

Nombre de la práctica	Ejercicios de normalización			No.	2
Asignatura:	Taller De Bases De Datos	Carrera:	Ingeniería sistemas	en	Duración de la práctica (Hrs)

Nombre del alumno: Jesús Navarrete Martínez

Grupo: 3501

I. Competencia(s) específica(s):

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

III. Material empleado:

- Equipo de computo

IV. Desarrollo de la práctica:

Ejercicio 1:

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

- 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 realizados.

Primera Forma Normal

Tabla_Libro						
ISBN	titulo	nombre_autor	apellido1_autor	apellido2_autor	fecha_publicacion	editorial

Tabla_Cliente									
id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	calle	localidad	municipio	codigo_postal	estado	pais

Tabla_Presatmo			
id_prestamo	id_cliente	fecha_prestamo	fecha_devolucion



Segunda Forma Normal

Tabla_Libro			
ISBN	titulo	año_publicacion	id_editorial

Tabla_Cliente										
id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	calle	localidad	municipio	codigo_postal	estado	pais	telefono

Tabla_Prestamo			
id_prestamo	id_cliente	fecha_prestamo	fecha_devolucion

Tabla_Autor			
id_autor	nombre_a	apellido1_a	apellido2_a

Tabla_Libro_Autor	
ISBN	id_autor

Tabla_Libro_Prestamo	
ISBN	id_prestamo

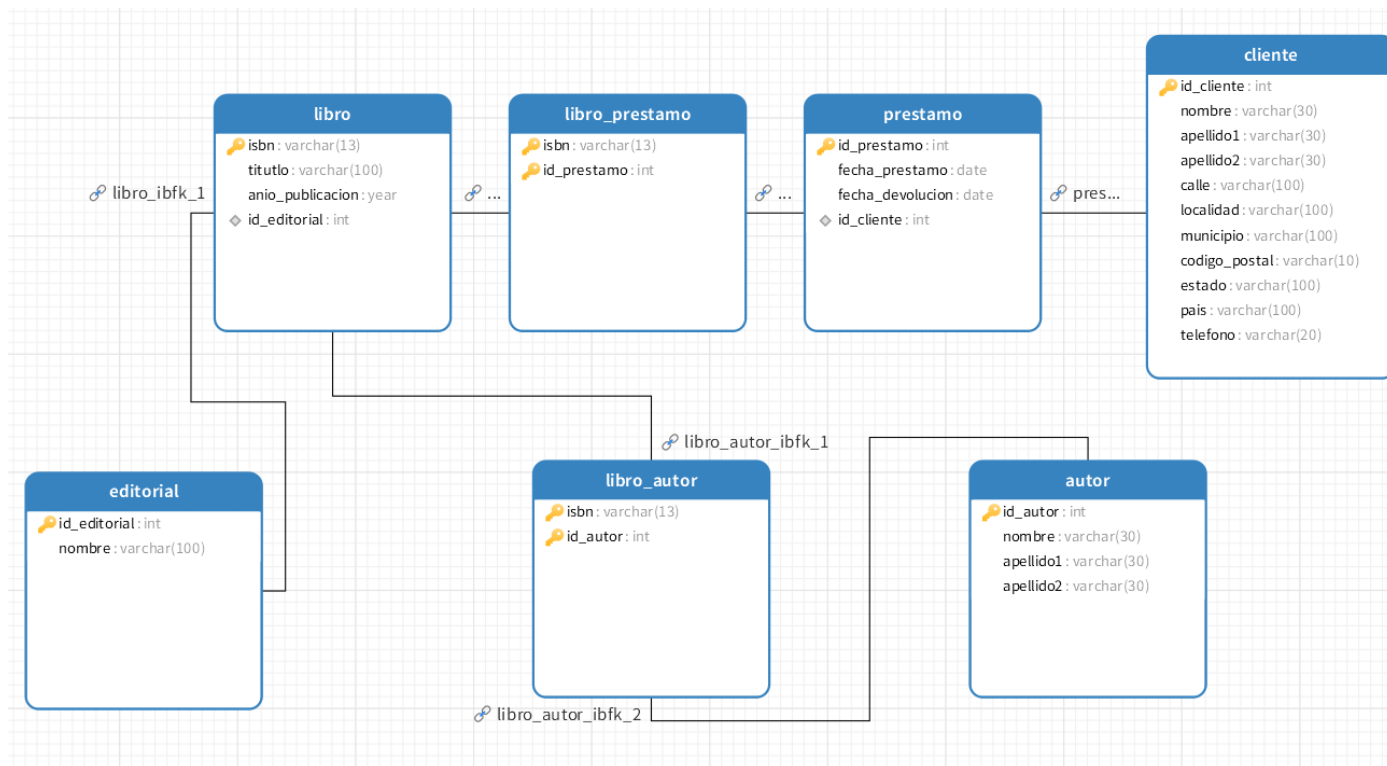


Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
1 CREATE TABLE libro (  
2 isbn VARCHAR(13) PRIMARY KEY NOT NULL CHECK (isbn>0),  
3 titutlo VARCHAR(100) NOT NULL,  
4 anio_publicacion YEAR,  
5 id_editorial INT,  
6 FOREIGN KEY ( id_editorial) REFERENCES editorial(id_editorial) ON UPDATE CASCADE ON DELETE RESTRICT  
7 );  
8  
9  
10 CREATE TABLE editorial (  
11 id_editorial INT PRIMARY KEY AUTO_INCREMENT,  
12 nombre VARCHAR(100)  
13 );  
14  
15 CREATE TABLE cliente (  
16 id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
17 nombre VARCHAR(30) NOT NULL,  
18 apellido1 VARCHAR (30) NOT NULL,  
19 apellido2 VARCHAR (30) NOT NULL,  
20 calle VARCHAR(100) NOT NULL,  
21 localidad VARCHAR(100) NOT NULL,  
22 municipio VARCHAR(100) NOT NULL,  
23 codigo_postal VARCHAR (10) NOT NULL,  
24 estado VARCHAR(100) NOT NULL,  
25 pais VARCHAR(100) NOT NULL,  
26 telefono VARCHAR(20) NOT NULL  
27 );  
28  
29 CREATE TABLE prestamo (  
30 id_prestamo INT PRIMARY KEY AUTO_INCREMENT,  
31 fecha_prestamo DATE NOT NULL,  
32 fecha_devolucion DATE NOT NULL,  
33 id_cliente INT,  
34 FOREIGN KEY ( id_cliente) REFERENCES cliente (id_cliente) ON UPDATE CASCADE ON DELETE RESTRICT  
35 );  
36  
37 CREATE TABLE autor (  
38 id_autor INT PRIMARY KEY AUTO_INCREMENT,  
39 nombre VARCHAR(30) NOT NULL,  
40 apellido1 VARCHAR (30) NOT NULL,  
41 apellido2 VARCHAR (30) NOT NULL  
42 );  
43  
44 CREATE TABLE libro_autor (  
45 isbn VARCHAR (13),  
46 id_autor INT,  
47 PRIMARY KEY (isbn, id_autor),  
48 FOREIGN KEY ( isbn) REFERENCES libro(isbn) ON UPDATE CASCADE ON DELETE RESTRICT,  
49 FOREIGN KEY ( id_autor) REFERENCES autor(id_autor) ON UPDATE CASCADE ON DELETE RESTRICT  
50 );  
51
```



Tablas Resultantes:



Ejercicio 2:

Diseñar y modificar un esquema para un sistema de ventas.

- **Objetivo:** Crear y modificar un esquema de base de datos para un sistema de ventas. El sistema debe gestionar la información de productos, clientes, ventas y los detalles de cada venta.

Primera Forma Normal

Tabla Producto			
id_producto	nombre	precio	categoria

Tabla_Cliente											
id_cliente	nombre_C	Apellido1_C	Apellido2_C	Email	calle	localidad	municipio	codigo_postal	estado	pais	telefono

Tabla_Venta		
id_venta	id_cliente	Fecha

Tabla_Detalle_Venta			
id_detalle	id_venta	id_producto	cantidad



Segunda Forma Normal

Tabla Producto			
id_producto	nombre	precio	id_categoria

Tabla_Cliente											
id_cliente	nombre_C	Apellido1_C	Apellido2_C	Email	calle	localidad	municipio	codigo_postal	estado	pais	telefono

Tabla_Venta		
id_venta	id_cliente	Fecha

Tabla_Detalle_Venta			
id_detalle	id_venta	id_producto	cantidad

Tabla_Categoria	
id_categoria	nombre

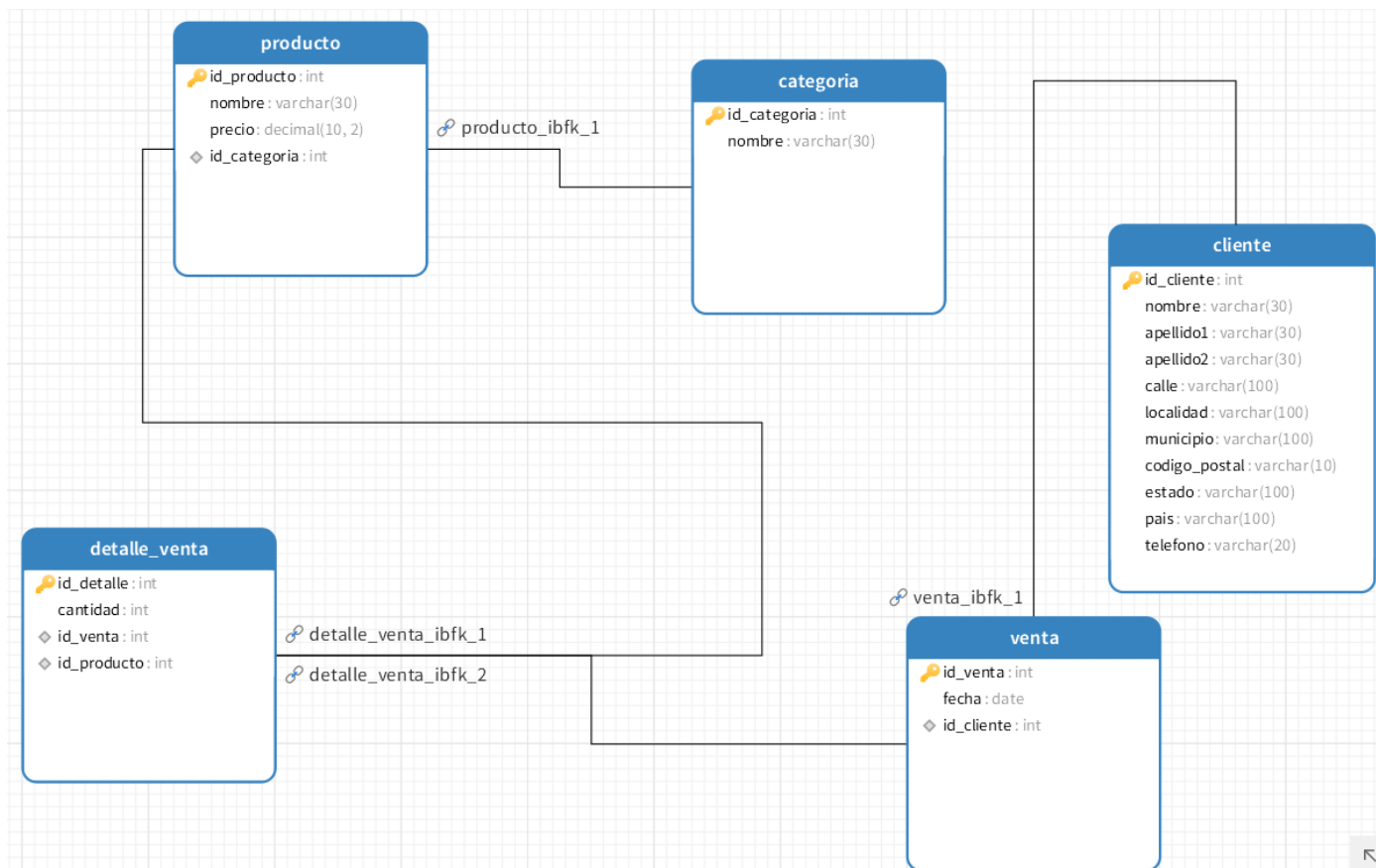


Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
1 CREATE TABLE producto (  
2   id_producto INT PRIMARY KEY AUTO_INCREMENT,  
3   nombre VARCHAR (30) NOT NULL,  
4   precio DECIMAL(10, 2) NOT NULL CHECK (precio>0),  
5   id_categoria INT,  
6   FOREIGN KEY ( id_categoria) REFERENCES categoria(id_categoria) ON UPDATE CASCADE ON DELETE RESTRICT  
7 );  
8  
9 CREATE TABLE categoria (  
10  id_categoria INT PRIMARY KEY AUTO_INCREMENT,  
11  nombre VARCHAR (30) NOT NULL  
12 );  
13  
14 CREATE TABLE cliente (  
15  id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
16  nombre VARCHAR(30) NOT NULL,  
17  apellido1 VARCHAR (30) NOT NULL,  
18  apellido2 VARCHAR (30) NOT NULL,  
19  calle VARCHAR(100) NOT NULL,  
20  localidad VARCHAR(100) NOT NULL,  
21  municipio VARCHAR(100) NOT NULL,  
22  codigo_postal VARCHAR (10) NOT NULL,  
23  estado VARCHAR(100) NOT NULL,  
24  pais VARCHAR(100) NOT NULL,  
25  telefono VARCHAR(20) NOT NULL  
26 );  
27  
28 CREATE TABLE venta(  
29  id_venta INT PRIMARY KEY AUTO_INCREMENT,  
30  fecha DATE NOT NULL,  
31  id_cliente INT,  
32  FOREIGN KEY ( id_cliente) REFERENCES cliente(id_cliente) ON UPDATE CASCADE ON DELETE RESTRICT  
33 );  
34  
35 CREATE TABLE detalle_venta (  
36  id_detalle INT PRIMARY KEY AUTO_INCREMENT,  
37  cantidad INT NOT NULL CHECK (cantidad>0),  
38  id_venta INT NOT NULL,  
39  id_producto INT NOT NULL,  
40  FOREIGN KEY (id_venta) REFERENCES venta(id_venta),  
41  FOREIGN KEY (id_producto) REFERENCES producto(id_producto)  
42 );
```



Tablas Resultantes:



Ejercicio 3:

Diseñar y modificar un esquema para un sistema hospitalario.

- **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.

Primera Forma Normal

Tabla_Paciente					
id_paciente	nombre_P	apellido1_P	apellido2_P	fecha_nacimiento	sexo

Tabla_cita				
id_cita	id_paciente	id_medico	fecha_cita	hora_cita

Tabla_Medico			
id_medico	nombre_m	apellido1_m	apellido2_m

Segunda Forma Normal

Tabla_Paciente					
id_paciente	nombre_P	apellido1_P	apellido2_P	fecha_nacimiento	sexo

Tabla_cita				
id_cita	id_paciente	id_medico	fecha_cita	hora_cita

Tabla_Medico			
id_medico	nombre_m	apellido1_m	apellido2_m

Tabla_Especialidad	
id_especialidad	nombre

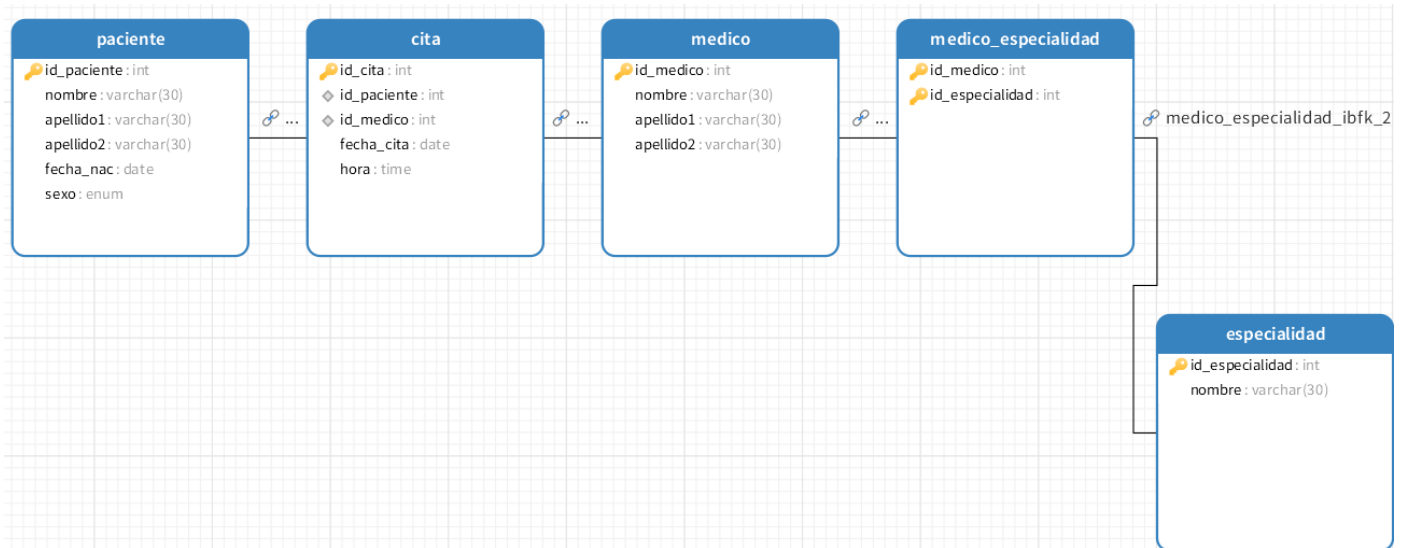
Tabla_Medico_Especialidad	
id_medico	id_especialidad



Creación de tablas y definición de atributos para la base de datos mediante lenguaje DDL.

```
1 CREATE TABLE paciente (  
2   id_paciente INT PRIMARY KEY AUTO_INCREMENT,  
3   nombre VARCHAR(30) NOT NULL,  
4   apellido1 VARCHAR (30) NOT NULL,  
5   apellido2 VARCHAR (30) NOT NULL,  
6   fecha_nac DATE NOT NULL,  
7   sexo ENUM('Masculino', 'Femenino', 'Otro', 'No especificado') NOT NULL  
8 );  
9  
10 CREATE TABLE cita (  
11   id_cita INT PRIMARY KEY AUTO_INCREMENT,  
12   id_paciente INT,  
13   id_medico INT,  
14   fecha_cita DATE NOT NULL,  
15   hora TIME NOT NULL,  
16   FOREIGN KEY (id_paciente) REFERENCES paciente(id_paciente),  
17   FOREIGN KEY (id_medico) REFERENCES medico(id_medico)  
18 );  
19  
20 CREATE TABLE medico (  
21   id_medico INT PRIMARY KEY AUTO_INCREMENT,  
22   nombre VARCHAR(30) NOT NULL,  
23   apellido1 VARCHAR (30) NOT NULL,  
24   apellido2 VARCHAR (30) NOT NULL  
25 );  
26  
27 CREATE TABLE especialidad (  
28   id_especialidad INT PRIMARY KEY AUTO_INCREMENT,  
29   nombre VARCHAR(30) NOT NULL  
30 );  
31  
32 CREATE TABLE medico_especialidad (  
33   id_medico INT,  
34   id_especialidad INT,  
35   PRIMARY KEY (id_medico, id_especialidad),  
36   FOREIGN KEY ( id_medico) REFERENCES medico(id_medico) ON UPDATE CASCADE ON DELETE RESTRICT,  
37   FOREIGN KEY ( id_especialidad) REFERENCES especialidad(id_especialidad) ON UPDATE CASCADE ON DELETE RESTRICT  
38 );
```

Tablas Resultantes:



V. Conclusiones:

En esta práctica, he comprendido la importancia de la **normalización de bases de datos** como un proceso fundamental para optimizar el diseño de tablas, minimizando la redundancia de datos y garantizando su integridad. Al aplicar las diferentes formas normales, logre dividir correctamente la información en entidades independientes, eliminando dependencias funcionales y asegurando un almacenamiento eficiente.

El proceso de creación de la base de datos en **MySQL** me permitió llevar a la práctica los principios de normalización, aplicando restricciones como claves primarias y foráneas para asegurar relaciones adecuadas entre tablas. Esto garantiza la consistencia de los datos y facilita su manipulación mediante consultas más sencillas y eficientes.