

Nombre de la práctica	Problemario 2 ejercicios de BD			No.	2
Asignatura:	Taller de Bases de Datos	Carrera :	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	2

I. Nombre del alumno: Francisco David Colin Llra

II. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado: Laptop, gestor de bases de datos

IV. Desarrollo de la práctica:

Creación 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**.

```
INSERT productos (nombre, categoria, precio, stock)  
VALUES  
    ("Gansito", "Panecillo", 20.0, 10),  
    ("Pingüinos", "Panecillo", 25.0, 20),  
    ("Doritos Dinamita", "Frituras", 18.0, 40),  
    ("Sabritas Naturales", "Frituras", 18.0, 20),  
    ("Takis", "Frituras", 20.0, 20);
```

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

```
INSERT INTO clientes (nombre, correo)VALUES
  ("Francisco", "ejemplo1@gmail.com"),
  ("Jose", "ejemplo2@gmail.com"),
  ("Antonio", "ejemplo3@gmail.com");
```

3.- Inserta 2 pedidos hechos por diferentes clientes.

```
-- entidad fuerte, pedido
INSERT INTO pedidos (cliente_id, total) VALUES
  (1, 65.0),
  (2, 36.0);
-- entidad debil, detalle_pedidos depende de pedidos
INSERT INTO detalle_pedidos (pedido_id, producto_id, cantidad, precio_unitario) VALUES
  (1, 1, 2, 20.0),
  (1, 2, 1, 25.0),
  (2, 3, 2, 18.0);
```

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;
```

123 producto_id ▼	A-Z nombre ▼	123 precio ▼	123 stock ▼
2	Pingüinos	25	20
3	Doritos Dinamita	18	40
4	Sabritas Naturale	18	20
5	Takis	20	20



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

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

```
SELECT clientes.nombre, pedidos.pedido_id, pedidos.fecha_pedido, pedidos.total
FROM pedidos
JOIN clientes ON pedidos.cliente_id = clientes.cliente_id
WHERE clientes.nombre = 'Francisco';
```

A-Z nombre	123 pedido_id	🕒 fecha_pedido	123 total
Francisco	1	2024-10-16 19:28:51	65

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.

```
-- aqui especifico de que tabla voy a sacar los datos, ya que se usan varias
SELECT productos.nombre, SUM(detalle_pedidos.cantidad) AS total_vendido
FROM detalle_pedidos
JOIN productos ON detalle_pedidos.producto_id = productos.producto_id
GROUP BY productos.producto_id;
```

A-Z nombre	123 total_vendido
Gansito	2
Pingüinos	1
Doritos Dinamita	2

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.

```
-- use el 1.15 ya que asi me devuelve el valor y no el porcentaje a sumar  
UPDATE productos SET precio = precio * 1.15 WHERE categoria = 'Panecillo';
```

A-Z nombre ▼	A-Z categoria ▼	123 precio ▼
Gansito	Panecillo	23
Pingüinos	Panecillo	28.75

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

Instrucción: Asegúrate de que el nuevo correo sea único.

```
-- este correo no existe  
UPDATE clientes SET correo = 'nuevo_correo@gmail.com' WHERE cliente_id = 1;
```

A-Z nombre ▼	A-Z correo ▼
Francisco	nuevo_correo@gmail.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** .

```
UPDATE productos SET stock=30 WHERE producto_id=1;
```

A-Z nombre ▼	123 stock ▼
Gansito	30

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`.

```
-- ya que la entidad de detalles es debil por que depende de
-- la entidad productos esta la debemos eliminar primero, si no, no se podria
DELETE FROM detalle_pedidos WHERE pedido_id = 1;

-- ahora si seguimos con la de productos
DELETE FROM pedidos WHERE pedido_id = 1;
```

pedido_id	cliente_id	fecha_pedido	total	detalle_id	pedido_id	producto_id	cantidad	precio_unitario
2	2	2024-10-16 19:28:51	36	3	2	3	2	18

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`.

```
-- Primero eliminamos los detalles de los pedidos del cliente por la entidad debil
DELETE FROM detalle_pedidos WHERE pedido_id IN (SELECT pedido_id FROM pedidos WHERE cliente_id = 1);

-- Luego eliminamos los pedidos del cliente, esta es entidad fuerte de detalles pedidos
-- pero entidad debil de cliente
DELETE FROM pedidos WHERE cliente_id = 1;

-- ahora si eliminamos al cliente
DELETE FROM clientes WHERE cliente_id = 1;
```

cliente_id	nombre	correo	fecha_registro
2	Jose	ejemplo2@gmail.com	2024-10-16
3	Antonio	ejemplo3@gmail.com	2024-10-16



V. Conclusiones:

En esta practica mejore con la practica mi habilidad para el manejo de las bases de datos y algunos conceptos que aun no entendía para que servían tanto con esto me quedan algo mas claros y pues la verdad me agrada ver como la información se mantiene y que no se le puede eliminar sin antes poner las entidades débiles ya que si nos vamos a el modelado asi lo indica desde ahi por ser entidad débil y fuerte o alfenos asi lo he envidio yo y pues me agrado bastante esta practica.