

Creación
de tablas
- SQL -

Lenguaje de definición de datos DDL:

Se compone de varias sentencias:

- CREATE: crea objetos
- DROP: elimina objetos
- ALTER: modifica objetos
- GRANT: concede privilegios
- REVOKE: retira privilegios

* Creación de tablas:

Se utiliza CREATE TABLE para crear una nueva tabla en una BD.

Necesitaremos saber las columnas y los tipos de dato que vamos a crear.

Esquema:

```
CREATE TABLE nombre tabla (  
columna1 TIPO DE DATO (-)  
  
CONSTRAINT nombre restricción1 restricción de columna ,  
  
columna2 TIPO DE DATO (-)  
  
CONSTRAINT nombre restricción2 restricción de columna , ...  
  
CONSTRAINT nombre restricción3 restricción de tabla , ...  
  
);
```

Ejemplo:

```
CREATE TABLE persona (  
ID varchar2(10) PRIMARY KEY,  
Nombre varchar(15),  
Apellido varchar(15),  
edad number(3);
```

* Tipos de datos: hay muchos tipos de datos pero estos son los más usados.

- Varchar2: cadena de caracteres de longitud variable, entre paréntesis ponemos el nº total de caracteres.
- Char: cadena de caracteres de longitud fija, entre paréntesis se coloca el nº total de caracteres, si este no se completa se rellenará al tamaño indicado. Usado en id y cod.

· Number: datos numéricos. Los paréntes se ponen de la sig forma (n,m) donde n es el total de dígitos y m el total de decimales. Ej: number (4,2) → 13132
number (3,2) → 1.32
number (3) → 123

· DATE: almacena fecha y hora. Para cambiar el formato se usa:

ALTER SESSION SET NLS_DATE = 'DDMM/YYYY';

★ Restricciones:

Se utilizan para limitar el tipo de dato que puede recibir una columna de una tabla.

Las restricciones se pueden definir cuando se crea la tabla o posteriormente con la sentencia ALTER.

· Tipos de restricciones:

· PRIMARY KEY: solo puede haber 1 clave principal por tabla, aunque una clave principal pueda ser la unión de 2.
Es como unir NOT NULL y UNIQUE.

Ej:

nombrar
→
sin
CREATE TABLE persona (
ID number (5) PRIMARY KEY,
Nombre varchar2 (10),
Apellidos varchar2 (15));

nombrar
→
CREATE TABLE persona (
ID number (5) CONSTRAINT (pk_id) PRIMARY KEY,
Nombre varchar2 (10),
Apellidos varchar2 (15));

Hacer PK a dos
columnas
→
CREATE TABLE persona (
ID number (5),
Nombre varchar2 (10),
Apellidos varchar2 (15)
CONSTRAINT (pk_idnom) PRIMARY KEY (id, nombre));

• FOREIGN KEY: es la pk de otra tabla.

```
CREATE TABLE persona (  
  ID number(5) CONSTRAINT pk_id PRIMARY KEY,  
  Nombre varchar2(10),  
  Apellidos varchar2(15) );
```

Can nombre
→

```
CREATE TABLE domicilio (  
  cod_provincia number(2) PRIMARY KEY,  
  ID number(5),  
  capital varchar2(15),  
  CONSTRAINT fk_id FOREIGN KEY (ID) REFERENCES persona );
```

• NOT NULL: prohíbe usar datos nulos.

ES indiferente ponerle nombre.

```
CREATE TABLE persona (  
  ID number(5) PRIMARY KEY,  
  Nombre varchar2(10) CONSTRAINT nn_nom NOT NULL,  
  Apellidos varchar2(15) NOT NULL);
```

• NULL: acepta el uso de nulos.

```
CREATE TABLE persona (  
  ID varchar2(10) PRIMARY KEY,  
  Nombre varchar(15),  
  Apellido varchar(15) NULL,  
  edad number(3) );
```

- **DEFAULT**: se utiliza para establecer un valor predeterminado para una columna. No se puede hacer referencias a otras columnas o funciones PL/SQL.

```
CREATE TABLE persona (  
  ID varchar2(10) PRIMARY KEY,  
  Nombre varchar(15),  
  Apellido varchar(15) DEFAULT 'Antonio Machado',  
  Fecha DATE DEFAULT SYSDATE);
```

- **CHECK**: expresa una condición que ha de cumplirse para todas y cada una de las filas de la tabla, puede hacer referencia a una o varias columnas, pero no a valores de otras filas. No puede incluir subconsultas ni SYSDATE, UID...

```
CREATE TABLE persona (  
  ID varchar2(10) PRIMARY KEY,  
  Nombre varchar(15),  
  Apellido varchar(15),  
  edad number(3) CHECK (edad BETWEEN 5 AND 20),  
  curso number(1),  
  CONSTRAINT nom_mayus CHECK (nombre = UPPER(nombre)),  
  CONSTRAINT curso CHECK (curso in (1,2,3));
```

- **UNIQUE**: evita valores repetidos en la misma columna. Puede afectar a una o varias columnas. Admite valores NULL. Se crea un índice automáticamente.

```
1- CREATE TABLE persona (  
  ID varchar2(10) PRIMARY KEY,  
  Nombre varchar(15) UNIQUE,  
  edad number(3);
```

```
2- CREATE TABLE persona (  
  ID varchar2(10) PRIMARY KEY,  
  Nombre varchar(15),  
  edad number(3),  
  CONSTRAINT unica UNIQUE(nombre, edad));
```

Lenguaje comando:

% → LIKE 'A%' → "empieza por A"

Reemplaza cadena de caracteres

- → LIKE 'U%': → "2ª letra es la U"

Reemplaza 1 solo caracter

- Borrado de tablas: cuando se borra la tabla se suprimen también los índices y privilegios asociados a ella, pero siguen existiendo las vistas y los sinónimos de la tabla aunque dejan de funcionar.

DROP TABLE persona;

DROP TABLE persona CASCADE CONSTRAINTS;

- Modificación de estructura de tabla: sirve para cambiar (modificar, borrar, añadir) la estructura de la tabla, incluso las restricciones.

ALTER TABLE nombre_tabla

[ADD (columna [etc])] → Añadir columna

[MODIFY (columna [etc])] → Modificar columna

[DROP (columna [etc])] → Borrar columna

[ADD CONSTRAINT restricción] → Añadir restricción

[DROP CONSTRAINT restricción]; → Borrar restricción

Ej:

ALTER TABLE empleado ADD (
sexo char(1),
cod number(4));

ALTER TABLE empleado MODIFY (
sexo char(1) NOT NULL);

ALTER TABLE empleado ADD CONSTRAINT codigo-va UNIQUE (cod);
nom restricción

• Renombrar tabla: permite cambiar de nombre la tabla.

Si hay clave ajena o vista que hace referencia a la tabla el cambio de nombre generará un error.

RENAME nombre anterior tabla TO nombre nuevo tabla;

Las restricciones de integridad, los índices y permisos dados al objeto anterior se transfieren automáticamente al nuevo nombre.

Se invalidan los objetos que dependen del objeto renombrado.

• Inserción de datos:

INSERT INTO tabla [(columna 1)

VALUES (valor 1, valor 2 ...)

En columna 1... se especifican las columnas donde se van a introducir datos, si no se especifica se considerará por defecto todas las columnas.

Se representa en posicional, es decir el orden en el que están las columnas será el orden de inserción.

Los valores de carácter y fecha deben ir entre comillas simples.

INSERT INTO persona

VALUES (10 , 'Ana' , 'Oroza' , 24) ;

• Eliminación de filas de tablas: elimina todos las filas de la tabla. No admite ROLLBACK.

TRUNCATE [usuario].nombre tabla [DROP | REUSE] STORAGE ;

REUSE STORAGE mantiene reservado el espacio de esta fila.

DROP STORAGE el almacenamiento destinado a esta fila queda libre y disponible.

En caso de no especificar nada esta será la opción por defecto

TRUNCATE TABLE empleado;