

Problemario de operaciones CRUD #2

Creacion de la base de datos

```
CREATE DATABASE tienda_virtual;

USE tienda_virtual;

CREATE TABLE productos (
    producto_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    categoria VARCHAR(50),
    precio DECIMAL(10, 2),
    stock INT,
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE clientes (
    cliente_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100) UNIQUE,
    fecha_registro DATE DEFAULT CURDATE()
);

CREATE TABLE pedidos (
    pedido_id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    fecha_pedido DATETIME DEFAULT CURRENT_TIMESTAMP,
    total DECIMAL(10, 2),
    FOREIGN KEY (cliente_id) REFERENCES clientes(cliente_id)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

CREATE TABLE detalle_pedidos (
    detalle_id INT AUTO_INCREMENT PRIMARY KEY,
    pedido_id INT,
    producto_id INT,
    cantidad INT,
    precio_unitario DECIMAL(10, 2),
    FOREIGN KEY (pedido_id) REFERENCES pedidos(pedido_id)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    FOREIGN KEY (producto_id) REFERENCES productos(producto_id)
    ON UPDATE CASCADE ON DELETE RESTRICT
);
```

Ejercicios CREATE

1. Inserta 5 productos diferentes en la tabla `productos` .

Instrucción: Los productos deben incluir un nombre, categoría, precio y stock inicial.

```
INSERT INTO productos (nombre, categoria, precio, stock)
VALUES
('Tablet', 'Electrónica', 350.00, 20),
('Teclado', 'Electrónica', 40.00, 70),
('Pantalón', 'Ropa', 25.00, 80),
('Batidora', 'Electrodomésticos', 100.00, 25),
('Revista', 'Libros', 12.00, 150);
```

2. Registra 3 clientes en la tabla `clientes` .

Instrucción: Ingresa datos de nombre y correo para cada cliente. Asegúrate de que los correos sean únicos.

```
INSERT INTO clientes (nombre, correo)
VALUES
('María Fernández', 'maria.fernandez@gmail.com'),
('Luis Martínez', 'luis.martinez@yahoo.com'),
('Sofía Torres', 'sofia.torres@outlook.com');
```

3. Inserta 2 pedidos hechos por diferentes clientes.

Instrucción: Cada pedido debe tener al menos 2 productos, especifica la cantidad y el precio unitario de cada uno.

```
INSERT INTO pedidos (cliente_id, total)
VALUES
(1, 390.00), -- Pedido de María Fernández
(2, 137.00); -- Pedido de Luis Martínez

INSERT INTO detalle_pedidos (pedido_id, producto_id, cantidad, precio_unitario)
VALUES
(1, 1, 1, 350.00), -- María compra una tablet
(1, 2, 1, 40.00), -- María compra un teclado
(1, 4, 1, 100.00), -- María compra una batidora
(2, 3, 3, 25.00), -- Luis compra 3 pantalones
(2, 5, 1, 12.00); -- Luis compra una revista
```

Ejercicios READ

1. Obtén una lista de todos los productos que tienen un stock mayor a 10 unidades.

Instrucción: Muestra el `producto_id`, `nombre`, `precio` y `stock`.

```
SELECT producto_id, nombre, precio, stock
FROM productos
WHERE stock > 10;
```

2. Encuentra los pedidos realizados por un cliente en particular.

Instrucción: Muestra el `nombre` del cliente, `pedido_id`, `fecha_pedido` y el `total`.

```
SELECT c.nombre, p.pedido_id, p.fecha_pedido, p.total
FROM pedidos p
JOIN clientes c ON p.cliente_id = c.cliente_id
WHERE c.nombre = 'Luis Martínez';
```

3. Muestra el total de ventas por cada producto.

Instrucción: Agrupa por `producto_id` y muestra el `nombre` del producto y la cantidad total vendida en todos los pedidos.

```
SELECT p.nombre, SUM(d.cantidad) AS total_vendido
FROM productos p
JOIN detalle_pedidos d ON p.producto_id = d.producto_id
GROUP BY p.producto_id;
```

Ejercicios UPDATE

1. Actualiza el precio de todos los productos de una categoría aumentando un 15%.

Instrucción: Usa la columna `categoría` para filtrar los productos.

```
UPDATE productos
SET precio = precio * 1.05
WHERE categoria = 'Libros';
```

2. Modifica el correo de uno de los clientes por un nuevo correo electrónico.

Instrucción: Asegúrate de que el nuevo correo sea único.

```
UPDATE clientes
SET correo = 'nuevo.correo@gmail.com'
WHERE cliente_id = 1;
```

3. Corrige el stock de un producto cuyo stock actual es incorrecto. *Instrucción:* Busca el producto por su `producto_id` y actualiza el campo `stock`.

```
UPDATE productos SET stock = 25 WHERE producto_id = 3;
```

Ejercicios DELETE

1. Elimina todos los productos de la tabla `productos` que no tienen stock disponible.

Instrucción: Debes usar la columna `stock` para identificar productos con stock igual a 0.

```
DELETE FROM productos WHERE stock = 0;
```

2. Borra un pedido que fue cancelado por el cliente.

Instrucción: Elimina el pedido junto con todos los registros relacionados en la tabla `detalle_pedidos`.

```
DELETE FROM detalle_pedidos WHERE pedido_id = 1;

DELETE FROM pedidos WHERE pedido_id = 1;
```

3. Elimina un cliente que ha solicitado la eliminación de su cuenta.

Instrucción: Asegúrate de borrar primero los registros relacionados en la tabla `pedidos` y luego el cliente de la tabla `clientes`.

```
DELETE FROM pedidos WHERE cliente_id = 3;

DELETE FROM clientes WHERE cliente_id = 3;
```