



UNIDAD 6

Diseño físico de datos. Oracle



Bases de datos 2011-2012



Índice

- 1. Oracle. Características
- 2. SQL-Developer
- 3. Diseño físico
 - 3.1 Tipos de datos
 - 3.2 Creación de tablas. Restricciones.
 - 3.3 Supresión de tablas.
 - 3.4 Modificación de tablas.



Bases de datos



1. Oracle. Características

Es fundamental darse de alta en la página

<http://www.oracle.com/es/index.html>



Bases de datos

The screenshot shows the Oracle Spain homepage. At the top, there's a navigation bar with links like 'Productos y Servicios', 'Soluciones', 'Descargas', 'Tienda', 'Soporte', 'Formación', 'Partners', and 'Acerca de'. Below this is a main banner for 'IDENTITY MANAGEMENT' titled 'Security for the Mobile Workforce'. The banner includes a sub-headline 'Find out how Oracle Identity Management secures operations in the cloud and on the go.' and a button to 'Read the Oracle Magazine article'. Below the banner is a 'Newsroom' section with a headline 'Oracle anuncia Exalogic Elastic Cloud Software 2.0'. At the bottom, there's a footer with columns for 'Base De Datos', 'Middleware', 'Aplicaciones', 'Server and Storage Systems', and 'Industrias', each listing various Oracle products and services. A contact box on the right provides the Oracle 900 952 900 number and a 'Chat de Ventas en Vivo' button.



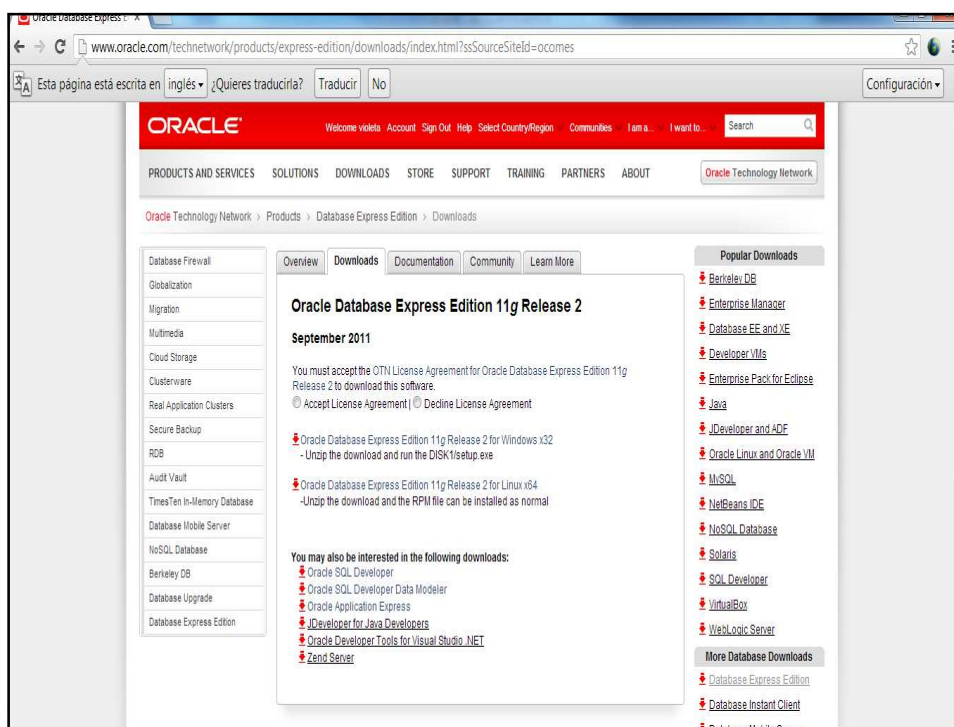
1. Oracle. Características

- En la sección Descargas podremos acceder a versiones gratuitas de Oracle.
- Utilizaremos la última versión:

Oracle 11G Express Edition



Bases de datos





Características

- Oracle Database 11g Express Edition (Oracle Database XE), creada sobre la base de código Oracle Database 11g Release 2 que puede desarrollarse, implementarse y distribuirse sin cargo; es fácil de descargar y de administrar.



Bases de datos




Características

- Oracle DB XE puede instalarse en máquinas host de cualquier tamaño con cualquier cantidad de CPUs (una base de datos por máquina)
- XE almacenará hasta 4GB de datos de usuarios, utilizará hasta 1GB de memoria, y utilizará una sola CPU en la máquina host.



Bases de datos




I.E.S.
Santiago
Hernández

Instalación


Please sign in

The requested username and password are incorrect. (If you are seeing this message, your browser configuration may be incorrect.)



Nombre: OracleXEUniv.exe
Tipo: Aplicación, 206 MB
De: download-lhnw.oracle.com

Aunque los archivos procedentes de Internet pueden ser útiles, este tipo de archivo puede llegar a dañar el equipo. Si no confía en el origen, no ejecute ni guarde este software. [¿Cuál es el riesgo?](#)



Bases de datos

Oracle Database 11g Express Edition - Install Wizard

Choose Destination Location

Select folder where setup will install files.


Setup will install Oracle Database 11g Express Edition in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

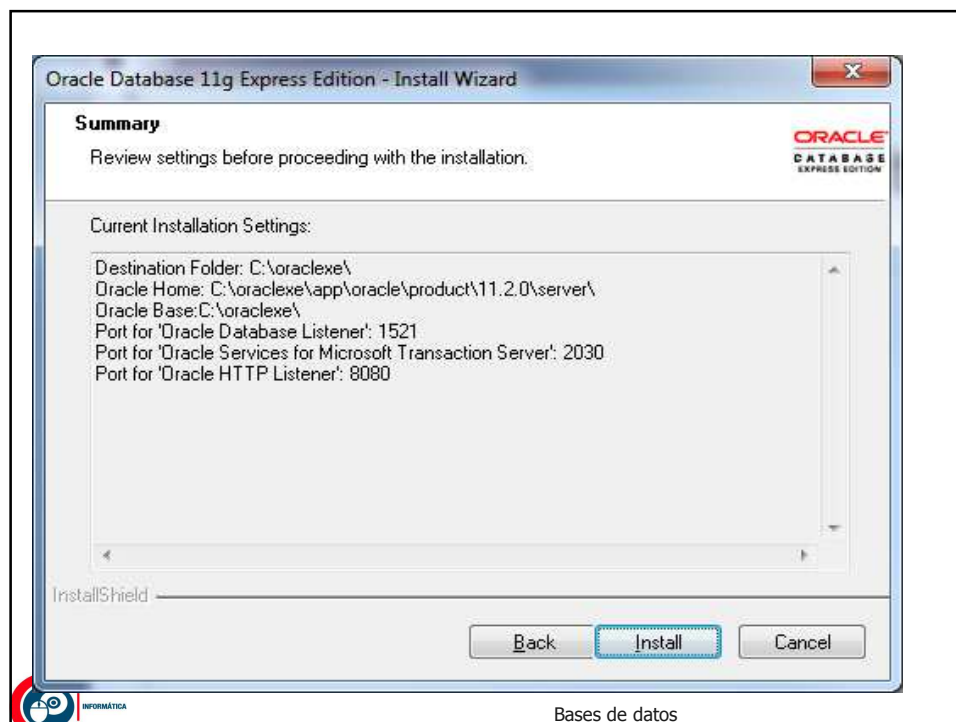
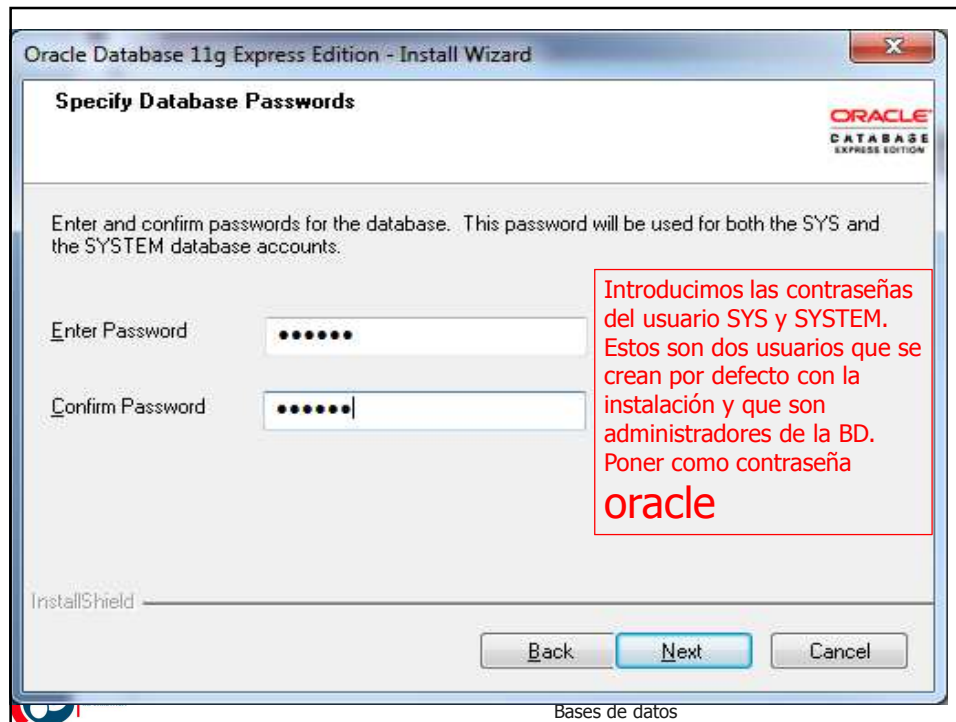
☒ Oracle Database 11g Express Edition 563556 K

Destination Folder
C:\oraclexe\

Space Required on C: 563556 K
Space Available on C: 100186964 K



Bases de datos





Instalación

- En la pantalla anterior nos indica las carpetas de instalación y puertos para los servicios instalados.
- Si abrimos la consola de servicios de windows utilizando el comando services.msc desde el menú de inicio, veremos los servicios que se han creado



Bases de datos



Servicios asociados a XE

Oracle Service Engine			
Nombre del servicio		Estado	Origen
OracleJobSchedulerXE		Deshabilitado	Sistema local
OracleMTSRecoveryService		Manual	Sistema local
OracleServiceXE	Iniciado	Automático	Sistema local
OracleXEClrAgent		Manual	Sistema local
OracleXETNSListener	Iniciado	Automático	Sistema local

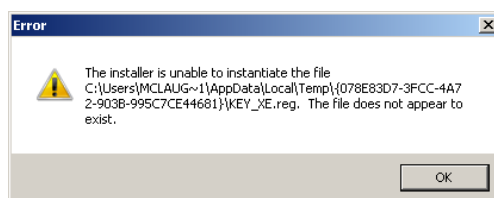


Bases de datos



Problemas

- La versión Oracle11g Express es de 32bits, no está la de 64bits para Windows7.
- Es probable que durante la instalación nos dé este error:



Bases de datos



Solución

- The physical file isn't missing, what failed to happen was that the installer didn't write a Registry Key's *Data* value. You can add the missing registry key's *Data* value, and then click *OK* on the *Error* dialog box to continue the installation.

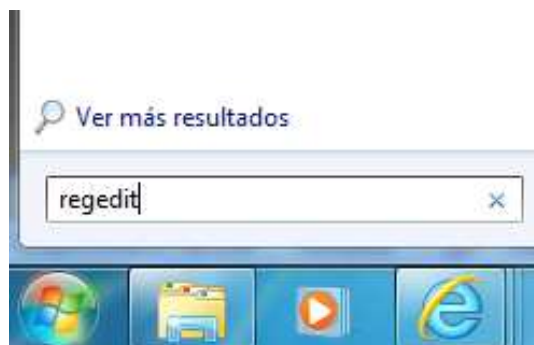


Bases de datos



Solución

- Accederemos al registro de windows7






Bases de datos



Solución

- Buscamos la clave

HKEY_CLASSES_ROOT\Installer\Products\
266B7A503A089BE4EAD1986A429434C
1\SourceList\Media

Name	Type	Data
 (Default)	REG_SZ	(value not set)
 1	REG_SZ	DISK1;1
 DiskPrompt	REG_SZ	[1]



Bases de datos



Solución

- Debes cambiar el valor de la entrada 1 por la dirección del fichero que ha aparecido en el error.



Bases de datos



Solución

C:\Users\hostname\AppData\Local\Temp\{078E83D7-3FCC-4A72-903B-995C7CE44681}\

Name	Type	Data
ab (Default)	REG_SZ	(value not set)
ab 1	REG_SZ	C:\Users\MCLAUG~1\AppData\Local\Temp\{078E83D7-3FCC-4A72-903B-995C7CE44681}\
ab DiskPrompt	REG_SZ	[1]



Bases de datos

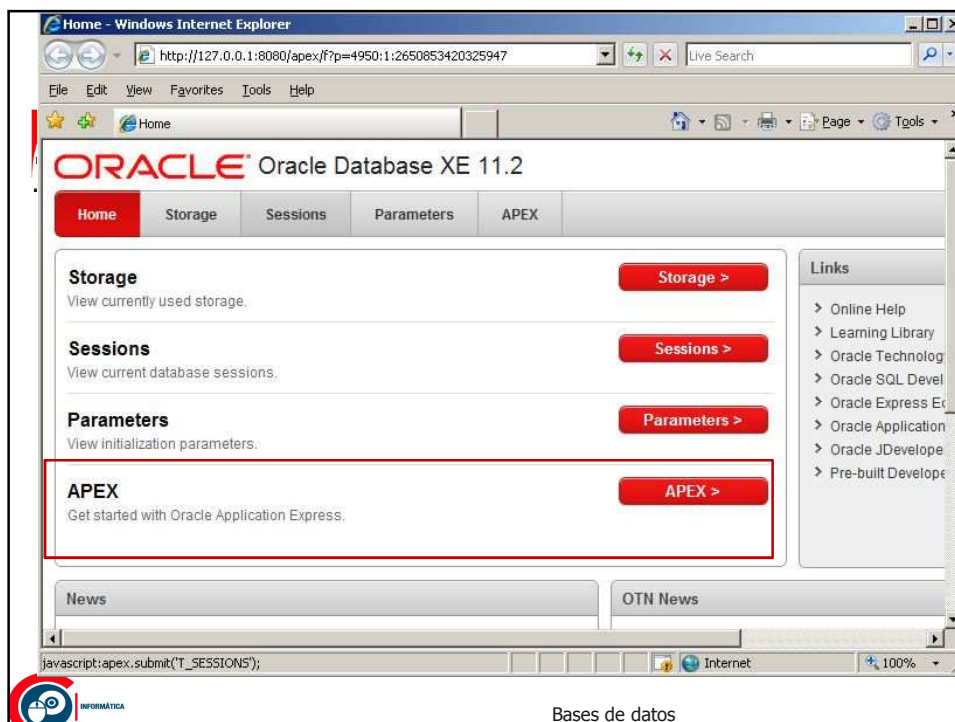


Instalación

- Una vez reiniciado el ordenador para que guarde los cambios del registro. Debes acceder al icono GetStarted



Bases de datos

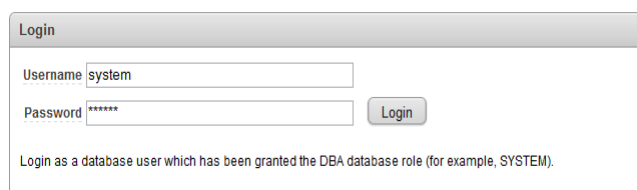


Bases de datos



Crear un nuevo usuario de trabajo

- Introduce como usuario system



Login

Username system

Password ***** Login

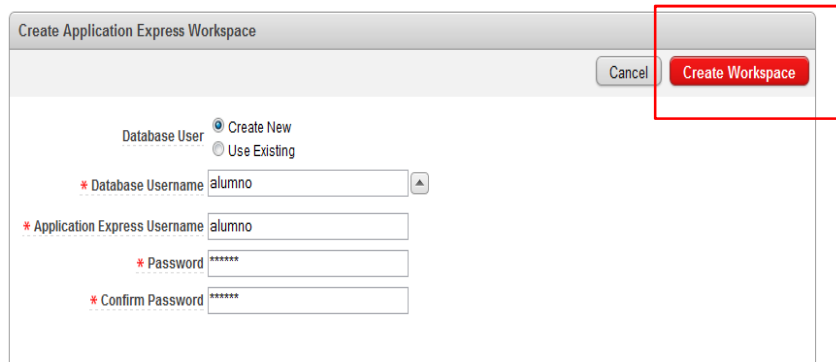
Login as a database user which has been granted the DBA database role (for example, SYSTEM).



Bases de datos



Crear un nuevo usuario de trabajo



Create Application Express Workspace

Cancel Create Workspace

Database User ☒ Create New ☐ Use Existing

* Database Username alumno

* Application Express Username alumno

* Password *****

* Confirm Password *****



Bases de datos



2. SQL Developer

- El objetivo fundamental de SQL Developer es proporcionar una interfaz más amigable para la consulta y programación de la BD Oracle.
- La funcionalidad disponible en SQL Developer es sólo parte de la disponible a través de comandos en SQL*Plus, pero se corresponde con las tareas más habituales de interacción, programación y depuración de código sobre la base de datos.



Bases de datos

The screenshot shows the Oracle SQL Developer website. At the top, there is a search bar and a 'Technology Network' link. Below this is a navigation bar with links: shortcuts, GETTING STARTED, DOWNLOADS, DOCUMENTATION, FORUMS, ARTICLES, SAMPLE CODE, and TUTORIALS. The main content area features a large section for 'Oracle SQL Developer' with a description: 'Oracle SQL Developer is a free and fully supported graphical tool for database development. With SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, as well as create and save your own. SQL Developer enhances productivity and simplifies your database development tasks.' To the right of this section is a 'FREE DOWNLOAD' button and a link to 'Download (09 June 08) Oracle SQL Developer 1 (Build 54.40)'. Below the main section, there are several links: 'What is Oracle SQL Developer?', 'SQL Developer OTN Forum', 'SQL Developer Exchange', 'SQL Developer 1.5', 'White papers & Supporting Documents (Oct 08)', and 'Online Demonstrations'. On the far right, there is a 'Testimonials' section with a quote: 'We've given up all of our life for other tools. We simply do need them anymore. Oracle Developer does it all for us. I saved a lot of money because free. It's also given our development staff a standard and they love it. It's made tri...'. At the bottom left, there is a small logo and the text 'INFORMÁTICA'. At the bottom right, the text 'Bases de datos' is visible.

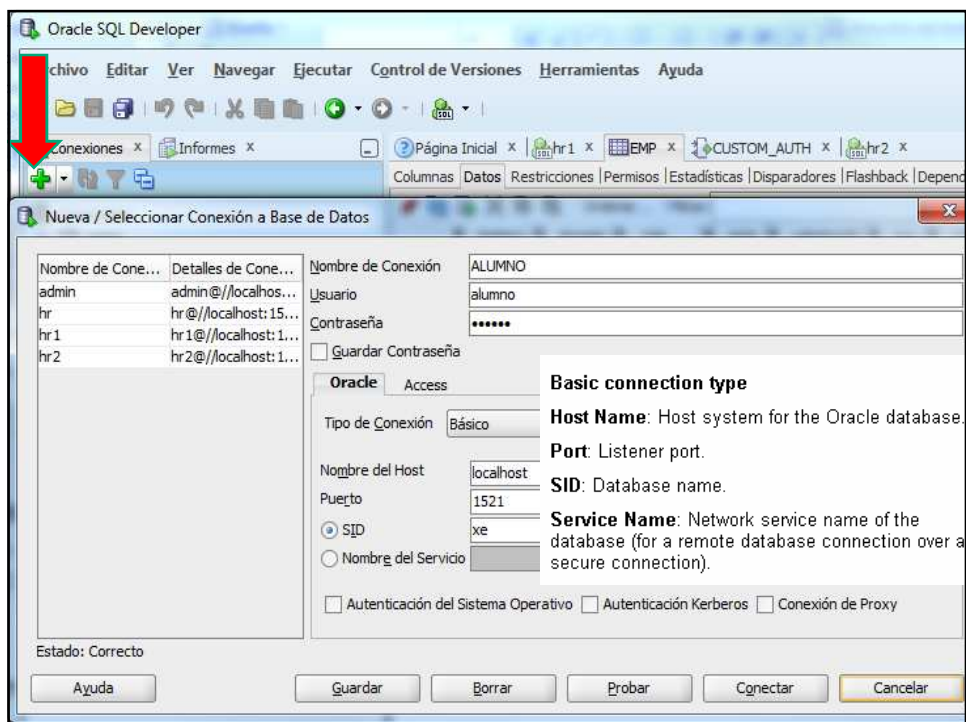


Instalación de SQL Developer

- Descomprimir el archivo ZIP descargado y ejecutar sqldeveloper.exe.
- Nos conectaremos a Oracle a través SQLDeveloper con los siguientes datos.



Bases de datos





3. Diseño físico



Bases de datos



CONCEPTOS

- Utilizaremos el lenguaje DDL de descripción de datos.
- Órdenes CREATE, DROP y ALTER (Creación, borrado o modificación de objetos de la BD)



Bases de datos



3.1. TIPOS DE DATOS



Bases de datos

PALABRA CLAVE	TIPO	EJEMPLOS
VARCHAR2(tamaño)	Variables alfanuméricas, el tamaño expresado será el número máximo de caracteres de ese campo.	124555
VARCHAR(tamaño)		1234RTT Alfonso Rodríguez =,%i8
NUMBER (precisión,escala)	Variable numéricas. La precisión es el número máximo de dígitos a la izquierda de la coma decimal, y la escala es el número máximo de dígitos a la derecha de la coma decimal La precisión máxima es 38.	56
NUMBER		45,67
NUMBER(precisión)		2232,90
		56E8 -90 0,89E-7
DATE	Fechas	2006/12/31
TIMESTAMP	Fechas en Año, Mes y Día y horas, minutos y segundos	2006-10-03 12:09:24
BLOB	Un objeto binario de hasta 4 gigabytes	Una foto Un sonido Un video
CLOB	Una cadena de caracteres de hasta 4 gigabytes	
BFILE	Un localizador a un fichero fuera de la base de datos	
LONG	Una cadena de caracteres muy larga de hasta 2 gigabytes.	Un capítulo de un libro.
RAW(tamaño)	Un objeto binario de hasta 2000 bytes	Una foto pequeña.





VARCHAR2 (tamaño)

- Permite almacenar cadenas de caracteres de longitud variable.
- La longitud máxima que se puede definir en una columna VARCHAR2 es de 4.000 caracteres o bytes y la mínima 1.
- La longitud de la cadena se define entre paréntesis. Por ejemplo: APELLIDO VARCHAR2 (10); que define la columna APELLIDO con una longitud máxima de 10 caracteres.
- Si se introduce una cadena de menor longitud que la que está definida, se almacenará con esa longitud y no se rellenará con blancos ni con ningún otro carácter a la derecha hasta completar la longitud definida.
- Si se introduce una cadena de mayor longitud a la fijada, Oracle devolverá un error.
- **El tipo VARCHAR2 necesita menos espacio que el tipo CHAR para almacenar cadenas de caracteres.**
- Es aconsejable utilizarla para cadenas cuyas longitudes difieran bastante de unas filas a otras, por ejemplo nombre, direcciones,



Bases de datos



CHAR(tamaño)

- Almacena caracteres de longitud fija, entre 1 y 2000 caracteres. La longitud de la cadena se define entre paréntesis. Por ejemplo: TEMA CHAR (15), que define la columna TEMA con una longitud de 15 caracteres.
- Las columnas tienen una longitud fija.
- Si se introduce una cadena de menor longitud que la definida, se rellenará con blancos a la derecha hasta que quede completa.
- Si se introduce una cadena de mayor longitud que la fijada, Oracle devolverá un error.
- Es aconsejable utilizarlo en campos que vayan a tener siempre la misma longitud (NIF, CIF,...)



Bases de datos



LONG

- Almacena cadenas de caracteres de longitud variable que contengan hasta 2 gigabytes de información. Se puede usar este tipo para almacenar textos muy grandes. Este tipo de dato está sujeto a algunas restricciones:
 - Sólo se puede definir una columna LONG por tabla.
 - No pueden aparecer en restricciones de integridad (*constraints*).
 - No sirve para indexar.
 - Una función almacenada no puede devolver un valor LONG.
 - No se puede utilizar como argumento de funciones.
 - No es posible su uso en cláusulas WHERE, GROUP BY, ORDER BY, CONNECT BY o DISTINCT, ni con operaciones de UNION, INTERSECT y MINUS.
 - No se puede referenciar como subconsulta en la creación de tablas ni inserción de filas.



Bases de datos



RAW y LONG RAW

- Estos tipos sirven para almacenar datos binarios. RAW es similar al tipo VARCHAR2, con la diferencia de que maneja cadenas de bytes en lugar de cadenas de caracteres. El tipo LONG RAW es similar al tipo LONG; se emplea para el almacenamiento de gráficos, sonidos, etc.
- Los tipos RAW soportan indexación, mientras que los LONG RAW no. El tamaño máximo para el tipo de datos RAW es de 255 bytes y para el tipo LONG RAW, 2 Gigabytes.



Bases de datos



ROWID

- Cadena hexadecimal que representa la dirección de una fila en su tabla. Cada fila de una tabla tiene una dirección que la identifica de forma única. Podemos consultar la dirección preguntando por la pseudocolumna ROWID.
- Esta columna utiliza una representación binaria de la localización física de la fila. Cuando se solicita esa columna, se convierte la representación binaria a un dato hexadecimal representado como VARCHAR2. Este dato se divide en tres campos: bloque.fila. fichero.
- Bloque es el bloque de datos en el que está contenida la fila, fila es el lugar que ocupa la fila dentro del bloque, fichero es el fichero de datos que contiene la fila.
- Consideremos este valor de ROWID: 0000003F.0000.0002. Significa que la primera fila (0000) está almacenada en el bloque Oracle 0000003F del fichero de datos identificado como 0002.

Bases de datos

Left Query: `select * from alum`

NOMBRE	EDAD	LOCALIDAD
JUAN	18	COSLADA
PEDRO	19	COSLADA
ANA	17	ALCALA
LUISA	18	TORREJÓN
MARÍA	20	MADRID
ERNESTO	21	MADRID

6 rows selected

Right Query: `select rowid from alum`

ROWID
AAAA2zAAEAAAAIoAAA
AAAA2zAAEAAAAIoAAB
AAAA2zAAEAAAAIoAAC
AAAA2zAAEAAAAIoAAD
AAAA2zAAEAAAAIoAAE
AAAA2zAAEAAAAIoAAF
AAAA2zAAEAAAAIoAAG

7 rows selected

Bases de datos



CLOB, NCLOB, BLOB

- **Tipo CLOB**
Objeto de gran tamaño de caracteres con una longitud de hasta 4 Gb
- **Tipo NCLOB**
Igual que CLOB pero para conjuntos de caracteres multibyte
- **Tipo BLOB**
Objeto de gran tamaño binario con una longitud de hasta 4 Gb



Bases de datos



NCHAR y NVARCHAR2

- Similares a CHAR Y VARCHAR2 solo que el tamaño de caracteres depende del juego de caracteres elegido. El máximo tamaño es de 4000 bytes.



Bases de datos



NUMBER(precisión, escala)

- Con este tipo se almacenan datos numéricos, tanto enteros como decimales, con o sin signo. Soporta hasta 38 dígitos de precisión es decir el rango va de 1 a 38 y escala representa el número de dígitos a la derecha del punto decimal el rango es de -84 a 127



Bases de datos



NUMBER(precisión, escala)

- Si se especifica una escala negativa, el número es redondeado tantos dígitos a la izquierda del punto decimal como se indicó en la escala.
- Si el número no tiene decimales, se puede omitir la escala.



Bases de datos



Es posible especificar números enteros usando este otro formato: **NUMBER (precisión)** , que representa un número entero de precisión indicada en el parámetro precisión, y escala 0 (número de dígitos decimales: 0). Equivale a **NUMBER (precisión, 0)**.



Bases de datos

Dato actual	Formato	Almacenamiento
Hernández 7456123.89	NUMBER	7456123.89
7456123.89	NUMBER(9)	7456124
7456123.89	NUMBER(9,2)	7456123.89
7456123.89	NUMBER(9,1)	7456123.9
7456123.8	NUMBER(6)	ERROR-valor mayor que el que permite la precisión especificada para esta columna
7456123.8	NUMBER(15,1)	7456123.8
7456123.89	NUMBER(7,-2)	7456100
7456123.89	NUMBER(-7,2)	ERROR-Especificador de precisión numérica está fuera de rango (1 a 38)



Bases de datos



TIPO DATE

- Se usa para almacenar información de fechas y horas. Para cada tipo DATE se almacena la siguiente información: Siglo/Año/Mes/Día/Hora/Minutos/Segundos.
- Por omisión, el valor para el formato de la fecha se especifica con el parámetro NLS_DATE_FORMAT, y es una cadena de caracteres como ésta: 'DD/MM/YY', que representa: día del mes/dos dígitos del mes/dos últimos dígitos del año.
- El formato de la fecha se puede cambiar mediante la orden ALTER SESSION y variando el parámetro NLS_DATE_FORMAT. Las funciones de fecha y este parámetro se verán después.
- Por ejemplo, para declarar una columna de tipo DATE utilizaremos: `FECHA_ALT DATE`



Bases de datos



3.2. CREACIÓN DE UNA TABLA

Antes de crear una tabla hay que planificar

- Su nombre (max 30 caracteres, el primero debe ser alfabético solo puede haber letras y nºs o el caracter _)
- Nombre de cada columna:
- Tipo de dato y tamaño de cada campo.
- Restricciones de integridad



Bases de datos



3.2. CREACIÓN DE TABLAS. Restricciones

- Una restricción de integridad será una regla que restringe el rango de valores para una o más columnas de la tabla. Si se produce un fallo, la base de datos tiene la capacidad de deshacer o cancelar cualquier transacción sospechosa.
- Existe también la restricción de integridad referencial, la cual garantiza que los valores de una columna/s de una tabla dependan de los valores de otra columna/s de otra tabla.
- El objetivo de las restricciones es que las aplicaciones o los usuarios que van a manipular los datos tengan menos trabajo, y que sea Oracle el que realice la mayor parte de las tareas de mantenimiento de la integridad de la base de datos

Bases de datos



La orden CREATE TABLE Permite definir distintos de restricciones sobre una tabla mediante la cláusula CONSTRAINT

- claves primarias
- claves ajenas
- obligatoriedad
- valores por defecto
- verificación de condiciones
- La cláusula CONSTRAINT puede restringir una sola columna(restricción de columna) o un grupo de columnas (restricción de tabla)
- Hay dos formas de especificar restricciones:
 - como parte de la definición de columnas
 - al final, una vez especificadas todas las columnas



Bases de datos



SINTAXIS con restricción de columna

```
CREATE TABLE nombre_tabla (
  Columna1  tipo_de_dato
    [CONSTRAINT nombrerestricción]
    [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT
    valor]
    [REFERENCES Nombretabla [(columna[,columna])]]
    [ON DELETE CASCADE|SET NULL]]
    [CHECK condición],
  Columna2  tipo_de_dato
  .....
) [TABLESPACE espacio_de_tabla];
```



Bases de datos



EJEMPLO1

```
CREATE TABLE EMPLEADO
(
  NOMBRE VARCHAR2(25),
  EDAD NUMBER,
  COD_PROVINCIA NUMBER(2) )
```

Este ejemplo crea la tabla empleado con 3 campos

Nombre de tipo varchar2

Edad de tipo numérico

Cod_provincia de tipo numérico



Bases de datos



Ejemplo2. PRIMARY KEY

Para definir el campo **nombre** como clave primaria escribiríamos:

```
CREATE TABLE EMPLEADO  
(  
  NOMBRE VARCHAR2(25) primary key,  
  EDAD NUMBER,  
  COD_PROVINCIA NUMBER(2)  
)
```



Bases de datos



Ejemplo2

Si queremos que un campo sea obligatorio, por ejemplo **edad**

```
CREATE TABLE EMPLEADO  
(  
  NOMBRE VARCHAR2(25) primary key,  
  EDAD NUMBER NOT NULL,  
  COD_PROVINCIA NUMBER(2)  
)
```



Bases de datos



Ejemplo3. REFERENCES

Supongamos que tenemos creada la tabla **provincias(#cod_provi, nombre, cod_postal)** y el campo **cod_provi** de la tabla empleado es clave ajena de **cod_provi**.

(Por lo tanto el valor de **cod_provi** en la tabla empleado será o bien NULL o bien tendrá que existir en la tabla provincias.)



Bases de datos



Ejemplo3

provincias(#cod_provi, nombre, cod_postal)

CREATE TABLE EMPLEADO

(

NOMBRE VARCHAR2(25) primary key,

EDAD NUMBER NOT NULL,

COD_PROVINCIAS NUMBER(2) REFERENCES
PROVINCIAS ON DELETE CASCADE



Bases de datos



Ejemplo 4. DEFAULT

Si queremos que el campo edad tenga como valor por defecto 18

```
CREATE TABLE EMPLEADO  
(  
  NOMBRE VARCHAR2(25) primary key,  
  EDAD NUMBER DEFAULT 18 NOT NULL,  
  COD_PROVINCIA NUMBER(2) REFERENCES  
    PROVINCIAS ON DELETE CASCADE  
)
```



Bases de datos



Ejemplo 5. CHECK

Si queremos que el campo edad además tenga que estar comprendido entre 18 y 65

```
CREATE TABLE EMPLEADO  
(  
  NOMBRE VARCHAR2(25) primary key,  
  EDAD NUMBER DEFAULT 18 NOT NULL CHECK  
    (EDAD BETWEEN 18 AND 65),  
  COD_PROVINCIA NUMBER(2) REFERENCES  
    PROVINCIAS ON DELETE CASCADE  
)
```



Bases de datos



Nombre de las restricciones

- Hasta ahora no les hemos dado un nombre a las restricciones, por defecto Oracle asigna automáticamente un nombre SYS_C00n , donde n es un nº asignado automáticamente.



Bases de datos

```

1 CREATE TABLE EMPLEADO5
2 (  NOMBRE          VARCHAR2(25) primary key,
3   EDAD   NUMBER DEFAULT 18 NOT NULL CHECK (EDAD BETWEEN 18 AND 35),
4   COD_PROVINCIAS  NUMBER(2) REFERENCES PROVINCIAS ON DELETE CASCADE );
5
6 insert into empleado5(nombre, edad) values ('juan', 19);
7 insert into empleado5(nombre, edad) values ('juan', 25);

```

Results Script Output Explain Autotrace DBMS Output OWA Output

CREATE TABLE succeeded.
1 rows inserted

Error starting at line 7 in command:
insert into empleado5(nombre, edad) values ('juan', 25)

Error report:
SQL Error: ORA-00001: restricción única (CASE02.SYS_C005544) violada
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

bases de datos

```

1 CREATE TABLE EMPLEADO7
2 ( NOMBRE VARCHAR2(25) constraint CP_emple7 primary key,
3  EDAD  NUMBER DEFAULT 18 NOT NULL CHECK (EDAD BETWEEN 18 AND 35),
4  COD_PROVINCIA NUMBER(2) REFERENCES PROVINCIAS ON DELETE CASCADE );
5
6 insert into empleado7(nombre, edad) values ('juan', 19);
7 insert into empleado7(nombre, edad) values ('juan', 25);

```

Results Script Output Explain Autotrace DBMS Output OWA Output

CREATE TABLE succeeded.
1 rows inserted

Error starting at line 7 in command:
insert into empleado7(nombre, edad) values ('juan', 25)

Error report:
SQL Error: ORA-00001: restricción única (CASE02.CP_EMPLE7) violada
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.



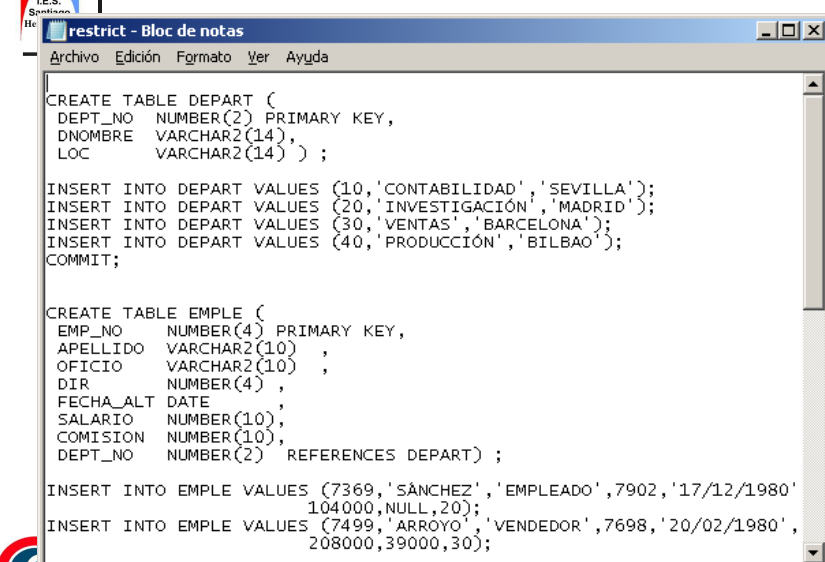
ON DELETE

- Oracle implementa las opciones
ON DELETE CASCADE
ON DELETE SET NULL

Si no se especifica ninguna de éstas, la opción por defecto es un borrado restringido RESTRICT



BORRADO RESTRINGIDO



```

CREATE TABLE DEPART (
  DEPT_NO NUMBER(2) PRIMARY KEY,
  DNOMBRE VARCHAR2(14),
  LOC VARCHAR2(14) );

INSERT INTO DEPART VALUES (10,'CONTABILIDAD','SEVILLA');
INSERT INTO DEPART VALUES (20,'INVESTIGACIÓN','MADRID');
INSERT INTO DEPART VALUES (30,'VENTAS','BARCELONA');
INSERT INTO DEPART VALUES (40,'PRODUCCIÓN','BILBAO');
COMMIT;

CREATE TABLE EMPL (
  EMP_NO NUMBER(4) PRIMARY KEY,
  APELLIDO VARCHAR2(10),
  OFICIO VARCHAR2(10),
  DIR NUMBER(4),
  FECHA_ALT DATE,
  SALARIO NUMBER(10),
  COMISION NUMBER(10),
  DEPT_NO NUMBER(2) REFERENCES DEPART );

INSERT INTO EMPL VALUES (7369,'SÁNCHEZ','EMPLEADO',7902,'17/12/1980',
104000,NULL,20);
INSERT INTO EMPL VALUES (7499,'ARROYO','VENDEDOR',7698,'20/02/1980',
208000,39000,30);
  
```

	1	2	3	4	5	6	7	8	9	10	11
	...	APELLIDO	...	OFICIO	...	DIR	...	SALARIO	...	DEPT_...	
1	...	CEREZO	DIRECTOR	7839	...	7839	...	318500	...	10	
2	...	REY	PRESIDENTE	(null)	650000	...	10	
3	...	MUÑOZ	EMPLEADO	7782	...	7782	...	169000	...	10	
4	...	JIMÉNEZ	DIRECTOR	7839	...	7839	...	386750	...	20	
5	...	FERNÁNDEZ	ANALISTA	7566	...	7566	...	390000	...	20	
6	...	ALONSO	EMPLEADO	7788	...	7788	...	143000	...	20	
7	...	SÁNCHEZ	EMPLEADO	7902	...	7902	...	104000	...	20	
8	...	GIL	ANALISTA	7566	...	7566	...	390000	...	20	
9	...	SALA	VENDEDOR	7698	...	7698	...	162500	...	30	
0	...	TOVAR	VENDEDOR	7698	...	7698	...	195000	0	30	
1	...	ARROYO	VENDEDOR	7698	...	7698	...	208000	...	30	
2	...	JIMENO	EMPLEADO	7698	...	7698	...	123500	...	30	
3	...	NEGRO	DIRECTOR	7839	...	7839	...	370500	...	30	
4	...	MARTÍN	VENDEDOR	7698	...	7698	...	162500	...	30	


DEPT_NO	DNOMBRE	LOC
10	CONTABILIDAD	SEVILLA
20	INVESTIGACIÓN	MADRID
30	VENTAS	BARCELONA
40	PRODUCCIÓN	BILBAO

Si borramos el departamento 40 de la tabla depart, se borrará sin problema ya que no hay empleados de alta en ese departamento.

Si tratamos de borrar el departamento 10 por ejemplo, obtendremos el siguiente mensaje de error y no podremos borrar ese departamento

```

DELETE FROM "CASE02"."DEPART" WHERE ROWID = 'AAAM5JA.AEAAAAMGAAC' AND ORA_ROWSCN = '899876
One error saving changes to table "CASE02"."DEPART":
Row 3: ORA-02292: restricción de integridad (CASE02.SYS_C005561) violada - registro secundario encontrado
  
```



BORRADO EN CASCADA

PRUEBACASCADE - Bloc de notas

```

CREATE TABLE DEPART (
  DEPT_NO NUMBER(2) PRIMARY KEY,
  DNOMBRE VARCHAR2(14),
  LOC VARCHAR2(14) );

INSERT INTO DEPART VALUES (10,'CONTABILIDAD','SEVILLA');
INSERT INTO DEPART VALUES (20,'INVESTIGACIÓN','MADRID');
INSERT INTO DEPART VALUES (30,'VENTAS','BARCELONA');
INSERT INTO DEPART VALUES (40,'PRODUCCIÓN','BILBAO');
COMMIT;

CREATE TABLE EMPL (
  EMP_NO NUMBER(4) PRIMARY KEY,
  APELLIDO VARCHAR2(10) ,
  OFICIO VARCHAR2(10) ,
  DIR NUMBER(4) ,
  FECHA_ALT DATE ,
  SALARIO NUMBER(10),
  COMISION NUMBER(10),
  DEPT_NO NUMBER(2) NOT NULL REFERENCES DEPART ON DELETE CASCADE );

INSERT INTO EMPL VALUES (7369,'SÁNCHEZ','EMPLEADO',7902,'17/12/1980',
104000,NULL,20);
INSERT INTO EMPL VALUES (7499,'ARROYO','VENDEDOR',7698,'20/02/1980',
208000,39000,30);
INSERT INTO EMPL VALUES (7521,'SALA','VENDEDOR',7698,'22/02/1981',

```

	APELLIDO	OFICIO	DIR	SALARIO	DEPT...
1	CEREZO	DIRECTOR	7839	318500	10
2	REY	PRESIDEN...	(null)	650000	10
3	MUÑOZ	EMPLEADO	7782	169000	10
4	JIMÉNEZ	DIRECTOR	7839	386750	20
5	FERNÁNDEZ	ANALISTA	7566	390000	20
6	ALONSO	EMPLEADO	7788	143000	20
7	SÁNCHEZ	EMPLEADO	7902	104000	20
8	GIL	ANALISTA	7566	390000	20
9	SALA	VENDEDOR	7698	162500	30
0	TOVAR	VENDEDOR	7698	195000	30
1	ARROYO	VENDEDOR	7698	208000	30
2	JIMENO	EMPLEADO	7698	123500	30
3	NEGRO	DIRECTOR	7839	370500	30
4	MARTÍN	VENDEDOR	7698	162500	30

DEPT_NO	DNOMBRE	LOC
10	CONTABILID...	SEVILLA
20	INVESTIGACI...	MADRID
30	VENTAS	BARCELONA
40	PRODUCCIÓN	BILBAO

Si borramos el departamento 10 de la tabla depart, automáticamente en la tabla emple desaparecerán las filas correspondientes a ese departamento, en concreto las filas 1, 2 y 3, obteniendo el resultado de la siguiente diapositiva





SINTAXIS con restricción de columna

```

CREATE TABLE nombre_tabla (
  Columna1    tipo_de_dato
    [CONSTRAINT nombrerestricción]
    [NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor]
    [REFERENCES Nombretabla [(columna[,columna])]]
    [ON DELETE CASCADE|SET NULL]
    [CHECK condición],
  Columna2    tipo_de_dato
    .....
) [TABLESPACE espacio_de_tabla];
  
```



Bases de datos



Columns Data Constraints Grants Statistics Column Statistics Triggers Dependence

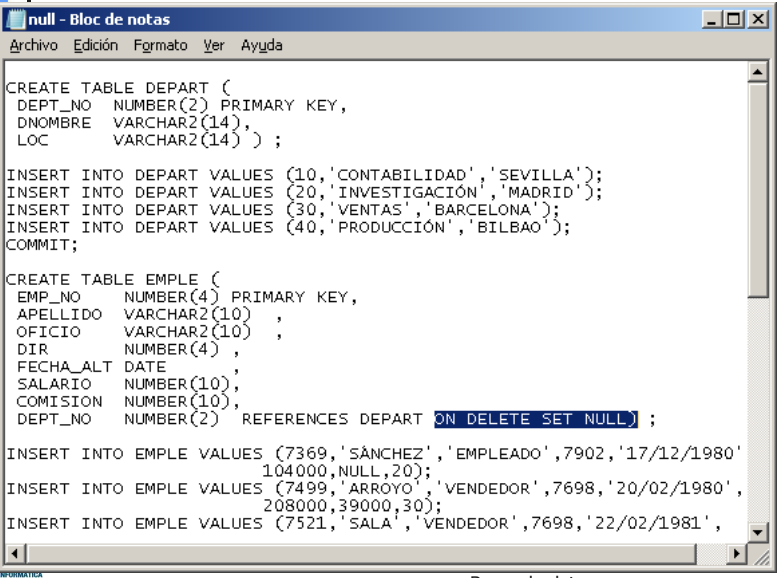
Sort... Filter: Enter Where Clause

		APELLIDO	OFICIO	DIR	SALARIO	DEPT_...
1	...	SÁNCHEZ	EMPLEADO	7902	104000	20
2	...	ARROYO	VENDEDOR	7698	208000	30
3	...	SALA	VENDEDOR	7698	162500	30
4	...	JIMÉNEZ	DIRECTOR	7839	386750	20
5	...	MARTÍN	VENDEDOR	7698	162500	30
6	...	NEGRO	DIRECTOR	7839	370500	30
7	...	GIL	ANALISTA	7566	390000	20
8	...	TOVAR	VENDEDOR	7698	195000	0 30
9	...	ALONSO	EMPLEADO	7788	143000	20
10	...	JIMENO	EMPLEADO	7698	123500	30
11	...	FERNÁNDEZ	ANALISTA	7566	390000	20



bases de datos

BORRADO CON PUESTA A NULOS



```

CREATE TABLE DEPART (
  DEPT_NO NUMBER(2) PRIMARY KEY,
  DNOMBRE VARCHAR2(14),
  LOC VARCHAR2(14) );

INSERT INTO DEPART VALUES (10,'CONTABILIDAD','SEVILLA');
INSERT INTO DEPART VALUES (20,'INVESTIGACIÓN','MADRID');
INSERT INTO DEPART VALUES (30,'VENTAS','BARCELONA');
INSERT INTO DEPART VALUES (40,'PRODUCCIÓN','BILBAO');
COMMIT;

CREATE TABLE EMPL (
  EMP_NO NUMBER(4) PRIMARY KEY,
  APELLIDO VARCHAR2(10) ,
  OFICIO VARCHAR2(10) ,
  DIR NUMBER(4) ,
  FECHA_ALT DATE ,
  SALARIO NUMBER(10),
  COMISION NUMBER(10),
  DEPT_NO NUMBER(2) REFERENCES DEPART ON DELETE SET NULL );

INSERT INTO EMPL VALUES (7369,'SÁNCHEZ','EMPLEADO',7902,'17/12/1980',
  104000,NULL,20);
INSERT INTO EMPL VALUES (7499,'ARROYO','VENDEDOR',7698,'20/02/1980',
  208000,39000,30);
INSERT INTO EMPL VALUES (7521,'SALA','VENDEDOR',7698,'22/02/1981',
  162500,0,30);
  
```

Bases de datos

	APELLIDO	OFICIO	DIR	SALARIO	DEPT...
1	CEREZO	DIRECTOR	7839	318500	10
2	REY	PRESIDEN...	(null)	650000	10
3	MUÑOZ	EMPLEADO	7782	169000	10
4	JIMÉNEZ	DIRECTOR	7839	386750	20
5	FERNÁNDEZ	ANALISTA	7566	390000	20
6	ALONSO	EMPLEADO	7788	143000	20
7	SÁNCHEZ	EMPLEADO	7902	104000	20
8	GIL	ANALISTA	7566	390000	20
9	SALA	VENDEDOR	7698	162500	30
0	TOVAR	VENDEDOR	7698	195000	30
1	ARROYO	VENDEDOR	7698	208000	30
2	JIMENO	EMPLEADO	7698	123500	30
3	NEGRO	DIRECTOR	7839	370500	30
4	MARTÍN	VENDEDOR	7698	162500	30

DEPT_NO	DNOMBRE	LOC
10	CONTABILID...	SEVILLA
20	INVESTIGACI...	MADRID
30	VENTAS	BARCELONA
40	PRODUCCIÓN	BILBAO

Si borramos el departamento 10 de la tabla depart, automáticamente en la tabla emple las filas donde aparezca dicho departamento pondrán el valor NULL en el campo dept_no, en concreto en las filas 1, 2 y 3, obteniendo el resultado de la siguiente diapositiva

Bases de datos

oracle EMPLE null.sql PRUEBACASCADE.SQL restric

Columns Data Constraints Grants Statistics Column Statistics Triggers Dependence

Sort... Filter: Enter Where Clause

		APELLIDO	OFICIO	DIR	SALARIO	DEPT_...
1	...	CEREZO	DIRECTOR	7839 ...	318500 ...	(null)
2	...	REY	PRESIDEN...	(null) ...	650000 ...	(null)
3	...	MUÑOZ	EMPLEADO	7782 ...	169000 ...	(null)
4	...	ARROYO	VENDEDOR	7698 ...	208000 ...	30
5	...	SALA	VENDEDOR	7698 ...	162500 ...	30
6	...	JIMENO	EMPLEADO	7698 ...	123500 ...	30
7	...	TOVAR	VENDEDOR	7698 ...	195000 0	30
8	...	MARTÍN	VENDEDOR	7698 ...	162500 ...	30
9	...	NEGRO	DIRECTOR	7839 ...	370500 ...	30
10	...	ALONSO	EMPLEADO	7788 ...	143000 ...	20
11	...	SÁNCHEZ	EMPLEADO	7902 ...	104000 ...	20
12	...	GIL	ANALISTA	7566 ...	390000 ...	20
13	...	JIMÉNEZ	DIRECTOR	7839 ...	386750 ...	20
14	...	FERNÁNDEZ	ANALISTA	7566 ...	390000 ...	20



- Oracle tampoco implementa directamente las opciones ON UPDATE CASCADE|SET NULL O SET DEFAULT
- Por defecto es una modificación restringida RESTRICT, de forma que no permite modificar el valor de una clave primaria si existen filas en otra o la misma tabla que la referencian





Por ejemplo: Podemos modificar el valor 40 en la tabla depart correspondiente al departamento de producción, ya que en la tabla emple no hay empleados en ese departamento, pero no nos permitirá cambiar el valor del departamento 10 ya que existen empleados en ese departamento

Al intentar confirmar la modificación da un error y hay que deshacer el cambio y dejar de nuevo el valor 10 en lugar de 13

	DEPT_NO	DNOMBRE	LOC
*1	13	CONTABILID...	SEVILLA
2	20	INVESTIGACI...	MADRID
3	30	VENTAS	BARCELONA
4	40	PRODUCCIÓN	BILBAO



- Si las restricciones se definen al final de la definición de columnas, entonces se puede hacer referencia a varias columnas en una única restricción (por ejemplo declarando dos columnas como clave primaria)





SINTAXIS con restricción de tabla

```
CREATE TABLE nombre_tabla (
  Columna1 tipo_de_dato,
  Columna2 tipo_de_dato,
  Columna3 tipo_de_dato,
  .....
  [CONSTRAINT nombrerestricción
    { [ UNIQUE] | [PRIMARY KEY] (columna [, columna] ) },
  [CONSTRAINT nombrerestricción
    [FOREIGN KEY (columna [,columna])
      REFERENCES nombretabla [(columna[,columna])]
      [ON DELETE CASCADE|SET NULL]],
  [CONSTRAINT nombrerestricción
    [CHECK (condición)]
  .....
  [TABLESPACES espacio_de_tabla];
```



Bases de datos



Ejemplo

```
CREATE TABLE empleado (
  nombre VARCHAR2(25),
  edad NUMBER,
  cod_provincia NUMBER(2),
  CONSTRAINT pk_empleado PRIMARY KEY (nombre),
  CONSTRAINT ck_edad CHECK (edad BETWEEN 18 AND 35),
  CONSTRAINT fk_empleado FOREIGN KEY (cod_provincia)
    REFERENCES provincias ON DELETE CASCADE
);
```



Bases de datos



- El nombre de las restricciones es opcional, es un nombre único que define el propietario del objeto o por defecto, el sistema.
- Se asigna en el momento de definir la restricción. Por defecto, la denominación es : SYS_C00n. Resulta más fácil identificar una violación de una restricción si se le da un nombre al definirla.



Bases de datos



Ejemplo anterior sin nombre de restricciones

```
CREATE TABLE empleado2 (
  nombre VARCHAR2(25),
  edad NUMBER,
  cod_provincia    NUMBER(2),
  PRIMARY KEY (nombre),
  CHECK (edad BETWEEN 18 AND 35),
  FOREIGN KEY (cod_provincia)
  REFERENCES provincias ON DELETE
  CASCADE
);
```



Bases de datos



- Veamos ahora una a una los distintos tipos de restricciones



Bases de datos



PRIMARY KEY

- Una clave primaria dentro de una tabla es una columna o un conjunto de columnas que identifican unívocamente a cada fila
- Debe ser única, no nula y obligatoria
- Como máximo, podemos definir una clave primaria por tabla
- Esta clave se puede referenciar por una columna o columnas de otra tabla (clave ajena)
- Cuando se crea una clave primaria en una tabla se crea un índice que facilita el acceso a la tabla



Bases de datos

Las restricciones en una tabla pueden ser impuestas de dos formas:

A nivel de campo:

```
CREATE TABLE Empleados(
  DNI          VARCHAR(10)          PRIMARY KEY,
  Nombre       VARCHAR(20),
  Edad        NUMBER(2,0)
);
```

O a nivel de tabla:

```
CREATE TABLE Empleados(
  DNI          VARCHAR(10),
  Nombre       VARCHAR(20),
  Edad        NUMBER(2,0),
  PRIMARY KEY (DNI)
);
```

Cuando la clave primaria está formada por dos o más campos, la restricción solo puede imponerse a nivel de tabla.

```
CREATE TABLE Empleados_Puesto(
  DNI          VARCHAR(10),
  Cod_puesto   VARCHAR(5),
  Hora        TIMESTAMP,
  PRIMARY KEY (DNI,CodPuesto)
);
```



```
CREATE TABLE ARTICULOS
(
  ARTICULO VARCHAR2(30),
  CATEGORIA NUMBER(1),
  PESO NUMBER(1),
  COD_FABRICANTE CHAR(3),
  PVP NUMBER(3),
  CONSTRAINT CP_ARTIC PRIMARY KEY (ARTICULO,
    CATEGORIA, PESO, COD_FABRICANTE),
  CONSTRAINT CA_ARTIC FOREIGN KEY(COD_FABRICANTE)
    REFERENCES FABRICANTE
)
```



Bases de datos

Clave ajenas. FOREIGN KEY



- Una clave ajena está formada por una o varias columnas que están asociadas a una clave primaria o bien unique, de otra o de la misma tabla.
- Se pueden definir varias claves ajenas.
- El valor de la columna o columnas que son claves ajenas debe ser NULL o igual a un valor de la clave referenciada (regla de integridad referencial)
- Si en
- Si deseamos un borrado en cascada o puesta a nulos añadiremos
ON DELETE CASCADE|SET NULL



Bases de datos



- La opción más común es que la clave ajena haga referencia a una clave primaria de otra o la misma tabla, en este caso no es necesario especificar **la columna de la tabla referenciada**
[FOREIGN KEY (columna [,columna])
REFERENCES nombretabla
[(columna[,columna])]



Bases de datos



```
CREATE TABLE FABRICANTE
(CODIGO CHAR(3) primary key,
NOMBRE VARCHAR2(20));
```

```
CREATE TABLE ARTICULOS2
(ARTICULO VARCHAR2(30),
CATEGORIA NUMBER(1),
PESO NUMBER(1),
COD_FABRICANTE CHAR(3),
CONSTRAINT CP_ARTIC1 PRIMARY KEY (ARTICULO,
CATEGORIA, PESO, COD_FABRICANTE),
CONSTRAINT CA_ARTIC1 FOREIGN
KEY(COD_FABRICANTE) REFERENCES FABRICANTE )
```

No es necesario indicar en la última fila
...REFERENCES FABRICANTE(CODIGO) ya que por defecto
hace referencia a la clave primaria



Bases de datos




```
CREATE TABLE FABRICANTE7
(CODIGO CHAR(3) PRIMARY KEY,
CODIGO2 CHAR(3) UNIQUE,
NOMBRE VARCHAR2(20));
```

```
CREATE TABLE ARTICULOS7
(ARTICULO VARCHAR2(30),
CATEGORIA NUMBER(1),
PESO NUMBER(1),
COD_FABRICANTE CHAR(3),
CONSTRAINT CP_ARTIC2 PRIMARY KEY (ARTICULO,
CATEGORIA, PESO, COD_FABRICANTE),
CONSTRAINT CA_ARTIC2 FOREIGN KEY(COD_FABRICANTE)
REFERENCES FABRICANTE7(CODIGO2) )
```

Si queremos que la clave ajena haga referencia a una columna
única en lugar de la clave primaria, debemos especificarlo
en la cláusula REFERENCES



Bases de datos



```


1 CREATE TABLE FABRICANTE
2 (CODIGO CHAR(3) UNIQUE,
3 CODIGO2 CHAR(3) UNIQUE,
4 NOMBRE VARCHAR2(20));
5
6 CREATE TABLE ARTICULOS
7 (ARTICULO VARCHAR2(30),
8 CATEGORIA NUMBER(1),
9 PESO NUMBER(1),
10 COD_FABRICANTE CHAR(3),
11 CONSTRAINT CP_ARTIC1 PRIMARY KEY (ARTICULO, CATEGORIA, PESO, COD_FABRICANTE),
12 CONSTRAINT CA_ARTIC1 FOREIGN KEY(COD_FABRICANTE) REFERENCES FABRICANTE
13 )
14

```

Results Script Output Explain Autotrace DBMS Output OWVA Output

Error starting at line 6 in command:
 REATE TABLE ARTICULOS
 ARTICULO VARCHAR2(30),
 CATEGORIA NUMBER(1),
 PESO NUMBER(1),
 COD_FABRICANTE CHAR(3),
 CONSTRAINT CP_ARTIC1 PRIMARY KEY (ARTICULO, CATEGORIA, PESO, COD_FABRICANTE),
 CONSTRAINT CA_ARTIC1 FOREIGN KEY(COD_FABRICANTE) REFERENCES FABRICANTE

Error at Command Line:12 Column:60
 Error report:
 SQL Error: ORA-02268: la tabla a la que se hace referencia no tiene ninguna clave primaria
 02268. 00000 - "referenced table does not have a primary key"
 Cause: The referenced table does not have a primary key.
 Action: Specify explicitly the referenced table unique key.



```

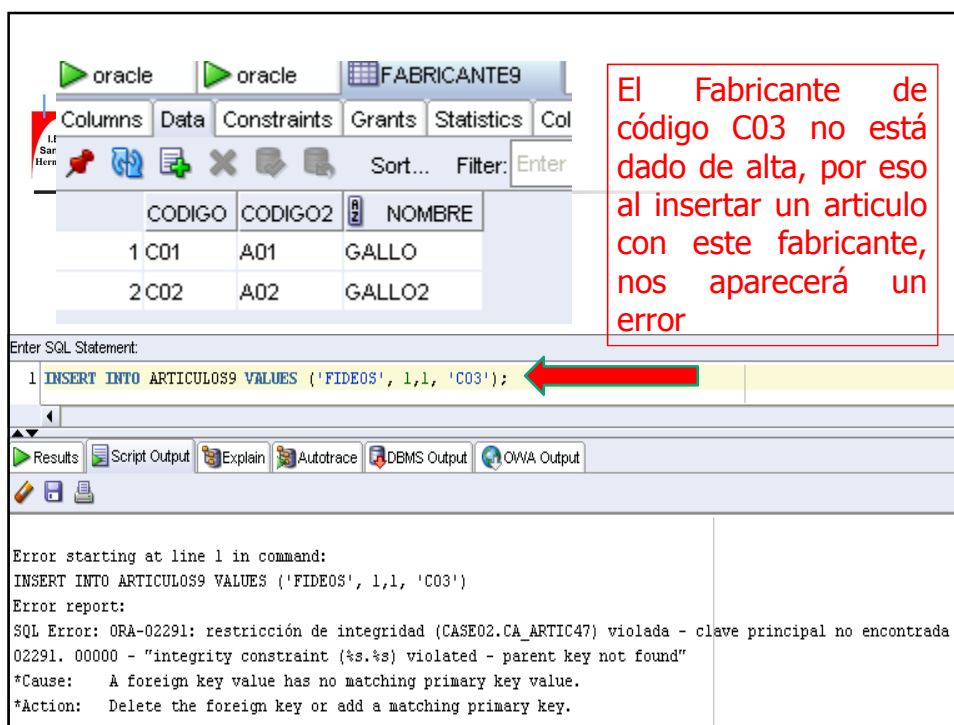
CREATE TABLE FABRICANTE9
(CODIGO CHAR(3) PRIMARY KEY,
CODIGO2 CHAR(3) UNIQUE,
NOMBRE VARCHAR2(20));

CREATE TABLE ARTICULOS9
(ARTICULO VARCHAR2(30),
CATEGORIA NUMBER(1),
PEO NUMBER(1),
COD_FABRICANTE CHAR(3),
CONSTRAINT CP_ARTIC47 PRIMARY KEY (ARTICULO, CATEGORIA, PESO, COD_FABRICANTE),
CONSTRAINT CA_ARTIC47 FOREIGN KEY(COD_FABRICANTE) REFERENCES FABRICANTE9
)

```

- Cuando insertemos valores en la tabla artículos, los valores dados para el código del fabricante deberán ser o bien nulo o estar dados de alta en la tabla FABRICANTES





The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'oracle', 'oracle', and 'FABRICANTE9'. Below these are tabs for 'Columns', 'Data', 'Constraints', 'Grants', 'Statistics', and 'Col'. A table with the following data is visible:

	CODIGO	CODIGO2	NOMBRE
1	C01	A01	GALLO
2	C02	A02	GALLO2

Below the table, the 'Enter SQL Statement:' area contains the following SQL command:

```
1 INSERT INTO ARTICULOS9 VALUES ('FIDEOS', 1,1, 'C03');
```

A red arrow points to the 'C03' value in the SQL command. To the right of the SQL command, a red box contains the following text:

El Fabricante de código C03 no está dado de alta, por eso al insertar un articulo con este fabricante, nos aparecerá un error

Below the SQL command, the 'Error report:' section shows the following error message:

```
SQL Error: ORA-02291: restricción de integridad (CASE02.CA_ARTIC47) violada - clave principal no encontrada
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:      A foreign key value has no matching primary key value.
*Action:     Delete the foreign key or add a matching primary key.
```

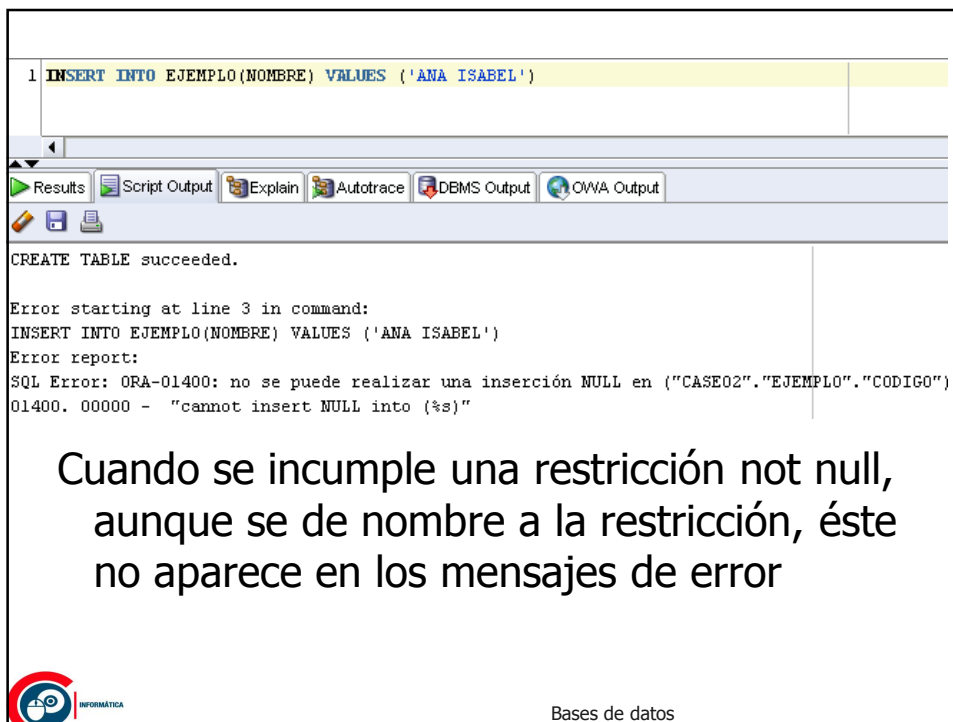


La restricción NOT NULL

- Es una restricción de columna, significa que ha de tener obligatoriamente un valor.

CREATE TABLE ejemplo (
 NOMBRE CHAR(20) NOT NULL,
 CODIGO CHAR(3) CONSTRAINT
 COD_NONULO NOT NULL)





1 **INSERT INTO EJEMPLO(NOMBRE) VALUES ('ANA ISABEL')**


Results Script Output Explain Autotrace DBMS Output OWA Output

CREATE TABLE succeeded.

Error starting at line 3 in command:
INSERT INTO EJEMPLO(NOMBRE) VALUES ('ANA ISABEL')

Error report:
 SQL Error: ORA-01400: no se puede realizar una inserción NULL en ("CASE02"."EJEMPLO"."CODIGO")
 01400. 00000 - "cannot insert NULL into (%s)"

Cuando se incumple una restricción not null, aunque se de nombre a la restricción, éste no aparece en los mensajes de error

 INFORMÁTICA


Bases de datos

Esta restricción acepta un nombre como en el caso de la clave primaria, pero no resulta muy útil porque para su modificación no se precisa un nombre de restricción.

```
CREATE TABLE Empleados(
  DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
  Nombre       VARCHAR(20)    NOT NULL,
  Edad        NUMBER(2,0)
);
```

```
CREATE TABLE Empleados(
  DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
  Nombre       VARCHAR(20)    NOT NULL,
  Apellidos    VARCHAR(30)    NOT NULL,
  Edad        NUMBER(2,0)
);
```

```
CREATE TABLE Empleados(
  DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
  Nombre       VARCHAR(20)    CONSTRAINT nn_nombre NOT NULL,
  Apellidos    VARCHAR(30)    NOT NULL,
  Edad        NUMBER(2,0)
);
```

 INFORMÁTICA

Bases de datos



Valores por defecto DEFAULT

- En el momento de crear una tabla, podemos asignar valores por defecto a las columnas. Si especificamos la cláusula DEFAULT a una columna, le proporcionamos un valor por omisión cuando el valor de la columna no se especifica en la cláusula INSERT
- Se pueden incluir: constantes, funciones SQL, SYSDATE. No se puede hacer referencia a columnas o a funciones PL/SQL



Bases de datos

Cuando una fila en un campo es normal que tenga siempre el mismo valor, conviene darle dicho valor por defecto. Esto quiere decir que al insertar una nueva fila, si no indicamos lo contrario el valor de la fila en ese campo será el predeterminado.

Por ejemplo:

```
CREATE TABLE Empleados(
DNI          VARCHAR(10)    PRIMARY KEY,
Nombre       VARCHAR(20)    NOT NULL,
Apellidos    VARCHAR(30),
Edad         NUMBER(2,0),
Cargo        VARCHAR(20) DEFAULT 'Administrativo'
);
```

Esto quiere decir que en todas las filas que insertemos se nos propondrá de forma automática el cargo de administrativo.

```
CREATE TABLE Empleados(
DNI          VARCHAR(10)    PRIMARY KEY,
Nombre       VARCHAR(20)    NOT NULL,
Apellidos    VARCHAR(30),
Edad         NUMBER(2,0),
FechaAlta    DATE DEFAULT SYSDATE
);
```

La función SYSDATE asigna la fecha actual del sistema operativo.



Ejemplo

```
CREATE TABLE EJEMPLO (
  DNI VARCHAR2(10) NOT NULL,
  NOMBRE VARCHAR2(30),
  FECHA DATE DEFAULT SYSDATE )
INSERT INTO EJEMPLO1 (DNI, NOMBRE) VALUES
('88888888G','ABDCABCD')
```

DNI	NOMBRE	FECHA
1 88888888G	ABDCABCD	01/10/08



La restricción CHECK

- Muchas columnas de tablas requieren valores limitados dentro de un rango o el cumplimiento de ciertas condiciones.
- La restricción CHECK actúa como una cláusula WHERE. Puede hacer referencia a una o más columnas, pero no a valores de otras filas. En una cláusula CHECK no cabe incluir subconsultas.





EJEMPLO

CREATE TABLE EJEMPLO2

(
DNI VARCHAR2(10) NOT NULL,
NOMBRE VARCHAR2(30) NOT NULL,
EDAD NUMBER(2),
CURSO NUMBER,
CONSTRAINT CLAVE_P PRIMARY KEY(DNI),
CONSTRAINT COMP_EDAD CHECK (EDAD BETWEEN 5 AND 20),
CONSTRAINT NOMBR_MAY (NOMBRE=UPPER(NOMBRE)),
CONSTRAINT COMP_EDAD CHECK (CURSO IN (1,2,3))
)



Bases de datos



EJEMPLO

CREATE TABLE EJEMPLO2

(
DNI VARCHAR2(10) PRIMARY KEY,
NOMBRE VARCHAR2(30) NOT NULL
CHECK((NOMBRE=UPPER(NOMBRE)),
EDAD NUMBER(2) CHECK (EDAD BETWEEN 5 AND 20),
CURSO NUMBER CHECK (CURSO IN (1,2,3))
)



Bases de datos



La restricción UNIQUE

- Evita valores repetidos en la misma columna.
- Puede contener una o varias columnas.
- Es similar a la restricción PRIMARY KEY, salvo que son posibles varias columnas UNIQUE definidas en una misma tabla.
- Admite valores NULL
- Al definir una restricción UNIQUE se crea un índice automáticamente



Bases de datos

VALORES ÚNICOS

Si queremos que un campo que no sea clave primaria, no acepte 2 filas con el mismo valor, pero que sí pueden quedar en blanco, se aplicaría la restricción UNIQUE, puede aplicarse a nivel de tabla, o a nivel de campo

La restricción puede aplicarse a un conjunto de campos como la clave primaria.

```
CREATE TABLE Empleados(
DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
Nombre       VARCHAR(20)    NOT NULL,
Apellidos    VARCHAR(30)    UNIQUE,
Edad         NUMBER(2,0),
);
```

```
CREATE TABLE Empleados(
DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
Nombre       VARCHAR(20)    NOT NULL,
Apellidos    VARCHAR(30),
Edad         NUMBER(2,0),
UNIQUE (Apellidos)
);
```

```
CREATE TABLE Empleados(
DNI          VARCHAR(10)    CONSTRAINT ClaveEmpl PRIMARY KEY,
Nombre       VARCHAR(20)    NOT NULL,
Apellidos    VARCHAR(30),
Edad         NUMBER(2,0),
UNIQUE (Nombre,Apellidos)
);
```



EJEMPLO

```
CREATE TABLE EJEMPLO4
(
  COD_ESTUDIANTE CHAR(3) PRIMARY KEY,
  DNI CHAR (9) UNIQUE,
  EDAD NUMBER(2)
)
```



Bases de datos



Vistas del diccionario de datos para las restricciones

- USER_CONSTRAINTS: Definiciones de restricciones de tablas propiedad del usuario
- Los campos de esta vista, entre otros son:

```
DESC USER_CONSTRAINTS
```

Name	Null	Type
OWNER	NOT NULL	VARCHAR2(30)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
CONSTRAINT_TYPE		VARCHAR2(1)
TABLE_NAME	NOT NULL	VARCHAR2(30)



Bases de datos



Ejemplo

```
SELECT CONSTRAINT_NAME, TABLE_NAME, CONSTRAINT_TYPE FROM
  USER_CONSTRAINTS WHERE TABLE_NAME='EMPLEADO2'
```

CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE
CP_ARTIC1	ARTICULOS2	P
CA_ARTIC1	ARTICULOS2	R

EL TIPO DE RESTRICCIÓN PUEDE SER C (CHECK),
P (PRIMARY KEY), U (UNIQUE), R (FOREIGN KEY)



Bases de datos



Otras vistas del diccionario

- **ALL_CONSTRAINTS:** definiciones de restricciones sobre tablas a las que puede acceder el usuario
- **DBA_CONSTRAINTS:** Todas las definiciones de restricciones sobre todas las tablas (hay que ser dba para poder consultarla)

Para información sobre columnas:

- **USER_CONS_COLUMNS:** información sobre las restricciones de columnas en tablas del usuario
- **ALL_CONS_COLUMNS:** información sobre las restricciones de columnas en tablas a las que puede acceder el usuario
- **DBA_CONS_COLUMNS:** información sobre todas las restricciones de columnas (hay que ser dba para consultarla)



Bases de datos



USER_CONS_COLUMNS

```
DESC USER_CONS_COLUMNS
```

Name	Null	Type
OWNER	NOT NULL	VARCHAR2(30)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
TABLE_NAME	NOT NULL	VARCHAR2(30)
COLUMN_NAME		VARCHAR2(4000)
POSITION		NUMBER

5 rows selected



Bases de datos



Ejemplo

```
SELECT * FROM USER_CONS_COLUMNS
WHERE TABLE_NAME='ARTICULOS2'
```

OWNER	CONSTRAINT_NAME	TABLE_NAME	COLUMN_NAME
CASE02	CA_ARTIC1	ARTICULOS2	COD_FABRICANTE
CASE02	CP_ARTIC1	ARTICULOS2	COD_FABRICANTE
CASE02	CP_ARTIC1	ARTICULOS2	PESO
CASE02	CP_ARTIC1	ARTICULOS2	CATEGORIA
CASE02	CP_ARTIC1	ARTICULOS2	ARTICULO



Bases de datos



Vistas del usuario

- User_tables: información acerca de las tablas del usuario
- User_objects: objetos que son propiedad del usuario
- User_catalog: tablas, vistas, sinónimos y secuencias del usuario



Bases de datos



CREACIÓN DE TABLA CON SELECT

Podemos crear una tabla a partir de otra existente en la BD, en este caso no es necesario especificar nombres de columna ni tipos. En cuanto a las restricciones solo crean las de tipo NOT NULL



Bases de datos



SINTAXIS

```
CREATE TABLE Nombretabla  
(  
  Columna1 [, columna2 [...]]  
)  
[TABLESPACE espacio_tabla]  
AS consulta
```



Bases de datos



Ejemplo

```
CREATE TABLE EMPLEADOS AS  
SELECT * FROM EMPLOYEES  
WHERE JOB_ID IN ('SA_MAN',  
  'SA_REP', 'INIT_PROG')
```

Crea la tabla empleados con los mismos campos que employees y los datos correspondientes a los empleados cuyo oficio es Jefe de Ventas, Representante de Ventas o Programador



Bases de datos



EJEMPLO

```
CREATE TABLE EMPLEADO
( NOMBRE VARCHAR2(25),
  EDAD NUMBER NOT NULL,
  COD_PROVINCIA NUMBER(2) CONSTRAINT NO_NULO NOT NULL,
  CONSTRAINT PK_EMPLEADO PRIMARY KEY (NOMBRE),
  CHECK (EDAD BETWEEN 18 AND 35) )
```

CREATE TABLE EMPLE2 AS SELECT * FROM EMPLEADO

Si CONSULTAMOS LAS RESTRICCIONES COMPROBAREMOS
QUE SOLO SE HAN MANTENIDO LAS RESTRICCIONES NOT
NULL

Bases de datos



EJEMPLO

```
CREATE TABLE EMPLEADO
( NOMBRE VARCHAR2(25),
  EDAD NUMBER NOT NULL,
  COD_PROVINCIA NUMBER(2) CONSTRAINT NO_NULO NOT NULL,
  CONSTRAINT PK_EMPLEADO PRIMARY KEY (NOMBRE),
  CHECK (EDAD BETWEEN 18 AND 35) )
```

**CREATE TABLE EMPLE2(NOM, EDAD2) AS SELECT
NOMBRE, EDAD FROM EMPLEADO**



Bases de datos



3. SUPRESIÓN DE TABLAS

- La orden de SQL, DROP TABLE suprime una tabla de la base de datos.
- Cada usuario puede borrar sus propias tablas
- Sólo el administrador de la base de datos o algún usuario con el privilegio DROP ANY TABLE puede borrar las tablas de otro usuario
- Al suprimir una tabla también se suprimen los índices y los privilegios asociados a ella.
- Las vistas y los sinónimos creados a partir de esta tabla dejan de funcionar, pero siguen existiendo en la base de datos por los que habría que eliminarlos



Bases de datos



FORMATO

DROP TABLE *nombretabla* [CASCADE CONSTRAINTS];



Bases de datos



Ejemplo

■ DROP TABLE PROVINCIAS

ORA-02449: claves únicas/primarias en la tabla referidas por claves ajenas

En este ejemplo no permite borrar la tabla provincias porque en la tabla empleados se referencia a la tabla provincias



Bases de datos



Ejemplo

DROP TABLE PROVINCIAS CASCADE CONSTRAINTS

Borra la tabla provincias con las restricciones de integridad referencial que remitan a la clave primaria de provincias



Bases de datos



Orden Truncate

- Permite suprimir todas las filas de una tabla y liberar el espacio ocupado sin que desaparezca la definición de la tabla.
- NO se puede anular con ROLLBACK.



Bases de datos



FORMATO

- **TRUNCATE TABLE** *nombretabla*
[{ DROP | REUSE } STORAGE];
- Truncate permite liberar el espacio utilizado por las filas suprimidas. Con la opción DROP STORAGE se libera todo el espacio.
- Con la opción REUSE STORAGE, mantendrá reservado el espacio para nuevas filas de la tabla.



Bases de datos



Ejemplo

- No se puede truncar una tabla cuya clave primaria sea referenciada por la clave ajena de otra tabla.

1 truncate table provincias

2

Results Script Output Explain Autotrace DBMS Output OWA Output

Error starting at line 1 in command:
truncate table provincias
Error report:
SQL Error: ORA-02266: claves únicas/primarias en la tabla referidas por claves ajenas

INFORMÁTICA

Bases de datos



4. MODIFICACIÓN DE UNA TABLA

La modificación de tablas permitirá:
añadir, modificar o eliminar columnas
de una tabla existente, así como añadir
o eliminar restricciones y activarlas o
desactivarlas.

ALTER TABLE {
ADD
MODIFY
DROP



Bases de datos



FORMATO

ALTER TABLE *nombretabla*

[**ADD** (columna [,columna] ...)]

[**MODIFY** (columna [, columna] ...)]

[**DROP COLUMN**(columna [, columna] ...)]

[**ADD CONSTRAINT** restricción]

[**DROP CONSTRAINT** restricción]);

[**ENABLE CONSTRAINT** restricción]

[**DISABLE CONSTRAINT** restricción]);



Bases de datos



4.1 Añadir columnas

- **ADD** se utiliza para añadir columnas a una tabla. Para ello hay que tener en cuenta que:
 - Si la columna no está definida como NOT NULL, se puede añadir en cualquier momento.
 - Si la nueva columna está definida como NOT NULL, y la tabla NO está vacía, tendremos que dar un valor por defecto previamente. Si la tabla está vacía podremos añadir la columna NOT NULL sin necesidad de dar un valor por defecto.



Bases de datos



Ejemplo

Table	Column	Tipo De Dato	Longitud	Precisión	Escala	Clave Primaria	Nulo	Valor Por Defecto
EMPLEADOS	<u>COD_EMPLE</u>	Number	-	2	0	1	-	-
	NOMBRE	Varchar2	25	-	-	-	-	-
	APELLIDO	Varchar2	25	-	-	-	✓	-
	SALARIO	Number	-	5	0	-	✓	-
	<u>COD_PROVI</u>	Number	-	7	2	-	✓	-
	<u>COD_EMPRE</u>	Number	-	2	0	-	✓	-

**ALTER TABLE EMPLEADOS ADD (SEXO CHAR(1),
EDAD NUMBER(2))**



Bases de datos



Tras modificar la tabla, su estructura es:

Table	Column	Tipo De Dato	Longitud	Precisión	Escala	Clave Primaria	Nulo	Valor Por Defecto	Coment
EMPLEADOS	<u>COD_EMPLE</u>	Number	-	2	0	1	-	-	-
	NOMBRE	Varchar2	25	-	-	-	-	-	-
	APELLIDO	Varchar2	25	-	-	-	✓	-	-
	SALARIO	Number	-	5	0	-	✓	-	-
	<u>COD_PROVI</u>	Number	-	7	2	-	✓	-	-
	<u>COD_EMPRE</u>	Number	-	2	0	-	✓	-	-
	<u>SEXO</u>	Char	1	-	-	-	✓	-	-
	<u>EDAD</u>	Number	-	2	0	-	✓	-	-



Bases de datos



Ejemplo

Sobre la tabla **jobs2** que contiene los mismos datos y campos que **jobs**, añadimos una nueva columna, grupo, que es obligatoria

```
ALTER TABLE JOBS2 ADD GRUPO CHAR(1) NOT NULL
```

ORA-01758: la tabla debe estar vacía para agregar la columna (NOT NULL) obligatoria



Bases de datos



Ejemplo

- En este caso no deja añadir una columna NOT NULL ya que la tabla jobs2 tiene datos y al agregar la columna **grupo**, al no haberle dado ningún valor, añadiría NULL en este campo para cada fila, por lo que contradice que deba ser NOT NULL. La solución es la siguiente:



Bases de datos



Solución

- ALTER TABLE JOBS2 ADD GRUPO CHAR(1) DEFAULT 'C' NOT NULL



Bases de datos



4.2 Modificar Columnas

MODIFY: podemos modificar columnas de una tabla, bien porque modifiquemos la longitud del campo, bien porque modifiquemos el tipo de dato, teniendo en cuenta que:

A. Aumentar/disminuir la longitud del campo (tipos char, varchar2,...).

- * Podremos aumentar la longitud de un campo en cualquier momento.
- * Si disminuimos, no podemos dar una longitud menor que el tamaño del máximo valor almacenado.



Bases de datos



4.2 Modificar Columnas

B. Modificar el tipo de dato

- Si la **columna es NULL** en todas las filas, se puede modificar el tipo de dato.
- Es posible aumentar o disminuir el nº de posiciones decimales de una columna de tipo NUMBER
- La opción MODIFY... NOT NULL solo será posible cuando la tabla no contenga ninguna fila con valor NULL en la columna que se modifica.



Bases de datos



Ejemplo

Tipo de Objeto **TABLE** Objeto **ALUMNOS**

Table	Column	Tipo De Dato	Longitud	Precisión	Escala	Clave Primaria
ALUMNOS	CODIGO	Number	-	3	0	1
	NOMBRE	Varchar2	21	-	-	-
	APELLIDO	Varchar2	30	-	-	-
	CURSO	Number	-	2	0	-
	FECHA_MATRI	Date	7	-	-	-

En la tabla alumnos, modificamos el campo apellido a una longitud máxima de 25

**ALTER TABLE ALUMNOS MODIFY APELLIDO
VARCHAR2(25)**



Bases de datos



Ejemplo

Si decidimos modificar el campo apellido con una longitud máxima de 1:

```
ALTER TABLE ALUMNOS MODIFY  
APELLIDO VARCHAR2(1)
```

ORA-01441: no se puede reducir la longitud de la columna porque algún valor es demasiado grande



Bases de datos



3.3 Borrar Columnas

DROP COLUMN: podemos borrar columnas de una tabla, teniendo en cuenta que:

- No se pueden borrar todas las columnas de una tabla.
- No se pueden eliminar las claves primarias referenciadas por claves ajenas.



Bases de datos



Ejemplo

ALTER TABLE ALUMNOS DROP COLUMN NOMBRE

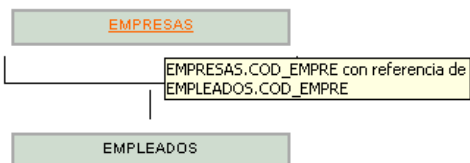
Borra la columna nombre. También podríamos borrar la columna código que es clave primaria ya que no está referenciada desde ninguna otra tabla.



Bases de datos



Ejemplo



ALTER TABLE EMPRESAS DROP COLUMN COD_EMPRE

Aparece el siguiente error, porque la tabla empresas está referenciada por la tabla empleados

ORA-12992: no se puede borrar la columna de claves principales



Bases de datos



3.4 Añadir y borrar restricciones

- Podemos añadir y borrar las siguientes restricciones de una tabla:

CHECK, PRIMARY KEY, NOT NULL Y FOREIGN KEY, UNIQUE

Se pueden borrar tanto las restricciones con nombre como las asignadas por el sistema SYS_C00n



Bases de datos



FORMATO

ALTERTABLE nombretabla [ADD CONSTRAINT nombrerestricción restricción]

ALTERTABLE nombretabla DROP CONSTRAINT nombrerestricción



Bases de datos



Ejemplo

```
ALTER TABLE ALUMNOS  
ADD CONSTRAINT CLAVEPRIM  
PRIMARY KEY (CODIGO)
```

La columna código pasa a ser clave primaria de ALUMNOS esta restricción se llamará CLAVEPRIM y cuando Oracle haga referencia a ella utilizará este nombre.



Bases de datos



Ejemplo

- Si no deseamos darle nombre a la restricción, el ejemplo anterior quedaría como sigue

```
ALTER TABLE ALUMNOS  
ADD PRIMARY KEY (CODIGO)
```



Bases de datos



Ejemplos

```
ALTER TABLE EMPLE ADD CONSTRAINT  
SALMAYOR CHECK (SALARIO>0)
```

```
ALTER TABLE EMPLE ADD CONSTRAINT  
CA_EMPLE FOREIGN KEY(DEPARTMENT_ID)  
REFERENCES DEPARTMENTS ON DELETE  
CASCADE
```



Bases de datos



Borrar restricciones

```
ALTER TABLE EMPLE DROP CONSTRAINT  
SALMAYOR
```



Bases de datos



3.5 Activar y desactivar restricciones

Por defecto las restricciones se activan al crearlas, se pueden desactivar de la siguiente forma:

- Para desactivar una restricción sin borrarla se usa DISABLE

```
ALTER TABLE table
```

```
DISABLE CONSTRAINT nombre_constraint
```

- Para activar una restricción de integridad actualmente desactivada se usa ENABLE

```
ALTER TABLE table
```

```
ENABLE CONSTRAINT nombre_constraint;
```



Bases de datos



Ejemplo

```
ALTER TABLE EMPLE DISABLE CONSTRAINT PK_EMPLE
```



Bases de datos



Ejemplo

- Para visualizar las constraints con su condición:

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, SEARCH_CONDITION FROM
  USER_CONSTRAINTS
WHERE TABLE_NAME = 'EMPLE';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION	
SYS_C001069	C	"EMP_NO"	IS NOT NULL
SYS_C001070	C	"DEPT_NO"	IS NOT NULL



Bases de datos



CAMBIO DE NOMBRE

- RENAME nombrea anterior TO nombrenuevo
- Las restricciones, índices, permisos, triggers se transfieren automáticamente al nuevo objeto. Oracle invalida todos los objetos que dependen del objeto renombrado como vistas, sinónimos, procedimientos,...



Bases de datos