

2. Ejercicios de modificación de datos

1. Ejercicios de INSERT, UPDATE y DELETE

Objetivo: Practicar la manipulación de datos en la base de datos de retail deportivo.

Datos de Ejemplo para Insertar

```
-- Insertar categorías
INSERT INTO categorias (nombre, descripcion) VALUES
('Running', 'Zapatillas y ropa para correr'),
('Fútbol', 'Balones, botas y equipamiento para fútbol'),
('Fitness', 'Accesorios para gimnasio y entrenamiento'),
('Natación', 'Bañadores, gafas y material para piscina'),
('Ciclismo', 'Bicicletas y componentes');

-- Insertar productos
INSERT INTO productos (nombre, descripcion, precio, stock, id_categoria, marca)
VALUES
('Zapatillas Run Fast', 'Zapatillas ligeras para corredores', 89.99, 50, 1,
'Nike'),
('Balón Adidas Champions', 'Balón oficial de la Champions League', 129.99, 30, 2,
'Adidas'),
('Mancuernas 5kg', 'Par de mancuernas de neopreno', 29.99, 100, 3, 'Decathlon'),
('Gafas de Natación Speed', 'Gafas de competición', 24.99, 40, 4, 'Speedo'),
('Bicicleta MTB Pro', 'Bicicleta montaña 21 velocidades', 599.99, 15, 5, 'Trek');

-- Insertar clientes
INSERT INTO clientes (nombre, apellidos, email, telefono, puntos_fidelidad) VALUES
('Juan', 'García', 'juan@email.com', '611223344', 100),
('María', 'López', 'maria@email.com', '655667788', 50),
('Carlos', 'Martínez', 'carlos@email.com', '699112233', 200);

-- Insertar pedidos
INSERT INTO pedidos (id_cliente, estado, total, metodo_pago) VALUES
(1, 'entregado', 219.98, 'tarjeta'),
(2, 'procesado', 59.98, 'paypal'),
(3, 'pendiente', 599.99, 'transferencia');

-- Insertar detalles de pedido
INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad, precio_unitario)
VALUES
(1, 1, 1, 89.99), -- Pedido 1: Zapatillas Run Fast
(1, 3, 1, 29.99), -- Pedido 1: Mancuernas 5kg
(2, 4, 2, 24.99), -- Pedido 2: 2x Gafas de Natación
(3, 5, 1, 599.99); -- Pedido 3: Bicicleta MTB Pro
```

Objetivo: Añadir nuevos registros a la base de datos.

1. **Insertar una nueva categoría** llamada "Tenis" con la descripción "Raquetas y calzado para tenis".
2. **Añadir un nuevo producto:**
 - Nombre: "Raqueta Pro Tour"
 - Descripción: "Raqueta profesional para competición"
 - Precio: 199.99
 - Stock: 20
 - Categoría: "Tenis"
 - Marca: "Wilson"
3. **Registrar un nuevo cliente:**
 - Nombre: "Laura"
 - Apellidos: "Fernández"
 - Email: "laura@email.com"
 - Teléfono: "644556677"
 - Puntos fidelidad: 0

◊ Ejercicios de UPDATE

Objetivo: Modificar datos existentes.

1. **Actualizar el precio** de las "Zapatillas Run Fast" a **79.99**.
2. **Incrementar el stock** de todos los productos de la categoría "Fitness" en **10 unidades**.
3. **Sumar 50 puntos** a los clientes que han realizado pedidos superiores a **100€**.

◊ Ejercicios de DELETE

Objetivo: Eliminar registros de forma controlada.

1. **Borrar los productos** con **stock menor a 5**.
2. **Eliminar pedidos cancelados** (estado = 'cancelado').
3. **Borrar clientes inactivos** (sin pedidos registrados).

2. Ejercicios con TRANSACCIONES y ROLLBACK

Objetivo: Aprender a gestionar operaciones atómicas y revertir cambios si hay errores.

◊ Ejemplo de Transacción Completa

```
-- Iniciar transacción
START TRANSACTION;

-- Paso 1: Insertar nuevo pedido
INSERT INTO pedidos (id_cliente, estado, total, metodo_pago)
VALUES (1, 'pendiente', 199.98, 'tarjeta');

-- Paso 2: Obtener el ID del pedido recién creado
```

```
SET @nuevo_pedido_id = LAST_INSERT_ID();

-- Paso 3: Añadir productos al pedido
INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad, precio_unitario)
VALUES (@nuevo_pedido_id, 2, 1, 129.99); -- Balón Adidas

INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad, precio_unitario)
VALUES (@nuevo_pedido_id, 4, 1, 24.99); -- Gafas de Natación

-- Paso 4: Actualizar stock de productos
UPDATE productos SET stock = stock - 1 WHERE id_producto = 2; -- Balón
UPDATE productos SET stock = stock - 1 WHERE id_producto = 4; -- Gafas

-- Verificar si hay suficiente stock
IF (SELECT stock FROM productos WHERE id_producto = 2) < 0 THEN
    -- Si no hay stock, deshacer cambios
    ROLLBACK;
    SELECT 'Error: No hay stock suficiente. Transacción cancelada.' AS mensaje;
ELSE
    -- Si todo está OK, confirmar cambios
    COMMIT;
    SELECT 'Pedido registrado correctamente.' AS mensaje;
END IF;
```

♦ Ejercicios con Transacciones

1. Realizar una compra con verificación de stock:

- Insertar un pedido con 2 productos.
- Actualizar el stock.
- Si algún producto no tiene stock suficiente, deshacer la operación.

2. Actualizar puntos de fidelidad:

- Sumar 10 puntos por cada 50€ gastados en pedidos entregados.
- Si ocurre un error, revertir los cambios.

3. Borrado seguro de clientes:

- Eliminar un cliente y todos sus pedidos.
- Usar una transacción para asegurar que si falla, no se borre nada.

3. Ejemplo de SAVEPOINT (Anulación Parcial)

```
START TRANSACTION;

-- Paso 1: Insertar categoría
INSERT INTO categorias (nombre, descripcion) VALUES ('Yoga', 'Accesorios para yoga');
```

```
-- Guardar punto de recuperación
SAVEPOINT sp_categoria;

-- Paso 2: Insertar productos (simular error)
INSERT INTO productos (nombre, precio, stock, id_categoria) VALUES
('Esterilla Yoga', 19.99, 30, 100); -- Error: categoría 100 no existe

-- Si hay error, volver al SAVEPOINT
IF ROW_COUNT() = 0 THEN
    ROLLBACK TO sp_categoria;
    SELECT 'Error: Categoría no existe. Se canceló la inserción de productos.' AS
mensaje;
ELSE
    COMMIT;
END IF;
```

4. Ejercicios de SAVEPOINT

1. Actualizar múltiples productos:

- Crear un **SAVEPOINT** después de actualizar el primer producto.
- Si falla la actualización del segundo, revertir solo esa parte.

2. Registro de pedido en dos pasos:

- Paso 1: Insertar pedido (**SAVEPOINT**).
- Paso 2: Añadir detalles. Si hay error, deshacer solo los detalles.

Resumen de Habilidades Practicadas

- ✓ **INSERT, UPDATE, DELETE** (manipulación de datos).
- ✓ **Transacciones** (COMMIT, ROLLBACK).
- ✓ **SAVEPOINT** (anulación parcial).
- ✓ **Gestión de errores** en operaciones críticas.