

MÓDULO 1(anexos):

Ejercicio SQL

Desarrolla una base de datos para una tienda de venta de fruta. necesitamos tener los productos
(nombre cantidad, precio). Clientes (nombre, apellidos, direccion facturación), compras
(relacion entre compradores y los productos almacenados en un histórico), proveedores
(relacion productos) ... agrega lo que consideres conveniente

Para desarrollar una base de datos para una tienda de venta de frutas, podemos utilizar un modelo relacional que permita gestionar eficientemente los productos, clientes, compras e inventario. A continuación, te presento un diseño básico de la base de datos, incluyendo las tablas necesarias, sus atributos y las relaciones entre ellas.

1. Entidades y Atributos

a. Productos

- **producto_id** (INT, PK): Identificador único del producto.
- **nombre** (VARCHAR): Nombre de la fruta.
- **cantidad** (INT): Cantidad disponible en el inventario.
- **precio** (DECIMAL): Precio por unidad de la fruta.
- **unidad_medida** (VARCHAR): Unidad de medida (por ejemplo, kg, unidad).

b. Clientes

- **cliente_id** (INT, PK): Identificador único del cliente.
- **nombre** (VARCHAR): Nombre del cliente.
- **apellidos** (VARCHAR): Apellidos del cliente.
- **direccion_facturacion** (VARCHAR): Dirección de facturación del cliente.
- **email** (VARCHAR): Correo electrónico del cliente.
- **telefono** (VARCHAR): Número de teléfono del cliente.

c. Proveedores

- **proveedor_id** (INT, PK): Identificador único del proveedor.
- **nombre** (VARCHAR): Nombre del proveedor.
- **direccion** (VARCHAR): Dirección del proveedor.
- **contacto** (VARCHAR): Persona de contacto.
- **telefono** (VARCHAR): Número de teléfono del proveedor.
- **email** (VARCHAR): Correo electrónico del proveedor.

d. Compras

- **compra_id** (INT, PK): Identificador único de la compra.
- **cliente_id** (INT, FK): Referencia al cliente que realizó la compra.
- **fecha_compra** (DATE): Fecha en la que se realizó la compra.
- **total** (DECIMAL): Total de la compra.

e. Detalle_Compra

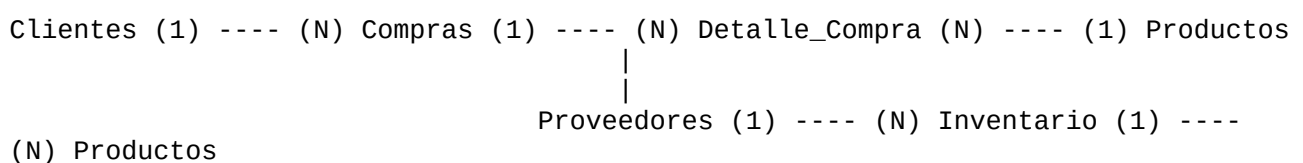
- **detalle_compra_id** (INT, PK): Identificador único del detalle de la compra.
- **compra_id** (INT, FK): Referencia a la compra.
- **producto_id** (INT, FK): Referencia al producto comprado.
- **cantidad** (INT): Cantidad del producto comprado.
- **precio_unitario** (DECIMAL): Precio unitario del producto en el momento de la compra.

f. Inventario

- **inventario_id** (INT, PK): Identificador único del registro de inventario.
- **producto_id** (INT, FK): Referencia al producto.
- **fecha_ingreso** (DATE): Fecha en la que el producto ingresó al inventario.
- **cantidad_ingresada** (INT): Cantidad de producto ingresada.
- **proveedor_id** (INT, FK): Referencia al proveedor del producto.

2. Relaciones

- **Clientes y Compras**: Un cliente puede realizar muchas compras (1).
- **Compras y Detalle_Compra**: Una compra puede tener muchos detalles de compra (1).
- **Productos y Detalle_Compra**: Un producto puede aparecer en muchos detalles de compra (1).
- **Proveedores y Inventario**: Un proveedor puede suministrar muchos productos (1).
- **Productos y Inventario**: Un producto puede tener muchos registros de inventario (1).

3. Diagrama de Relaciones

4. Ejemplo de Creación de Tablas

Aquí tienes un ejemplo de cómo podrías crear estas tablas en SQL:

```
CREATE TABLE Clientes (  
    cliente_id INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(50) NOT NULL,  
    apellidos VARCHAR(50) NOT NULL,  
    direccion_facturacion VARCHAR(100),  
    email VARCHAR(50),  
    telefono VARCHAR(20)  
);  
  
CREATE TABLE Proveedores (  
    proveedor_id INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(50) NOT NULL,  
    direccion VARCHAR(100),  
    contacto VARCHAR(50),  
    telefono VARCHAR(20),  
    email VARCHAR(50)  
);  
  
CREATE TABLE Productos (  
    producto_id INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(50) NOT NULL,  
    cantidad INT NOT NULL,  
    precio DECIMAL(10,2) NOT NULL,  
    unidad_medida VARCHAR(10)  
);  
  
CREATE TABLE Compras (  
    compra_id INT PRIMARY KEY AUTO_INCREMENT,  
    cliente_id INT NOT NULL,  
    fecha_compra DATE NOT NULL,  
    total DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (cliente_id) REFERENCES Clientes(cliente_id)  
);  
  
CREATE TABLE Detalle_Compra (  
    detalle_compra_id INT PRIMARY KEY AUTO_INCREMENT,  
    compra_id INT NOT NULL,  
    producto_id INT NOT NULL,  
    cantidad INT NOT NULL,  
    precio_unitario DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (compra_id) REFERENCES Compras(compra_id),  
    FOREIGN KEY (producto_id) REFERENCES Productos(producto_id)  
);  
  
CREATE TABLE Inventario (  
    inventario_id INT PRIMARY KEY AUTO_INCREMENT,  
    producto_id INT NOT NULL,  
    fecha_ingreso DATE NOT NULL,  
    cantidad_ingresada INT NOT NULL,  
    proveedor_id INT NOT NULL,  
    FOREIGN KEY (producto_id) REFERENCES Productos(producto_id),  
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(proveedor_id)  
);
```

5. Consideraciones Adicionales

- **Transacciones:** Asegúrate de manejar transacciones para mantener la integridad de los datos, especialmente en operaciones de compra y actualización de inventario.
- **Índices:** Considera la creación de índices en columnas clave para mejorar el rendimiento de las consultas.
- **Seguridad:** Implementa medidas de seguridad para proteger la información sensible de los clientes y proveedores.
- **Normalización:** Asegúrate de que la base de datos esté bien normalizada para evitar redundancias y anomalías en los datos.

Este diseño proporciona una base sólida para gestionar una tienda de venta de frutas. Puedes expandirlo según las necesidades específicas de tu negocio, como la gestión de descuentos, promociones, o la integración con sistemas de punto de venta.

Crea el fichero sql para agregarlo a Mariadb. No olvides CREATE database. Además incorpora Varios INSERT con datos de ejemplo en cada tabla. Define diferentes SELECT contra una tabla y contra varias relacionadas. Realiza un DELETE de registros en cascada

Para crear un fichero SQL que pueda ser ejecutado en MariaDB, primero debemos definir la estructura de la base de datos, incluyendo la creación de la base de datos, las tablas con sus relaciones, y luego insertar datos de ejemplo. También incluiremos consultas SELECT para recuperar datos de una sola tabla y de varias tablas relacionadas, así como un ejemplo de un DELETE en cascada.

A continuación, te proporciono un script SQL completo:

sql
Word Wrap
Copy

```
-- 1. Crear la base de datos
CREATE DATABASE IF NOT EXISTS TiendaFrutas;
USE TiendaFrutas;

-- 2. Crear tablas

CREATE TABLE Clientes (
    cliente_id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    direccion_facturacion VARCHAR(100),
    email VARCHAR(50),
    telefono VARCHAR(20)
);

CREATE TABLE Proveedores (
    proveedor_id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
```

```
    direccion VARCHAR(100),
    contacto VARCHAR(50),
    telefono VARCHAR(20),
    email VARCHAR(50)
);

CREATE TABLE Productos (
    producto_id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    cantidad INT NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    unidad_medida VARCHAR(10)
);

CREATE TABLE Compras (
    compra_id INT PRIMARY KEY AUTO_INCREMENT,
    cliente_id INT NOT NULL,
    fecha_compra DATE NOT NULL,
    total DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (cliente_id) REFERENCES Clientes(cliente_id)
    ON DELETE CASCADE
);

CREATE TABLE Detalle_Compra (
    detalle_compra_id INT PRIMARY KEY AUTO_INCREMENT,
    compra_id INT NOT NULL,
    producto_id INT NOT NULL,
    cantidad INT NOT NULL,
    precio_unitario DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (compra_id) REFERENCES Compras(compra_id)
    ON DELETE CASCADE,
    FOREIGN KEY (producto_id) REFERENCES Productos(producto_id)
    ON DELETE CASCADE
);

CREATE TABLE Inventario (
    inventario_id INT PRIMARY KEY AUTO_INCREMENT,
    producto_id INT NOT NULL,
    fecha_ingreso DATE NOT NULL,
    cantidad_ingresada INT NOT NULL,
    proveedor_id INT NOT NULL,
    FOREIGN KEY (producto_id) REFERENCES Productos(producto_id)
    ON DELETE CASCADE,
    FOREIGN KEY (proveedor_id) REFERENCES Proveedores(proveedor_id)
    ON DELETE CASCADE
);

-- 3. Insertar datos de ejemplo

-- Insertar clientes
INSERT INTO Clientes (nombre, apellidos, direccion_facturacion, email, telefono)
VALUES
('Juan', 'Pérez', 'Calle Falsa 123', 'juan.perez@example.com', '123456789'),
('María', 'Gómez', 'Avenida Siempreviva 742', 'maria.gomez@example.com',
'987654321');

-- Insertar proveedores
INSERT INTO Proveedores (nombre, direccion, contacto, telefono, email) VALUES
```

```
('Frutícola del Sur', 'Calle del Sur 45', 'José López', '555555555',
'contacto@fruticolasur.com'),
('Verduras y Frutas SA', 'Avenida Norte 32', 'Ana Martínez', '444444444',
'contacto@vyfsa.com');

-- Insertar productos
INSERT INTO Productos (nombre, cantidad, precio, unidad_medida) VALUES
('Manzana', 150, 1.20, 'kg'),
('Plátano', 200, 0.80, 'kg'),
('Naranja', 180, 1.00, 'kg');

-- Insertar compras
INSERT INTO Compras (cliente_id, fecha_compra, total) VALUES
(1, '2023-10-01', 12.00),
(2, '2023-10-02', 8.00);

-- Insertar detalle de compras
INSERT INTO Detalle_Compra (compra_id, producto_id, cantidad, precio_unitario)
VALUES
(1, 1, 10, 1.20),
(1, 2, 5, 0.80),
(2, 3, 8, 1.00);

-- Insertar inventario
INSERT INTO Inventario (producto_id, fecha_ingreso, cantidad_ingresada,
proveedor_id) VALUES
(1, '2023-09-25', 150, 1),
(2, '2023-09-26', 200, 2),
(3, '2023-09-27', 180, 1);

-- 4. Consultas SELECT

-- Seleccionar todos los productos
SELECT * FROM Productos;

-- Seleccionar todas las compras con detalles
SELECT c.compra_id, c.fecha_compra, c.total, dc.producto_id, dc.cantidad,
dc.precio_unitario
FROM Compras c
JOIN Detalle_Compra dc ON c.compra_id = dc.compra_id;

-- Seleccionar todas las compras de un cliente específico
SELECT c.compra_id, c.fecha_compra, c.total
FROM Compras c
WHERE c.cliente_id = 1;

-- 5. DELETE en cascada

-- Eliminar un cliente y todas sus compras y detalles de compra
DELETE FROM Clientes WHERE cliente_id = 1;

-- Al ejecutar el DELETE anterior, las compras y detalles de compra asociados al
cliente_id 1 también se eliminarán debido a la cláusula ON DELETE CASCADE en las
relaciones de las tablas.
```

Explicación del Script

1.**Creación de la Base de Datos:** Se crea una base de datos llamada **TiendaFrutas** y se selecciona para su uso.

2.**Creación de Tablas:** Se definen las tablas **Clientes**, **Proveedores**, **Productos**, **Compras**, **Detalle_Compra** e **Inventario** con sus respectivos atributos y claves foráneas. Las relaciones se establecen con **FOREIGN KEY** y la cláusula **ON DELETE CASCADE** para permitir la eliminación en cascada de registros relacionados.

3.**Inserción de Datos de Ejemplo:** Se insertan datos de ejemplo en cada tabla para simular un entorno de trabajo.

4.

Consultas SELECT:

- Se seleccionan todos los productos.
- Se seleccionan todas las compras con sus detalles.
- Se seleccionan las compras de un cliente específico.

5.**DELETE en Cascada:** Se elimina un cliente y, debido a la cláusula **ON DELETE CASCADE**, todas las compras y detalles de compra asociados al cliente también se eliminan.

Este script proporciona una base sólida para gestionar una tienda de venta de frutas, permitiendo la inserción, consulta y eliminación de datos de manera eficiente y segura.