Desafío – Acceso a datos normalizados SQL con Node y Express

Alumno: Nicolás Chávez Garzón
Grupo: 1

Contexto

Supongamos que tenemos una tabla llamada "Ventas" en una base de datos de una tienda de electrónicos con la siguiente estructura:

```javascript
CREATE TABLE Ventas (
    ID_Venta INT,
    Fecha DATE,
    Cliente VARCHAR(100),
    Telefono VARCHAR(20),
    Producto VARCHAR(100),
    Categoria VARCHAR(50),
    Precio DECIMAL(10,2),
    Cantidad INT,
    Total DECIMAL(10,2)
);
```

Esta tabla contiene información sobre ventas, clientes, productos y sus categorías. La tabla cumple con la 1FN, ya que:

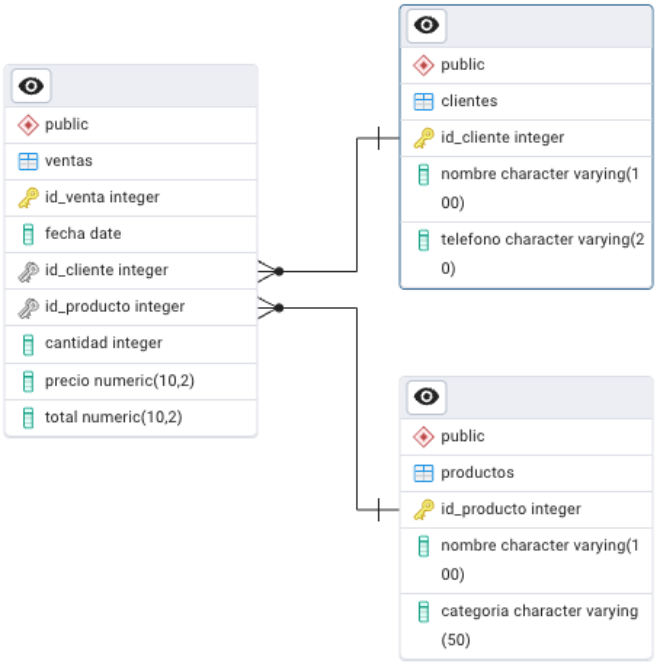
a. Cada columna contiene valores atómicos (no hay valores múltiples en una sola celda).
b. Cada columna tiene un nombre único.
c. Todos los valores en una columna son del mismo tipo de datos.
d. No hay grupos repetitivos.

Requerimientos

1. La tabla no cumple con la 2FN porque hay dependencias parciales. El Telefono depende solo del Cliente, y la Categoria depende solo del Producto. Para cumplir con la 2FN, debemos separar estas entidades en tablas propias. **Aplicar la 2FN, identificar y separar aquellas entidades que podrían ser propias.**

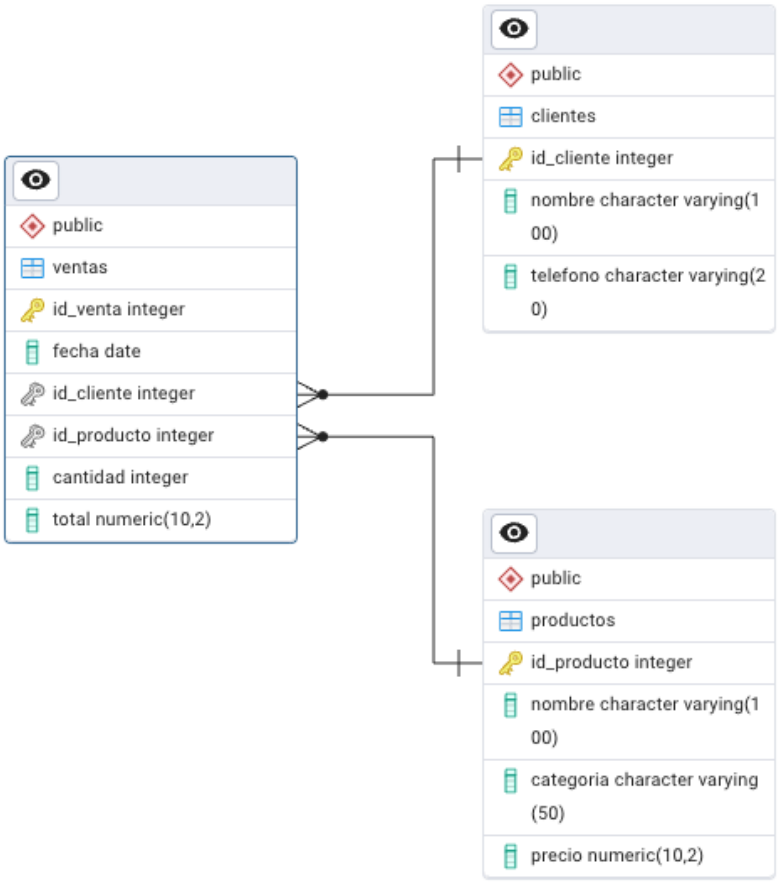
   Para normalizar y aplicar la 2FN, debemos crear 2 nuevas tablas llamadas Clientes y Productos:

- **Clientes**: Contiene las variables columnas ID_Cliente, Nombre y Telefono.
- **Productos**: Contiene las variables columnas ID_Producto, Nombre y Categoria.



2. Después de aplicar la 2FN, aún no se cumple la 3FN porque hay dependencias transitivas. El Precio depende del Producto, no de la clave primaria de la tabla Ventas. **Aplicar la 3FN en la tabla ventas para que cada atributo no clave dependa directamente de la clave primaria.**

En este caso, Precio depende solo de Producto, por lo que se mueve hacia dicha tabla. Además, la columna Total también se podría eliminar porque podría calcularse directamente producto de una query, pero se asume que hacerlo podría ser costoso computacionalmente y por lo tanto, se prefiere dejar como columna.



3. **Realiza la solución de este ejercicio escribiendo código SQL donde se declaren los atributos y tipo de datos para cada atributo.**

   Finalmente, la query para crear las tablas Clientes, Productos y Ventas que cumplen con 2FN y 3FN es la siguiente:

```sql
CREATE TABLE Clientes (
    ID_Cliente INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Telefono VARCHAR(20)
);

CREATE TABLE Productos (
    ID_Producto INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Categoria VARCHAR(50),
    Precio DECIMAL(10,2)
);

CREATE TABLE Ventas (
    ID_Venta INT PRIMARY KEY,
    Fecha DATE,
    ID_Cliente INT,
    ID_Producto INT,
    Cantidad INT,
    Total DECIMAL(10,2),
    FOREIGN KEY (ID_Cliente) REFERENCES Clientes(ID_Cliente),
    FOREIGN KEY (ID_Producto) REFERENCES Productos(ID_Producto)
);
```