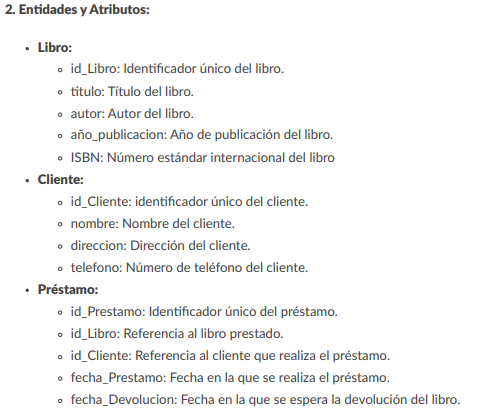
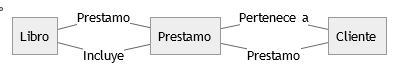
Modificación del esquema de la Base de Datos

Realiza los siguientes ejercicios en MySQL

Diseñar y modificar un esquema para una base de datos de biblioteca.

1. **Objetivo:** Diseñar en un esquema de la base de datos para gestionar la información de una biblioteca. El sistema debe ser capaz de almacenar datos sobre libros, los clientes y los préstamos realizado





**TABLAS INICIALES**

CREATE TABLE Libro (

ID\_Libro INT PRIMARY KEY,

Titulo VARCHAR(255), Autor VARCHAR(255),

Año\_Publicacion INT,

ISBN VARCHAR(20) );

CREATE TABLE Cliente (

ID\_Cliente INT PRIMARY KEY,

Nombre VARCHAR(255),

Direccion VARCHAR(255),

Telefono VARCHAR(20) );

CREATE TABLE Prestamo (

ID\_Prestamo INT PRIMARY KEY,

ID\_Libro INT,

ID\_Cliente INT,

Fecha\_Prestamo DATE,

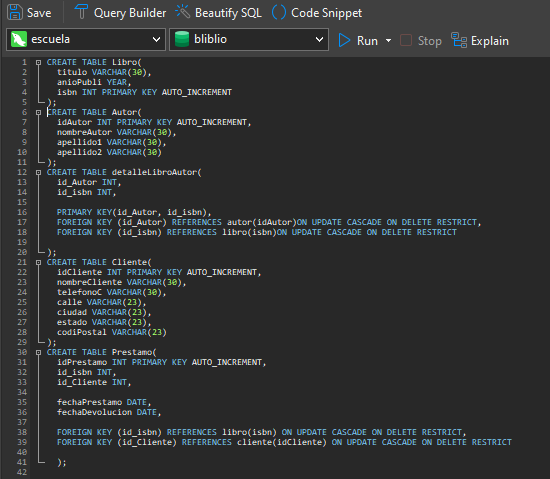
Fecha\_Devolucion DATE,

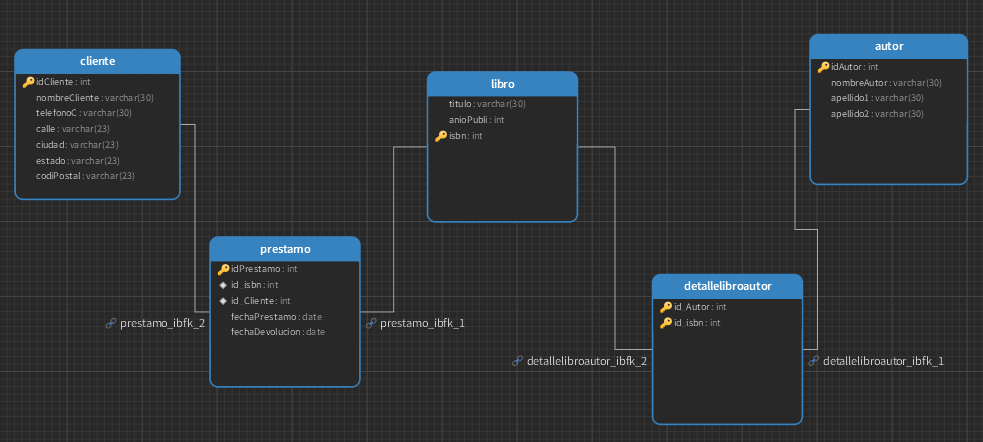
FOREIGN KEY (ID\_Libro) REFERENCES Libro(ID\_Libro),

FOREIGN KEY (ID\_Cliente) REFERENCES Cliente(ID\_Cliente) );

1. **Normalización y Modificación**

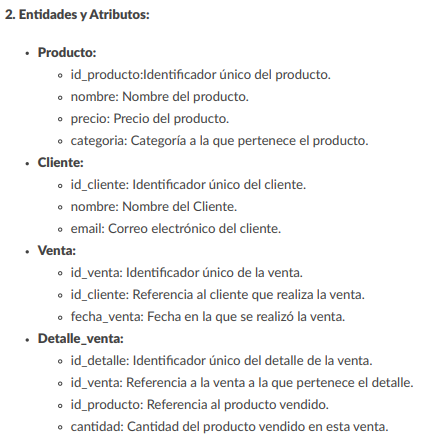
* Modifica el esquema para cumplir con las 3 formas normales.
* Descompón la tabla Libro si el ISBN puede repetirse o si los autores deben estar en una tabla separada.
* Normaliza la tabla Cliente si la dirección se repite o tiene múltiples componentes.
* SOLUCIÓN:

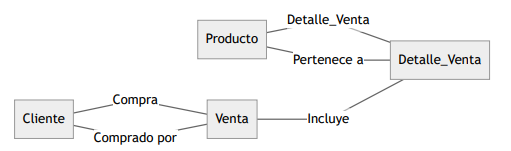




# Ejercicio 2:

**Objetivo:** Diseñar en un esquema de la base de datos para gestionar la información de una biblioteca. El sistema debe ser capaz de almacenar datos sobre libros, los clientes y los préstamos realizado.





**TABLAS INICIALES**

CREATE TABLE Producto (

ID\_Producto INT PRIMARY KEY,

Nombre VARCHAR(255),

Precio DECIMAL(10, 2), Categoria VARCHAR(100)

);

CREATE TABLE Cliente (

ID\_Cliente INT PRIMARY KEY,

Nombre VARCHAR(255),

Email VARCHAR(255) );

CREATE TABLE Venta (

ID\_Venta INT PRIMARY KEY,

ID\_Cliente INT,

Fecha\_Venta DATE,

FOREIGN KEY (ID\_Cliente) REFERENCES Cliente(ID\_Cliente) );

CREATE TABLE Detalle\_Venta (

ID\_Detalle INT PRIMARY KEY,

ID\_Venta INT,

ID\_Producto INT,

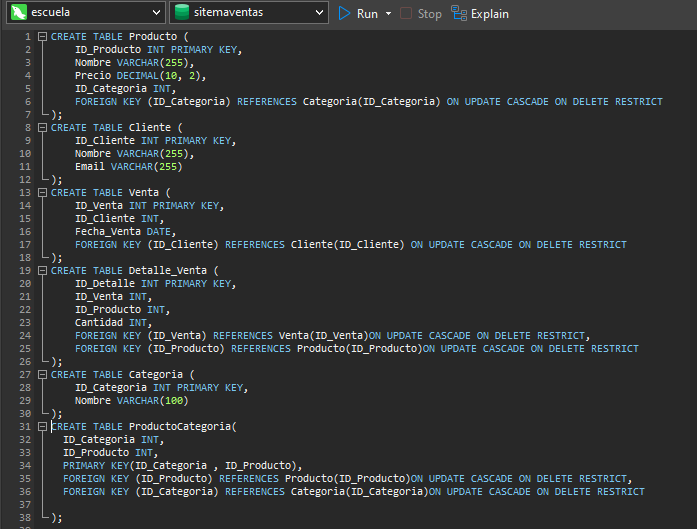
Cantidad INT,

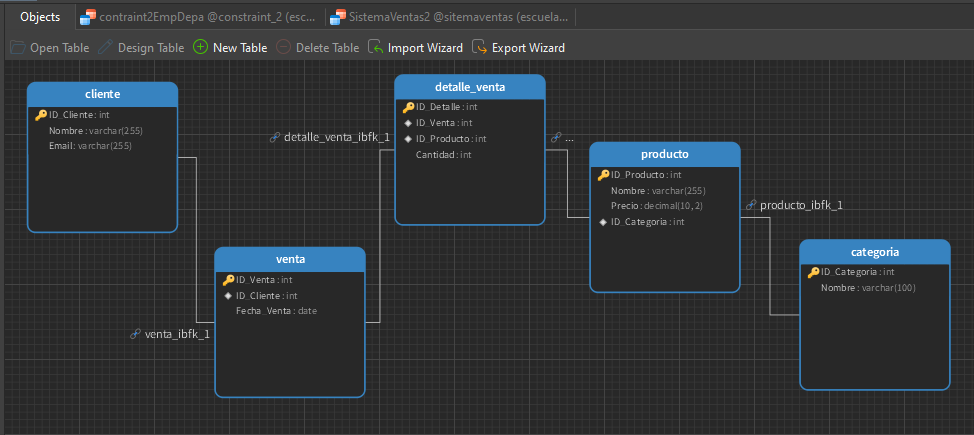
FOREIGN KEY (ID\_Venta) REFERENCES Venta(ID\_Venta),

FOREIGN KEY (ID\_Producto) REFERENCES Producto(ID\_Producto) );

1. **Normalización y Modificación:**

* Normaliza las tablas si hay redundancia en la Venta o en Detalle\_Venta
* Asegúrate de que la tabla Producto no contenga valores redundantes para la categoría.

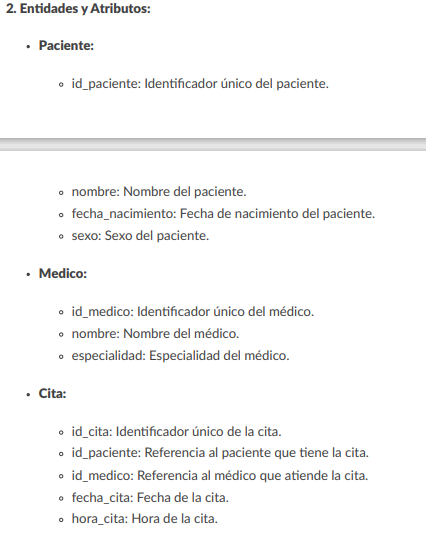




# Ejercicio 3:

**Objetivo:**

Crear y modificar un esquema de base de datos para un sistema de hospital. El sistema debe gestionar la información de pacientes, médicos y citas.



PACIENTE

REALIZA

CITA

ATIENDE

MEDICO

**TABLAS INICIALES**

CREATE TABLE Paciente (

ID\_Paciente INT PRIMARY KEY,

Nombre VARCHAR(255),

Fecha\_Nacimiento DATE,

Sexo CHAR(1) );

CREATE TABLE Medico (

ID\_Medico INT PRIMARY KEY,

Nombre VARCHAR(255),

Especialidad VARCHAR(100) );

CREATE TABLE Cita (

ID\_Cita INT PRIMARY KEY,

ID\_Paciente INT,

ID\_Medico INT,

Fecha\_Cita DATE,

Hora\_Cita TIME,

FOREIGN KEY (ID\_Paciente) REFERENCES Paciente(ID\_Paciente),

FOREIGN KEY (ID\_Medico) REFERENCES Medico(ID\_Medico) );

1. **Normalización y Modificación:**

* Revisa si hay redundancia en la tabla Cita o en la información del Médico.
* Considera agregar una tabla de Especialidad si hay múltiples médicos con la misma especialidad
* SOLUCION:

