

GESTIÓN DE BASE DE DATOS

UNIDAD 6: SQL (DDL Y DML)



NOELIA HUGUET CHACÓN



DDL (Data Definition Language)

- **CREATE**: Para crear nuevas bases de datos o tablas.
 - **CREATE DATABASE EjemploBD**
 - **CREATE TABLE Tabla1**
- **ALTER**: Para modificar las tablas, agregando o modificando la definición de los campos.
 - **ALTER TABLE Tabla1 ...**
- **DROP**: Para eliminar bases de datos o tablas.
 - **DROP DATABASE EjemploBD**
 - **DROP TABLE Tabla1**



CREATE TABLE

- SINTAXIS:

CREATE TABLE Tabla1 (Campo1 tipodatos1 restriccion1, Campo2 tipodatos2 restriccion2, ..., CampoN tipodatosN restriccionN)

- Donde los tipos de datos pueden ser:

- **Texto** (Varchar, char, text...)
- **Número** (int, decimal, float...)
- **Fecha/Hora** (date, time)
- **Booleanos** (Bool)



CREATE TABLE

- Donde las restricciones de los campos pueden ser:

[NOT NULL]
[DEFAULT valor_predeterminado]
[AUTO_INCREMENT][PRIMARY KEY] [Referencia]
[PRIMARY KEY] (columna1, columna2...)
[UNIQUE]
[CONSTRAINT nombre] FOREIGN KEY [nombre_índice][referencia]

REFERENCES nombre_tabla [(columnas_índice)]
[ON DELETE {RESTRICT|CASCADE|SET NULL|NO ACTION|SET DEFAULT}]
[ON UPDATE {RESTRICT|CASCADE|SET NULL|NO ACTION|SET DEFAULT}]



TIPOS DE DATOS EN MySQL

TIPO DATO	DESCRIPCIÓN
bigint	-9223372036854775808 a 9223372036854775807 normal. 0 a 18446744073709551615 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
blob	(Objetos grandes binarios). Tiene capacidad para 65.535 bytes de datos.
boolean	Especifica un valor booleano, tipo TRUE o FALSE, verdadero o falso.
char	Tiene una cadena de longitud fija (puede contener letras, números y caracteres especiales). El tamaño fijo se especifica entre paréntesis. Puede almacenar hasta 255 caracteres.
date	Una fecha con Formato tipo: AAAA-MM-DD
datetime	Una combinación de fecha y hora. Formato: AAAA-MM-DD HH: MI: SS Nota: El rango admitido es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'
decimal	UN DOUBLE almacenado como una cadena, lo que permite un punto decimal fijo. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño.
double	Un número grande con un punto decimal flotante. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño.
enum	Permiten ingresar una lista de valores posibles. Puede enumerar hasta 65535 valores en una lista ENUM. Si se inserta un valor que no está en la lista, se insertará un valor en blanco.
float	Un pequeño número con un punto decimal flotante. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño.
int	-2147483648 a 2147483647 normal. 0 a 4294967295, La cantidad máxima de dígitos se puede especificar entre paréntesis.

longblob	(Objetos grandes binarios). Contiene hasta 4,294,967,295 bytes de datos.
longtext	Tiene una cadena con una longitud máxima de 4.294.967.295 caracteres.
mediumblob	(Objetos grandes binarios). Almacena hasta 16.777.215 bytes de datos.
mediumint	-8388608 a 8388607 normal. 0 a 16777215, La cantidad máxima de dígitos se puede especificar entre paréntesis.
mediumtext	Tiene una cadena con una longitud máxima de 16,777,215 caracteres.
set	Similar a ENUM, excepto que SET puede contener hasta 64 elementos de lista y puede almacenar más de una opción
smallint	Tipo de dato entero con valor mínimo -32768, valor máximo - 32767
text	Sostiene una cadena con una longitud máxima de 65.535 caracteres.
time	Un tiempo. Formato: HH: MI: SS
timestamp	Una marca de tiempo. Los valores de TIMESTAMP se almacenan como el número de segundos ('1970-01-01 00:00:00' UTC).
tinyint	-128 a 127 normal. 0 a 255. La cantidad máxima de dígitos se puede especificar entre paréntesis.
tinytext	Tiene una cadena con una longitud máxima de 255 caracteres.
varchar	Tiene una cadena de longitud variable (puede contener letras, números y caracteres especiales). El tamaño máximo se especifica entre paréntesis. Puede almacenar hasta 255 caracteres.
year	Un año en formato de dos o cuatro dígitos.



CREATE TABLE

- Crear la siguiente tabla con el nombre **CLIENTES**:

CAMPO	TIPO DATOS
NOMBRE	VARCHAR (20)
APELLIDOS	VARCHAR (30)

✓ ***CREATE TABLE CLIENTES (NOMBRE VARCHAR(20), APELLIDOS VARCHAR(30))***



CREATE TABLE

- Crear la siguiente tabla con el nombre CLIENTES:

CAMPO	TIPO DATOS
ID_CLIENTE	INT / AUTO_INCREMENT / PRIMARY KEY
NOMBRE	VARCHAR (20)
APELLIDOS	VARCHAR (30)
EDAD	INT (2)
FECHA_NACIMIENTO	DATE

- ✓ *CREATE TABLE CLIENTES (ID_CLIENTE INT AUTO_INCREMENT PRIMARY KEY, NOMBRE VARCHAR(20), APELLIDOS VARCHAR(30), EDAD INT(2), FECHA_NACIMIENTO DATE)*
- ✓ *CREATE TABLE CLIENTES (ID_CLIENTE INT AUTO_INCREMENT, NOMBRE VARCHAR(20), APELLIDOS VARCHAR(30), EDAD INT(2), FECHA_NACIMIENTO DATE, PRIMARY KEY (ID_CLIENTE))*



CREATE TABLE

- Crear la siguiente tabla con el nombre **CLIENTES**:

CAMPO	TIPO DATOS
ID_CLIENTE	INT / AUTO_INCREMENT / PRIMARY KEY
NOMBRE	VARCHAR (20)
APELLIDOS	VARCHAR (30)
DNI	VARCHAR (9) / UNICO / NOT NULL
POBLACION	VARCHAR (30) / VALOR POR DEFECTO VALENCIA

✓ CREATE TABLE CLIENTES (ID_CLIENTE INT AUTO_INCREMENT PRIMARY KEY, NOMBRE VARCHAR(20), APELLIDOS VARCHAR(30), DNI VARCHAR(9) UNIQUE NOT NULL, POBLACION VARCHAR (30) DEFAULT 'VALENCIA')



CREATE TABLE

- Crear la siguiente tabla con el nombre COMPRAS:

CAMPO	TIPO DATOS
ID_COMPRA	INT / AUTO_INCREMENT / PRIMARY KEY
ARTICULO	VARCHAR (30)
CANTIDAD	INT (5)
CLIENTE	INT / RELACIONADO CON LA TABLA CLIENTE

✓ ***CREATE TABLE COMPRAS (ID_COMPRAS INT AUTO_INCREMENT PRIMARY KEY, ARTICULO VARCHAR(20), CANTIDAD INT(5), CLIENTE INT, CONSTRAINT REL1 FOREIGN KEY (CLIENTE) REFERENCES CLIENTES (ID_CLIENTE))***



CREATE TABLE

- Crear la siguiente tabla con el nombre COMPRAS:

CAMPO	TIPO DATOS
ID_COMPRA	INT / AUTO_INCREMENT / PRIMARY KEY
ARTICULO	VARCHAR (30)
CANTIDAD	INT (5)
CLIENTE	INT / RELACIONADO CON LA TABLA CLIENTE ELIMINADOSE EN CASCADA

✓ ***CREATE TABLE COMPRAS (ID_COMPRAS INT AUTO_INCREMENT PRIMARY KEY, ARTICULO VARCHAR(20), CANTIDAD INT(5), CLIENTE INT, CONSTRAINT REL1 FOREIGN KEY (CLIENTE) REFERENCES CLIENTES (ID_CLIENTE) ON DELETE CASCADE ON UPDATE CASCADE)***



ALTER

- **ALTER**: Para modificar las tablas, agregando o modificando la definición de los campos.

- Para **añadir** una columna a una tabla:

```
ALTER TABLE Tabla1 ADD COLUMN NombreColumna tipodato;
```

- Para **eliminar** una columna de una tabla:

```
ALTER TABLE Tabla1 DROP COLUMN NombreColumna;
```

- Para **modificar** una columna de una tabla:

```
ALTER TABLE Tabla1 MODIFY COLUMN NombreColumna tipodato;
```



ALTER

- Añadir a la tabla de COMPRAS la columna de PRECIO que es decimal.

ALTER TABLE COMPRAS ADD COLUMN PRECIO DECIMAL

- Eliminar de la tabla de COMPRAS la columna de PRECIO.

ALTER TABLE COMPRAS DROP COLUMN PRECIO

- Modificar de la tabla de COMPRAS la columna de Cantidad que es *int* por *decimal*.

ALTER TABLE COMPRAS MODIFY COLUMN CANTIDAD DECIMAL

- Añadir el valor por defecto (0) de la columna de Cantidad de la tabla Compras.

ALTER TABLE COMPRAS ALTER COLUMN CANTIDAD SET DEFAULT 0



ALTER

- Eliminar la clave principal de la tabla de CLIENTES (ID_CLIENTE)

ALTER TABLE CLIENTES DROP PRIMARY KEY

- Añadir como clave primaria en la tabla clientes el ID_CLIENTE.

ALTER TABLE CLIENTES ADD PRIMARY KEY (ID_CLIENTE)

- Eliminar la clave ajena de la tabla de COMPRAS.

ALTER TABLE COMPRAS DROP FOREIGN KEY rel1

- Añadir la relación entre compras y clientes.

ALTER TABLE COMPRAS ADD CONSTRAINT REL2 FOREIGN KEY (CLIENTE) REFERENCES CLIENTES(ID_CLIENTE)



DROP

- **DROP**: Para eliminar bases de datos o tablas.
- Para **ELIMINAR** una base de datos

```
DROP DATABASE NombreBaseDatos
```

- Para **ELIMINAR** una tabla:

```
DROP TABLE NombreTabla;
```



DROP

- Eliminar la tabla de COMPRAS

DROP TABLE COMPRAS

- Eliminar la base de datos del Ejemplo

DROP DATABASE EJEMPLO