

## CASO PRÁCTICO

### CREACIÓN DE BASES DE DATOS, USUARIOS Y TABLAS

#### Contexto

Eres desarrollador en una compañía que vende y distribuye productos de informática. Los dueños de la empresa te han pedido que crees una base de datos que les permita almacenar toda la información necesaria para gestionar clientes, productos y pedidos.

1. Crear una base de datos llamada "tienda".

```
CREATE DATABASE tienda;
```

2. Crear la tabla "clientes" con la siguiente especificación.

Columnas:(ID, NOMBRE, APELLIDOS, TELEFONO, DNI, CALLE, NUMERO, CP, POBLACION, PROVINCIA)

- id: Número entero autoincrementable. Clave primaria.
- nombre: Texto de 255 caracteres como máximo. No nulo.
- apellidos: Texto de 255 caracteres como máximo. No nulo.
- telefono: Número entero. No nulo.
- dni: Texto de 9 caracteres como máximo. No nulo.
- dir\_calle: Texto de 255 caracteres como máximo. No nulo.
- dir\_numero: Número entero. Puede ser nulo.
- dir\_cp: Texto de 5 caracteres como máximo. No nulo.
- dir\_poblacion: Texto de 255 caracteres como máximo. No nulo.
- dir\_provincia: Texto de 255 caracteres como máximo. No nulo.

```
CREATE TABLE clientes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    apellidos VARCHAR(255) NOT NULL,  
    telefono INT NOT NULL,  
    dni VARCHAR(9) NOT NULL,  
    dir_calle VARCHAR(255) NOT NULL,  
    dir_numero INT,  
    dir_cp VARCHAR(5) NOT NULL,  
    dir_poblacion VARCHAR(255) NOT NULL,  
    dir_provincia VARCHAR(255) NOT NULL  
);
```

3. Crear la tabla “pedidos” con la siguiente especificación.

Columnas: (ID, FK\_CLIENTE\_ID, IMPORTE\_TOTAL, FECHA\_Y\_HORA)

- id: Número entero autoincrementable. Clave primaria.
- fk\_cliente\_id: Clave foránea con el ID del cliente al que pertenece el pedido. No nulo.
- importe\_total: Número con decimales. No nulo.
- fecha\_y\_hora: Timestamp. No nulo. Valor por defecto: la fecha y hora actual.

```
CREATE TABLE pedidos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    fk_cliente_id INT NOT NULL,  
    importe_total FLOAT NOT NULL,  
    fecha_y_hora TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (fk_cliente_id) REFERENCES clientes(id)  
);
```

4. Crear la tabla “productos” con la siguiente especificación.

Columnas: (ID, NOMBRE, DESCRIPCIÓN, PRECIO, PESO, STOCK)

- id: Número entero autoincrementable. Clave primaria.
- nombre: Texto de 255 caracteres como máximo. No nulo.
- descripcion: Texto de 255 caracteres como máximo. Puede ser nulo.
- precio: Número con decimales. No nulo.
- peso: Número con decimales. No nulo.
- stock: Número entero. No nulo.

```
CREATE TABLE productos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    descripcion VARCHAR(255) NOT NULL,  
    precio FLOAT NOT NULL,  
    peso FLOAT NOT NULL,  
    stock INT NOT NULL  
);
```

5. Crear la tabla “productos\_pedido” con la siguiente especificación.

Columnas: (FK\_PEDIDO\_ID, FK\_PRODUCTO\_ID, CANTIDAD)

- fk\_pedido\_id: Clave foránea con el ID del pedido. No nulo. Clave primaria.
- fk\_producto\_id: Clave foránea con el ID del producto. No nulo. Clave primaria.
- cantidad: Número entero. No nulo. Valor por defecto: 0.

```
CREATE TABLE productos_pedido (  
    fk_pedido_id INT NOT NULL,  
    fk_producto_id INT NOT NULL,  
    cantidad INT NOT NULL DEFAULT 0,  
    PRIMARY KEY (fk_pedido_id, fk_producto_id),  
    FOREIGN KEY (fk_pedido_id) REFERENCES pedidos(id),  
    FOREIGN KEY (fk_producto_id) REFERENCES productos(id),  
);
```

Una vez creadas las tablas, resuelve lo siguiente:

6. Indica la instrucción SQL necesaria para: eliminar la columna “peso” de la tabla productos.

```
ALTER TABLE productos  
DROP COLUMN peso;
```

7. Indica la instrucción SQL necesaria para: añadir una columna llamada estado en la tabla pedidos. Texto de 50 caracteres como máximo. No nulo.

```
ALTER TABLE pedidos  
ADD COLUMN estado VARCHAR(50);
```

8. Indica la instrucción SQL necesaria para: eliminar la tabla Productos\_Pedido.

```
DROP TABLE productos_pedido;
```

9. Indica la instrucción SQL necesaria para: cambiar el nombre de la columna CP por CODIGO\_POSTAL, en la tabla Clientes.

```
ALTER TABLE clientes  
CHANGE COLUMN dir_cp CODIGO_POSTAL VARCHAR(5);
```

10. Inserta 3 o 4 registros en cada tabla

```
INSERT INTO clientes (nombre, apellidos, telefono, dni, dir_calle, dir_numero,  
CODIGO_POSTAL, dir_poblacion, dir_provincia)  
VALUES ('Pedro', 'Martino Garcia', 555666333, '12345678H', 'tenor', 24, '50018',  
'Casetas', 'Zaragoza');
```

```
INSERT INTO clientes (nombre, apellidos, telefono, dni, dir_calle, dir_numero,  
CODIGO_POSTAL, dir_poblacion, dir_provincia)  
VALUES ('Juan', 'Perez Hevia', 554665313, '14365658H', 'ochoa', 14, '50010', 'Utebo',  
'Zaragoza');
```

```
INSERT INTO clientes (nombre, apellidos, telefono, dni, dir_calle, dir_numero,  
CODIGO_POSTAL, dir_poblacion, dir_provincia)  
VALUES ('Diana', 'Gil Cueto', 234661393, '10363678D', 'hispanidad', 5, '50000',  
'Alfocea', 'Zaragoza');
```

```
INSERT INTO pedidos (fk_cliente_id, importe_total)  
VALUES (1, 20000);
```

```
INSERT INTO pedidos (fk_cliente_id, importe_total)  
VALUES (10, 40000);
```

```
INSERT INTO pedidos (fk_cliente_id, importe_total)  
VALUES (23, 20000);
```

```
INSERT INTO productos (nombre, descripcion, peso, precio, stock)  
VALUES ('camiseta', 'trail', 240, 100, 3);
```

```
INSERT INTO productos (nombre, descripcion, peso, precio, stock)  
VALUES ('casco', 'nieve', 220, 80, 5);
```

```
INSERT INTO productos (nombre, descripcion, peso, precio, stock)  
VALUES ('pantalon', 'montaña', 440, 200, 2);
```

11. Por último, debes indicar la sentencia utilizada para crear 2 usuarios llamados Usuario1 y Usuario2. Al primero de ellos debes otorgarle todos los permisos. Al Usuario2 debes darle permisos de selección y modificación para, posteriormente, quitarle el permiso de modificación.

```
CREATE USER 'Usuario1' @ 'localhost' IDENTIFIED BY 'asdfgh';  
GRANT ALL PRIVILEGES ON tienda.* TO 'Usuario1'@'localhost';  
FLUSH PRIVILEGES;  
CREATE USER 'Usuario2'@'localhost' IDENTIFIED BY '123456';  
GRANT SELECT, UPDATE ON tienda.* TO 'Usuario2'@'localhost';  
FLUSH PRIVILEGES;  
REVOKE UPDATE ON tienda.* FROM 'Usuario2'@'localhost';  
FLUSH PRIVILEGES;
```