

# Tipos de Sentencias SQL

## LDD-LMD-LCD

1

- Lenguaje de Definición de Datos.

2

- Lenguaje de Manipulación de Datos.

3

- Lenguaje de Control de Datos.

# Lenguaje de Definición de Datos.

## DDL

SENTENCIA	DESCRIPCIÓN
CREATE TABLE	Añade una nueva tabla a la base de datos.
DROP TABLE	Suprime una tabla de la base de datos.
ALTER TABLE	Modifica la estructura de una tabla existente.
CREATE VIEW	Añade una nueva vista a la base de datos.
DROP VIEW	Suprime una vista de la base de datos.
CREATE INDEX	Construye un índice para una columna.
CREATE SYNOYM	Define un alias para un nombre de tabla.
DROP SYNONYM	Suprime un alias para un nombre de tabla.

# Lenguaje de Definición de Datos.

## DDL

### CREATE

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas, vistas y procedimientos almacenados.

Ejemplo (crear una tabla)

```
CREATE TABLE 'CUSTOMERS';
```

# Lenguaje de Definición de Datos.

## DDL

### ALTER

Este comando permite modificar la estructura de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

Ejemplo (agregar columna a una tabla)

```
ALTER TABLE 'ALUMNOS' ADD EDAD INT UNSIGNED;
```

# Lenguaje de Definición de Datos.

## DDL

### DROP

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

### Ejemplo

```
DROP TABLE 'ALUMNOS';
```

# Lenguaje de Definición de Datos.

## DDL

### TRUNCATE

Este comando solo aplica a tablas y su función es borrar el contenido completo de la tabla especificada. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

### Ejemplo

```
TRUNCATE TABLE 'NOMBRE_TABLA';
```

# Lenguaje de Manipulación de Datos.

## DML

SENTENCIA	DESCRIPCIÓN
SELECT	Recupera datos de la base de datos.
INSERT	Añade nuevas filas de datos a la base de datos.
DELETE	Suprime filas de datos de la base de datos.
UPDATE	Modifica datos existentes en la base de datos.

# Lenguaje de Manipulación de Datos.

## DML

### SELECT

La sentencia SELECT nos permite consultar los datos almacenados en una tabla de la base de datos.

Forma básica

**SELECT [ALL | DISTINCT ]**

**<nombre\_campo> [{,<nombre\_campo>}]**

**FROM <nombre\_tabla>|<nombre\_vista>**

**[{,<nombre\_tabla>|<nombre\_vista>}]**

**[WHERE <condición> [{ AND|OR <condición>}]]**

**[GROUP BY <nombre\_campo> [{,<nombre\_campo>}]]**

**[HAVING <condición>[{ AND|OR <condición>}]]**

**[ORDER BY <nombre\_campo>|<indice\_campo> [ASC | DESC]**

**[{,<nombre\_campo>|<indice\_campo> [ASC | DESC ]}]**



# Lenguaje de Manipulación de Datos.

## DML

**SELECT:** Palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.

**ALL:** Indica que queremos seleccionar todos los valores. Es el valor por defecto y no suele especificarse casi nunca.

**DISTINCT:** Indica que queremos seleccionar sólo los valores distintos.

**FROM:** Indica la tabla (o tablas) desde la que queremos recuperar los datos. En el caso de que exista más de una tabla se denomina a la consulta "consulta combinada" o "join". En las consultas combinadas es necesario aplicar una condición de combinación a través de una cláusula WHERE.

**WHERE:** Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Admite los operadores lógicos AND y OR.

**GROUP BY:** Especifica la agrupación que se da a los datos. Se usa siempre en combinación con funciones agregadas.

**HAVING:** Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Su funcionamiento es similar al de WHERE pero aplicado al conjunto de resultados devueltos por la consulta. Debe aplicarse siempre junto a GROUP BY y la condición debe estar referida a los campos contenidos en ella.

**ORDER BY:** Presenta el resultado ordenado por las columnas indicadas. El orden puede expresarse con ASC (orden ascendente) y DESC (orden descendente). El valor predeterminado es ASC.

# Lenguaje de Manipulación de Datos.

## DML

### INSERT

Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Forma básica

```
INSERT INTO 'tablatura' ('columnaA',['columnaB,... '])  
VALUES ('valor1', ['valor2,...'])
```

O también se puede utilizar como:

```
INSERT INTO tablatura VALUES ('valor1','valor2')
```

# Lenguaje de Manipulación de Datos.

## DML

### UPDATE

Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo

```
UPDATE My_table SET field1 = 'updated value asd' WHERE  
field2 = 'N';
```

# Lenguaje de Manipulación de Datos.

## DML

### DELETE

Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla.

Forma básica

**DELETE FROM tabla WHERE columna1 = 'valor1';**

# Lenguaje de Control de Datos.

## DCL

SENTENCIA	DESCRIPCIÓN
<b>Control de acceso:</b>	
GRANT	Concede privilegios de acceso a usuarios.
REVOKE	Suprime privilegios de acceso a usuarios
<b>Control de transacciones:</b>	
COMMIT	Finaliza la transacción actual.
ROLLBACK	Deshace la transacción actual.