

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 ('Smartphone X',
'Electrónica', 699.99, 50),
('Laptop Pro', 'Computadoras', 1200.00, 30),
('Teclado Mecánico', 'Accesorios', 80.50, 100),
('Audífonos Inalámbricos', 'Electrónica', 150.00, 75),
('Monitor 4K', 'Pantallas', 350.99, 25);
```

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
('Juan Perez', 'juan.perez@example.com'),
('Ana Gomez', 'ana.gomez@example.com'),
('Carlos Ruiz', 'carlos.ruiz@example.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.

````sql`

```
INSERT INTO pedidos (cliente_id, total) VALUES
(1, 899.99), -- Pedido de Juan
(2, 1350.00); -- Pedido de Ana
```

-- Detalle de los pedidos

```
INSERT INTO detalle_pedidos (pedido_id, producto_id, cantidad, precio_unitario) VALUES
(1, 1, 1, 699.99), -- Juan compró 1 Smartphone X
(1, 3, 5, 80.50), -- Juan compró 5 Teclados Mecánicos
(2, 2, 1, 1200.00), -- Ana compró 1 Laptop Pro
(2, 5, 2, 350.99); -- Ana compró 2 Monitores 4K
```

## 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``.

````sql`

```
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 = 'Juan Perez';
```

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

Ejercicios UPDATE

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

**Instrucción:* Usa la columna ``categoria`` para filtrar los productos.

````sql`

```
UPDATE productos
SET precio = precio * 1.15
WHERE categoria = 'Electrónica';
```

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 = 'juan.nuevo@example.com'
WHERE correo = 'juan.perez@example.com';
```

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`.

```sql

```
UPDATE productos
SET stock = 40
WHERE producto_id = 1;
```

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`.

```
```sql
```

```
DELETE FROM detalle_pedidos
```

```
WHERE pedido_id IN (SELECT pedido_id FROM pedidos WHERE cliente_id = 2);
```

```
DELETE FROM pedidos
```

```
WHERE cliente_id = 2;
```

```
DELETE FROM clientes
```

```
WHERE cliente_id = 2;
```