

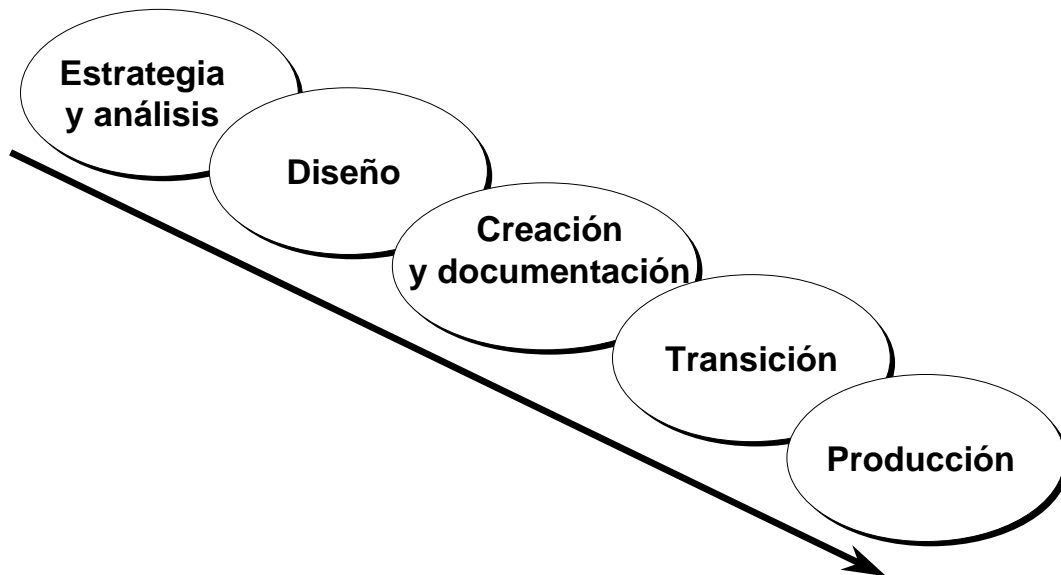
I

Introducción

ORACLE®

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Ciclo de Vida del Desarrollo del Sistema



ORACLE

I-9

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Ciclo de Vida del Desarrollo del Sistema

Desde el concepto hasta la producción, puede desarrollar una base de datos utilizando el ciclo de vida del desarrollo del sistema, que contiene varias etapas de desarrollo. Este enfoque de arriba hacia abajo y sistemático del desarrollo de base de datos transforma los requisitos de información de negocios en una base de datos operativa

Estrategia y Análisis

- Estudie y analice los requisitos de negocio. Entreviste a usuarios y directores para identificar los requisitos de información. Incorpore las declaraciones de objetivos para la empresa y las aplicaciones, así como las especificaciones futuras de sistema.
- Cree modelos del sistema. Transfiera la descripción de negocios a una representación gráfica de necesidades y reglas de la información de negocios. Confirme y refine el modelo con analistas y expertos.

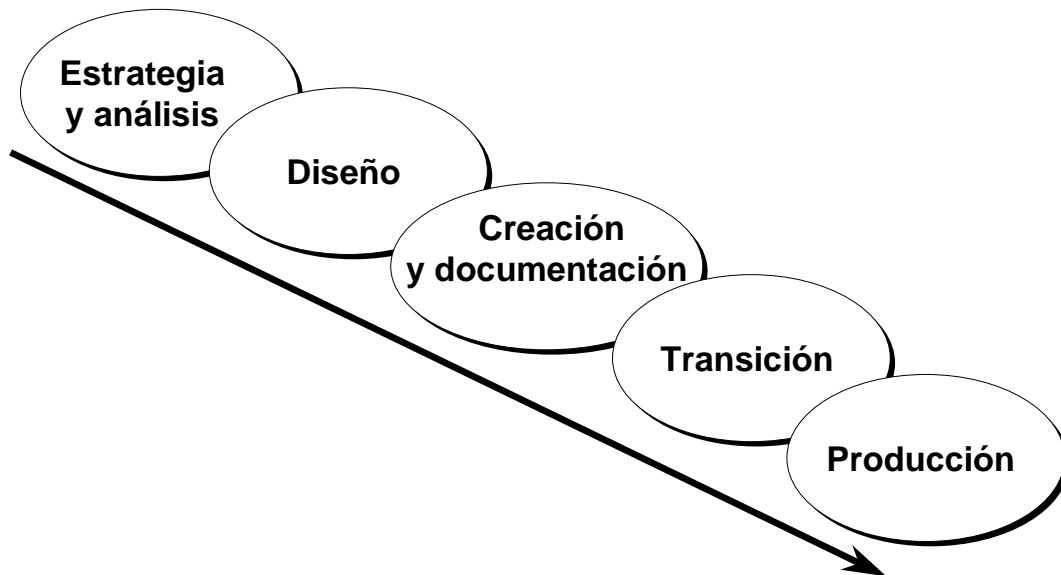
Diseño

Diseñe la base de datos basándose en el modelo desarrollado en la fase de estrategia y análisis.

Creación y Documentación

- Cree el sistema prototipo. Escriba y ejecute los comandos para crear las tablas y objetos de soporte para la base de datos.
- Desarrolle la documentación de usuario, el texto de ayuda y los manuales de operaciones de soporte para el uso y el funcionamiento del sistema.

Ciclo de Vida del Desarrollo del Sistema



ORACLE

I-10

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Ciclo de Vida del Desarrollo del Sistema (continuación)

Transición

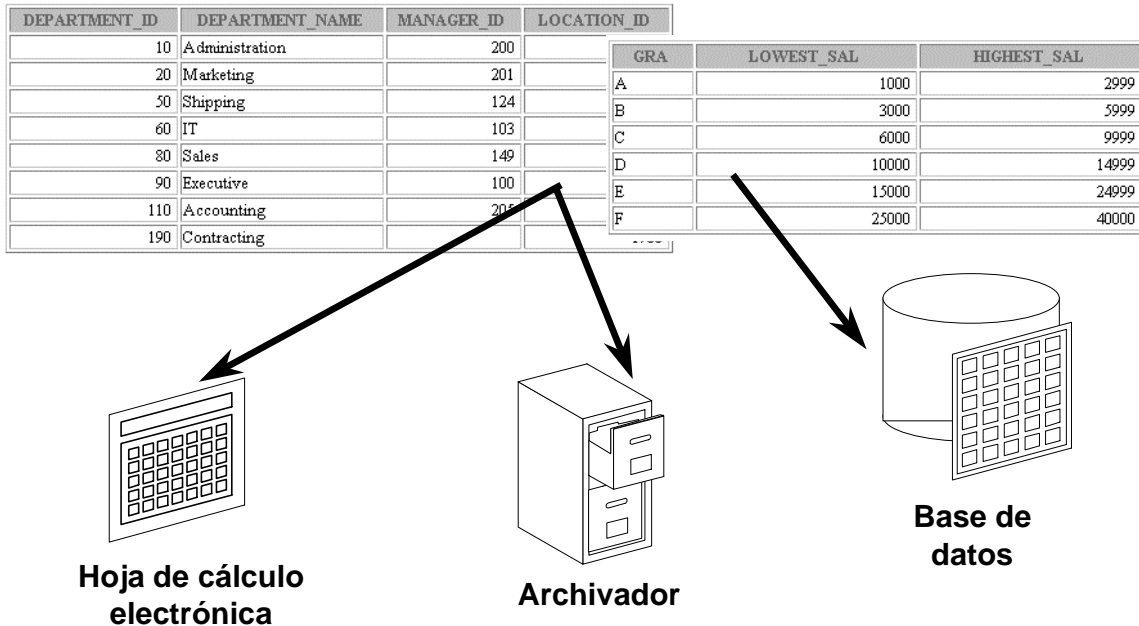
Refine el prototipo. Mueva una aplicación a producción con prueba de aceptación del usuario, conversión de datos existentes y operaciones paralelas. Realice las modificaciones que sean necesarias.

Producción

Transfiera el sistema a los usuarios. Opere el sistema de producción. Controle su rendimiento y mejore y refine el sistema.

Nota: Las diversas fases del ciclo de vida del desarrollo del sistema se pueden llevar a cabo iterativamente. Este curso se centra en la fase de creación del ciclo de vida del desarrollo del sistema.

Almacenamiento de Datos en Diferentes Soportes



ORACLE

I-11

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Almacenamiento de la Información

Todas las organizaciones tienen alguna necesidad de información. Una biblioteca mantiene una lista de miembros, libros, fechas de vencimiento y multas. Una compañía necesita guardar información de sus empleados, departamentos y salarios. Estas informaciones se denominan *datos*.

Las organizaciones pueden almacenar datos en diferentes soportes y en formatos distintos, como un documento impreso en un archivador o datos almacenados en hojas de cálculo electrónicas o en bases de datos.

Una *base de datos* es una recopilación organizada de información.

Para gestionar bases de datos, necesita sistemas de gestión de bases de datos (DBMS). Un DBMS es un programa que almacena, recupera y modifica datos en la base de datos a petición. Hay cuatro tipos principales de bases de datos: *jerárquica*, *de red*, *relacional* y, más recientemente, *relacional de objetos*.

Concepto de Base de Datos Relacional

- **Dr. E.F. Codd propuso el modelo relacional para sistemas de base de datos en 1970.**
- **Es la base para el sistema de gestión de bases de datos relacionales (RDBMS).**
- **El modelo relacional consta de:**
 - **Recopilación de objetos o relaciones**
 - **Juego de operadores que actúan en las relaciones**
 - **Integridad de datos para obtener precisión y consistencia**

ORACLE

I-12

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Modelo Relacional

Dr. E. F. Codd perfiló por primera vez los principios del modelo relacional en un artículo de junio de 1970 titulado “A Relational Model of Data for Large Shared Data Banks” (Un modelo relacional de datos para grandes bancos de datos compartidos). En este documento, Dr. Codd propuso el modelo relacional para sistemas de bases de datos.

Los modelos más populares en aquellos tiempos eran el jerárquico y el de red, o incluso sencillas estructuras de datos de archivo plano. Los sistemas de gestión de bases de datos relacionales (RDBMS) se generalizaron rápidamente, en especial por su facilidad de uso y la flexibilidad de su estructura. Además, muchos proveedores innovadores, como Oracle, agregaron a RDBMS una serie de potentes productos de usuario y de desarrollo de aplicaciones, lo que ofrecía una solución completa.

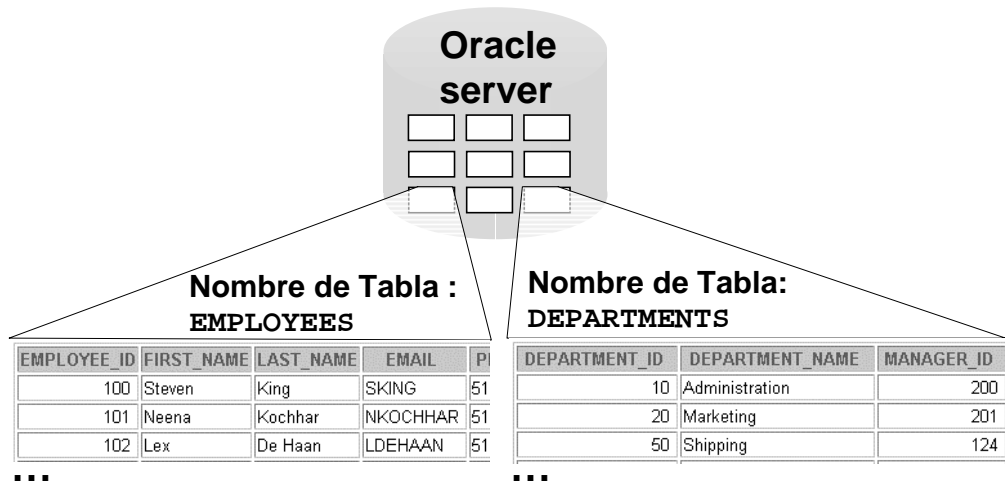
Componentes del Modelo Relacional

- Recopilaciones de objetos o relaciones que almacenan los datos
- Un juego de operadores que pueden actuar en las relaciones para producir otras relaciones
- Integridad de datos para obtener precisión y consistencia

Para obtener más información, consulte E. F. Codd, *The Relational Model for Database Management Version 2* (Reading, Mass.: Addison-Wesley, 1990).

Definición de una Base de Datos Relacional

Una base de datos relacional es una recopilación de relaciones o tablas de dos dimensiones.



ORACLE

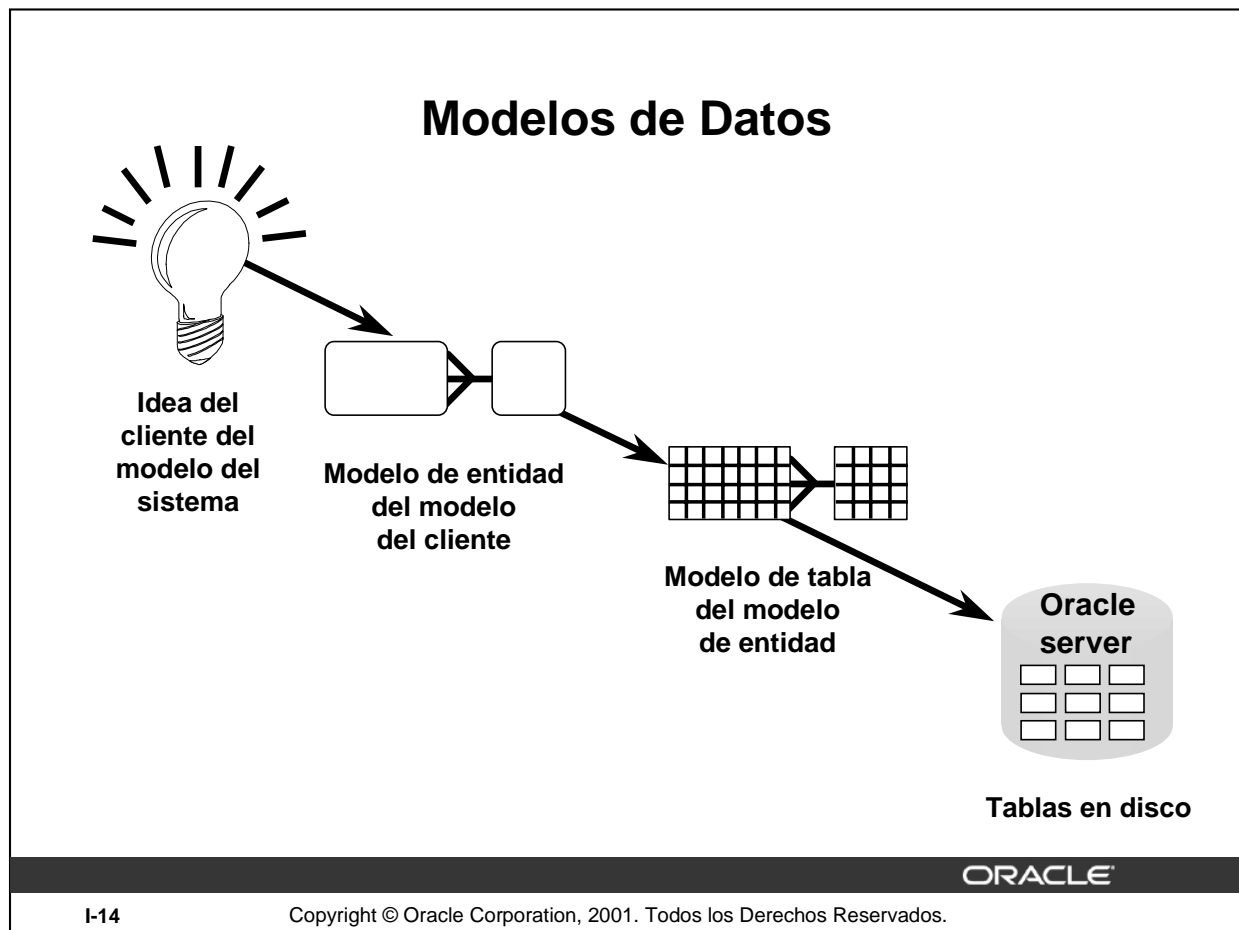
I-13

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Definición de una Base de Datos Relacional

Una base de datos relacional utiliza relaciones o tablas de dos dimensiones para almacenar información.

Por ejemplo, si desea almacenar información acerca de todos los empleados de la compañía, en una base de datos relacional puede crear varias tablas para almacenar diferentes informaciones acerca de los empleados, como por ejemplo, una tabla de empleados, una de departamentos y otra de salarios.



Modelos de Datos

Los modelos son una piedra angular del diseño. Los ingenieros crean un modelo de un automóvil para trabajar en los detalles antes de enviarlo a producción. Del mismo modo, los diseñadores del sistema desarrollan modelos para estudiar ideas y mejorar la comprensión del diseño de base de datos.

Objetivo de los Modelos

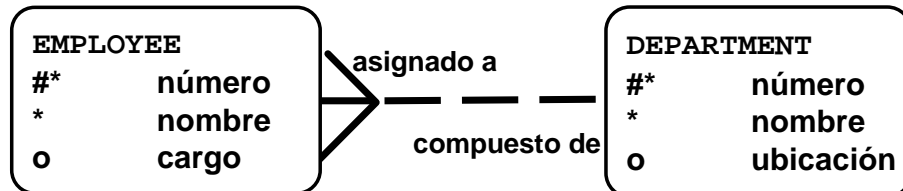
Los modelos ayudan a comunicar los conceptos que las personas tienen en su mente. Se pueden utilizar para:

- Comunicar
- Clasificar
- Describir
- Especificar
- Investigar
- Evolucionar
- Analizar
- Imitar

El objetivo es producir un modelo que se ajuste a una multitud de estos usos, que pueda entenderlo un usuario final y que contenga suficientes detalles para que un desarrollador pueda crear un sistema de bases de datos.

Modelo de Relación entre Entidades

- Cree un diagrama de relación entre entidades a partir de descripciones o especificaciones de negocios



- Supuesto
 - "... Asigne uno o más empleados a un departamento ..."
 - "... Algunos departamentos aún no tienen empleados asignados. ..."

ORACLE

Modelo ER

En un sistema efectivo, los datos se dividen en categorías o entidades discretas. Un modelo de relación entre entidades (ER) es una ilustración de diversas entidades en un negocio y la relación entre ellas. Un modelo ER se deriva de las descripciones o especificaciones de negocios y se crean durante la fase de análisis del ciclo de vida del desarrollo del sistema. Los modelos ER separan la información requerida por un negocio de las actividades realizadas dentro de un negocio. Aunque los negocios pueden cambiar sus actividades, el tipo de información tiende a mantenerse constante. Por lo tanto, las estructuras de datos también tienden a ser constantes.

Ventajas del Modelo ER

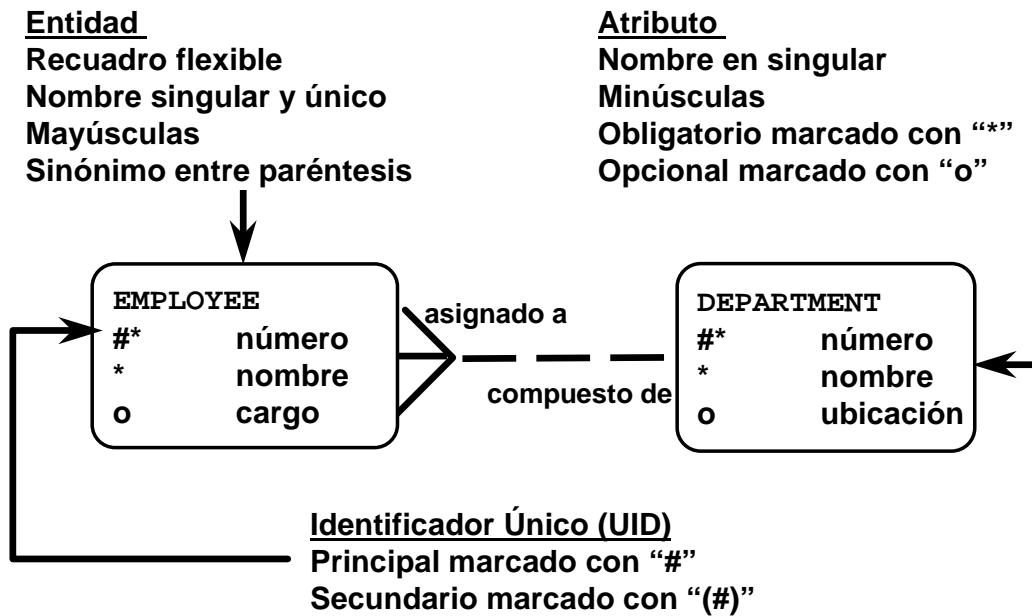
- Documenta información para la organización en un formato claro y preciso.
- Proporciona una imagen clara del ámbito del requisito de información.
- Proporciona un mapa gráfico de fácil comprensión para el diseño de la base de datos.
- Ofrece un marco efectivo para integrar múltiples aplicaciones.

Modelo ER (continuación)

Componentes Clave

- Entidad: Una cosa significativa acerca de la cual se debe tener información, por ejemplo, departamentos, empleados y órdenes.
- Atributo: Algo que describe o cualifica una entidad. Por ejemplo, para la entidad de empleado, los atributos serían el número de empleado, así como el nombre, el cargo, la fecha de contratación, el número de departamento, etc. Cada uno de los atributos es necesario u opcional. Este estado se denomina *opcionalidad*.
- Relación: Asociación especificada entre entidades que muestra la opcionalidad y el grado, por ejemplo, empleados y departamentos, y órdenes y artículos.

Convenciones de Modelo de Relaciones entre Entidades



ORACLE

I-17

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Modelo ER (continuación)

Entidades

Para representar una entidad en un modelo, utilice las siguientes convenciones:

- Recuadro flexible de cualquier dimensión
- Nombre de entidad singular y único
- Nombre de entidad en mayúsculas
- Nombres sinónimos opcionales en mayúsculas y entre paréntesis: ()

Atributos

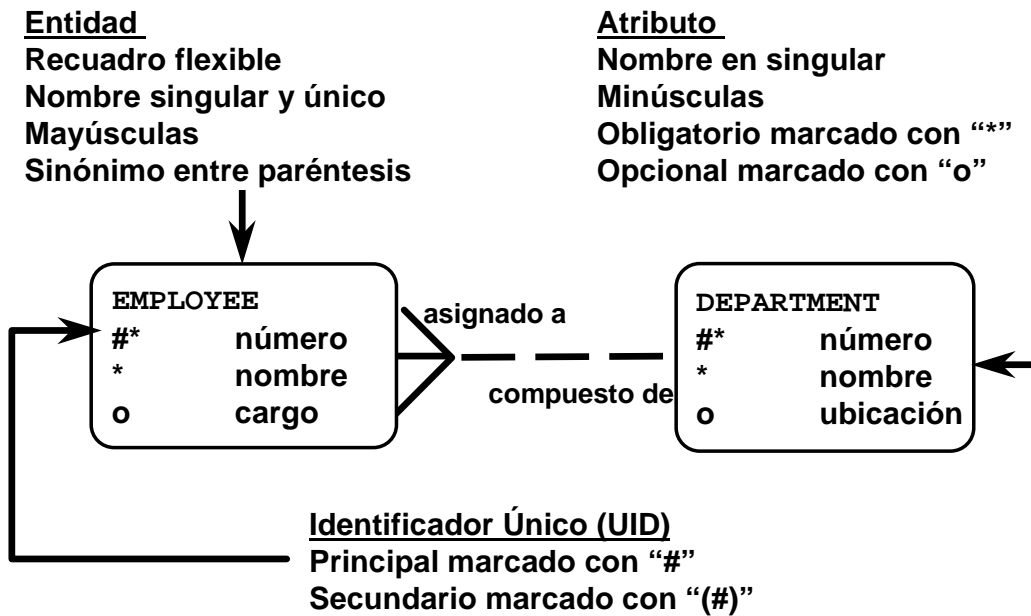
Para representar un atributo en un modelo, utilice las siguientes convenciones:

- Utilice nombres en singular y en minúsculas
- Etiquete los atributos obligatorios, o valores que se deben conocer, con un asterisco: *.
- Etiquete los atributos opcionales, o valores que se pueden conocer, con la letra o.

Relaciones

Símbolo	Descripción
Línea discontinua	Elemento opcional que indica que “puede ser”
Línea continua	Elemento obligatorio que indica que “debe ser”
Pata de gallo	Elemento de grado que indica “uno o más”
Línea sencilla	Elemento de grado que indica “uno y sólo uno”

Convenciones de Modelo de Relaciones entre Entidades



ORACLE

I-18

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Modelo ER (continuación)

Relaciones

Cada dirección de la relación contiene:

- Una etiqueta, por ejemplo, *enseñado* o *asignado a*
- Una opcionalidad, ya sea *debe* o *puede ser*
- Un grado, ya sea *uno y sólo uno* o *uno o más*

Note: El término *cardinalidad* es sinónimo del término *grado*.

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

Nota: La convención se debe leer en el sentido de las agujas del reloj.

Identificadores Únicos

Un identificador único (UID) es cualquier combinación de atributos o relaciones, o de ambos elementos, que sirva para distinguir incidencias de una entidad. Cada incidencia de entidad debe ser identificable de forma única.

- Etiquete cada atributo que sea parte del UID con una almohadilla: #.
- Etiquete los UID secundarios con una almohadilla entre paréntesis: (#).

Relación de Varias Tablas

- Cada fila de datos de una tabla se identifica de forma única con una clave primaria (PK).
- Se pueden relacionar de forma lógica datos de varias tablas utilizando claves ajenas (FK).

Nombre de Tabla: **EMPLOYEES**

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
174	Ellen	Abel	80
142	Curtis	Davies	50
102	Lex	De Haan	90
104	Bruce	Ernst	60
202	Pat	Fay	20
206	William	Gietz	110

Clave primaria

Clave ajena

Nombre de Tabla: **DEPARTMENTS**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

Clave primaria

ORACLE

I-19

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Relación de Varias Tablas

Cada tabla contiene datos que describen exactamente una entidad. Por ejemplo, la tabla **EMPLOYEES** contiene información acerca de los empleados. Las categorías de datos se enumeran horizontalmente en la parte superior de cada tabla, y los casos individuales se enumeran debajo verticalmente.

Utilizando un formato de tabla, se puede visualizar, entender y utilizar fácilmente la información.

Como los datos de entidades diferentes se almacenan en tablas distintas, es posible que necesite combinar dos o más tablas para responder a una pregunta concreta. 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 de los empleados) y de la tabla **DEPARTMENTS** (que contiene información de los departamentos). Con un RDBMS, puede relacionar los datos de una tabla con los de otra utilizando las claves ajenas. Una clave ajena es una columna o un juego de columnas que hacen referencia a la clave primaria de la misma tabla o de otra.

Puede utilizar la capacidad para relacionar los datos de una tabla con los de otra para organizar la información en unidades separadas y manejables. Los datos de los empleados se pueden mantener separados de forma lógica de los datos de los departamentos almacenándolos en una tabla distinta.

Instrucciones para Claves Primarias y Claves Secundarias

- No se pueden duplicar valores en una clave primaria.
- Las claves primarias, por lo general, no se pueden cambiar.
- Las claves ajenas se basan en valores de datos y son punteros meramente lógicos, no físicos.
- El valor de una clave ajena debe coincidir con el valor de una clave primaria o única existente, de lo contrario debe ser nulo.
- Una clave ajena debe hacer referencia a una columna de clave primaria o única.

Introducción a Oracle9i: SQL I-19

Terminología de Bases de Datos Relacionales

②	③	④			
					⑥
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	5800		50
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

ORACLE

I-20

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Terminología Utilizada en una Base de Datos Relacional

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

La transparencia muestra los contenidos de la *tabla* o *relación* EMPLOYEES. Los números indican:

1. Una *fila* o tabla sencilla que representa todos los datos necesarios para un empleado concreto. Cada fila de la tabla debe identificarse con una clave primaria, lo que no permite que existan filas duplicadas. El orden de las filas no es significativo; se especifica al recuperar los datos.
2. Una *columna* o un atributo que contiene el número de empleado. Este número identifica a un empleado *único* de la tabla EMPLOYEES. En este ejemplo, la columna de número de empleado se ha designado como la *clave primaria*. Una clave primaria debe contener un valor y ese valor debe ser único.
3. Una columna que no es un valor clave. Una columna representa un tipo de dato de la tabla: en el ejemplo, el salario de todos los empleados. El orden de las columnas no es significativo al almacenar los datos, se especifica al recuperarlos.
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 entre sí. Una clave ajena hace referencia a una clave primaria o única de la misma tabla o de otra. En el ejemplo, DEPARTMENT_ID identifica *de forma única* un departamento de la tabla DEPARTMENTS.
5. Un campo puede no contener ningún valor. Esto se conoce como valor nulo. En la tabla EMPLOYEES, sólo los empleados con un rol de representante de ventas tienen un valor en el campo COMMISSION_PCT (comisión).
6. Un *campo* se puede encontrar en la intersección de una fila y una columna. Sólo puede contener un valor.

Propiedades de las Bases de Datos Relacionales

Una base de datos relacional:

- Se puede acceder a ella y modificarla ejecutando sentencias de lenguaje estructurado de consulta (SQL).
- Contiene una recopilación de tablas sin punteros físicos.
- Utiliza un juego de operadores.

ORACLE

I-21

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Propiedades de una Base de Datos Relacional

En una base de datos relacional, no se especifica la ruta de acceso a las tablas y no es necesario conocer la organización física de los datos.

Para acceder a la base de datos, se ejecuta una sentencia de lenguaje estructurado de consulta (SQL), que es el lenguaje estándar ANSI para bases de datos relacionales operativas. El lenguaje contiene un amplio juego de operadores para particionar y combinar relaciones. La base de datos se puede modificar utilizando las sentencias SQL.

Comunicación con un RDBMS Utilizando SQL

**SQL statement
is entered.**

```
SELECT department_name  
FROM departments;
```

**Se envía la sentencia
a Oracle Server.**

**Oracle
server**

DEPARTMENT_NAME
Administration
Marketing
Shipping
IT
Sales
Executive
Accounting
Contracting

ORACLE

I-22

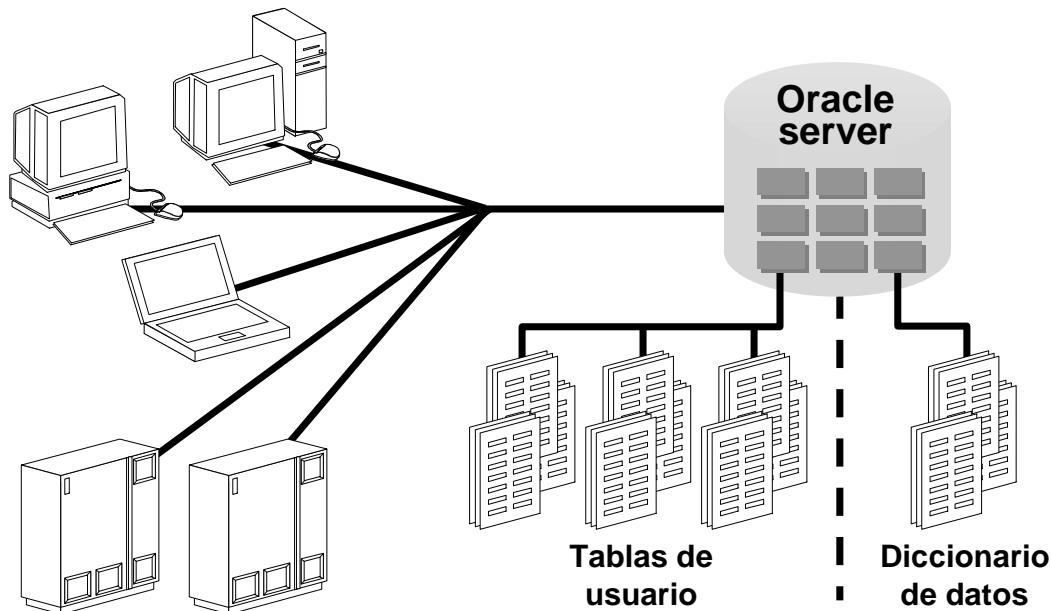
Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Lenguaje Estructurado de Consulta

Al utilizar SQL, se puede comunicar con Oracle Server. SQL tiene las siguientes ventajas:

- Es eficiente.
- Es fácil de aprender y de usar.
- Es funcionalmente completo (con SQL, puede definir, recuperar y manipular datos en las tablas).

Sistema de Gestión de Bases de Datos Relacionales



I-23

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Sistema de Gestión de Bases de Datos Relacionales

Oracle Server ofrece un RDBMS flexible llamado Oracle9i. Al utilizar sus funciones, puede almacenar y gestionar datos con todas las ventajas de la estructura relacional, más PL/SQL, un motor que le proporciona la capacidad de almacenar y ejecutar unidades de programa. Oracle9i también soporta Java y XML. Oracle Server ofrece las opciones de recuperar datos basándose en técnicas de optimización. Incluye funciones de seguridad que controlan la forma de acceso y de utilización de una base de datos. Entre otras funciones se encuentran la consistencia y la protección de los datos mediante mecanismos de bloqueo.

Oracle9i Server proporciona un enfoque abierto, global e integrado a la gestión de información. Oracle Server consta de instancia de Oracle Server y una base de datos Oracle. Cada vez que se inicia una base de datos, se asigna un área global del sistema (SGA) y se inician los procesos en segundo plano de Oracle. El área global del sistema es un área de memoria utilizada para la información de base de datos compartida por los usuarios de la base de datos. La combinación de los procesos en segundo plano y los buffers de memoria se conoce como instancia Oracle.

Sentencias SQL

SELECT	Recuperación de datos
INSERT UPDATE DELETE MERGE	Lenguaje de manipulación de datos (DML)
CREATE ALTER DROP RENAME TRUNCATE	Lenguaje de definición de datos (DDL)
COMMIT ROLLBACK SAVEPOINT	Control de transacciones
GRANT REVOKE	Lenguaje de control de datos (DCL)

ORACLE

I-24

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Sentencias SQL

Oracle SQL cumple con los estándares aceptados por la industria. Oracle Corporation asegura el cumplimiento futuro con los estándares en desarrollo implicando a personal clave en los comités de estándares SQL. Los comités aceptados por la industria son el Instituto Nacional Americano de Normalización (ANSI) y la Organización Internacional de Normalización (ISO). Tanto ANSI como ISO han aceptado SQL como lenguaje estándar para bases de datos relacionales.

Sentencia	Description
SELECT	Recupera datos de la base de datos.
INSERT UPDATE DELETE MERGE	Introduce nuevas filas, cambia filas existentes y elimina las filas no deseadas de las tablas de la base de datos, respectivamente. Se conocen de forma conjunta como <i>lenguaje de manipulación de datos (DML)</i> .
CREATE ALTER DROP RENAME TRUNCATE	Configuran, cambian y eliminan estructuras de datos de las tablas. Se conocen de forma conjunta como <i>lenguaje de definición de datos (DDL)</i> .
COMMIT ROLLBACK SAVEPOINT	Gestionan los cambios realizados a las sentencias DML. Los cambios en los datos se pueden agrupar en transacciones lógicas.
GRANT REVOKE	Otorga o elimina derechos de acceso a la base de datos Oracle y las estructuras contenidas en ella. Se conocen de forma conjunta como <i>lenguaje de control de datos (DCL)</i> .

Tablas Utilizadas en el Curso

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	240
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	170
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	170
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	42
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	58
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	35
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	31
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	26

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

DEPARTMENTS

0.121.2004	09-JUL-98	ST_CLERK	25
1.44.1344.429018	29-JAN-00	SA_MAN	105
1.44.1644.429067	11-MAY-96	SA_DEP	110

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

JOB_GRADES

ORACLE

Tablas Utilizadas en el Curso

En este curso se utilizan las siguientes tablas principales:

- La tabla EMPLOYEES, que ofrece detalles de todos los empleados.
- La tabla DEPARTMENTS, que ofrece detalles de todos los departamentos.
- La tabla JOB_GRADES, que ofrece detalles de los salarios para distintas categorías.

Nota: La estructura y los datos para todas las tablas se proporcionan en el Apéndice B.

Resumen

- **Oracle9i Server es la base de datos para computación en Internet.**
- **Oracle9i se basa en el sistema de gestión de bases de datos relacionales de objetos.**
- **Las bases de datos relacionales se componen de relaciones, se gestionan con operaciones relacionales y se rigen por restricciones de integridad de datos.**
- **Con Oracle Server, puede almacenar y gestionar información utilizando el lenguaje SQL y el motor PL/SQL.**

ORACLE

I-26

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

Resumen

Los sistemas de gestión de bases de datos relacionales se componen de objetos o relaciones. Se gestionan con operaciones y se rigen por restricciones de integridad de datos.

Oracle Corporation crea productos y servicios que cubren las necesidades de sistemas de gestión de bases de datos relacionales. Los productos principales son: Oracle9i Database Server, con el que puede almacenar y gestionar información utilizando SQL y Oracle9i Application Server con el que puede ejecutar todas las aplicaciones.

SQL

Oracle Server soporta SQL del estándar ANSI y contiene extensiones. SQL es el lenguaje utilizado para comunicarse con el servidor para acceder, manipular y controlar datos.

1

Escritura de Sentencias SQL SELECT Básicas

ORACLE®

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Enumerar las capacidades de las sentencias SQL `SELECT`**
- **Ejecutar una sentencia `SELECT` básica**
- **Diferenciar entre sentencias SQL y comandos `iSQL*Plus`**

ORACLE

1-2

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Objetivo de la Lección

Para extraer datos de la base de datos, es necesario utilizar la sentencia `SELECT` de lenguaje estructurado de consulta (SQL). Puede tener que restringir las columnas que se muestran. Esta lección describe todas las sentencias SQL necesarias para realizar estas acciones.

Es posible que desee crear sentencias `SELECT` que se puedan utilizar más de una vez. Esta lección también trata el entorno `iSQL*Plus` donde puede ejecutar sentencias SQL.

Nota: `iSQL*Plus` es un nuevo producto de Oracle9i. Se trata de un entorno de explorador donde se ejecutan comandos SQL. En versiones anteriores de Oracle, `SQL*Plus` era el entorno por defecto donde se ejecutaban los comandos SQL. `SQL*Plus` todavía está disponible y se describe en el Apéndice C.

Capacidades de las Sentencias SQL SELECT

Proyección

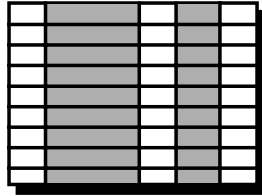


Tabla 1

Selección

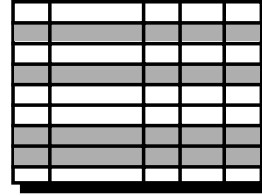


Tabla 1

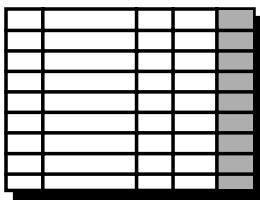


Tabla 1

Unión

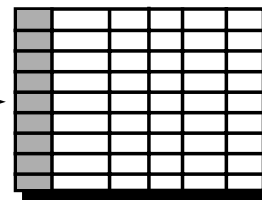


Tabla 2

ORACLE

Capacidades de las Sentencias SQL SELECT

Una sentencia `SELECT` recupera información de la base de datos. Mediante una sentencia `SELECT`, puede hacer lo siguiente:

- **Proyección:** Puede utilizar la capacidad de proyección de SQL para seleccionar las columnas de una tabla que desee que su consulta le devuelva. Puede elegir tantas columnas de la tabla como sea necesario.
- **Selección:** Puede utilizar la capacidad de selección de SQL para seleccionar las filas de una tabla que desee que una consulta le devuelva. Puede usar diferentes criterios para restringir las filas visibles.
- **Unión:** Puede utilizar la capacidad de unión de SQL para recopilar datos que están almacenados en diferentes tablas creando un enlace entre ellos. Aprenderá más sobre uniones en una lección posterior.

Sentencias SELECT Básicas

```
SELECT    *|{[DISTINCT] column|expression [alias],...}  
FROM      table;
```

- **SELECT** identifica las columnas *what*
- **FROM** identifica la tabla *which*

ORACLE

1-4

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Sentencias SELECT Básicas

En su forma más simple, una sentencia SELECT debe incluir lo siguiente:

- Una cláusula SELECT, que especifica las columnas que se han de mostrar
- Una cláusula FROM, que especifica la tabla que contiene las columnas listadas 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 expresión o columna especificada
alias	asigna cabeceras diferentes a las columnas seleccionadas
FROM table	especifica la tabla que contiene las columnas

Nota: A lo largo de este curso, las palabras *cláusula*, *sentencia* y *palabra clave* se usan según lo siguiente:

- Una *palabra clave* hace referencia a un elemento SQL individual.
Por ejemplo, SELECT y FROM son palabras clave.
- Una *cláusula* es una parte de una sentencia SQL.
Por ejemplo, SELECT employee_id, last_name, ... es una cláusula.
- Una *sentencia* es una combinación de dos o más cláusulas.
Por ejemplo, SELECT * FROM employees es una sentencia SQL.

Introducción a Oracle9i: SQL 1-4

Selección de Todas las Columnas

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

ORACLE

Selección de Todas las Columnas de Todas las Filas

Puede visualizar todas las columnas de datos en una tabla si escribe un asterisco (*) detrás de la palabra clave **SELECT**. En el ejemplo de la transparencia, la tabla **DEPARTMENT** contiene cuatro columnas: **DEPARTMENT_ID**, **DEPARTMENT_NAME**, **MANAGER_ID** y **LOCATION_ID**. La tabla contiene siete filas, una para cada departamento.

También puede visualizar todas las columnas de la tabla si enumera todas las columnas después de la palabra clave **SELECT**. Por ejemplo, la siguiente sentencia SQL, al igual que el ejemplo de la transparencia, muestra todas las columnas y filas de la tabla **DEPARTMENTS**:

```
SELECT department_id, department_name, manager_id, location_id  
FROM departments;
```


Selección de Columnas Específicas

```
SELECT department_id, location_id
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

ORACLE

Selección de Columnas Específicas de Todas las Filas

Puede utilizar la sentencia `SELECT` para visualizar columnas específicas de la tabla si indica los nombres de columna, separados por comas. El ejemplo de la transparencia muestra todos los números de departamento y de ubicación de la tabla de `DEPARTMENTS`.

En la cláusula `SELECT`, especifique las columnas que desee en el orden en que quiera que aparezcan en el resultado. Por ejemplo, para mostrar la ubicación, antes que el número de departamento, de izquierda a derecha, utilice la siguiente sentencia:

```
SELECT location_id, department_id
FROM departments;
```

LOCATION_ID	DEPARTMENT_ID
1700	10
1800	20
1500	50

...

8 rows selected.

Escritura de Sentencias SQL

- Las sentencias SQL no son sensibles a mayúsculas/minúsculas.
- Las sentencias SQL pueden ocupar una o más líneas.
- Las palabras clave no se pueden abreviar ni dividir entre líneas.
- Las cláusulas suelen estar colocadas en líneas separadas.
- Los sangrados se utilizan para mejorar la legibilidad.

ORACLE

1-7

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Escritura de Sentencias SQL

Utilizando las sencillas reglas e instrucciones siguientes, puede construir sentencias válidas que sean fáciles tanto de leer como de 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 ni abreviar.
- Normalmente las cláusulas están colocadas en líneas separadas por motivos de legibilidad y facilidad de edición.
- Los sangrados se deben utilizar para que los códigos sean más legibles.
- Generalmente, las palabras clave se introducen en mayúsculas; todas las demás palabras, como los nombres de tabla y columnas se introducen en minúsculas.

Ejecución de Sentencias SQL

Mientras utiliza *iSQL*Plus*, haga clic en el botón **Execute** para ejecutar el comando o los comandos en la ventana.

Valores por Defecto de Cabecera de Columna

- **iSQL*Plus:**
 - Justificación de cabecera por defecto: Centro
 - Visualización de cabecera por defecto: Mayúsculas
- **SQL*Plus:**
 - Las cabeceras de la columna de fecha y carácter están justificadas a la izquierda
 - Las cabeceras de columna de número están justificadas a la derecha
 - Visualización de cabecera por defecto: Mayúsculas

ORACLE

1-8

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Valores por Defecto de Cabecera de Columna

En iSQL*Plus, las cabeceras de columna se muestran en mayúsculas y centradas.

```
SELECT last_name, hire_date, salary
FROM employees;
```

LAST_NAME	HIRE_DATE	SALARY
King	17-JUN-87	24000
Kochhar	21-SEP-89	17000
De Haan	13-JAN-93	17000
Hunold	03-JAN-90	9000
Ernst	21-MAY-91	6000
■ ■ ■		
Higgins	07-JUN-94	12000
Gietz	07-JUN-94	8300

20 rows selected.

Puede sustituir la presentación de la cabecera de columna por un alias. Posteriormente, en esta lección se tratan los alias de columna.

Expresiones Aritméticas

Cree expresiones con datos de número y fecha mediante operadores aritméticos.

Operador	Descripción
+	Suma
-	Resta
*	Multipliación
/	División

ORACLE

1-9

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Expresiones Aritméticas

Puede que necesite modificar la forma en la que se muestra la información, realizar cálculos, o examinar supuestos hipotéticos. Todo ello es posible si se utilizan las expresiones aritméticas. Una expresión aritmética puede contener nombres de columna, valores numéricos constantes y operadores aritméticos.

Operadores Aritméticos

La transparencia enumera los operadores aritméticos disponibles en SQL. Puede utilizar operadores aritméticos en cualquier cláusula de una sentencia SQL excepto en la cláusula FROM.

Uso de Operadores Aritméticos

```
SELECT last_name, salary, salary + 300
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300
...		
Hartstein	13000	13300
Fay	6000	6300
Higgins	12000	12300
Gietz	8300	8600

20 rows selected.

ORACLE

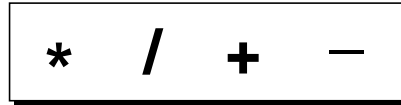
Uso de Operadores Aritméticos

El ejemplo de la transparencia utiliza el operador de suma para calcular un incremento del salario de \$300 para todos los empleados y muestra una nueva columna `SALARY+300` en el resultado.

Observe que la columna calculada resultante `SALARY+300` no es una columna nueva de la tabla `EMPLOYEES`, sino sólo de visualización. Por defecto, el nombre de una columna nueva surge del cálculo que la generó, en este caso, `salary+300`.

Nota: El servidor Oracle9i ignora los espacios en blanco anteriores y posteriores al operador aritmético.

Prioridad de Operador



- La multiplicación y la división tienen prioridad sobre la suma y la resta.
- Los operadores de idéntica prioridad se evalúan de izquierda a derecha.
- Los paréntesis se utilizan para forzar evaluaciones prioritarias y para clarificar sentencias.

ORACLE

Prioridad de Operador

Si una expresión aritmética contiene más de un operador, la multiplicación y la división se evalúan en primer lugar. Si los operadores incluidos en una expresión son de idéntica prioridad, entonces la evaluación se hace de izquierda a derecha.

Puede utilizar paréntesis para forzar que la expresión incluida entre paréntesis se evalúe en primer lugar.

Prioridad de Operador

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Hunold	9000	108100
Ernst	6000	72100

...

Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100
Gietz	8300	99700

20 rows selected.

ORACLE

Prioridad de Operador (continuación)

El ejemplo de la transparencia muestra el apellido, el salario y la remuneración anual de los empleados. La remuneración anual se calcula como 12 multiplicado por el salario mensual, más una bonificación única de \$100. Observe que la multiplicación se realiza antes que la suma.

Nota: Utilice paréntesis para reforzar el orden de prioridad estándar y para mejorar la claridad. Por ejemplo, la expresión de la transparencia se puede escribir como $(12 * salary) + 100$ sin que haya ningún cambio en el resultado.

Uso de Paréntesis

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Hunold	9000	109200
Ernst	6000	73200

• • •

Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200
Gietz	8300	100800

20 rows selected.

ORACLE

Uso de Paréntesis

Puede sustituir las reglas de prioridad utilizando paréntesis para especificar el orden de ejecución de los operadores.

El ejemplo de la transparencia muestra el apellido, el salario y la remuneración anual de los empleados. La remuneración anual se calcula como salario mensual más una bonificación mensual de \$100 multiplicado por 12. Debido a los paréntesis, la suma tiene prioridad sobre la multiplicación.

Definición de un Valor Nulo

- Un valor nulo es un valor no disponible, no asignado, desconocido, o no aplicable.
- Un valor nulo no es lo mismo que cero ni que un espacio en blanco.

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

ORACLE

Valores Nulos

Si una fila no tiene el valor de datos de una columna determinada, se dice que ese valor es *nulo*, o que contiene un null.

Un valor nulo es un valor no disponible, no asignado, desconocido, o no aplicable. Un valor nulo no es lo mismo que cero ni que un espacio. Cero es un número y un espacio 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 el director de ventas o el representante de ventas puede percibir una comisión. Los demás empleados no tienen derecho a ganar comisiones. Un valor nulo representa este hecho.

Valores Nulos en Expresiones Aritméticas

Las expresiones aritméticas que contienen un valor nulo tienen una evaluación nula.

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
...	
Zlotkey	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

ORACLE

Valores Nulos (continuación)

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

En el ejemplo de la transparencia, el empleado King no obtiene ninguna comisión. Debido a que la columna COMMISSION_PCT de la expresión aritmética es nula, el resultado es null.

Para obtener más información, consulte *Oracle9i SQL Reference*, “Basic Elements of SQL”

Definición de un Alias de Columna

Un alias de columna:

- **Cambia el nombre de una cabecera de columna.**
- **Resulta útil con cálculos.**
- **Se sitúa inmediatamente detrás del nombre de la columna, también puede existir la palabra clave opcional `AS` entre el nombre de la columna y el alias.**
- **Requiere comillas dobles si contiene espacios, caracteres especiales o si es sensible a mayúsculas/minúsculas.**

ORACLE

Alias de Columna

Cuando muestra el resultado de una consulta, `iSQL*Plus` normalmente utiliza el nombre de la columna seleccionada como la cabecera de columna. Es posible que esta cabecera no sea descriptiva y por tanto, sea difícil de comprender. Puede cambiar la cabecera de una columna mediante un alias de columna.

Especifique el alias tras la columna en la lista `SELECT` utilizando un espacio como separador. Por defecto, las cabeceras de alias aparecen en mayúsculas. Si el alias contiene espacios o caracteres especiales (como `#` o `$`), o es sensible a mayúsculas/minúsculas, escriba el alias entre comillas dobles (" ").

Uso de Alias de Columna

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

20 rows selected.

ORACLE

Alias de Columna (continuación)

El primer ejemplo muestra los nombres y los porcentajes de las comisiones de todos los empleados. Observe que la palabra clave opcional AS se ha usado delante del nombre alias de columna. El resultado de la consulta es el mismo se utilice la palabra clave AS o no. Observe también que la sentencia SQL tiene los nombres, las comisiones y los alias de columna en minúsculas, mientras que el resultado de la consulta muestra la cabecera de columna en mayúsculas. Como se mencionó en una transparencia anterior, las cabeceras de columna aparecen en mayúsculas por defecto.

El segundo ejemplo muestra los apellidos y los salarios anuales de todos los empleados. Debido a que Annual Salary contiene un espacio, se ha escrito entre comillas dobles. Observe que la cabecera de columna en el resultado es exactamente igual que el alias de columna.

Operador de Concatenación

Un operador de concatenación:

- **Concatena columnas o cadenas de caracteres a otras columnas.**
- **Está representado por dos barras verticales (||).**
- **Crea una columna resultante que es una expresión de caracteres.**

ORACLE

Operador de Concatenación

Puede enlazar columnas a otras columnas, expresiones aritméticas o valores constantes para crear una expresión de caracteres mediante el operador de concatenación (||). Las columnas a cada lado del operador se combinan para hacer una única columna de resultados.

Uso del Operador de Concatenación

```
SELECT last_name||job_id AS "Employees"  
FROM employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
HunoldIT_PROG
ErnstIT_PROG
LorentzIT_PROG
MourgosST_MAN
RajsST_CLERK

...
20 rows selected.

ORACLE

Operador de Concatenación (continuación)

En el ejemplo, LAST_NAME y JOB_ID están concatenados y se les ha asignado el alias Employees. Observe que el código del puesto y el apellido del empleado están combinados para formar una única columna de resultados.

La palabra clave AS delante del alias hace que la cláusula SELECT sea más fácil de leer.

Cadenas de Caracteres Literales

- Un literal es un carácter, un número o una fecha incluida en la lista `SELECT`.
- Los valores literales de caracteres y fecha se deben escribir entre comillas simples.
- Cada cadena de caracteres tiene una salida para cada fila devuelta.

ORACLE

1-20

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Cadenas de Caracteres Literales

Un literal es un carácter, un número o una fecha que está incluido en la lista `SELECT` y que no es un nombre de columna o un alias de columna. Un literal está impreso para cada fila devuelta. Las cadenas literales de texto de formato libre se pueden incluir en el resultado de la consulta y son tratados de la misma manera que una columna en la lista `SELECT`.

Mientras que los literales de caracteres y fecha *se deben* escribir entre comillas simples (' '), los literales numéricos no.

Uso de Cadenas de Caracteres Literales

```
SELECT last_name || ' is a ' || job_id
       AS "Employee Details"
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

■ ■ ■
20 rows selected.

ORACLE

Cadenas de Caracteres Literales (continuación)

El ejemplo de la transparencia muestra los códigos de los cargos y los apellidos de todos los empleados. La columna tiene la cabecera Employee Details. Observe los espacios entre las comillas simples en la sentencia SELECT. Los espacios mejoran la legibilidad del resultado.

En el siguiente ejemplo, el apellido y el salario de cada empleado están concatenados con un literal para dotar de más significado a las filas devueltas.

```
SELECT last_name || ': 1 Month salary = ' || salary Monthly
FROM   employees;
```

MONTHLY
King: 1 Month salary = 24000
Kochhar: 1 Month salary = 17000
De Haan: 1 Month salary = 17000
Hunold: 1 Month salary = 9000
Ernst: 1 Month salary = 6000
Lorentz: 1 Month salary = 4200
Mourgos: 1 Month salary = 5800
Rajs: 1 Month salary = 3500

■ ■ ■
20 rows selected.

Filas Duplicadas

La visualización por defecto de las consultas son todas las filas, incluidas las filas duplicadas.

```
SELECT department_id  
FROM employees;
```

DEPARTMENT_ID	
	90
	90
	90
	60
	60
	60
	50
	50
	50

...

20 rows selected.

ORACLE

Filas Duplicadas

A menos que indique lo contrario, *iSQL*Plus* muestra los resultados de una consulta sin eliminar filas duplicadas. El ejemplo de la transparencia muestra todos los números de departamento de la tabla `EMPLOYEES`. Observe que los números de departamento están repetidos.

Eliminación de Filas Duplicadas

Elimine filas duplicadas mediante la palabra clave DISTINCT de la cláusula SELECT.

```
SELECT DISTINCT department_id
FROM employees;
```

DEPARTMENT_ID	
	10
	20
	50
	60
	80
	90
	110

8 rows selected.

ORACLE

1-23

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Filas Duplicadas (continuación)

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

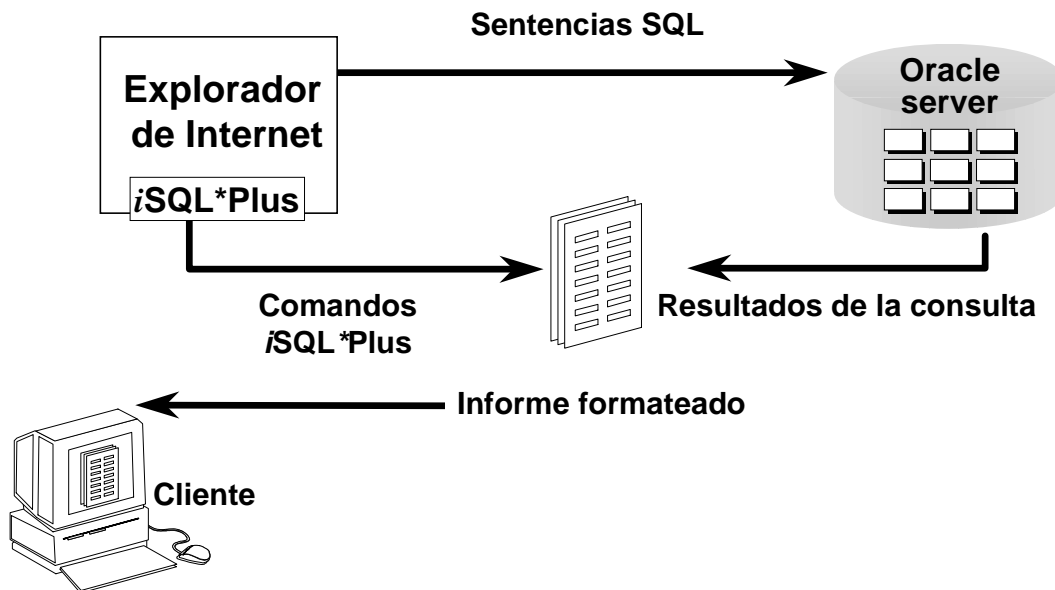
Puede especificar múltiples columnas detrás del cualificador DISTINCT. Este cualificador afecta a todas las columnas seleccionadas y el resultado es cada distinta combinación de columnas.

```
SELECT DISTINCT department_id, job_id
FROM employees;
```

DEPARTMENT_ID	JOB_ID
10	AD_ASST
20	MK_MAN
20	MK_REP
50	ST_CLERK
50	ST_MAN
60	IT_PROG
...	
	SA_REP

13 rows selected.

Interacción de SQL y iSQL*Plus



ORACLE

1-24

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

SQL e iSQL*Plus

SQL es un lenguaje de comandos para comunicar con Oracle Server desde cualquier herramienta o aplicación. Oracle SQL contiene muchas extensiones.

*iSQL*Plus* es una herramienta de Oracle que reconoce y ejecuta sentencias SQL en Oracle Server y que contiene su propio lenguaje de comandos.

Funciones de SQL

- Puede ser utilizado por distintos usuarios, incluidos aquellos con poca o ninguna experiencia en programación
- Es un lenguaje no procedural
- Reduce la cantidad de tiempo requerido para crear y mantener sistemas
- Es un lenguaje propio del inglés

Funciones de iSQL*Plus

- Se accede a él desde un explorador
- Acepta entradas ad hoc de sentencias
- Proporciona edición en línea para la modificación de sentencias SQL
- Controla definiciones de entorno
- Formatea los resultados de las consultas en un informe básico
- Accede a bases de datos locales y remotas

Sentencias SQL frente a Comandos *iSQL*Plus*

SQL

- Lenguaje
- Estándar ANSI
- Las palabras clave no se pueden abreviar
- Las sentencias manipulan definiciones de tabla y datos en la base de datos

**Sentencias
SQL**

*iSQL*Plus*

- Entorno
- Propiedad de Oracle
- Las palabras clave se pueden abreviar
- Los comandos no permiten la manipulación de valores en la base de datos
- Se ejecuta en un explorador
- Al estar cargado de forma centralizada, no se tiene que implementar en cada máquina

**Comandos
*iSQL*Plus***

ORACLE

1-25

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

SQL e *iSQL*Plus* (continuación)

La siguiente tabla compara SQL e *iSQL*Plus*:

SQL	<i>iSQL*Plus</i>
Es un lenguaje para comunicar con Oracle Server y acceder a datos.	Reconoce sentencias SQL y las envía al servidor.
Está basado en el SQL estándar American National Standards Institute (ANSI)	Es la interfaz propiedad de Oracle de ejecución de sentencias SQL.
Manipula definiciones de tabla y datos en la base de datos.	No permite la manipulación de valores en la base de datos.
No tiene carácter de continuación.	Tiene un guión (-) como carácter de continuación si el comando no es más largo de una línea.
No se puede abreviar.	Se puede abreviar.
Utiliza funciones para realizar algunos formatos.	Utiliza comandos para formatear datos.

Visión General de *iSQL*Plus*

Después de conectarse a *iSQL*Plus*, puede:

- **Describir la estructura de tabla.**
- **Editar la sentencia SQL.**
- **Ejecutar SQL desde *iSQL*Plus*.**
- **Guardar sentencias SQL en archivos y agregar sentencias SQL a archivos.**
- **Ejecutar sentencias almacenadas en archivos guardados.**
- **Cargar comandos desde un archivo de texto a la ventana *iSQL*Plus* Edit.**

ORACLE[®]

1-26

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

iSQL*Plus

*iSQL*Plus* es un entorno en el cual puede:

- Ejecutar sentencias SQL para recuperar, modificar, agregar y eliminar datos de la base de datos.
- Formatear, realizar cálculos, almacenar e imprimir los resultados de las consultas en forma de informes.
- Crear archivos de comandos para almacenar sentencias SQL y poder volver a utilizarlos en el futuro.

Los comandos *iSQL*Plus* se pueden dividir en las siguientes categorías principales:

Categoría	Objetivo
Entorno	Afecta al comportamiento general de las sentencias SQL de la sesión.
Formato	Formatea resultados de consultas.
Manipulación de archivos	Guarda sentencias en archivos de comandos de texto y ejecuta sentencias desde archivos de comandos de texto.
Ejecución	Envía sentencias SQL desde el explorador a Oracle Server.
Editar	Modifica sentencias SQL en la ventana Edit.
Interacción	Le permite crear y transferir variables a sentencias SQL, imprimir valores de variables e imprimir mensajes en la pantalla.
Varios	Tiene distintos comandos para conectar con la base de datos, manipular el entorno <i>iSQL*Plus</i> y mostrar definiciones de columna.

Conexión a iSQL*Plus

Desde el entorno del explorador de Windows:

The screenshot shows a Netscape browser window with the following elements:

- Browser toolbar: Back, Forward, Reload, Home, Search, Netscape, Print, Security, Shop, Stop.
- Address bar: Go to: <http://hgxxxx-lap1.xxoracle.com/isqlplus>
- Page content:
 - Oracle logo and "iSQL*Plus" text.
 - Username:
 - Password:
 - Connection Identifier:
 - Privilege: User (dropdown menu)
 - Log In and Clear buttons.
 - Help link (question mark icon).

ORACLE®

1-27

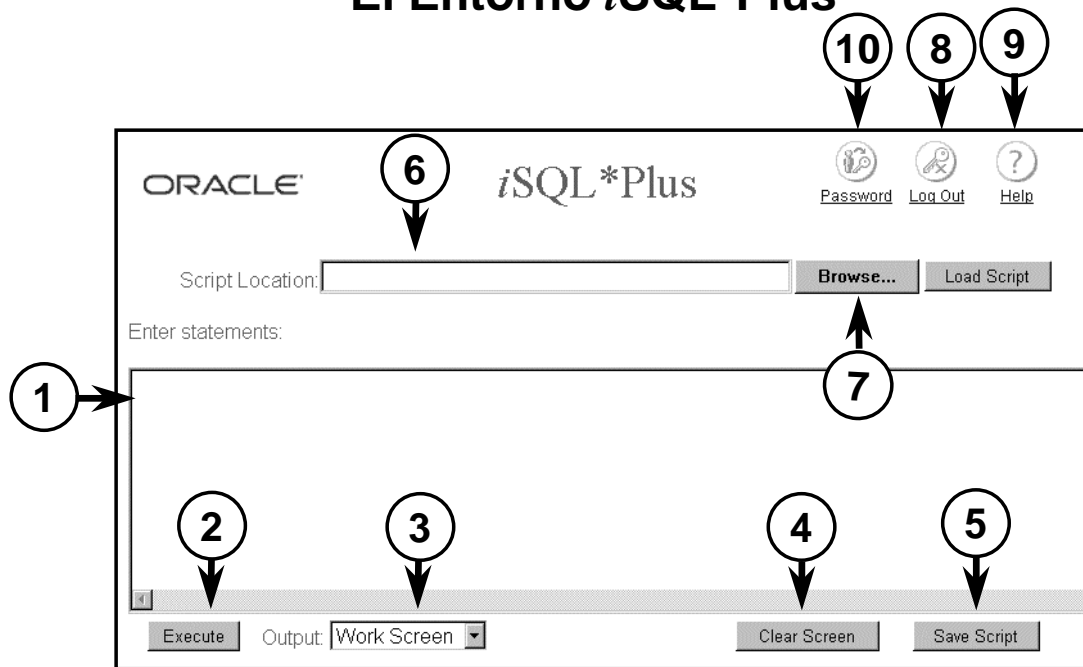
Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Conexión a iSQL*Plus

Para conectarse a través de un entorno de explorador:

1. Inicie el explorador.
2. Introduzca la dirección URL del entorno iSQL*Plus.
3. Introduzca valores en los campos Username, Password y Oracle Connection Identifier.

El Entorno iSQL*Plus



El Entorno iSQL*Plus

Dentro del explorador de Windows, la ventana iSQL*Plus tiene varias áreas claves:

1. La ventana Edit: El área donde escribe las sentencias SQL y los comandos iSQL*Plus.
2. El botón Execute : Haga clic para ejecutar las sentencias y comandos en la ventana Edit.
3. Opción Output : Tiene como valor por defecto Work Screen, que muestra los resultados de la sentencia SQL debajo de la ventana Edit. Las otras opciones son File o Window. File guarda los contenidos en un archivo especificado. Window coloca el resultado en la pantalla, pero en una ventana distinta.
4. Botón Clear Screen: Haga clic para borrar texto de la ventana Edit.
5. Botón Save Script: Guarda el contenido de la ventana Edit en un archivo.
6. Script Locator: Identifica el nombre y la ubicación de un archivo de comandos que desee ejecutar.
7. Botón Browse: Se utiliza para buscar un archivo de comandos mediante el recuadro de diálogo Windows File Open.
8. Icono Exit: Haga clic para finalizar la sesión iSQL*Plus y regresar a la ventana iSQL*Plus LogOn.
9. Icono Help: Proporciona acceso a la documentación iSQL*Plus Help.
10. Botón Password: Se utiliza para cambiar de contraseña.

Visualización de la Estructura de Tabla

Utilice el comando `DESCRIBE` de *iSQL*Plus* para ver la estructura de una tabla.

```
DESC[RIBE] tablename
```

ORACLE

1-29

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Visualización de la Estructura de Tabla

En *iSQL*Plus*, puede visualizar la estructura de una tabla mediante el comando `DESCRIBE`. El comando muestra los nombres de columna y los tipos de dato, así como si una columna *debe* contener datos.

En la sintaxis:

tablename es el nombre de cualquier tabla, vista o sinónimo existente accesible para el usuario

Visualización de la Estructura de Tabla

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

ORACLE

1-30

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Visualización de la Estructura de Tabla (continuación)

El ejemplo de la transparencia muestra la información de la estructura de la tabla DEPARTMENTS.

En el resultado:

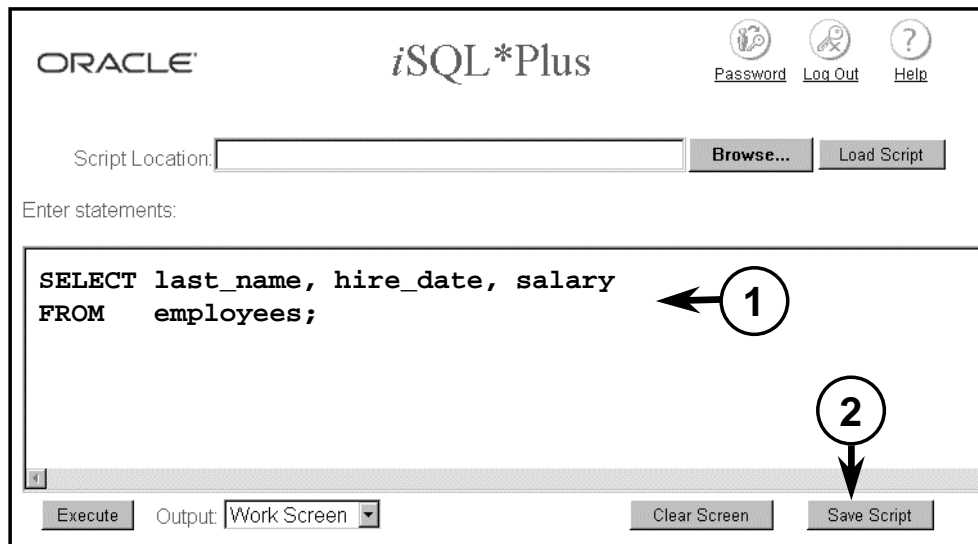
Null? indica si una columna *debe* contener datos; NOT NULL indica que una columna debe contener datos

Type muestra el tipo de dato de una columna

Los tipos de dato se describen en la siguiente tabla:

Tipo de dato	Descripción
NUMBER (<i>p</i> , <i>s</i>)	Valor numérico que tiene un número de dígitos máximo <i>p</i> , con dígitos <i>s</i> a la derecha de la coma decimal
VARCHAR2 (<i>s</i>)	Valor de carácter de longitud variable de tamaño máximo <i>s</i>
DATE	Valor de fecha y hora entre el 1 de enero de 4712 a.c. y el 31 de diciembre de 9999 d.c.
CHAR (<i>s</i>)	Valor de carácter de longitud fija de tamaño <i>s</i>

Interacción con Archivos de Comandos

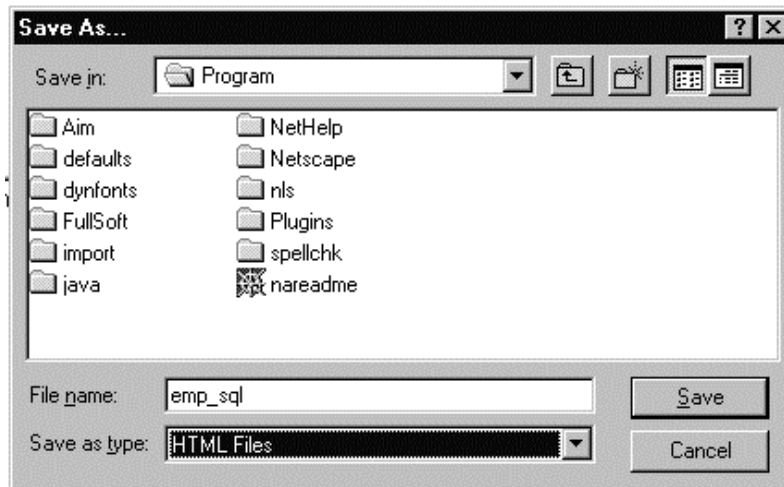


Interacción con Archivos de Comandos

Introducción de Sentencias y Comandos en un Archivo de Comandos de Texto

Puede guardar comandos y sentencias de la ventana Edit de iSQL*Plus en un archivo de comandos de texto como se indica a continuación:

1. Escriba la sentencia SQL en la ventana Edit de iSQL*Plus.
2. Haga clic en el botón Save Script. Se abre el recuadro de diálogo Windows File Save. Identifique el nombre del archivo. Éste toma por defecto la extensión .html. Puede cambiar el tipo de archivo por un archivo de texto o guardarlo como archivo .sql.



Interacción con Archivos de Comandos

The screenshot shows the iSQL*Plus web interface. At the top, there is an Oracle logo and the text 'iSQL*Plus'. To the right are links for 'Password', 'Log Out', and 'Help'. Below the header, there is a 'Script Location' field containing 'D:\temp\emp_sql.htm', a 'Browse...' button, and a 'Load Script' button. Below this is a text area labeled 'Enter statements:' containing the SQL query: `SELECT last_name, hire_date, salary FROM employees;`. At the bottom, there is an 'Execute' button, an 'Output' dropdown menu set to 'Work Screen', a 'Clear Screen' button, and a 'Save Script' button. Three numbered callouts are present: '1' points to the 'Script Location' field, '2' points to the 'Load Script' button, and '3' points to the 'Execute' button.

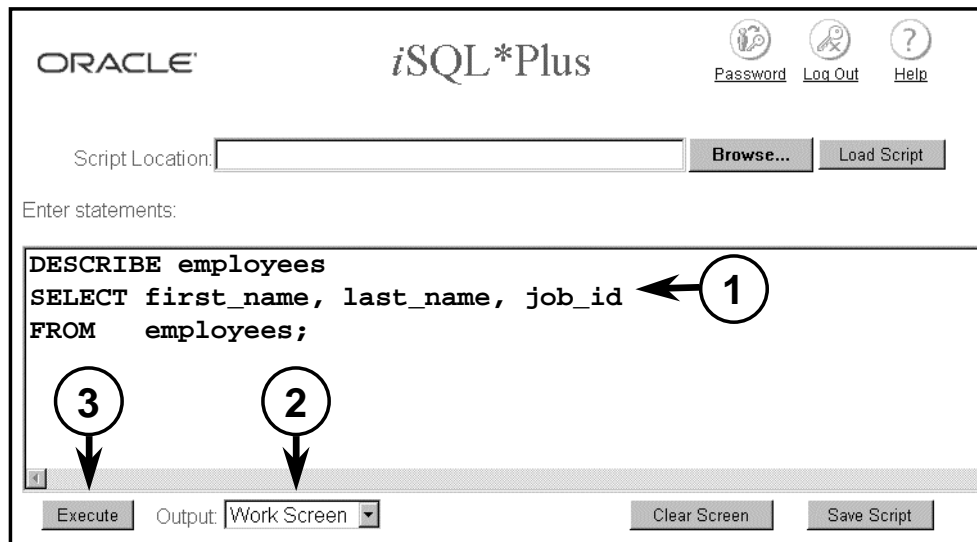
Interacción con Archivos de Comandos

Uso de Sentencias y Comandos de un Archivo de Comandos en iSQL*Plus

Puede utilizar sentencias y comandos guardados previamente de un archivo de comandos en iSQL*Plus como se indica a continuación:

1. Introduzca el nombre del archivo de comandos y la ubicación. También puede hacer clic en el botón Browse para buscar la ubicación y el nombre del archivo de comandos.
2. Haga clic en el botón Load Script. Los contenidos del archivo se cargan en la ventana Edit de iSQL*Plus.
3. Haga clic en el botón Execute para ejecutar los contenidos de la ventana Edit de iSQL*Plus.

Interacción con Archivos de Comandos

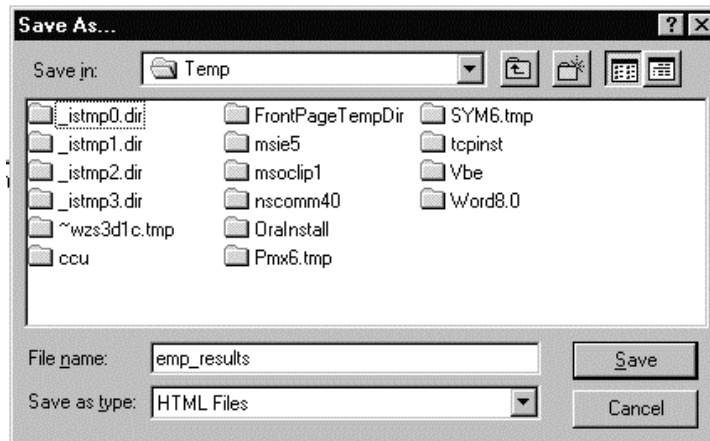


Interacción con Archivos de Comandos

Guardar Resultados en un Archivo

Puede guardar los resultados generados por una sentencia SQL o comando iSQL*Plus en un archivo:

1. Escriba las sentencias SQL y los comandos iSQL*Plus en la ventana Edit de iSQL*Plus.
2. Cambie la opción Output a Save.
3. Haga clic en el botón Execute para ejecutar los contenidos de la ventana Edit de iSQL*Plus. Se abre el recuadro de diálogo Windows File Save. Identifique el nombre del archivo. Éste toma por defecto la extensión .html. Puede cambiar el tipo de archivo. Los resultados se envían al archivo especificado.



Resumen

En esta lección, debería haber aprendido a:

- **Escribir una sentencia `SELECT` que:**
 - Devuelva todas las filas y las columnas de una tabla.
 - Devuelva columnas especificadas de una tabla.
 - Utilice alias de columna para asignar cabeceras de columna descriptivas.
- **Utilice el entorno `iSQL*Plus` para escribir, guardar y ejecutar sentencias SQL y comandos `iSQL*Plus`.**

```
SELECT  *|{[DISTINCT] column/expression [alias],...}  
FROM    table;
```

ORACLE®

1-34

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Sentencias `SELECT`

En esta lección, debería haber aprendido a recuperar datos de una tabla de base de datos con la sentencia `SELECT`.

```
SELECT  *|{[DISTINCT] column [alias],...}  
FROM    table;
```

En la sintaxis:

<code>SELECT</code>	es una lista de una o más columnas
<code>*</code>	selecciona todas las columnas
<code>DISTINCT</code>	suprime los duplicados
<code>column/expression</code>	selecciona la expresión o columna especificada
<code>alias</code>	asigna cabeceras diferentes a las columnas seleccionadas
<code>FROM table</code>	especifica la tabla que contiene las columnas

`iSQL*Plus`

`iSQL*Plus` es un entorno de ejecución que puede utilizar para enviar sentencias SQL al servidor de la base de datos y editarlas y guardarlas. Las sentencias se pueden ejecutar desde el prompt de SQL o desde un archivo de comandos.

Nota: El entorno SQL*Plus se trata en el Apéndice C.

Visión General de la Práctica 1

Esta práctica cubre los siguientes temas:

- **Selección de todos los datos de tablas diferentes**
- **Descripción de la estructura de las tablas**
- **Realización de cálculos aritméticos y especificación de nombres de columna**
- **Uso de *iSQL*Plus***

ORACLE

1-35

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Visión General de la Práctica 1

Ésta es la primera de muchas prácticas. Las soluciones (si las necesita) se pueden encontrar en el Apéndice A. Las prácticas pretenden introducir todos los temas tratados en la lección. Las preguntas 2–4 se contestan en papel.

En cualquier práctica puede haber preguntas “Si tiene tiempo” o “Si desea una comprobación adicional”. Contéstelas únicamente si ha terminado las otras preguntas en el tiempo asignado y si desea otra comprobación de sus habilidades.

Realice las prácticas de manera lenta y precisa. Puede experimentar guardando y ejecutando archivos de comandos. Si tiene alguna duda en cualquier momento, pregunte al instructor.

Preguntas para Contestar en Papel

En las preguntas 2-4, marque con un círculo Verdadero o Falso.

Práctica 1

1. Inicie una sesión *iSQL**Plus utilizando el identificador de usuario y la contraseña proporcionados por el instructor.

2. Los comandos *iSQL**Plus acceden a la base de datos.

Verdadero/Falso

3. La siguiente sentencia SELECT se ejecuta correctamente:

```
SELECT last_name, job_id, salary AS Sal
FROM employees;
```

Verdadero/Falso

4. La siguiente sentencia SELECT se ejecuta correctamente:

```
SELECT *
FROM job_grades;
```

Verdadero/Falso

5. Hay cuatro errores de codificación en esta sentencia. ¿Puede identificarlos?

```
SELECT      employee_id, last_name
sal x 12    ANNUAL SALARY
FROM        employees;
```

6. Muestre la estructura de la tabla DEPARTMENTS. Seleccione todos los datos de la tabla.

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

Práctica 1 (continuación)

- Muestre la estructura de la tabla `EMPLOYEES`. Cree una consulta para mostrar el apellido, el código de cargo, la fecha de contratación y el número de empleado para cada empleado, con el número de empleado en primer lugar. Proporcione un alias para la columna. Guarde la sentencia SQL en un archivo llamado `lab1_7.sql`.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

- Ejecute la consulta del archivo `lab1_7.sql`.

EMPLOYEE_ID	LAST_NAME	JOB_ID	STARTDATE
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
149	Zlotkey	SA_MAN	29-JAN-00
174	Abel	SA_REP	11-MAY-96
176	Taylor	SA_REP	24-MAR-98
...			
206	Gietz	AC_ACCOUNT	07-JUN-94

20 rows selected.

Práctica 1 (continuación)

9. Cree una consulta para mostrar códigos de cargo únicos de la tabla EMPLOYEES.

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
IT_PROG
MK_MAN
MK_REP
SA_MAN
SA_REP
ST_CLERK
ST_MAN

12 rows selected.

Si tiene tiempo, complete los siguientes ejercicios:

10. Copie la sentencia de lab1_7.sql en la ventana iSQL*Plus Edit. Llame a las cabeceras de columna Emp #, Employee, Job y Hire Date, respectivamente. Vuelva a ejecutar la consulta.

Emp #	Employee	Job	Hire Date
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
...			
206	Gietz	AC_ACCOUNT	07-JUN-94

20 rows selected.

Práctica 1 (continuación)

11. Muestre el apellido concatenado con el identificador de cargo, separados por una coma y un espacio y llame a la columna `Employee and Title`.

Employee and Title
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK
■ ■ ■
Gietz, AC_ACCOUNT

20 rows selected.

Si desea una comprobación adicional, complete el siguiente ejercicio:

12. Cree una consulta para mostrar todos los datos de la tabla `EMPLOYEES`. Separe cada columna con una coma. Llame a la columna `THE_OUTPUT`.

THE_OUTPUT
100,Steven,King,SKING,515.123.4567,AD_PRES,,17-JUN-87,24000,,90
101,Neena,Kochhar,NKOCHHAR,515.123.4568,AD_VP,100,21-SEP-89,17000,,90
102,Lex,De Haan,LDEHAAN,515.123.4569,AD_VP,100,13-JAN-93,17000,,90
103,Alexander,Hunold,AHUNOLD,590.423.4567,IT_PROG,102,03-JAN-90,9000,,60
104,Bruce,Ernst,BERNST,590.423.4568,IT_PROG,103,21-MAY-91,6000,,60
107,Diana,Lorentz,DLORENTZ,590.423.5567,IT_PROG,103,07-FEB-99,4200,,60
124,Kevin,Mourgos,KMOURGOS,650.123.5234,ST_MAN,100,16-NOV-99,5800,,50
141,Trenna,Rajs,TRAJS,650.121.8009,ST_CLERK,124,17-OCT-95,3500,,50
142,Curtis,Davies,CDAVIES,650.121.2994,ST_CLERK,124,29-JAN-97,3100,,50
143,Randall,Matos,RMATOS,650.121.2874,ST_CLERK,124,15-MAR-98,2600,,50
144,Peter,Vargas,PVARGAS,650.121.2004,ST_CLERK,124,09-JUL-98,2500,,50
■ ■ ■
206,William,Gietz,WGIETZ,515.123.8181,AC_ACCOUNT,205,07-JUN-94,8300,,110

20 rows selected.

2

Restricción y Ordenación de Datos

ORACLE

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Limitar las filas recuperadas por una consulta**
- **Ordenar las filas recuperadas por una consulta**

ORACLE

2-2

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Objetivo de la Lección

Al recuperar datos de la base de datos, es posible que deba restringir las filas de datos que se muestran o especificar el orden en el que se muestran las filas. Esta lección explica las sentencias SQL que debe utilizar para realizar estas acciones.

Limitación de Filas Mediante una Selección

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

...

20 rows selected.

**“recuperar todos
los empleados
del departamento 90”**



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

ORACLE

Limitación de Filas Mediante una Selección

En el ejemplo de la transparencia, suponga que desea mostrar todos los empleados del departamento 90. Las filas con el valor 90 en la columna DEPARTMENT_ID son las únicas que se devuelven. Este método de restricción es la base de la cláusula WHERE en SQL.

Limitación de las Filas Seleccionadas

- Restrinja las filas devueltas utilizando la cláusula **WHERE**.

```
SELECT    *|{[DISTINCT] column/expressión [alias],...}  
FROM      table  
[WHERE    condición(s)];
```

- La cláusula **WHERE** sigue a la cláusula **FROM**.

ORACLE

2-4

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Limitación de las Filas Seleccionadas

Puede restringir las filas devueltas por la consulta utilizando la cláusula **WHERE**. Esta cláusula contiene una condición que se debe cumplir y sigue directamente a la cláusula **FROM**. Si la condición es verdadera, se devuelve la fila que cumple la condición.

En la sintaxis:

WHERE	restringe la consulta a las filas que cumplen una condición
<i>condición</i>	está formado por nombres de columna, expresiones, constantes y un operador de comparación

La cláusula **WHERE** puede comparar valores de columnas, valores literales, expresiones aritméticas o funciones. Consta de tres elementos:

- Nombre de columna
- Condición de comparación
- Nombre de 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 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

ORACLE

Uso de la Cláusula WHERE

En el ejemplo, la sentencia `SELECT` recupera el nombre, el identificador de cargo y el número de departamento de todos los empleados cuyos identificadores de cargo sean `SA_REP`.

Observe que se ha especificado el cargo `SA_REP` en mayúsculas para asegurarse de que coincide con la columna de identificador de cargo de la tabla `EMPLOYEES`. Las cadenas de caracteres son sensibles a mayúsculas/minúsculas.

Cadenas de Caracteres y Fechas

- Las cadenas de caracteres y los valores de fechas se escriben entre comillas simples.
- Los valores de caracteres son sensibles a mayúsculas/minúsculas y los de fecha, al formato.
- El formato de fecha por defecto es DD-MON-RR.

```
SELECT last_name, job_id, department_id
FROM   employees
WHERE  last_name = 'Whalen';
```

ORACLE

Cadenas de Caracteres y Fechas

En la cláusula WHERE, las cadenas de caracteres y las fechas se deben escribir entre comillas simples (' '), pero no las constantes numéricas.

Las búsquedas de caracteres son sensibles a mayúsculas/minúsculas. En el siguiente ejemplo, no se devuelve ninguna fila, pues la tabla EMPLOYEES almacena todos los apellidos con mayúsculas y minúsculas mezcladas:

```
SELECT last_name, job_id, department_id
FROM   employees
WHERE  last_name = 'WHALEN';
```

Las bases de datos Oracle almacenan fechas en formato numérico interno, representando el siglo, el año, el mes, el día, las horas, los minutos y los segundos. El formato de fecha por defecto es DD-MON-RR.

Nota: El cambio del formato de fecha por defecto se trata en una lección posterior.

Condiciones de Comparación

Operador	Significado
=	Igual que
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
<>	No igual a

ORACLE

2-7

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Condiciones de Comparación

Las condiciones de comparación se utilizan en condiciones que comparan una expresión con otro valor o expresión. Se usan en la cláusula `WHERE` con el siguiente formato:

Sintaxis

```
... WHERE expr operator value
```

Por Ejemplo

```
... WHERE hire_date='01-JAN-95'  
... WHERE salary>=6000  
... WHERE last_name='Smith'
```

No se puede utilizar un alias en la cláusula `WHERE`.

Nota: Los símbolos `!=` y `^=` también pueden representar la condición *no igual a*.

Uso de Condiciones de Comparación

```
SELECT last_name, salary
FROM   employees
WHERE  salary <= 3000;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

ORACLE

Uso de las Condiciones de Comparación

En el ejemplo, la sentencia `SELECT` recupera el apellido y el salario de la tabla `EMPLOYEES`, si el salario del empleado es menor o igual que 3000. Observe que existe un valor explícito para la cláusula `WHERE`. El valor explícito 3000 se compara con el valor de salario de la columna `SALARY` de la tabla `EMPLOYEES`.

Otras Condiciones de Comparación

Operador	Significado
BETWEEN ...AND...	Entre dos valores (ambos inclusive),
IN(set)	Coincide con cualquiera de una lista de valores
LIKE	Coincide con un patrón de caracteres
IS NULL	Es un valor nulo

ORACLE

Uso de la Condición BETWEEN

Utilice la condición **BETWEEN** para mostrar filas incluidas en un rango de valores.

```
SELECT last_name, salary
FROM   employees
WHERE  salary BETWEEN 2500 AND 3500;
```



Límite inferior Límite superior

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

ORACLE

2-10

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

La Condición BETWEEN

Puede mostrar filas incluidas en un rango de valores utilizando la condición de rango **BETWEEN**. El rango que especifique contiene un límite inferior y uno superior.

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

Los valores especificados en la condición **BETWEEN** están incluidos. Debe especificar en primer lugar el límite inferior.

Uso de la Condición IN

Utilice la condición IN de pertenencia para comprobar si hay valores en una lista.

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

ORACLE

La Condición IN

Para comprobar si hay valores en un juego especificado de valores, utilice la condición IN. Esta condición también se conoce como *condición de pertenencia*.

El ejemplo de la transparencia muestra números de empleado, apellidos, salarios y números de empleado del director para todos los empleados cuyos números de empleado del director sean 100, 101 o 201.

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 en la cláusula WHERE:

```
SELECT employee_id, manager_id, department_id
FROM   employees
WHERE  last_name IN ('Hartstein', 'Vargas');
```

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

Uso de la Condición LIKE

- Utilice la condición **LIKE** para realizar búsquedas con comodines de valores válidos de la cadena de búsqueda.
- Las condiciones de búsqueda pueden contener caracteres literales o números:
 - % indica cero o muchos caracteres.
 - _ indica un carácter.

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

ORACLE

2-12

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

La Condición LIKE

No siempre conocerá el valor exacto para buscar. Puede seleccionar filas que coincidan con un patrón de caracteres utilizando la condición **LIKE**. La operación de coincidencia del patrón de caracteres se denomina búsqueda con *comodines*. Se pueden utilizar dos símbolos para construir la cadena de búsqueda.

Símbolo	Descripción
%	Representa cualquier secuencia de cero o más caracteres.
_	Representa un único carácter.

La sentencia **SELECT** de la transparencia devuelve el nombre del empleado de la tabla **EMPLOYEES** para cualquier empleado cuyo nombre empiece por *S*. Observe que la *S* está en mayúsculas. No se devolverán nombres que empiecen por *s*.

La condición **LIKE** se puede utilizar como método abreviado de algunas comparaciones **BETWEEN**.

El siguiente ejemplo muestra los apellidos y las fechas de contratación de todos los empleados contratados entre enero y diciembre de 1995:

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date LIKE '%95';
```

Uso de la Condición LIKE

- Puede combinar caracteres de coincidencia de patrones.

```
SELECT last_name  
FROM employees  
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

- Puede utilizar el identificador **ESCAPE** para buscar los símbolos % y _ reales.

ORACLE

Combinación de Caracteres Comodín

Los símbolos % y _ se pueden utilizar en cualquier combinación con caracteres literales. El ejemplo de la transparencia muestra los nombres de todos los empleados cuyos apellidos tienen una *o* como segundo carácter.

La opción **ESCAPE**

Si necesita una coincidencia exacta de los caracteres % y _ reales, utilice la opción **ESCAPE**. Esta opción especifica cuál es 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 '\';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
149	Zlotkey	SA_MAN
174	Abel	SA_REP
176	Taylor	SA_REP
178	Grant	SA_REP

La opción **ESCAPE** identifica la barra invertida (\) como el carácter de escape. En el patrón, el carácter de escape precede al carácter de subrayado (_). Esto hace que Oracle Server interprete al carácter de subrayado literalmente.

Uso de las Condiciones NULL

Compruebe si hay valores nulos con el operador IS NULL.

```
SELECT last_name, manager_id
FROM   employees
WHERE  manager_id IS NULL;
```

LAST_NAME	MANAGER_ID
King	

ORACLE

2-14

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Las Condiciones NULL

Las condiciones NULL incluyen la condición IS NULL y la condición IS NOT NULL.

La condición IS NULL comprueba si hay valores nulos. Un valor nulo significa que el valor no está disponible, no está asignado, es desconocido o no es aplicable. Por lo tanto, no puede comprobar con =, pues un valor nulo no puede ser igual o distinto de ningún valor. El ejemplo de la transparencia recupera los apellidos y los directores de todos los empleados que no tienen director.

Para otro ejemplo, para mostrar apellidos, identificador de cargo y comisiones para 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;
```

LAST_NAME	JOB_ID	COMMISSION_PCT
King	AD_PRES	
Kochhar	AD_VP	
...		
Higgins	AC_MGR	
Gietz	AC_ACCOUNT	

16 rows selected.

Condiciones Lógicas

Operador	Significado
AND	Devuelve TRUE si las <i>dos</i> condiciones componentes son verdaderas
OR	Devuelve TRUE si <i>alguna</i> de las condiciones componentes es verdadera
NOT	Devuelve TRUE si la siguiente condición es falsa

ORACLE

Condiciones Lógicas

Las condiciones lógicas combinan el resultado de dos condiciones componentes para producir un resultado único basado en ellas o invierten el resultado de una única condición. Se devuelve una fila sólo si el resultado global de la condición es verdadero. En SQL están disponibles tres operadores lógicos:

- AND
- OR
- NOT

Todos los ejemplos mostrados hasta ahora han especificado solamente una condición en la cláusula WHERE. Puede utilizar varias condiciones en una cláusula WHERE utilizando los operadores AND y OR.

Uso del Operador AND

AND requiere que las dos condiciones sean verdaderas.

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >=10000
AND    job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

ORACLE

2-16

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

El Operador AND

En el ejemplo, las dos condiciones deben ser verdaderas para que se seleccione un registro. Por lo tanto, solamente se seleccionarán los empleados cuyo cargo contenga la cadena MAN y ganen más de \$10.000.

Todas las búsquedas de caracteres son sensibles a mayúsculas/minúsculas. No se devolverán filas si MAN no está en mayúsculas. Las cadenas de caracteres se deben escribir entre comillas.

Tabla de Verdad de 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

Uso del Operador OR

OR requiere que una de las condiciones sea verdadera.

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

ORACLE

El Operador OR

En el ejemplo, cualquiera de las dos condiciones puede ser verdadera para que se seleccione un registro. Por lo tanto, se seleccionará cualquier empleado cuyo identificador de cargo contenga MAN o gane más de \$10.000.

Tabla de Verdad de 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_id
FROM   employees
WHERE  job_id
       NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

ORACLE

2-18

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

El Operador NOT

El ejemplo de la transparencia muestra el apellido y el identificador de cargo de todos los empleados cuyos identificadores de cargo *no sean* IT_PROG, ST_CLERK o SA_REP.

Tabla de Verdad de NOT

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

Orden de Evaluación	Operador
1	Operadores aritméticos
2	Operador de concatenación
3	Condiciones de comparación
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	condición lógica NOT
7	condición lógica AND
8	condición lógica OR

Las reglas de prioridad se sustituyen mediante el uso de paréntesis.

ORACLE

Reglas de Prioridad

Las reglas de prioridad determinan el orden en el que se evalúan y se calculan las expresiones. La tabla enumera el orden de prioridad por defecto. Puede sustituir el orden por defecto escribiendo entre paréntesis las expresiones que desee calcular en primer lugar.

Reglas de Prioridad

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = 'SA_REP'
OR     job_id = 'AD_PRES'
AND    salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

ORACLE

Ejemplo de la Prioridad del Operador AND

En el ejemplo de la transparencia, hay dos condiciones:

- La primera condición es que el identificador de cargo es AD_PRES y el salario mayor que 15.000.
- La segunda condición es que el identificador de cargo es SA_REP.

Por lo tanto, la sentencia SELECT indica lo siguiente:

“Seleccionar la fila si un empleado es presidente y gana más de \$15.000 o si el empleado es representante de ventas.”

Reglas de Prioridad

Utilice paréntesis para forzar la prioridad.

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  (job_id = 'SA_REP'
OR      job_id = 'AD_PRES')
AND    salary > 15000;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

ORACLE

Uso de Paréntesis

En el ejemplo, hay dos condiciones:

- La primera condición es que el identificador de cargo es AD_PRES o SA_REP.
- La segunda condición es que el salario es mayor que \$15.000.

Por lo tanto, la sentencia SELECT indica lo siguiente:

“Seleccionar la fila si un empleado es presidente o representante de ventas y gana más de \$15.000.”

Cláusula ORDER BY

- Ordene filas con la cláusula ORDER BY
 - ASC: orden ascendente, por defecto
 - DESC: orden descendente
- La cláusula ORDER BY aparece en último lugar en la sentencia SELECT.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...
20 rows selected.

ORACLE

2-22

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

La Cláusula ORDER BY

El orden de filas devuelto por una consulta no está definido. Se puede utilizar la cláusula ORDER BY para ordenar las filas. Si lo hace, debe ser la última de la sentencia SQL. Puede especificar una expresión o alias o posición de columna como condición de orden.

Sintaxis

```
SELECT      expr
FROM        table
[WHERE      condition(s)]
[ORDER BY {column, expr} [ASC|DESC]];
```

En la sintaxis:

ORDER BY	especifica el orden en el que se muestran las filas recuperadas
ASC	ordena las filas en orden ascendente (es el orden por defecto)
DESC	ordena las filas en orden descendente

Si no se utiliza la cláusula ORDER BY, el orden no está definido y es posible que Oracle Server no recupere las filas en el mismo orden dos veces para la misma consulta. Utilice esta cláusula para mostrar las filas en un orden específico.

Ordenación en Orden Descendente

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
Zlotkey	SA_MAN	80	29-JAN-00
Mourgos	ST_MAN	50	16-NOV-99
Grant	SA_REP		24-MAY-99
Lorentz	IT_PROG	60	07-FEB-99
Vargas	ST_CLERK	50	09-JUL-98
Taylor	SA_REP	80	24-MAR-98
Matos	ST_CLERK	50	15-MAR-98
Fay	MK_REP	20	17-AUG-97
Davies	ST_CLERK	50	29-JAN-97

...

20 rows selected.

ORACLE

Ordenación por Defecto de Datos

El orden por defecto es ascendente:

- Los valores numéricos se muestran con los valores más bajos primero; por ejemplo, 1 a 999.
- Los valores de fecha se muestran con la fecha anterior primero; por ejemplo, 01-ENE-92 antes de 01-ENE-95.
- Los valores de caracteres se muestran en orden alfabético; por ejemplo, A en primer lugar y Z en último.
- Los valores nulos se muestran los últimos para las secuencias ascendentes y los primeros para las descendentes.

Reversión al Orden por Defecto

Para revertir 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 transparencia ordena el resultado según el último empleado contratado.

Ordenación según Alias de Columna

```
SELECT employee_id, last_name, salary*12 annsal  
FROM employees  
ORDER BY annsal;
```

EMPLOYEE_ID	LAST_NAME	ANNSAL
144	Vargas	30000
143	Matos	31200
142	Davies	37200
141	Rajs	42000
107	Lorentz	50400
200	Whalen	52800
124	Mourgos	69600
104	Ernst	72000
202	Fay	72000
178	Grant	84000

...
20 rows selected.

ORACLE

Ordenación según Alias de Columna

Puede utilizar un alias de columna en la cláusula ORDER BY. El ejemplo de la transparencia ordena los datos según el salario anual.

Ordenación según Múltiples Columnas

- El orden de la lista `ORDER BY` es el de ordenación.

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC;
```

LAST_NAME	DEPARTMENT_ID	SALARY
Whalen	10	4400
Hartstein	20	13000
Fay	20	6000
Mourgos	50	5800
Rajs	50	3500
Davies	50	3100
Matos	50	2600
Vargas	50	2500

...

20 rows selected.

- Puede ordenar según una columna que no se encuentre en la lista `SELECT`.

ORACLE

Ordenación según Múltiples Columnas

Puede ordenar el resultado de la consulta según más de una columna. El límite de ordenación es el número de columnas de la tabla dada.

En la cláusula `ORDER BY`, especifique las columnas y separe los nombres de columna mediante comas. Si desea revertir el orden de una columna, especifique `DESC` después de su nombre. También puede ordenar según columnas no incluidas en la cláusula `SELECT`.

Ejemplo

Muestre los apellidos y los salarios de todos los empleados. Ordene el resultado según número de departamento y después en orden descendente según salario.

```
SELECT last_name, salary
FROM employees
ORDER BY department_id, salary DESC;
```

Resumen

En esta lección, debería haber aprendido a:

- Utilizar la cláusula **WHERE** para restringir filas de salida
 - Utilizar las condiciones de comparación
 - Utilizar las condiciones **BETWEEN**, **IN**, **LIKE** y **NULL**
 - Aplicar los operadores lógicos **AND**, **OR** y **NOT**
- Utilizar la cláusula **ORDER BY** para ordenar filas de salida

```
SELECT      * | {[DISTINCT] column/expression [alias],...}  
FROM        table  
[WHERE      condition(s)]  
[ORDER BY   {column, expr, alias} [ASC|DESC]];
```

ORACLE

Resumen

En esa lección, debería haber aprendido a restringir y ordenar las filas devueltas por la sentencia **SELECT**. También debería haber aprendido a implementar varios operadores y condiciones.

Visión General de la Práctica 2

Esta práctica cubre los siguientes temas:

- Selección de datos y cambio del orden de las filas mostradas
- Restricción de filas utilizando la cláusula `WHERE`
- Ordenación de filas utilizando la cláusula `ORDER BY`

ORACLE

2-27

Copyright © Oracle Corporation, 2001. Todos los derechos reservados.

Visión General de la Práctica 2

Esta práctica le ofrece diversos ejercicios que utilizan las cláusulas `WHERE` y `ORDER BY`.

Práctica 2

1. Cree una consulta para mostrar el apellido y el salario de los empleados que ganan más de \$12.000. Coloque la sentencia SQL en un archivo de texto llamado `lab2_1.sql`. Ejecute la consulta.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hartstein	13000

2. Cree una consulta para mostrar el apellido del empleado y el número de departamento para el número de empleado 176.

LAST_NAME	DEPARTMENT_ID
Taylor	80

3. Modifique `lab2_1.sql` para mostrar el apellido y el salario para todos los empleados cuyos salarios no están comprendidos entre \$5.000 y \$12.000. Coloque la sentencia SQL en un archivo de texto llamado `lab2_3.sql`.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Lorentz	4200
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Whalen	4400
Hartstein	13000

10 rows selected.

Práctica 2 (continuación)

- Muestre el apellido del empleado, el identificador de cargo y la fecha de inicio de los empleados contratados entre el 20 de febrero de 1998 y el 1 de mayo de 1998. Ordene la consulta en orden ascendente por fecha de inicio.

LAST_NAME	JOB_ID	HIRE_DATE
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

- Muestre el apellido y el número de departamento de todos los empleados de los departamentos 20 y 50 en orden alfabético por apellido.

LAST_NAME	DEPARTMENT_ID
Davies	50
Fay	20
Hartstein	20
Matos	50
Mourgos	50
Rajs	50
Vargas	50

7 rows selected.

- Modifique lab2_3.sql para enumerar el apellido y el salario de los empleados que ganan entre \$5.000 y \$12.000, y están en el departamento 20 ó 50. Etiquete las columnas Employee y Monthly Salary, respectivamente. Vuelva a guardar lab2_3.sql como lab2_6.sql. Ejecute la sentencia en lab2_6.sql.

Employee	Monthly Salary
Mourgos	5800
Fay	6000

Práctica 2 (continuación)

7. Muestre el apellido y la fecha de contratación de todos los empleados contratados en 1994.

LAST_NAME	HIRE_DATE
Higgins	07-JUN-94
Gietz	07-JUN-94

8. Muestre el apellido y el cargo de todos los empleados que no tienen director.

LAST_NAME	JOB_ID
King	AD_PRES

9. Muestre el apellido, el salario y la comisión para todos los empleados que ganan comisiones. Ordene los datos en orden descendente de salarios y comisiones.

LAST_NAME	SALARY	COMMISSION_PCT
Abel	11000	.3
Zlotkey	10500	.2
Taylor	8600	.2
Grant	7000	.15

Si tiene tiempo, complete los siguientes ejercicios:

10. Muestre el apellido de todos los empleados que tengan la *a* como tercera letra.

LAST_NAME
Grant
Whalen

11. Muestre el apellido de todos los empleados que tengan una *a* y una *e* en el apellido.

LAST_NAME
De Haan
Davies
Whalen
Hartstein

Práctica 2 (continuación)

Si desea una comprobación adicional, complete los siguientes ejercicios:

12. Muestre el apellido, el cargo y el salario de todos los empleados cuyos cargos sean representantes de ventas o encargados de stock y cuyos salario no sean iguales a \$2.500, \$3.500 ni \$7.000.

LAST_NAME	JOB_ID	SALARY
Davies	ST_CLERK	3100
Matos	ST_CLERK	2600
Abel	SA_REP	11000
Taylor	SA_REP	8600

13. Modifique lab2_6.sql para mostrar el apellido, el salario y la comisión para todos los empleados cuyas comisiones son el 20 %. Vuelva a guardar lab2_6.sql como lab2_13.sql. Vuelva a ejecutar la sentencia en lab2_13.sql.

Employee	Monthly Salary	COMMISSION_PCT
Zlotkey	10500	.2
Taylor	8600	.2

