

# **Lenguaje de Definición de Datos**

# Introducción

- Usaremos el lenguaje de definición de datos, también llamado DDL.
- Manejaremos ordenes, CREATE, ALTER Y DROP, tablas.
- Manejaremos las restricciones, algo muy importante y útil

# Base de datos

- En Oracle no existe el concepto de Base de datos, las tablas son del usuario que las crea.
- Podemos tener un usuario para cada base de datos, y para nombrar a una tabla necesitaremos el nombre del usuario, un punto y el nombre de la tabla.

# Creación básica de tablas

Tendremos en cuenta los siguientes aspectos:

Elección del **NOMBRE DE TABLA**, debe identificar su contenido, máximo 30 caracteres y único. No será palabra reservada Oracle. Comenzará por letra. Ej: *ALUMNOS*

Elección de los **NOMBRES DE COLUMNA**, han de ser autodescriptivos. Ejs: *DNI, NOMBRE*.

Elección de los **TIPOS DE DATOS** y **TAMAÑOS** de las columnas. Solo se usan tipos numéricos si van a realizarse operaciones aritméticas con la columna.

Definición de las **RESTRICCIONES** necesarias: claves primarias y ajenas, columnas obligatorias, valores por defecto, rangos de valores, etc.

# Tipos de datos de ORACLE

Tipo de datos	Descripción
<b>VARCHAR2(size)</b>	Cadena de caracteres de longitud variable. Máximo 2000 caracteres
<b>CHAR(size)</b>	Cadena de caracteres de longitud fija. Tamaño máximo: 255 caracteres.
<b>NUMBER(p,s)</b>	Número entero y real hasta de 38 dígitos.
<b>DATE</b>	valores de fecha y hora.
<b>LONG</b>	Cadena de caracteres de longitud variable. Máximo 2 GB
<b>CLOB</b>	Cadena de caracteres de longitud variable. Máximo 4 GB
<b>RAW and LONG RAW</b>	Igual que char y long pero en binario
<b>BLOB</b>	Objeto binario de hasta 4 gigabytes
<b>BFILE</b>	Datos almacenados en un fichero externo; hasta 4 gigabytes

# Sintaxis de creación básica de tablas

```
CREATE TABLE nombre_tabla  
(column_1          tipo_dato( ),  
 column_2          tipo_dato( ),  
 .....  
) ;
```

Ejemplo:

```
CREATE TABLE alumnos_07  
(num_matri        NUMBER(6),  
 nombre            VARCHAR2(15),  
 fecha_nac         DATE,  
 direccion         VARCHAR2(50)  
) ;
```

# Vistas del diccionario de datos asociadas a las tablas

**USER\_TABLES**, información sobre tablas propiedad del usuario  
CAT es un sinónimo de USER\_TABLES.

**ALL\_TABLES**, información sobre las tablas que son propiedad del usuario o el usuario tiene permisos sobre ellas.

**DBA\_TABLES**, información sobre todas las tablas existentes. Solo se puede consultar con privilegios de administración.

La forma de consultar el diccionario de datos incluye con frecuencia el uso de filtros en el WHERE para mostrar solo la información de interés. Es importante entender el significado de las columnas de las vistas del diccionario de datos.

# Restricciones. Integridad de Datos.

- **INTEGRIDAD**, los datos antes de almacenarse en BD han de ajustarse a ciertas restricciones.
- **Las restricciones** hacen que el usuario o las aplicaciones tengan menos trabajo a la hora de manipular los datos y sea Oracle el encargado de las tareas de mantenimiento de la integridad de la BD.



# Restricción de clave primaria

- Columna o conjunto de columnas que identifican unívocamente cada fila de una tabla.
- Única, no nula y obligatoria.
- Solo se define una por tabla.
- Automáticamente se crea un índice que facilita el acceso a la tabla.

# Restricción de clave primaria. Formatos.

A nivel de columna:

```
CREATE TABLE nombre_tabla
```

```
(  
    nombrecolum_1 TIPO_DATO() [CONSTRAINT nombre_constraint] PRIMARY  
    KEY,  
    .....  
);
```

o bien a nivel de tabla:

```
CREATE TABLE nombre_tabla
```

```
(  
    colum_1          TIPO_DATO(),  
    .....  
    [CONSTRAINT nombre_constraint] PRIMARY KEY(colum1[,...]),  
    .....  
);
```

# Restricción de clave ajena

Formada por:

Una o varias columnas asociadas a una clave primaria de otra o de la misma tabla.

Se pueden definir tantas claves ajenas como sean necesarias.

El valor debe ser igual a un valor de la clave referenciada: regla de integridad referencial.

# Restricción de clave ajena. Formatos.

A nivel de columna:

La clave ajena se define en la descripción de la columna usando la palabra clave REFERENCES

```
CREATE TABLE nombre_tabla
(
  • column1 tipo_dato() [CONSTRAINT nombre_restriccion] REFERENCES
    nombre_tabla [(columna)] [ON DELETE opcion_borrado]
  ....
);
```

# Restricción de clave ajena. Formatos.

A nivel de tabla:

La clave ajena se define **al final** de todas las columnas usando las cláusulas **FOREIGN KEY** y **REFERENCES**.

```
CREATE TABLE nombre_tabla
(
    column1      tipo_dato(),
    .....
    [CONSTRAINT nombre_restriccion]
    FOREIGN KEY(columna [, columna..])
    REFERENCES nombre_tabla [(columna [, columna..])]
    [ON DELETE opcion_borrado]
    ....
);
```

# Restricción de clave ajena. Formatos.

Las posibles opciones del ON DELETE:

- ON DELETE CASCADE: al borrar el padre (primary key) se borra el hijo (foreign key)
- ON DELETE SET NULL al borrar padre (primary key) se pone a nulo el hijo (foreign key)
- No permite eliminar o modificar el padre (primary key) si existe un hijo. Es la opción por defecto, por lo que no habría que poner nada

# Restricción de clave ajena. Formatos.

A nivel de tabla:

La clave ajena se define **al final** de todas las columnas usando las cláusulas **FOREIGN KEY** y **REFERENCES**.

```
CREATE TABLE nombre_tabla
(
    column1      tipo_dato(),
    .....
    [CONSTRAINT nombre_restriccion]
    FOREIGN KEY(columna [, columna..])
    REFERENCES nombre_tabla [(columna [, columna..])]
    [ON DELETE opcion_borrado]
    ....
);
```

# Restricción de obligatoriedad

Asociada a una columna, significa que no puede tener valores nulos.

```
CREATE TABLE nombre_tabla  
(  
    column1 tipo_dato() [CONSTRAINT nombre_restriccion] NOT NULL,  
    ....  
);
```



# Restricción de valor por defecto

Sirve para asignar valores por defecto a una columna, cuando no se especifica valor en la cláusula INSERT.

Se pueden incluir constantes, funciones SQL y las variables UID y SYSDATE.

Es el único tipo de restricción que no lleva nombre puesto que no puede ser violada.

Formato:

```
CREATE TABLE nombre_tabla
(
    column1      tipo_dato() DEFAULT valor_por defecto,
    ....
);
```

# Restricción de verificación de condiciones

Sirve para comprobar el cumplimiento de ciertas condiciones en los valores que puede tomar una columna.

No se pueden incluir ni UID, SYSDATE y USER.

Pero si es pueden incluir funciones como UPPER() y LOWER(), LTRIM(), RTRIM(), TRIM()

Admite operadores relacionales (<, >=, <, <=, =, !=) y operadores lógicos (AND, OR)

*Formato a nivel de columna:*

```
CREATE TABLE nombre_tabla
(
  column1 tipo_dato() [CONSTRAINT nombre_restriccion] CHECK (condicion),
  .....
);
```

# Restricción de verificación de condiciones

En ORACLE los valores numéricos se escriben tal cual, pero los valores de tipo cadena van encerrados entre comilla simples.

Check permite comparar cadenas con el operador like, en el que % representa a cualquier cosa, y \_ representa un carácter.

# Restricción de verificación de condiciones

*Formato a nivel de tabla:*

```
CREATE TABLE nombre_tabla
(
    column1          tipo_dato(),
    .....
    [CONSTRAINT nombre_restriccion] CHECK (condicion),
    .....
);
```

# Restricción de unicidad

Evita valores repetidos en una columna. Puede contener una o varias columnas, se crea automáticamente un índice por esa columna, admite valores NULL si no se especifica lo contrario.

Formato a nivel de columna:

```
CREATE TABLE nombre_tabla  
(  
    column1 tipo_dato() [CONSTRAINT nombre_restriccion] UNIQUE,  
    .....  
);
```

# Restricción de unicidad

*Formato a nivel de tabla:*

```
CREATE TABLE nombre_tabla
(
    column1      tipo_dato(),
    .....
    [CONSTRAINT nombre_restriccion] UNIQUE (columna [,columna...]),
    .....
);
```

# Create Table. Restricciones a nivel de columna.

```
CREATE TABLE nombre_tabla  
( colum_1          TIPO_DATO( )  
[CONSTRAINT nombre_constraint]  
[NOT NULL] [UNIQUE] [PRIMARY KEY] [DEFAULT valor]  
[REFERENCES nombre_tabla [(column_1 [,column_n])]  
[ON DELETE CASCADE]]  
[CHECK condicion]),  
.....  
);
```

```
CREATE TABLE emple_7  
(nombre          VARCHAR2(25)    PRIMARY KEY,  
edad            NUMBER          CHECK (edad BETWEEN 18 AND 35),  
cod_provincia   NUMBER(2)       REFERENCES PROVINCIAS  
ON DELETE CASCADE  
);
```

## Create Table. Restricciones a nivel de tabla.

```
CREATE TABLE nombre_tabla
( colum_1          TIPO_DATO( ),
.....,
  colum_n          TIPO_DATO( ),
[CONSTRAINT nombre_constraint]
{[UNIQUE] | [PRIMARY KEY] (COLUMN1, )},
[CONSTRAINT nombre_constraint]
[FOREIGN KEY (COLUMN,...)]
REFERENCES nombre_tabla [(colum...)]
[ON DELETE CASCADE]],
[CONSTRAINT nombre_constraint]
[CHECK (condicion)],
.....
);
```



# Las restricciones en el Diccionario de Datos

Las vistas del diccionario de datos de Oracle con información sobre restricciones definidas en tablas son las siguientes:

**USER\_CONSTRAINTS** (tablas de usuario)

**ALL\_CONSTRAINTS** (tablas a las que puede acceder el usuario)

**DBA\_CONSTRAINTS** (todas las restricciones sobre todas las tablas)

*Tipo de restricción:*

<b>C</b>	CHECK	<b>R</b>	FOREIGN KEY
<b>P</b>	PRIMARY KEY	<b>U</b>	UNIQUE

# Las restricciones en el Diccionario de Datos

Para obtener una información más detallada sobre restricciones en las columnas:

**USER\_CONS\_COLUMNS** (restricciones en columnas de tablas de usuario)

**ALL\_CONS\_COLUMNS** (restricciones en columnas de tablas a las que puede acceder el usuario)

**DBA\_CONS\_COLUMNS** ( todas las restricciones de las columnas)

# Supresión de tablas. DROP

## Formato:

**DROP TABLE** [usuario.]nombre\_tabla;

Cada usuario puede borrar sus propias tablas.

El DBA u otro usuario con privilegios de DROP ANY TABLE, podrán borrar cualquier tabla.

Se borran también los índices y privilegios asociados a ellas.

Las vistas y sinónimos dejan de funcionar, con lo que habría que borrarlos.

# Supresión de tablas. TRUNCATE

Elimina filas de una tabla, liberando el espacio asociado.

No desaparece la definición de la tabla.

Orden DDL, no genera ROLLBACK.

No activa disparadores DELETE.

No se puede truncar una tabla cuya clave primaria sea referenciada por una ajena, antes tendremos que desactivar las restricciones

## **Formato:**

```
TRUNCATE TABLE [usuario.]nombre_tabla  
[{REUSE|DROP} STORAGE];
```

# Modificación de tablas. ALTER

**ALTER TABLE**, nos permitirá:

Añadir, modificar o eliminar columnas de una tabla.

Añadir o eliminar restricciones.

Activar o desactivar restricciones

# Modificación de tablas. ALTER

**Formato:**

```
ALTER TABLE nombre_tabla  
{ADD (columna)  
[MODIFY (column [...])]  
[DROP COLUMN (column.....)]  
[ADD CONSTRAINT restricción]  
[DROP CONSTRAINT restricción]  
[DISABLE CONSTRAINT restricción]  
[ENABLE CONSTRAINT restricción]  
};
```

# Añadir, modificar o eliminar columnas

**ADD**, añadir columnas a tabla, tendremos en cuenta:

Si la columna **no tiene restricción** NOT NULL, se puede añadir en cualquier momento.

**Si tiene restricción** NOT NULL:

- 1- Se añade la columna sin restricción.
- 2- Se da valor a la columna en todas las filas.
- 3- Se añade la restricción NOT NULL

# Añadir, modificar o eliminar columnas

**MODIFY**, modifica una o más columnas existentes en una tabla. Tendremos en cuenta:

- 1- Se puede aumentar longitud de columna en cualquier momento.
- 2- Si disminuimos la longitud esta no puede ser de menor tamaño que el valor máximo almacenado.
- 3- Se puede aumentar o disminuir el valor de los decimales en columna NUMBER.
- 4- Si columna es NULL en toda la tabla, se puede modificar la longitud y el tipo de dato.
- 5- **MODIFY... NOT NULL**, cuando no haya ningun valor NULO en la columna que se modifica.



## **Añadir, modificar o eliminar columnas**

**DROP COLUMN**, para borrar columna de tabla.

Tendremos en cuenta:

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

# Adición y Borrado de Restricciones

Se pueden añadir y borrar las restricciones:

PRIMARY KEY

NOT NULL

FOREIGN KEY

UNIQUE

CHECK

**ALTER TABLE nombre\_tabla**

**ADD CONSTRAINT nombre\_constraint;**

**ALTER TABLE nombre\_tabla**

**DROP CONSTRAINT nombre\_constraint;**

# Activar y Desactivar Restricciones

```
ALTER TABLE nombre_tabla  
DISABLE CONSTRAINT nombre_restriccion;
```

```
ALTER TABLE nombre_tabla  
ENABLE CONSTRAINT nombre_restriccion;
```

Las restricciones por defecto se activan al crearlas, aunque se pueden desactivar en el momento de crearlas.

```
ALTER TABLE emp  
ADD CONSTRAINT UNI_ename  
UNIQUE (ename) DISABLE;
```