por ejemplo, my\_tables.sql). Puede hacer clic en **Examinar** para seleccionar un directorio para este archivo. (La ruta de archivo predeterminada para las operaciones de exportación se especifica en las preferencias de usuario de SQL Developer para la [base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) de [datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) ).

**Conexión** : Seleccione la conexión de la base de datos con los objetos a exportar.

**Opciones de DDL** : opciones que afectan a las declaraciones de DDL en el archivo de salida:

* **Mostrar esquema** : si esta opción está marcada, el nombre del esquema se incluye en las sentencias CREAR. Si esta opción no está marcada, el nombre del esquema no se incluye en las sentencias CREATE, lo cual es conveniente si desea volver a crear los objetos exportados bajo un esquema que tiene un nombre diferente.
* **Almacenamiento** : si esta opción está marcada, las cláusulas de ALMACENAMIENTO en las definiciones de los objetos de la base de datos se conservan en las declaraciones DDL exportadas. Si no desea utilizar las definiciones de almacenamiento actuales (por ejemplo, si volverá a crear los objetos en un entorno de sistema diferente), desmarque esta opción.
* **Terminador** : Si esta opción está marcada, se inserta un carácter de terminador de línea al final de cada línea.
* **Impresión bonita** : si esta opción está marcada, las declaraciones tienen un formato atractivo en el archivo de salida y el tamaño del archivo será más grande de lo que sería.
* **Incluir palabra clave BYTE** : si esta opción está marcada, las especificaciones de longitud de columna se refieren a bytes; Si esta opción no está marcada, las especificaciones de longitud de columna se refieren a caracteres.
* **Agregar fuerza a las vistas** : si esta opción está marcada, la opción FUERZA se agrega a cualquier sentencia CREAR VISTA, lo que hace que cada vista se cree incluso si contiene errores.
* **Incluir declaración de caída** : Si esta opción está marcada, se incluye una instrucción DROP antes de cada instrucción CREATE, para eliminar cualquier objeto existente con los mismos nombres. Sin embargo, es posible que desee desmarcar esta opción y crear una secuencia de comandos separada que se pueda ejecutar para eliminar una versión anterior de sus objetos antes de la creación. Esto evita la posibilidad de eliminar accidentalmente un objeto que no tenía intención de soltar.
* **Incluir concesiones** : si esta opción está marcada, las sentencias GRANT se incluyen para cualquier objeto de concesión en los objetos exportados. (Sin embargo, las concesiones sobre objetos que son propiedad del esquema SYS nunca se exportan).
* **Incluir automáticamente objetos dependientes** : si esta opción está marcada, los objetos que dependen de los objetos especificados para la exportación también se exportan. Para usuarios no privilegiados, solo se exportan los objetos dependientes en su esquema; Para usuarios privilegiados, todos los objetos dependientes son exportados.

**Tipos para exportar panel**

Especifique los tipos de objeto a exportar y las opciones para la operación de exportación.

**Tipos de objetos** : marque los tipos de objetos que desea exportar. Puede hacer clic en *Alternar todo* para marcar y desmarcar todos los tipos individuales. Debe seleccionar al menos un tipo de objeto. Tenga en cuenta también lo siguiente:

* **Dependencias** (en Tablas): si esta opción está marcada, las restricciones para cada tabla se definen como restricciones en línea en la sentencia CREATE TABLE; y si existen índices o activadores para una tabla, también se incluyen en la sentencia CREATE TABLE.
* **Restricciones** : Si esta opción está marcada, cualquier restricción para cada tabla se define en instrucciones ALTER TABLE separadas en lugar de en la instrucción CREATE TABLE.
* **Datos** : si esta opción está marcada, se incluyen instrucciones para insertar los datos de una tabla o vista exportada. Si esta opción no está marcada, las declaraciones no se incluyen para insertar los datos de una tabla o vista exportada; es decir, solo se incluyen las declaraciones DDL. Si marca Datos, se incluyen declaraciones para insertar todos los datos en todas las tablas en el esquema seleccionado, a menos que use la pestaña Datos de filtro para limitar los datos que se migrarán.

**Panel especificar objetos**

Puede limitar los tipos u objetos, y los objetos dentro de los tipos seleccionados, para la operación de exportación.

**Tipo de objeto** : seleccione Todos para todos los tipos de objeto, o un tipo específico de objeto.

**Ir** : haga clic en Ir para mostrar una lista de objetos que cumplen con los criterios de selección para la conexión seleccionada. Use las teclas de flecha para mover los objetos seleccionados o todos los objetos del cuadro de objetos disponibles al cuadro de objetos seleccionados.

**Especificar panel de datos**

Puede limitar los datos para la operación de exportación.

**Ir** : haga clic en Ir para mostrar una lista de las tablas disponibles y use las teclas de flecha para mover las tablas seleccionadas o todas las tablas del cuadro disponible al cuadro seleccionado.

Luego, seleccione una tabla, ingrese el texto del filtro (una cláusula WHERE sin la palabra clave WHERE) y haga clic en **Aplicar filtro** .

**Panel Resumen de exportación**

Puede revisar la información que se utilizará para crear el archivo de salida, que contendrá sentencias para exportar objetos y datos de la base de datos de acuerdo con sus especificaciones.

Para volver y realizar cambios, haga clic en **Atrás** según sea necesario.

Para crear el archivo de salida, haga clic en **Finalizar** . El archivo también se muestra en una ventana de hoja de cálculo SQL, donde puede ejecutarlo como un script y realizar otras operaciones.

### Exportar: Filtro de datos avanzado

Este cuadro de diálogo se muestra cuando hace clic en Avanzadas en el cuadro de diálogo [Exportar (Objetos y datos de la base de datos)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJFIFF) .

**Esquema** : seleccione el nombre del esquema para ver los objetos disponibles en los que puede especificar un filtro.

**Filtro** : una cláusula WHERE que especifica la condición o condiciones para filtrar datos del objeto seleccionado.

**Aplicar filtro** : haga clic para aplicar el filtro especificado.

Cuando haya terminado de aplicar cualquier filtro, haga clic en **Aplicar** .

### Error de exportación

Este cuadro de diálogo se muestra cuando intenta exportar algunos o todos los objetos de uno o más tipos de objetos de base de datos a un archivo que contiene sentencias de SQL, pero no incluyó alguna información esencial, que podría incluir uno o más de los siguientes:

* La conexión de la base de datos. Para **Conexión** , seleccione la conexión de base de datos desde la cual se exportarán los objetos.
* El nombre del archivo de salida. Mire la pestaña **Opciones** y asegúrese de haber especificado un archivo.
* Uno o más objetos o tipos de objetos. Mire la pestaña **Objetos** y asegúrese de haber seleccionado (marcado) al menos un objeto o tipo de objeto.

### Exportar datos de la tabla

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en el nombre de una tabla, una pantalla de datos de la tabla, un conjunto de resultados de la hoja de cálculo SQL o el resultado del informe, y selecciona Exportar y luego un formato de exportación. Puede exportar algunos o todos los datos a un archivo o al portapapeles del sistema. Para restringir la salida a columnas específicas, use la [pestaña Columnas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACJACIG) . Para restringir la salida basada en una condición de la cláusula WHERE, use la [pestaña Dónde](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIDDIB) .

**Pestaña de formato**

**Formato** : determina el formato de las entradas escritas en el archivo de salida especificado: Insertar para sentencias SQL INSERT, XML para etiquetas y datos XML, SQL LOADER para un archivo de control de SQL \* Loader, o CSV para valores separados por comas, incluida una fila de encabezado para columna identificadores

**Salida** : el archivo escribe la salida en un archivo que usted especifique; El Portapapeles coloca la salida en el portapapeles del sistema, de modo que pueda pegarlo en un archivo, una línea de comando u otra ubicación apropiada para el formato.

**Archivo** : Si la salida es a un archivo, haga clic en Examinar para seleccionar el directorio o la carpeta y para especificar el nombre y la extensión del archivo. La ruta del archivo se coloca en el cuadro Archivo. (La ruta de archivo predeterminada para las operaciones de exportación se especifica en las preferencias de usuario del desarrollador de SQL para la [base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) de [datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDBHCAG) ). Las extensiones de archivo estándar son .sql para el formato de inserción, .xml para el formato XML, .ctl para el formato LOADER de SQL y .csv para el formato CSV.

**Pestaña de columnas**

Puede especificar si la salida debe incluir datos de todas las columnas o solo de las columnas marcadas. (Nota: para las columnas CLOB, solo se exportan los primeros 32 KB de cualquier CLOB).

**Donde pestaña**

Puede restringir la salida ingresando una cláusula WHERE válida para una consulta en la tabla, sin la palabra clave WHERE. Por ejemplo, para restringir los datos exportados a filas donde una columna llamada RATING contiene un valor mayor que 5, especifique:rating > 5

### Configuración del localizador externo

Este cuadro de diálogo se muestra si hace clic en Configuración del localizador externo al crear un repositorio de CVS. Especifique la información requerida para conectarse al repositorio remoto cuando el método por el cual el cliente tendrá acceso y se autentique en el servidor es Externo.

Establecer shell remoto: si esta opción está marcada, se accede a los repositorios externos a través de una utilidad de shell remota, generalmente rsh (la opción predeterminada) o ssh.

**Establecer shell remoto** : si esta opción está marcada, especifique el nombre del programa CVS en el servidor remoto. (Es poco probable que deba cambiarse de la configuración predeterminada, y solo debe cambiarse en colaboración con el administrador del servidor remoto CVS).

### Herramientas externas

Este cuadro de diálogo se muestra cuando hace clic en Herramientas y luego en Herramientas externas. Muestra información sobre herramientas externas definidas por el usuario que están integradas con la interfaz del desarrollador de SQL.

**Buscar herramientas** : busca las herramientas que Oracle ofrece para su consideración y las agrega a la lista si aún no están incluidas.

**Nuevo** : inicia un asistente para definir una nueva herramienta externa (consulte [Crear / Editar herramienta externa](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAAHG) ).

**Editar** : muestra un cuadro de diálogo para editar la herramienta externa seleccionada (ver [Crear / Editar herramienta externa](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAAHG) ).

### Crear / Editar herramienta externa

Esta interfaz se muestra como un asistente si está creando una nueva herramienta externa, y como un cuadro de diálogo si está editando una herramienta externa existente (consulte [Herramientas externas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHHCFE) ).

**Opciones de programas externos**

**Programa ejecutable** : Ruta del ejecutable del programa para la herramienta.

**Argumentos** : Argumentos (parámetros) que deben pasarse al programa. Puede hacer clic en **Insertar** para insertar una macro para el argumento (consulte [Insertar macro](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGIFHH) ).

**Ejecutar directorio** : Directorio en el que se ejecutará el programa. Puede hacer clic en **Insertar** para insertar una macro para el directorio (consulte [Insertar macro](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGIFHH) ).

**Ejemplo de comando** : muestra de solo lectura del comando para ejecutar el programa.

**Opciones de pantalla**

Especifique cómo debe aparecer la herramienta externa cuando se muestra en los elementos del menú o de la barra de herramientas.

**Leyenda para elementos del menú** : la cadena de texto que aparecerá para cualquier elemento del menú que llame a la herramienta externa. Para indicar el carácter mnemotécnico, use el símbolo "antes" del carácter. Por ejemplo: & Mytool para que la "M" esté subrayada y se use como mnemotécnica

**Texto de información sobre herramientas** : Texto de la información sobre herramientas que se mostrará cuando el puntero del mouse se desplace sobre el ícono de la herramienta en una barra de herramientas.

**Ubicación del icono** : ruta del archivo del icono asociado a la herramienta. Haga clic en Examinar para especificar un archivo de gráficos o Use Predeterminado para usar el icono predeterminado (si anteriormente especificó un icono no predeterminado).

**Vista previa** : una visualización de solo lectura del elemento del menú y su icono asociado.

**Opciones de integración**

Especifique cómo la herramienta externa se integrará con SQL Developer.

**Agregar elementos a menús** : marque los menús en los que desea incluir un elemento para esta herramienta.

**Agregar botones a las barras de herramientas** : para agregar el ícono de esta herramienta a la barra de herramientas principal de SQL Developer, revise la barra de herramientas principal.

**Después de las salidas de la herramienta** : para que SQL Developer vuelva a cargar los archivos abiertos después de que la herramienta salga, marque Recargar archivos abiertos.

**Opciones de disponibilidad**

Specify when the external tool is enabled. In contexts where the tool is not enabled, its menu item and icon are grayed out.

**Always**: Makes the tool always available.

**When a File is Selected or Open in the Editor**: Makes the tool available only when a file is selected or open, such as when the SQL Worksheet is open.

**When Specific File Types are Selected**: Makes the tool available only when files of the specified type or types are selected. Use the arrow buttons to move desired types from Available Types to Selected Types.

### Choose Offline Options

This dialog box is displayed when you click Migration, then MySQL, SQL Server, and Sybase Offline Capture, then Create Database Capture Scripts. It specifies options for creating an offline capture properties (.ocp) file, which you can later load and run by clicking Migration, then MySQL, SQL Server, and Sybase Offline Capture, then Load Database Capture Script Output.

**Output Directory**: Converted model containing tables whose data is to be moved to the corresponding Oracle database tables.

**Generate for**: *Windows Batch File* generates a .bat file to be run on Windows systems; *Linux Shell Scripts* generates .sh files to be run on Linux systems.

Para las migraciones de MySQL, si genera archivos .sh, también debe ejecutar el siguiente comando para hacer que los archivos .sh sean ejecutables y el archivo .ocp con capacidad de escritura:

chmod 755 \*

**Plataforma** : la versión de MySQL, Microsoft SQL Server o Sybase Adaptive Server para la cual generar los scripts.

### Característica requerida

Este cuadro de diálogo se muestra si intenta utilizar una función de desarrollador de SQL que requiere la licencia de la función especificada para la base de datos Oracle. Si usted no tiene una licencia para la función especificada, debe hacer clic en **No** .

Si tiene una licencia para la función en la base de datos o bases de datos en las que planea usar la función esta vez, puede hacer clic en **Sí** . Si tiene una licencia para la función en todas las bases de datos en las que planea usarla ahora y en el futuro, puede habilitar **Omitir este mensaje la próxima vez** y hacer clic en **Sí** .

Para adquirir cualquier licencia requerida, comuníquese con su representante de ventas de Oracle o su distribuidor autorizado de Oracle, o vaya a la Tienda Oracle para comprar en línea.

### Filtrar

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en un nodo de tipo de objeto (como Tablas) en el navegador de Conexiones y selecciona Aplicar filtro. Use este cuadro para limitar la cantidad de objetos de ese tipo que se muestran, de acuerdo con uno o más criterios de filtro que especifique. Para cada criterio, especifique lo siguiente:

* Nombre del criterio (por ejemplo, OBJECT\_NAME para una tabla)
* Operador (por ejemplo, LIKE)
* Valor de comparación (por ejemplo, EM%)
* Opción sensible a mayúsculas y minúsculas para la comparación de datos de caracteres

Por ejemplo, para mostrar solo las tablas con nombres que comienzan con EM, especifique: OBJECT\_NAME LIKE EM%(con el signo de porcentaje como un carácter comodín)

Para agregar otro criterio de filtro, haga clic en el icono Agregar (+); para eliminar un criterio, selecciónelo y haga clic en el icono Eliminar (X); para mover un criterio hacia arriba o hacia abajo en la lista, selecciónelo y use los íconos de flecha.

Para aplicar los criterios de filtro a la pantalla del navegador de conexiones, haga **clic** en **Aceptar** .

Para eliminar los efectos de aplicar un filtro, haga clic con el botón derecho en el nodo de tipo de objeto en la pantalla del navegador de Conexiones y seleccione **Borrar filtro** .

### Insertar Macro

Este cuadro de diálogo se muestra cuando hace clic en Insertar al especificar las opciones de programas externos (consulte [Crear / Editar herramienta externa](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCAAHG) ). Le permite insertar una cadena de texto de muestra en el campo relevante para la opción de programa externo; A continuación, puede editar esa cadena para satisfacer sus necesidades. (Esto es algo análogo al uso de fragmentos de código para insertar cadenas de texto en la hoja de cálculo de SQL).

Seleccione el tipo de macro deseado, lea su descripción para asegurarse de que es lo que desea y haga **clic** en **Aceptar** . Para algunas macros, se incluye una muestra de expansión.

### Archivos modificados externamente

Los filtros de este cuadro de diálogo se muestran cuando una aplicación externa ha modificado un archivo que ha abierto en SQL Developer. Se le preguntará si desea volver a cargar el archivo modificado externamente.

Si hace clic en **Sí** , el archivo modificado externamente sobrescribe los cambios que haya realizado en SQL Developer. Si hace clic en **No** , el archivo modificado externamente será sobrescrito por su versión cuando guarde el archivo en SQL Developer.

### Tipos de objetos de filtro

Este cuadro de diálogo filtra (restringe) los tipos de objetos que se mostrarán para el esquema asociado con el usuario seleccionado.

**Tipos de objetos disponibles** : enumera los tipos de objetos que están disponibles para agregarse a la pantalla.

**Tipos de objetos mostrados** : **muestra** los tipos de objetos que se incluyen en la pantalla.

Para agregar un tipo de objeto a la pantalla, selecciónelo en Tipos de objetos disponibles y haga clic en el icono Agregar (>); para eliminar un tipo de objeto de la pantalla, selecciónelo en Tipos de objetos mostrados y haga clic en el icono Eliminar (<). Para mover todos los tipos de objetos disponibles para mostrar (o al revés), use el icono Agregar todos (>>) o Eliminar todos (<<).

### Esquemas de filtro

Este cuadro de diálogo le permite restringir los esquemas que se muestran en Otros usuarios en el navegador de Conexiones.

**Esquemas disponibles** : **enumera** los esquemas que actualmente no se muestran en Otros usuarios en el navegador de Conexiones, pero que están disponibles para agregarse a la lista de usuarios mostrados.

**Esquemas mostrados** : **muestra** los esquemas que se incluirán en la pantalla debajo de Otros usuarios en el navegador de Conexiones.

Para agregar un esquema a la pantalla, selecciónelo en Esquemas disponibles y haga clic en el icono Agregar (>); para eliminar un esquema de la pantalla, selecciónelo en Esquemas mostrados y haga clic en el icono Eliminar (<). Para mover todos los esquemas de disponibles para mostrar (o al revés), use el icono Agregar todos (>>) o Eliminar todos (<<).

**Solo mostrar esquemas con objetos visibles** : limita la visualización a esquemas disponibles que tengan cualquier objeto de base de datos que sea visible para el usuario de la base de datos asociado con la conexión actual.

### Error de filtro

Este cuadro de diálogo se muestra si no especificó ningún dato para una operación de exportación. Asegúrese de especificar las opciones de Datos de filtro que seleccionan algunos datos para la operación de exportación.

### Buscar / reemplazar texto

Este cuadro de diálogo especifica una cadena de texto para buscar, opcionalmente una cadena de texto de reemplazo y opciones de búsqueda.

**Texto para buscar** : Cadena de texto para buscar.

**Reemplazar con** : si marca esta opción, ingrese una cadena de texto para reemplazar la cadena de texto que se está buscando.

**Opciones** : Opciones para controlar el comportamiento de búsqueda: **Match Case** hace que la búsqueda distinga entre mayúsculas y minúsculas; **Buscar desde el principio** inicia la búsqueda al principio y no en el cursor de texto; **Resaltar** todas las apariciones resalta todas las apariciones de la cadena de búsqueda en lugar de solo la primera; **Envolver alrededor de** búsquedas en los saltos de línea; **Palabra completa Solo** encuentre la cadena de búsqueda solo si es una palabra completa y no solo una parte de una palabra; **Expresiones regulares** significa que la cadena de búsqueda es una expresión regular; **Solo texto seleccionado** significa buscar solo en el bloque de texto que ha seleccionado.

**Dirección** : **Adelante** inicia la búsqueda desde el cursor en la dirección del flujo de texto normal; **Retroceso** inicia la búsqueda desde el cursor en la dirección opuesta al flujo de texto normal.

### Buscar resultado

Este cuadro se muestra si especifica el texto para buscar en el cuadro de diálogo [Buscar / Reemplazar texto](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACGECHE) que no se encuentra en la hoja de trabajo de SQL.

Si cree que el texto está en la hoja de trabajo, vuelva a intentar la consulta y verifique la ortografía del texto para buscar.

### Generar Oracle DDL

Este cuadro de diálogo se muestra cuando hace clic en Migración, luego en Generación de secuencias de comandos, luego en Generar DDL de Oracle Especifica el modelo convertido para el cual generar las declaraciones del DDL (lenguaje de definición de datos) de Oracle. La operación produce un archivo de script SQL \* Plus que utiliza para la generación fuera de línea: es decir, puede ejecutar el script para crear los objetos adecuados en la base de datos Oracle.

Cuando la operación está en curso, un cuadro muestra los tipos de objeto y la cantidad de objetos de cada tipo para los cuales se generan las declaraciones DDL.

Una vez que finaliza la operación, un cuadro muestra el directorio en el que se han creado dos archivos: un archivo .sql que contiene sentencias DDL (como CREAR TABLA y CREAR O REEMPLAZAR VISTA) que crean los objetos de esquema migrados en la base de datos Oracle, y a. Archivo ctl que contiene la instrucción SQL \* Plus @ para invocar el archivo .sql.

**Modelos convertidos** : modelo convertido que contiene objetos para los que generar sentencias DDL de Oracle.

### Generar archivos de movimiento de datos sin conexión

Este cuadro de diálogo se muestra cuando hace clic en Migración, luego en Generación de secuencias de comandos, luego en Generar secuencias de datos de movimiento de datos. Especifica el modelo convertido y el directorio de destino si está realizando una migración de datos fuera de línea, que se explica en [Transferencia de datos fuera de línea](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJFJBH) .

**Modelo convertido** : modelo convertido que contiene tablas cuyos datos se moverán a las tablas de base de datos de Oracle correspondientes.

**Directorio** : Ruta en la que se crean los archivos que contienen los datos y las especificaciones de SQL \* Loader.

### Ir al marcador

Use este cuadro para especificar el marcador al que ir en la función o el procedimiento seleccionado. Después de ingresar el marcador y hacer clic en **Ir** , se resalta la línea asociada con ese marcador.

### Ir al número de línea

Use este cuadro para especificar el número de línea al que ir en la función o el procedimiento seleccionado. Después de ingresar el número de línea y hacer clic en el ícono Ir, esa línea se resalta.

### Ir a número de línea: Error

Este cuadro de error le indica que ingresó un número de línea no válido en el cuadro Ir a número de línea, probablemente porque ingresó un número de línea mayor que el de la última línea en la función o procedimiento.

### Importar a CVS

Esta interfaz se muestra cuando hace clic en Versiones, luego en CVS y luego en Importar módulo. Le permite importar archivos locales en el repositorio como un módulo CVS.

**Módulo**

Seleccione el nombre de la conexión, ingrese un nombre para el módulo y, opcionalmente, ingrese un comentario descriptivo sobre la operación de importación.

**Etiquetas**

Seleccione el nombre de la conexión, ingrese un nombre para el módulo y, opcionalmente, ingrese un comentario descriptivo sobre la operación de importación.

**Fuentes**

**Carpeta de origen** : ubicación desde la cual se copiarán los archivos para la operación de importación.

**Filtros**

Puede configurar los filtros que se utilizarán para excluir carpetas y archivos de la operación de importación. Use las teclas de flecha para mover los filtros seleccionados o todos los filtros entre los Filtros disponibles y los Filtros seleccionados.

Para crear un filtro y agregarlo a la lista Filtros seleccionados, haga clic en **Nuevo** para mostrar el cuadro de diálogo [Crear filtro](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBAIBD) .

**Opciones**

Puede especificar las opciones que se utilizarán durante la operación de importación.

**Usar hora de modificación de archivo** : si esta opción está marcada, la hora de modificación del archivo se utiliza como hora de importación. Si esta opción no está marcada, el momento en que se realiza la operación de importación se utiliza como el tiempo de importación.

**Realizar la** comprobación del **módulo** : si esta opción está marcada, los módulos se desprotegen después de que se importan.

**Resumen**

Puede revisar la información que se utilizará para realizar la operación de importación.

Para volver y realizar cambios, haga clic en **Atrás** según sea necesario.

Para realizar la operación de importación, haga clic en **Finalizar** .

### Cargar asignaciones de teclas preestablecidas

Este cuadro de diálogo se muestra al hacer clic en Cargar preset al especificar las preferencias de la clave del acelerador para SQL Developer. Puede cargar un conjunto de asignaciones de teclas predefinidas para ciertos sistemas y aplicaciones de edición externas. Si carga cualquier asignación de teclas preestablecida que entre en conflicto con los cambios que haya realizado, se sobrescribirán.

Puede especificar Predeterminado para restablecer las asignaciones de la clave del acelerador a los valores predeterminados de SQL Developer para su sistema.

### Inicia sesión en CVS

Use este cuadro de diálogo para iniciar sesión en el repositorio CVS especificado. Debe conocer la contraseña del usuario especificado.

**Conectar automáticamente al inicio** : si esta opción está marcada, se inicia una operación de inicio de sesión cuando inicia SQL Developer.

### Modificar valor

Este cuadro de diálogo se muestra cuando hace clic con el botón derecho en una variable en el panel Datos o Datos inteligentes durante la depuración y selecciona Modificar valor. Puede modificar el valor del elemento de datos seleccionado (valor primitivo, cadena o puntero de referencia) durante la depuración. Nota: no puede deshacer la acción después de hacer clic en Aceptar, así que tenga cuidado al realizar cambios.

**Valor actual** : El valor del elemento de datos.

**Nuevo valor** : el nuevo valor para el elemento de datos (ingrese o seleccione de una lista desplegable).

* Para un valor primitivo, puede ingresar un nuevo valor.
* Para un puntero de referencia, puede ingresar la dirección de memoria de un objeto o matriz existente. Para establecer un puntero de referencia en nulo, ingrese 0 como una dirección de memoria.
* Para una cadena, puede ingresar un nuevo valor de cadena o la dirección de memoria de una cadena existente.

**Interpretar nuevo valor como dirección de objeto** : si esta opción está marcada, la entrada Nuevo valor se interpreta como un puntero de dirección de memoria a un objeto o matriz en el montón del programa que está depurando. Para una cadena, esta casilla debe estar marcada si el valor que ingresa en el campo Nuevo valor es la dirección de memoria de una cadena existente

### Detalles de movimiento de datos

Este cuadro de diálogo se muestra cuando hace clic en Migración, luego en Migrar datos. Especifica la información de origen y destino para la migración de datos en línea, que se explica en [Migración de los datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDJCGGI) .

**Conexión de origen** : conexión de base de datos desde la cual se migrarán los datos.

**Conexión de destino** : conexión de base de datos a la que se migrarán los datos.

**Modelo convertido** : modelo convertido que contiene tablas cuyos datos se moverán a las tablas de base de datos de Oracle correspondientes.

**Use nombres calificados del modelo convertido para insertar** : Si esta opción está marcada, los nombres de los objetos se califican por el nombre del esquema.

### Nuevo Procedimiento (Refactorización)

Este cuadro de diálogo se muestra si está editando un procedimiento, seleccione una o más sentencias PL / SQL, haga clic con el botón derecho y seleccione Refactorización, luego Extraiga el procedimiento. Las declaraciones seleccionadas se encapsulan en el procedimiento a crear.

**Definido localmente** : para un procedimiento independiente, define el código recientemente refactorizado en la sección de definición del procedimiento original.

**Almacenado** : para un procedimiento independiente, define el código recientemente refactorizado en un nuevo procedimiento independiente.

**Nombre** : Nombre del procedimiento para encapsular las declaraciones seleccionadas. Para un procedimiento empaquetado, el texto del procedimiento recién extraído se coloca inmediatamente después del procedimiento actual.

### No se ha encontrado ningún objeto

Este cuadro de diálogo se muestra si no se puede encontrar ningún objeto que satisfaga la operación solicitada, como intentar realizar una operación "Describir" cuando el objeto seleccionado actualmente no es válido para una declaración DESCRIBE de SQL \* Plus.

### Ningún objeto seleccionado

Este cuadro de diálogo se muestra si no se seleccionó ningún objeto para la operación solicitada, como intentar realizar una operación "Describir" cuando no se selecciona ningún objeto en la Hoja de cálculo de SQL.

### Abrir documento

Este es un cuadro estándar para seleccionar un archivo para abrir: use **Ubicación** para navegar (doble clic) en la carpeta con el archivo para abrir, luego haga clic en el archivo para seleccionarlo.

### Informe solo de Oracle

Este cuadro de diálogo se muestra si selecciona una conexión de base de datos no Oracle (de terceros) para un informe que se aplica solo a las conexiones de base de datos Oracle. Asegúrese de seleccionar una conexión de Oracle.

### Autenticación Proxy Oracle

Este cuadro de diálogo se muestra si habilita la opción Conexión proxy en el cuadro de diálogo [Crear / Editar / Seleccionar conexión de base de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACICJAF) . Para obtener una explicación de la autenticación de proxy, consulte [Conexiones con autenticación de proxy](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDECIII) .

**Tipo de proxy** : *Nombre de usuario* para la autenticación por nombre de usuario de proxy y contraseña, o *Nombre distinguido* para la autenticación por nombre de usuario de proxy y nombre distinguido.

**Usuario proxy** : Nombre del usuario que se usará para la autenticación para esta conexión.

**Contraseña de proxy** (si el tipo de proxy es el nombre de usuario): contraseña para el usuario de proxy especificado.

**Nombre distinguido** (si el Tipo de proxy es Nombre distinguido): Nombre distinguido para el usuario proxy especificado.

### Pegar

Este cuadro de diálogo se muestra si hace clic en **Editar** y luego en **Pegado extendido** . Muestra una lista de elementos del portapapeles, de modo que puede seleccionar el contenido que desea pegar. Haga **clic** en **Aceptar** para pegar el contenido seleccionado en la ubicación actual.

**Elementos del portapapeles** : elementos del portapapeles con contenido de las operaciones de copia. Por lo general, muestra la primera línea del contenido.

**Contenido del elemento** : el contenido del elemento del portapapeles seleccionado.

### Aviso de privilegio para la migración

Este cuadro de diálogo se muestra si hace clic en Verificar en el cuadro Migración rápida y el usuario de la base de datos para la conexión no tiene todos los privilegios necesarios para una migración multisquema. Para las migraciones multisquemas, este usuario debe otorgar el rol de RECURSO con la opción ADMINISTRADOR; y a este usuario también se le deben otorgar los privilegios CREAR ROL, CREAR USUARIO y ALTER ANY TRIGGER, todos con la opción ADMIN.

Si está realizando una migración de un solo esquema, puede ignorar esta advertencia.

### Consultor de construcción

El cuadro del Generador de consultas se muestra cuando hace clic con el botón derecho en la Hoja de trabajo SQL y selecciona el **Generador de consultas** . Puede usar este cuadro para crear una instrucción SELECT arrastrando y soltando nombres de tablas y vistas, y especificando gráficamente columnas y otros elementos de la consulta. Cuando termina de construir la consulta, la instrucción SELECT resultante se inserta en la Hoja de trabajo de SQL.

Las capacidades del Generador de consultas se agrupan en las siguientes pestañas.

**Seleccionar columnas**

Use la pestaña Seleccionar columnas para seleccionar tablas y vistas, luego columnas dentro de ellas, para usarlas en la consulta. Utilice el árbol de conexiones a la izquierda para encontrar las tablas y vistas deseadas en el esquema o esquemas apropiados, y haga doble clic en cada tabla y vista deseadas.

Dentro de cada tabla o vista seleccionada, haga clic para seleccionar las columnas que desee (todas o algunas específicas) para incluir en la consulta.

**Crear Cláusula Donde**

Use la pestaña Crear Cláusula Donde para seleccionar, para cada columna en la cláusula WHERE, el nombre de la columna, el operador y el valor. Por ejemplo, es posible que desee seleccionar solo las filas donde AUTHOR\_LAST\_NAME contiene Melville o donde RATING> 5.

**Mostrar SQL**

Use la pestaña Mostrar SQL para ver una visualización de solo lectura de la consulta que refleje lo que ha especificado hasta ahora.

**Ver resultados**

Use la pestaña Ver para probar la consulta en su forma actual. Haga clic en el ícono Execute Statement para ejecutar la consulta.

**Actualizar** : especifica el intervalo de actualización: el número de segundos entre cada vez que la consulta se vuelve a ejecutar automáticamente y se actualiza la pantalla de resultados. Un valor de cero (0) significa que la consulta no se vuelve a ejecutar automáticamente después de la ejecución inicial.

### Archivos recientes

Este cuadro de diálogo muestra los archivos recientemente abiertos en SQL Developer.

**Archivos** : una lista de archivos abiertos en SQL Developer, con el archivo más reciente primero. La opción Mostrar todo determina si la lista incluye solo archivos abiertos de forma implícita o archivos abiertos de forma implícita o explícita.

**Mostrar todo** : si esta opción está marcada, la lista incluye archivos abiertos explícita e implícitamente; Si esta opción no está marcada, la lista solo incluye archivos abiertos implícitamente. Los archivos abiertos explícitamente son aquellos que usted abrió directamente; los archivos abiertos de manera implícita son aquellos que SQL Developer abrió para admitir su trabajo (por ejemplo, mientras estaba depurando).

### Crear repositorio

Este cuadro de diálogo se muestra si hace clic en Migración, luego en Administración del repositorio y luego en Crear repositorio.

**Crear repositorio** : Nombre de la conexión de base de datos que se usará para crear un repositorio de migración. Los objetos asociados con el repositorio de migración se crean en el esquema del usuario asociado con la conexión seleccionada.

### Eliminar o Truncar Repositorio

El cuadro de diálogo Eliminar repositorio se muestra si hace clic en Migración, luego en Administración del repositorio y luego en Eliminar repositorio; el cuadro de diálogo Truncar repositorio se muestra si hace clic en Migración, luego en Administración del repositorio y luego en Truncar repositorio.

La eliminación de un repositorio elimina todos los objetos de esquema que se utilizan para el repositorio de migración. Al truncar un repositorio se eliminan todos los datos de los objetos de esquema que se utilizan para el repositorio de migración, pero no se eliminan los objetos del esquema en sí, lo que lo deja con un repositorio vacío.

**Repositorio** : nombre de la conexión de la base de datos en la que se eliminará o truncará el repositorio de migración.

### Capture el XML de Microsoft Access Exporter

Este cuadro de diálogo se muestra si hace clic en Migración y luego en Capture Exporter XML.

**Ruta del archivo** : **Ruta del** archivo al archivo .xml que se produjo cuando ejecutó la versión adecuada de la herramienta del exportador para Microsoft Access (cuando hizo clic en Migraciones, luego en Microsoft Access Exporter y luego en la versión adecuada para su versión de Microsoft Access).

### Cambiar nombre de variable local

Este cuadro de diálogo se muestra si hace clic con el botón derecho en el nombre de una variable en la pantalla del código fuente de una función o procedimiento, selecciona Refactorización y luego Renombra la variable local. Especifique el nuevo nombre deseado para la variable.

### Cambiar el procedimiento

Este cuadro de diálogo se muestra si intenta cambiar el nombre de un procedimiento. Especifique un nuevo nombre único para el procedimiento.

### Seleccionar repositorio actual

Este cuadro de diálogo se muestra si hace clic en Migración, luego en Administración de repositorio y luego en Seleccionar repositorio actual. Puede usar este cuadro de diálogo para volver a conectarse a un repositorio de migración después de que se haya desconectado (usando Migración, luego Administración de repositorio, luego Desconectar repositorio de migración). Además, si tiene varios repositorios de migración y puede usar este cuadro de diálogo para cambiar de uno a otro.

**Seleccione Repositorio actual** : Nombre de la conexión de la base de datos con el repositorio de migración que se utilizará para todas las operaciones relacionadas con la migración de bases de datos de terceros a Oracle.

### No se puede capturar la tabla

Este cuadro de diálogo se muestra si intenta capturar una base de datos de terceros antes de establecer y conectarse a un repositorio de migración actual.

Si no existe un repositorio de migración, cree uno haciendo clic en **Migración** , luego en **Administración de repositorio** , luego en **Crear repositorio** .

Para convertir un repositorio de migración existente en el actual, haga clic con el botón derecho en su conexión en el navegador de Connections y seleccione **Associate Migration Repository** .

Para abrir una conexión al repositorio de migración, expanda el nodo para su conexión en el navegador de conexiones.

### Restablecer contraseña caducada

Este cuadro de diálogo se muestra si intenta crear una nueva conexión de base de datos o abrir una conexión existente, y si la contraseña asociada con la utilizada para la conexión ha caducado. También se muestra solo si hay un controlador OCI (grueso) disponible; Si un controlador OCI no está disponible, se muestra un mensaje de error en lugar de este cuadro de diálogo.

Para restablecer la contraseña, ingrese la nueva contraseña, confírmela y haga **clic** en **Aceptar** .

### Revisión Lister

Este cuadro de diálogo se muestra si hace clic en Revisar lista en el cuadro de diálogo [Rama / Etiqueta](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACHADHJ) . Contiene una lista de revisiones en el repositorio.

Seleccione la revisión que desee utilizar y haga **clic** en **Aceptar** .

### Ejecutar / Depurar / Perfil PL / SQL

Use este cuadro para especificar valores de parámetros para ejecutar, depurar o crear perfiles de una función o procedimiento PL / SQL. (Si especifica un paquete, seleccione una función o un procedimiento en el paquete). Una operación de perfil ejecuta la función o el procedimiento y recopila estadísticas de ejecución; también requiere estructuras auxiliares en el esquema del usuario. Para obtener información, consulte [Uso del Perfilador Jerárquico PL / SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHDCBF) .

**Comentario** (solo perfil): comentario descriptivo que se incluirá en el perfil de ejecución.

**Destino** : nombre de la función o procedimiento para ejecutar o ejecutar en modo de depuración. (Solo tiene una opción si especificó un paquete que tiene más de un subprograma).

**Parámetros** : Lista de cada parámetro para el objetivo especificado. El modo de cada parámetro puede ser IN (el valor se pasa), OUT (el valor se devuelve) o IN / OUT (el valor se pasa, y el resultado de la acción o la acción del procedimiento se almacena en el parámetro).

**Bloque PL / SQL** : Un bloque de código PL / SQL creado por SQL Developer. Debe cambiar las especificaciones formales de los parámetros IN y IN / OUT en este bloque a los valores reales que desea utilizar para ejecutar o depurar la función o el procedimiento.

Por ejemplo, para especificar 10 como el valor para un parámetro de entrada llamado in\_rating, cambie IN\_RATING => IN\_RATINGa IN\_RATING => 10.

Cuando hace clic en **Aceptar** , SQL Developer ejecuta la función o el procedimiento.

Si está depurando una función o un procedimiento, se muestran la barra de herramientas de depuración y una o más ventanas para obtener información relacionada con la depuración, como se explica en [Funciones y procedimientos de ejecución y depuración](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDGIGHA) .

### Crear / Editar punto de interrupción

Use este cuadro para crear o editar un punto de interrupción para usar al depurar una función o procedimiento PL / SQL.

**Pestaña de definición**

Especifique la definición del punto de ruptura.

**Tipo de punto de interrupción** : Tipo de punto de interrupción, lo que indica cuándo se producirá el punto de interrupción. Las opciones incluyen romper cuando ocurre uno de los siguientes: una línea específica de código (Fuente); clase de excepción u otra clase; Método, archivo, o reloj.

**Detalles** del punto de interrupción: las opciones dependen del tipo de punto de interrupción.

**Nombre del** grupo de punto de interrupción: grupo de punto de interrupción en el que se incluye este punto de interrupción. Los grupos de punto de interrupción se pueden editar, habilitar y deshabilitar.

**Pestaña condiciones**

Especifique cualquier condición que se aplique al punto de interrupción.

**Condición** : una condición de SQL (cláusula WHERE sin la palabra clave WHERE) que restringe cuando se produce el punto de interrupción. Por ejemplo, para especificar que la condición debe ocurrir solo cuando status\_code es mayor que 10, especifique:

código de estado> 10

**Opciones de subprocesos** : puede especificar si el punto de interrupción se produce para todos los subprocesos, o solo cuando el punto de interrupción es alcanzado por subprocesos que tienen o no un nombre específico.

**Recuento** de pasadas: el número de veces que el depurador debe permitir que la ejecución pase por el punto de interrupción antes de que se produzca el punto de interrupción.

**Pestaña de acciones**

Especifique las acciones a realizar cuando se produce el punto de interrupción. Las opciones que especifique anulan cualquier valor predeterminado en el panel [Depurador: Puntos de interrupción: Acciones predeterminadas](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDCBBGC) para [las Preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) .

**Detener ejecución** : detiene la ejecución cuando se produce el punto de interrupción.

**Pitido** : emite un **pitido** cuando se produce el punto de interrupción.

**Registro de ocurrencia de punto de** interrupción: envía un mensaje a la ventana de registro cuando se produce el punto de interrupción. También puede especificar que se incluya lo siguiente en cada pantalla: una etiqueta y una condición para evaluar.

**Habilitar / deshabilitar un grupo de** puntos de interrupción: habilita o deshabilita el grupo de puntos de interrupción especificado cuando se produce este punto de interrupción.

### Guardar / Guardar como

Este es un cuadro estándar para guardar información en un archivo: use **Ubicación** para navegar (haciendo doble clic) la carpeta en la que se guardará el archivo, luego especifique el nombre del archivo (incluida cualquier extensión) y, si es necesario, el tipo de archivo.

### Guardar archivos

Este cuadro le pregunta si desea guardar los archivos especificados antes de que ocurra otra acción (por ejemplo, guardar los procedimientos que había estado editando antes de desconectarse).

### No se pueden guardar archivos

Este cuadro le informa que SQL Developer no puede guardar el archivo o archivos especificados. Para cancelar el intento de guardar los archivos y volver a editar el objeto relevante, haga clic en Cancelar.

### Guardar configuración de estilo

Este cuadro de diálogo se muestra al hacer clic en Guardar como en el [Editor de código:](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEEADH) panel de [colores de la sintaxis](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEEADH) al configurar [las preferencias de desarrollador de SQL](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDEDCJD) . Puede guardar la configuración de color especificada como un esquema de color con nombre, que lo agrega a la lista desplegable para **Esquema** en ese panel.

### Diferencias de esquema de origen o error de destino

Este cuadro de error se muestra si hace clic en Aplicar antes de especificar el origen o el destino, o ambos, para una operación de diferencias de esquema.

Haga **clic** en **Aceptar** para cerrar el cuadro de error, luego siga las instrucciones para realizar la operación de diferencias de esquema, como se explica en [Diferencias de esquema de base](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIGDDD) de [datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACIGDDD) .

### Falló la ejecución del script

Este cuadro de error se muestra si la secuencia de comandos generada por el procedimiento de migración rápida falla antes de completar su ejecución. El panel Generar muestra el error que causó el error.

Para cerrar el cuadro de error y abrir el script en una ventana de hoja de cálculo SQL, donde puede editar el texto y ejecutar el script corregido, haga clic en **Sí** ; o para cerrar el cuadro de error sin necesidad de abrir la secuencia de comandos en una ventana de la hoja de SQL, haga clic en **No** .

### Generación de script completa

Este cuadro de información se muestra después de generar el script de control y los archivos relacionados para realizar una captura fuera de línea de una base de datos de terceros, como se explica en [Captura fuera](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDHFDFH) de [línea](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/migration.htm#CHDHFDFH) .

Haga **clic** en **Aceptar** para cerrar el cuadro de error. Luego, ejecute el script de control para generar una salida que contenga el modelo convertido.

### Establecer asignación de datos

Este cuadro de diálogo se muestra si hace clic con el botón derecho en un modelo capturado y selecciona Establecer asignación de datos. Puede usar este cuadro de diálogo para especificar las asignaciones de tipo de datos de origen al migrar la base de datos de terceros especificada a Oracle. Si está editando una asignación existente, puede cambiar solo el tipo de datos de Oracle, la precisión y la información de escala.

**Mostrar solo los tipos de datos utilizados en el modelo de origen** : si marca esta opción, solo se muestran los tipos de datos utilizados en el modelo capturado seleccionado. Si no marca esta opción, se muestran todos los tipos de datos válidos para la base de datos de origen (de terceros).

**Tipo de datos de origen** : Nombre del tipo de datos en la base de datos de terceros.

**Tipo de datos de Oracle** : Nombre del tipo de datos en la base de datos Oracle.

**Tipo** : *Sistema* para un tipo de datos definido por el sistema, o *Usuario* para un tipo de datos definido por el usuario.

**Agregar nueva regla** : muestra el cuadro de diálogo [Agregar / Editar regla](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCHDEH) , para especificar una asignación para otro tipo de datos.

**Editar regla** : muestra el cuadro de diálogo [Agregar / Editar regla](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACCHDEH) , para editar la asignación seleccionada.

**Eliminar regla** : elimina la asignación seleccionada.

### Agregar / Editar Regla

Este cuadro de diálogo se muestra si hace clic en Agregar nueva regla o Editar regla en el cuadro de diálogo [Establecer asignación de datos](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBAJBE) , que se utiliza para especificar asignaciones de tipos de datos de origen al migrar una base de datos de terceros especificada a Oracle.

**Tipo de datos de origen** : Nombre del tipo de datos en la base de datos de terceros.

**Tipo de datos de Oracle** : Nombre del tipo de datos en la base de datos Oracle.

**Precisión** y **escala** : valores de precisión y escala que se utilizarán para el tipo de datos de origen y el tipo de datos de Oracle durante la conversión.

### Establecer codificación

Este cuadro de diálogo se muestra si hace clic con el botón derecho en una conexión CVS y selecciona Establecer codificación. Especifique un juego de caracteres para la conexión. El conjunto de caracteres que elija se aplica a la codificación de archivos bajo el control de CVS a través de esa conexión.

**Plataforma predeterminada (conversiones de nueva línea)** : utiliza el conjunto de caracteres especificado para la plataforma / sistema operativo. Las conversiones de nueva línea para archivos que cruzan diferentes plataformas se manejan automáticamente.

**Configuración global de IDE** : utiliza el conjunto de caracteres predeterminado para el entorno de desarrollo integrado (IDE).

**Otros** : utiliza el conjunto de caracteres seleccionado.

### Establecer pausa Continuar

Este cuadro de diálogo se muestra si ingresa la instrucción SQL \* Plus SET PAUSE ON en la hoja de trabajo de SQL y luego ejecuta el contenido de la hoja de trabajo como una secuencia de comandos. Después de procesar la instrucción SET PAUSE ON, la ejecución se detiene (y se muestra este cuadro de diálogo) después de cada instrucción hasta que se procesa la instrucción SET PAUSE OFF.

Para que la ejecución continúe en la siguiente declaración, haga **clic** en **Aceptar** .

### Iniciar sesión (buscando actualizaciones)

Este cuadro de diálogo se muestra si alguna de las actualizaciones que seleccionó durante el proceso de verificación de actualizaciones se encuentra en un sitio remoto que requiere que inicie sesión. Actualmente, todas las actualizaciones se encuentran en Oracle Technology Network (OTN), por lo que debe ingresar su Nombre de usuario y contraseña de OTN.

**Nombre de usuario** : su nombre de usuario en el sitio remoto.

**Contraseña** : Su contraseña en el sitio remoto.

**Registrarse** : Si no tiene una cuenta en el sitio remoto, haga clic en este enlace.

**Buscar contraseña** : si tiene una cuenta en el sitio remoto pero no puede recordar su contraseña, haga clic en este enlace.

### Vista de un solo registro

El uso principal de este cuadro, que se muestra al hacer clic con el botón derecho en la cuadrícula de visualización de un objeto y al seleccionar Vista de registro único, es editar los datos de una tabla o vista, un registro a la vez. Después de cambiar los datos en cualquier celda de una fila, puede aplicar los cambios haciendo clic en Aplicar o navegando a otro registro. (Para las cuadrículas que no son de datos, las celdas son de solo lectura).

**Iconos de navegación** : Primero (<<) se mueve al primer registro, Anterior (<) se mueve al registro anterior, Siguiente (>) se mueve al siguiente registro, y Último (>) se mueve al último registro.

**Aplicar** : aplica los cambios realizados en el registro de datos actual.

**Cancelar** : cancela los cambios realizados en el registro de datos actual y cierra el cuadro.

### Guardar fragmento (definido por el usuario)

Utilice este cuadro para crear un fragmento de código definido por el usuario. Para obtener información sobre cómo crear fragmentos definidos por el usuario, incluidas las opciones para las categorías de fragmentos, consulte [Fragmentos definidos por el usuario](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/intro.htm#CHDHHIJG) .

**Categoría** : **categoría** existente o nueva en la que colocar el fragmento. Para crear una nueva categoría (definida por el usuario), escriba el nombre de la categoría en lugar de seleccionar un nombre de categoría de la lista.

**Nombre** : nombre del fragmento, ya que se mostrará cuando los usuarios vean la lista de fragmentos disponibles en la categoría especificada. Si un fragmento de código suministrado por Oracle existente tiene el mismo nombre en la misma categoría, la definición de fragmento de código definido por el usuario reemplaza la definición suministrada por Oracle.

**Información sobre herramientas** : texto de información **sobre herramientas** opcional que se mostrará cuando el puntero del mouse permanezca brevemente sobre el nombre del fragmento en la visualización de fragmentos en la categoría especificada.

**Fragmento** : texto que se insertará para este fragmento.

### Editar fragmentos (definidos por el usuario)

Este cuadro muestra cualquier fragmento de código definido por el usuario existente y le permite agregar, editar o eliminar fragmentos de código definido por el usuario.

Para editar un fragmento de código definido por el usuario existente, seleccione su fila y haga clic en el icono **Editar** fragmento de **usuario** , que muestra el cuadro de diálogo [Guardar fragmento (definido por el usuario)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBADJA) .

Para crear un nuevo fragmento definido por el usuario, haga clic en el icono **Agregar** fragmento de **usuario** , que muestra el cuadro de diálogo [Guardar fragmento (definido por el usuario)](https://docs.oracle.com/cd/E11882_01/doc.112/e12152/dialogs.htm#BACBADJA) .

Para eliminar un fragmento de código definido por el usuario, seleccione su fila y haga clic en el icono **Eliminar** fragmento de **usuario** .

### No es posible abrir el archivo

Este cuadro le informa que SQL Developer no puede realizar la operación de exportación a la ubicación y al archivo que especificó. La causa podría ser que no tiene permiso para escribir en esa ubicación.

### Versión de base de datos no compatible

Este cuadro se muestra si intenta crear una conexión a una versión de base de datos que no es compatible con el Desarrollador de SQL, como la versión 8.1 de Oracle Database. Para obtener información sobre las versiones de bases de datos compatibles con SQL Developer, consulte la [*Guía de instalación de Oracle Database SQL Developer*](http://www.oracle.com/pls/topic/lookup?ctx=db112&id=RPTIG) .

### Windows

Este cuadro de diálogo se muestra si hace clic con el botón derecho en la pestaña de una ventana en el área de visualización de la ventana principal de SQL Developer.

**Windows** : una lista de las ventanas en el área de visualización.

**Activar** : Activa (cambia el foco a) la ventana seleccionada.

**Cerrar** : cierra la ventana seleccionada.