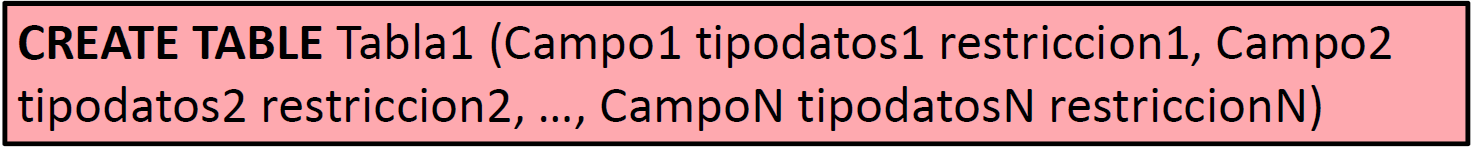
**SQL – DDL**

**CREATE: Crear nuevas bases de datos o tablas**

* CREATE DATABASE NombreBaseDatos
* CREATE TABLE NombreTabla



Restricciones de los campos:

Texto

El contenido generado por IA puede ser incorrecto.

Creación de tabla incluyendo relación:

Texto

El contenido generado por IA puede ser incorrecto.

**ALTER: Modificar tablas agregando/modificando definición de campos**

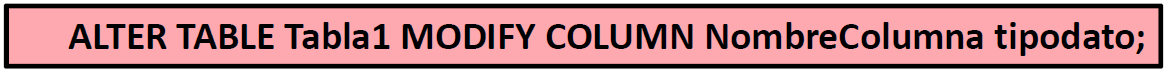
Añadir una columna:



Eliminar una columna:



Modificar una columna:



Añadir a la tabla Compras la columna Precio, DECIMAL:

**ALTER TABLE** Compras **ADD COLUMN** Precio DECIMAL

Eliminar de la tabla de Compras la columna Precio:

**ALTER TABLE** Compras **DROP COLUMN** Precio

Modificar de la tabla Comprar 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 Cantidad de la tabla Compras:

**ALTER TABLE** Compras **ALTER COLUMN** Cantidad **SET** DEFAULT 0

Eliminar la clave principal de la tabla 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 Compras:

**ALTER TABLE** Clientes **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: Eliminar bases o tablas**

Eliminar una base de datos:



**DROP TABLE** Compras

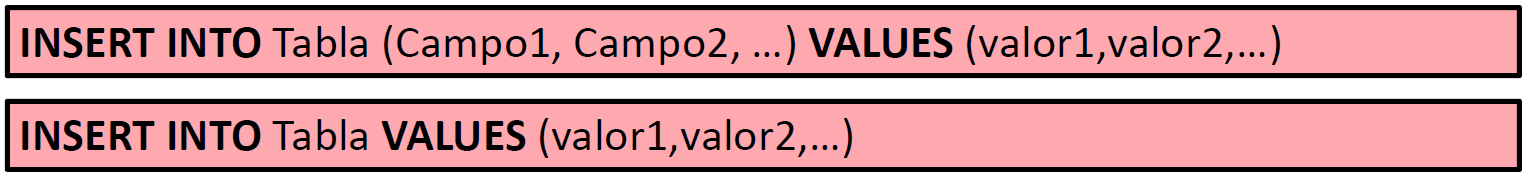
Eliminar una tabla:



**DROP DATABASE** NombreBaseDatos

**SQL DML**

Sintaxis para introducir un registro:

****

**INSERT INTO** Clientes **VALUES** (NULL, 'Maria', 'Garcia Castillo', 35, 'Valencia')

**INSERT INTO** Clientes(Código, Nombre, Apellidos, Edad, Población) **VALUES** (NULL, 'María', 'García Castillo', 35, 'Valencia')

Insertar un valor por defecto:

**INSERT INTO** Clientes **VALUES** (NULL, 'Sonia', 'Carretero Ruíz', 28, DEFAULT)

No sabemos algún campo o no queremos rellenarlo:

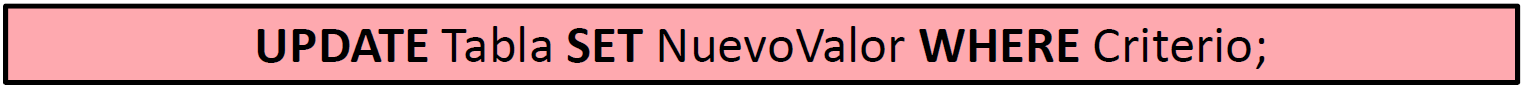
**INSERT INTO** Clientes **VALUES** (NULL, 'Sergio', 'Miralles Saez', NULL, ' Alaquàs')

Solo se requiere rellenar los campos necesarios:

**INSERT INTO** Clientes (Código, Nombre, Apellidos, Edad) **VALUES** (NULL, 'Silvia', 'Ramírez Vilches', 29)

**UPDATE – Actualización de tablas**

Sintaxis:



Actualizar las personas que viven en Alaquàs ahora viven en Torrent:

**UPDATE** Clientes **SET** Población = “Torrent” **WHERE** Población = “Alaquàs”

Actualizar edad de todos los clientes que tienen entre 30 y 50 años, añadiéndole 3 años más:

**UPDATE** Clientes **SET** Edad = Edad + 3 **WHERE** Edad **BETWEEN** 30 AND 50

**Creación de nuevas tablas a partir de una tabla con criterios especificados**

****

Crear una tabal a partir de la tabla Clientes, con los clientes que son de Valencia. La nueva tabla se llama Clientes\_Valencia:

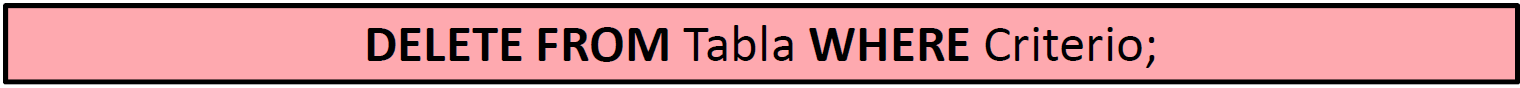
**CREATE TABLE** Clientes\_Valencia **SELECT** \* **FROM** Clientes **WHERE** Población = “Valencia”

Igual pero solo con el Código, Nombre y Apellidos:

**CREATE TABLE** Clientes\_Valencia **SELECT** Código, Nombre, Apellidos **FROM** Clientes **WHERE** Población = “Valencia”

**DELETE – Eliminar registros dependiendo del criterio**

Sintaxis:



Eliminar los clientes de Valencia de la tabla Clientes:

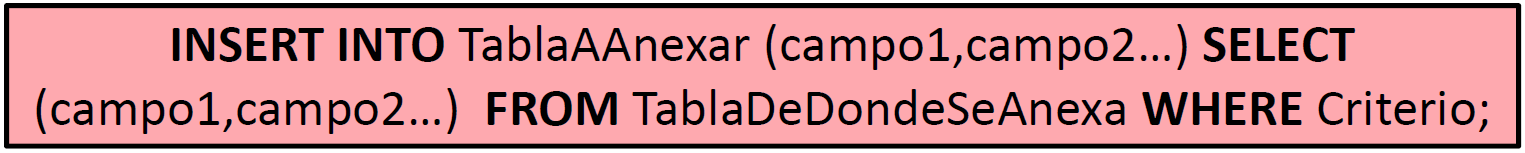
**DELETE FROM** Clientes **WHERE** Población = “Valencia”

Eliminar de la tabla Clientes\_Valencia los que se llamen José y tengan + 45 años.

**DELETE FROM** Clientes\_Valencia **WHERE** Nombre = “Jose” **AND** Edad > 45

**INSERT INTO – Anexar datos de otra tabla**

Sintaxis:

****

Anexar a tabla Clientes los datos e la tabla Clientes\_Valencia, todos los campos:

**INSERT INTO** Clientes **SELECT** \* **FROM** Clientes\_Valencia

Solo anexar campos Nombre y Edad:

**INSERT INTO** Clientes(Nombre, Edad) **SELECT** Nombre, Edad **FROM** Clientes Valencia

Anexar los campos Nombre, Apellidos y Edad, pero solo los que sean mayores de 50:

**INSERT INTO** Clientes (Nombre, Apellidos, Edad) **SELECT** Nombre, Apellidos, Edad **FROM** Clientes\_Valencia **WHERE** Edad > 50

**TRIGGERS**

Sintaxis:

Texto

El contenido generado por IA puede ser incorrecto.

2 palabras clave:

* OLD: Valor anterior al disparo.
* NEW: Valor posterior al disparo.

Según la operación se utilizarán 1 o 2:

* INSERT: Solo NEW.
* UPDATE: OLD y NEW.
* DELETE: Solo OLD.

Cada vez que alguien inserte un nuevo cliente en la tabla Clientes, en la tabla Auditoría\_clientes se insertara una nueva entrada:

**CREATE TRIGER** AudClient\_Insertar

**AFTER INSERT**

**ON** Clientes

**FOR EACH ROW**

**INSERT INTO** Auditoría\_clientes (Nombre\_nuevo, Seccion\_nuevo, usuario, Fecha\_modif, Proceso, Id\_cliente)

**VALUES** (NEW.Nombre, NEW.Seccion, CURRENT\_USER(), NOW, NEW.Accion, NEW.Id\_cliente)

Cada vez que alguien actualice un nuevo cliente en la tabla Clientes, en la tabla Auditoría\_clientes se insertara una nueva entrada:

**CREATE TRIGER** AudClient\_Modificar

**BEFORE UPDATE**

**ON** Clientes

**FOR EACH ROW**

**INSERT INTO** Auditoría\_clientes (Nombre\_ant, Seccion\_ant, Nombre\_nuevo, Seccion\_nuevo, usuario, Fecha\_modif, Proceso, Id\_cliente)

**VALUES** (OLD.Nombre, OLD.Seccion, NEW.Nombre, NEW.Seccion, CURRENT\_USER(), NOW, NEW.Accion, NEW.Id\_cliente)

Cada vez que alguien borre un nuevo cliente en la tabla Clientes, en la tabla Auditoría\_clientes se insertara una nueva entrada.

**CREATE TRIGER** AudClient\_Eliminar

**BEFORE DELETE**

**ON** Clientes

**FOR EACH ROW**

**INSERT INTO** Auditoría\_clientes (Nombre\_ant, Seccion\_ant, usuario, Fecha\_modif, Proceso, Id\_cliente)

**VALUES** (OLD.Nombre, OLD.Seccion, CURRENT\_USER(), NOW, OLD.Id\_cliente)

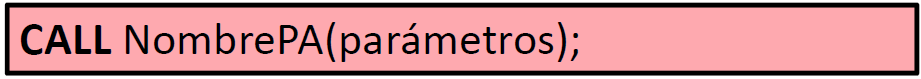
**PROCEDIMIENTOS**

Sintaxis para instrucción SQL:

Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.

Ejecución en la base de datos:



Procedimientos sin parámetros – Consulta de Clientes de valencia:

**CREATE PROCEDURE Clientes\_Valencia()**

**SELECT** \* **FROM** Clientes **WHERE** Población = “Valencia”

Procedimientos con parámetros – Consulta de Clientes de valencia:

**CREATE PROCEDURE** Act\_Precio(Precio\_Nue DECIMAL(5,2), Cod\_Art INT(6))

**UPDATE** Articulos **SET** Precio = Precio\_Nuev **WHERE** Codigo\_articulo = Cod\_Art

**Procedimiento con más de una instrucción SQL**

Sintaxis:

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

Declaración de variables en Procedimientos – Calcular la edad según el año de nacimiento:

**DELIMITER //**

**CREATE PROCEDURE** EdadPersona(Año\_nac INT)

**BEGIN**

**DECLARE** Año\_act INT DEFAULT 2021;

**DECLARE** Edad INT;

**SET** Edad = Año\_act – Año\_nac;

**SELECT** Edad;

**END; //**

**DELIMITER ;**

Triggers condicionales – Actualizar precio, pero controlando que no sea negativo ni > 900

**DELIMITER //**

**CREATE TRIGGER** Act\_Precio **BEFORE UPDATE ON** Articulos **FOR EACH ROW**

**BEGIN**

**IF** (New.Precio<0) **THEN**

**SET** New.Precio = 1;

**ELSEIF** (New.Precio > 900) **THEN**

**SET** New.Precio = 900;

**END IF;**

**END; //**

**DELIMITER ;**