



Nombre de la práctica	Modificación del esquema de la base de datos			No.	1
Asignatura:	Taller de Base de datos	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: Vanesa Hernández Martínez

GRUPO: 3501

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

Actividades en aula de clases y en equipo personal

III. Material empleado:

- Laptop
- Navicat

Ejercicio 1

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 realizados.

2. Entidades y Atributos:

• Libro:

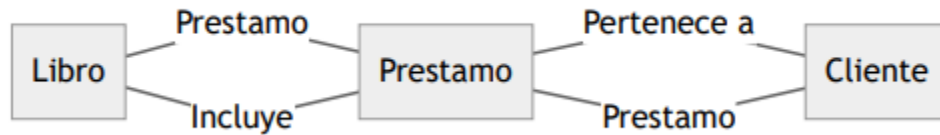
- id_Libro: Identificador único del libro.
- titulo: Título del libro.
- autor: Autor del libro.
- año_publicacion: Año de publicación del libro.
- ISBN: Número estándar internacional del libro

• Cliente:

- id_Cliente: identificador único del cliente.
- nombre: Nombre del cliente.
- direccion: Dirección del cliente.
- telefono: Número de teléfono del cliente.

• Préstamo:

- id_Prestamo: Identificador único del préstamo.
- id_Libro: Referencia al libro prestado.
- id_Cliente: Referencia al cliente que realiza el préstamo.
- fecha_Prestamo: Fecha en la que se realiza el préstamo.
- fecha_Devolucion: Fecha en la que se espera la devolución del libro.



3. 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.

Normalización

LIBRO			
ISBN	titulo	año_publicacion	id_editorial

PRESTAMO			
id_Prestamo	id_Cliente	fecha_Prestamo	fecha_Devolucion

EDITORIAL	
id_editorial	nombre

CLIENTE								
id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	calle	localidad	estado	codigo_postal	pais

Autor			
id_autor	nombre_autor	apellido1_autor	apellido2_autor

LIBRO AUTOR	
ISBN	id_autor

LIBRO PRESTAMO	
ISBN	id_prestamo

Código

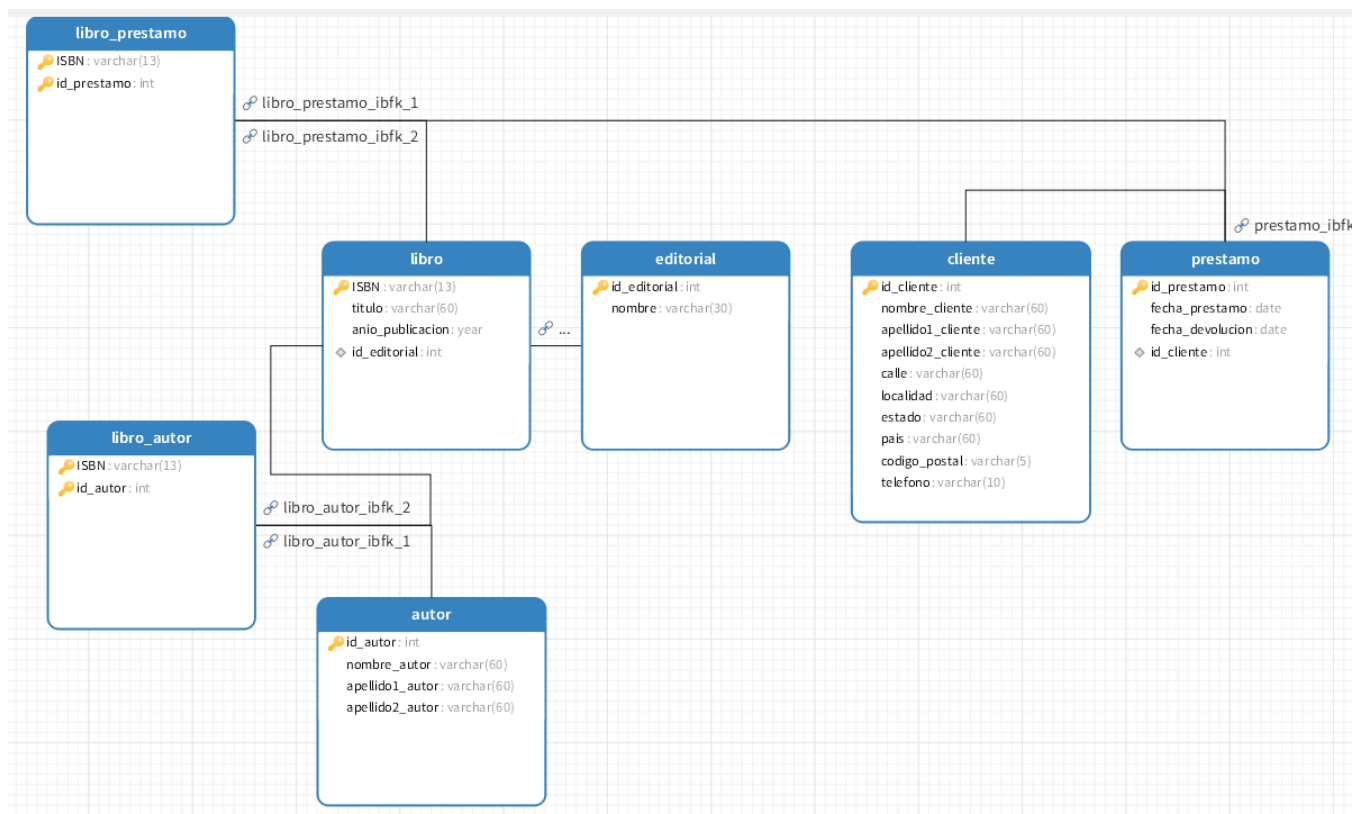
```

1 CREATE TABLE Libro(
2   ISBN VARCHAR(13) PRIMARY KEY NOT NULL CHECK(ISBN>0),
3   titulo VARCHAR(60) NOT NULL,
4   año_publicacion YEAR,
5   id_editorial INT,
6   FOREIGN KEY (id_editorial) REFERENCES editorial(id_editorial) ON UPDATE CASCADE ON DELETE RESTRICT
7 );
8
9 CREATE TABLE Editorial(
10  id_editorial 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_cliente VARCHAR(60) NOT NULL,
17     apellido1_cliente VARCHAR(60) NOT NULL,
18     apellido2_cliente VARCHAR(60) NOT NULL,
19     calle VARCHAR(60) NOT NULL,
20     localidad VARCHAR(60) NOT NULL,
21     estado VARCHAR(60) NOT NULL,
22     pais VARCHAR(60) NOT NULL,
23     codigo_postal VARCHAR(5) NOT NULL,
24     telefono VARCHAR(10) NOT NULL
25 );
26
27 CREATE TABLE Prestamo(
28     id_prestamo INT PRIMARY KEY AUTO_INCREMENT,
29     fecha_prestamo DATE NOT NULL,
30     fecha_devolucion 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 Autor(
36     id_autor INT PRIMARY KEY AUTO_INCREMENT,
37     nombre_autor VARCHAR(60) NOT NULL,
38     apellido1_autor VARCHAR(60) NOT NULL,
39     apellido2_autor VARCHAR(60) NOT NULL
40 );
41
42 CREATE TABLE Libro_Autor(
43     ISBN VARCHAR(13),
44     id_autor INT,
45     PRIMARY KEY (ISBN,id_autor),
46     FOREIGN KEY (ISBN) REFERENCES libro(ISBN) ON UPDATE CASCADE ON DELETE RESTRICT,
47     FOREIGN KEY (id_autor) REFERENCES autor(id_autor) ON UPDATE CASCADE ON DELETE RESTRICT
48 );
49
50 CREATE TABLE Libro_Prestamo(
51     ISBN VARCHAR(13),
52     id_prestamo INT,
53     PRIMARY KEY (ISBN,id_prestamo),
54     FOREIGN KEY (ISBN) REFERENCES libro(ISBN) ON UPDATE CASCADE ON DELETE RESTRICT,
55     FOREIGN KEY (id_prestamo) REFERENCES prestamo(id_prestamo) ON UPDATE CASCADE ON DELETE RESTRICT
56 );
57
```

Tablas



- **NOT NULL:** Obliga a que los campos tengan siempre un valor. Por ejemplo, ISBN y titulo en la tabla Libro no pueden estar vacíos.
- **CHECK:** Impone una condición sobre los datos. En el caso de CHECK(ISBN > 0), garantiza que el ISBN sea mayor a 0 (aunque este CHECK es redundante porque ISBN es un VARCHAR y no numérico).
- **YEAR:** Es un tipo de dato que almacena solo años (por ejemplo, anio_publicacion guarda el año en que se publicó el libro).
- **FOREIGN KEY, ON UPDATE CASCADE, ON DELETE RESTRICT:** Relacionan tablas. ON UPDATE CASCADE actualiza automáticamente los cambios en claves foráneas, y ON DELETE RESTRICT impide eliminar un registro si está siendo referenciado en otra tabla.

Ejercicio 2

Diseñar y modificar un esquema para un sistema de ventas.

1. 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.

2. Entidades y Atributos:

- **Producto:**

- id_producto: Identificador único del producto.
- nombre: Nombre del producto.
- precio: Precio del producto.
- categoria: Categoría a la que pertenece el producto.

- **Cliente:**

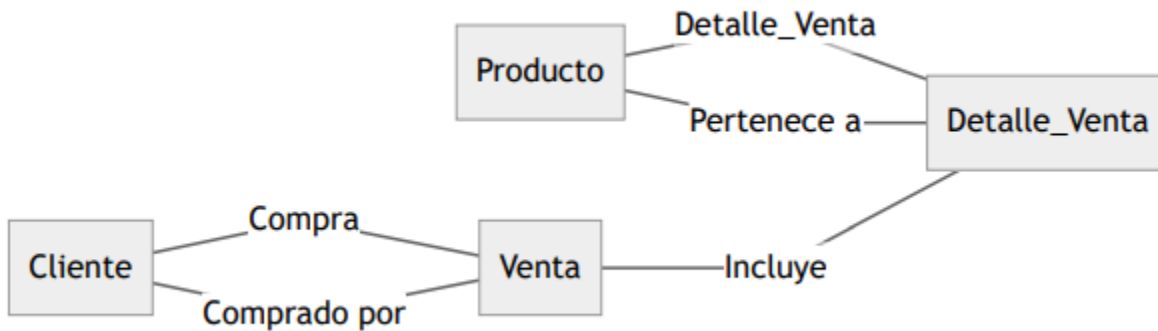
- id_cliente: Identificador único del cliente.
- nombre: Nombre del Cliente.
- email: Correo electrónico del cliente.

- **Venta:**

- id_venta: Identificador único de la venta.
- id_cliente: Referencia al cliente que realiza la venta.
- fecha_venta: Fecha en la que se realizó la venta.

- **Detalle_venta:**

- id_detalle: Identificador único del detalle de la venta.
- id_venta: Referencia a la venta a la que pertenece el detalle.
- id_producto: Referencia al producto vendido.
- cantidad: Cantidad del producto vendido en esta venta.



3. 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.

Normalización

PRODUCTO			
id_producto	nombre	precio	id_categoria

Cliente										
id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	Email	calle	localidad	estado	codigo_postal	pais	telefono

Venta		
id_venta	id_cliente	Fecha

DETALLE VENTA			
id_detalle	id_venta	id_producto	cantidad

CATEGORIA	
id_categoria	nombre

Código

```

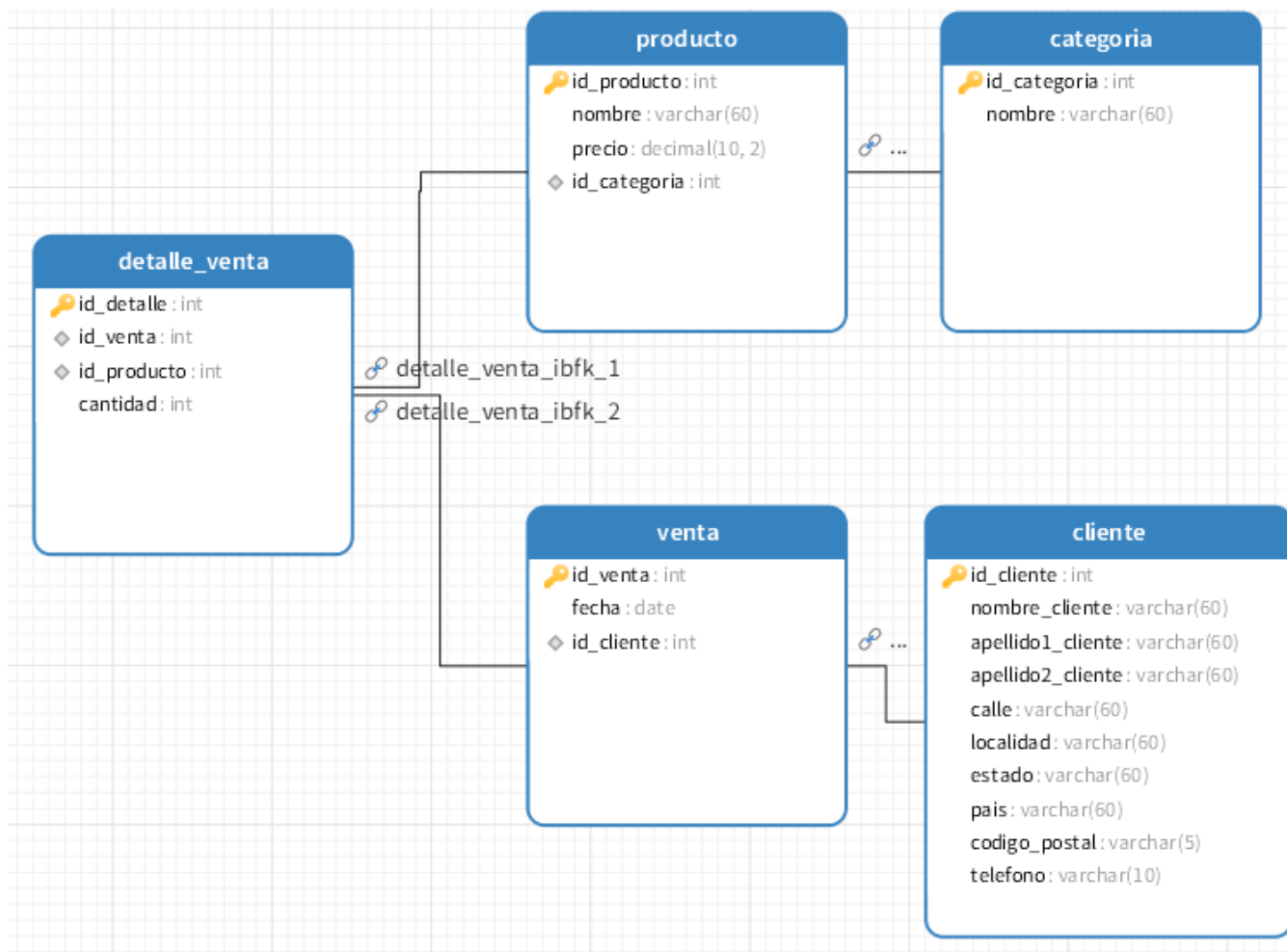
1 CREATE TABLE Producto(
2     id_producto INT PRIMARY KEY NOT NULL,
3     nombre VARCHAR(60) NOT NULL,
4     precio DECIMAL(10, 2) NOT NULL CHECK (precio > 0),
5     id_categoria INT,
6     FOREIGN KEY (id_categoria) REFERENCES categoria(id_categoria) ON UPDATE CASCADE ON DELETE RESTRICT
7 );
8
9 CREATE TABLE Categoria(
10     id_categoria INT PRIMARY KEY NOT NULL,
11     nombre VARCHAR(60) NOT NULL
12 );
13

```



```
14 CREATE TABLE Cliente(  
15     id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
16     nombre_cliente VARCHAR(60) NOT NULL,  
17     apellido1_cliente VARCHAR(60) NOT NULL,  
18     apellido2_cliente VARCHAR(60) NOT NULL,  
19     calle VARCHAR(60) NOT NULL,  
20     localidad VARCHAR(60) NOT NULL,  
21     estado VARCHAR(60) NOT NULL,  
22     pais VARCHAR(60) NOT NULL,  
23     codigo_postal VARCHAR(5) NOT NULL,  
24     telefono VARCHAR(10) NOT NULL  
25 );  
26  
27 CREATE TABLE Venta(  
28     id_venta INT PRIMARY KEY AUTO_INCREMENT,  
29     fecha DATE NOT NULL,  
30     id_cliente INT,  
31     FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente) ON UPDATE CASCADE ON DELETE RESTRICT  
32 );  
33  
34 CREATE TABLE Detalle_Venta (  
35     id_detalle INT PRIMARY KEY AUTO_INCREMENT,  
36     id_venta INT NOT NULL,  
37     id_producto INT NOT NULL,  
38     cantidad INT NOT NULL CHECK(cantidad > 0),  
39     FOREIGN KEY (id_venta) REFERENCES venta(id_venta) ON UPDATE CASCADE ON DELETE RESTRICT,  
40     FOREIGN KEY (id_producto) REFERENCES producto(id_producto) ON UPDATE CASCADE ON DELETE RESTRICT  
41 );
```

Tablas



- **NOT NULL:** Obliga a que los campos siempre tengan un valor. Por ejemplo, id_producto, nombre y precio en la tabla Producto no pueden estar vacíos.
- **CHECK:** Valida que los datos cumplan una condición. En CHECK

```
(precio > 0)
```

, asegura que el precio del producto sea mayor a 0, y en CHECK

```
(cantidad > 0)
```

 se asegura que la cantidad en Detalle_Venta sea positiva.
- **DECIMAL(10, 2):** Define el formato numérico para almacenar valores monetarios. Por ejemplo, precio DECIMAL(10, 2) permite precios de hasta 10 dígitos en total, con 2 decimales.
- **FOREIGN KEY, ON UPDATE CASCADE, ON DELETE RESTRICT:** Relacionan tablas. Si cambian las claves referenciadas, las actualizaciones se propagan automáticamente (**ON UPDATE CASCADE**). Las eliminaciones están restringidas si hay registros relacionados (**ON DELETE RESTRICT**).

Ejercicio 3

Diseñar y modificar un esquema para un sistema hospitalario.

1. 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.

2. Entidades y Atributos:

- **Paciente:**

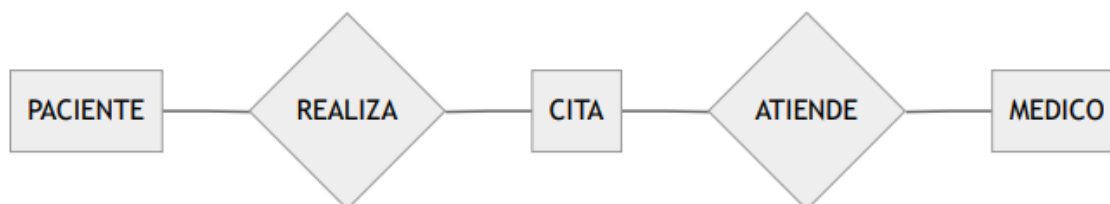
- id_paciente: Identificador único del paciente.
- nombre: Nombre del paciente.
- fecha_nacimiento: Fecha de nacimiento del paciente.
- sexo: Sexo del paciente.

- **Medico:**

- id_medico: Identificador único del médico.
- nombre: Nombre del médico.
- especialidad: Especialidad del médico.

- **Cita:**

- id_cita: Identificador único de la cita.
- id_paciente: Referencia al paciente que tiene la cita.
- id_medico: Referencia al médico que atiende la cita.
- fecha_cita: Fecha de la cita.
- hora_cita: Hora de la cita.





Normalización

id_paciente	nombre_paciente	apellido1_paciente	apellido2_paciente	fecha_nacimiento	sexo

id_medico	nombre_medico	apellido1_medico	apellido2_medico

CITA				
id_cita	id_paciente	id_medico	fecha_cita	hora_cita

Especialidad	
id_especialidad	nombre

Medico_Especialidad	
id_especialidad	id_medico

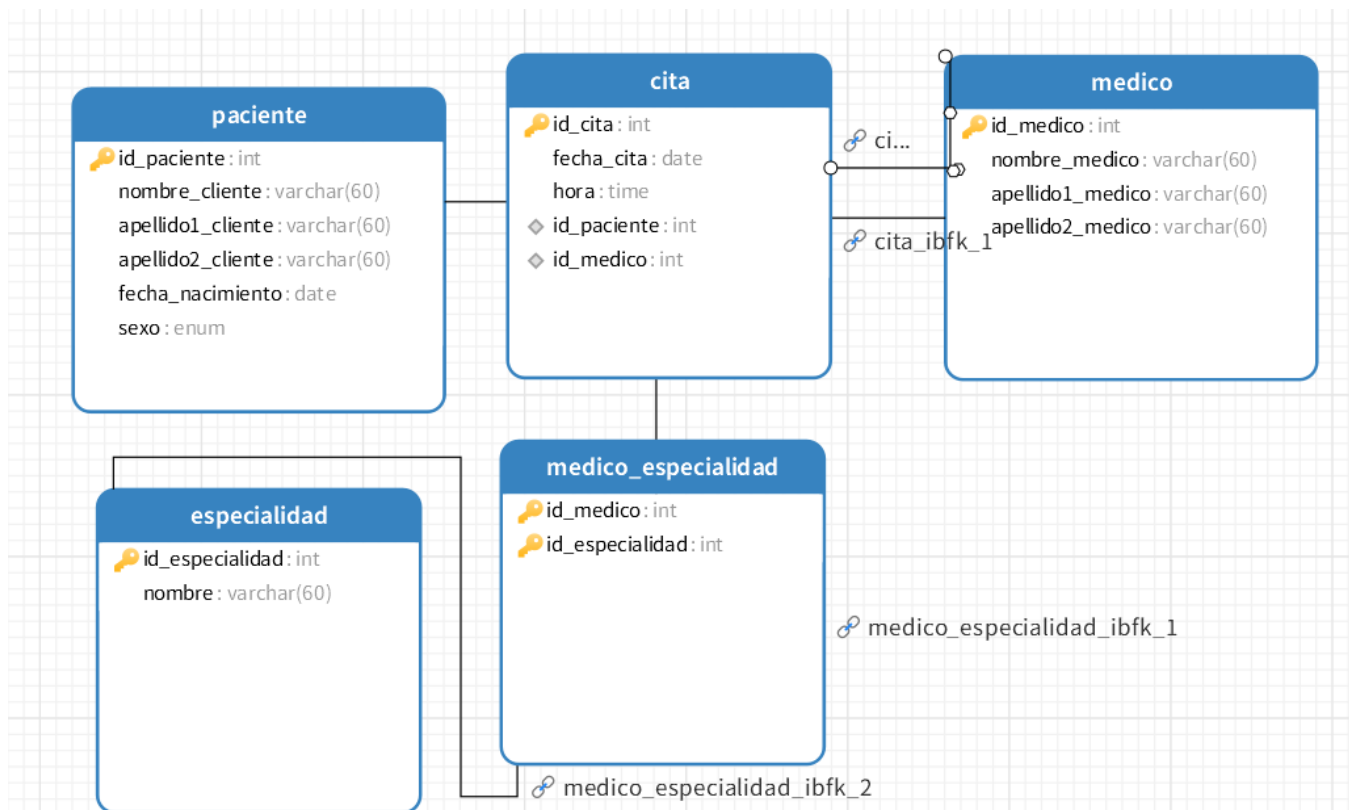
Código

```

1 CREATE TABLE Paciente(
2     id_paciente INT PRIMARY KEY AUTO_INCREMENT,
3     nombre_cliente VARCHAR(60) NOT NULL,
4     apellido1_cliente VARCHAR(60) NOT NULL,
5     apellido2_cliente VARCHAR(60) NOT NULL,
6     fecha_nacimiento DATE NOT NULL,
7     sexo ENUM('Masculino', 'Femenino', 'Otro', 'No especifico') NOT NULL
8 );
9
10 CREATE TABLE Medico(
11     id_medico INT PRIMARY KEY AUTO_INCREMENT,
12     nombre_medico VARCHAR(60) NOT NULL,
13     apellido1_medico VARCHAR(60) NOT NULL,
14     apellido2_medico VARCHAR(60) NOT NULL
15 );
16 CREATE TABLE Cita(
17     id_cita INT PRIMARY KEY AUTO_INCREMENT,
18     fecha_cita DATE NOT NULL,
19     hora TIME NOT NULL,
20     id_paciente INT NOT NULL,
21     id_medico INT NOT NULL,
22     FOREIGN KEY (id_paciente) REFERENCES paciente(id_paciente) ON UPDATE CASCADE ON DELETE RESTRICT,
23     FOREIGN KEY (id_medico) REFERENCES medico(id_medico) ON UPDATE CASCADE ON DELETE RESTRICT
24 );
25
26 CREATE TABLE Especialidad(
27     id_especialidad INT PRIMARY KEY AUTO_INCREMENT,
28     nombre VARCHAR(60) NOT NULL
29 );
30
31 DROP TABLE Especialidad;
32 CREATE TABLE Medico_Especialidad(
33     id_medico INT NOT NULL,
34     id_especialidad INT NOT NULL,
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 );
39

```

Tablas



- **NOT NULL**: Obliga a que ciertos campos no puedan estar vacíos, como `nombre_cliente`, `fecha_nacimiento`, y `sexo` en la tabla Paciente.
- **ENUM**: Define una lista de valores predefinidos. En `sexo` `ENUM('Masculino', 'Femenino', 'Otro', 'No especifico')`, el valor del campo `sexo` debe ser uno de estos valores.
- **AUTO_INCREMENT**: Genera automáticamente un valor único para campos como `id_paciente`, `id_medico`, `id_cita`, etc., en cada nueva inserción.
- **FOREIGN KEY, ON UPDATE CASCADE, ON DELETE RESTRICT**: Relaciona tablas como Cita con Paciente y Medico, y Medico con Especialidad. Si se actualizan las claves referenciadas, el cambio se propaga automáticamente (`ON UPDATE CASCADE`), y se restringe la eliminación de registros que tienen relaciones activas (`ON DELETE RESTRICT`).

Conclusión

En esta práctica se asegura la integridad de los datos y relaciones mediante el uso adecuado de restricciones como NOT NULL para evitar valores vacíos en campos esenciales, CHECK para validar datos (como precios o cantidades positivas), y FOREIGN KEY para crear relaciones entre tablas. Las opciones ON UPDATE CASCADE y ON DELETE RESTRICT garantizan que las actualizaciones se propaguen de manera consistente y que los datos no se eliminen si están siendo utilizados en otras tablas, preservando la integridad referencial. Además, el uso de AUTO_INCREMENT facilita la generación automática de identificadores únicos, mientras que ENUM permite definir valores específicos para ciertos campos, como el género de un paciente. Estas prácticas aseguran un manejo robusto, eficiente y seguro de la información.