1. \*\*mobiliario\*\*

- \*\*id\_mobiliario\*\* (PK) (int, auto incrementable)

- nombre (strig)

- estatus(bool, )

2. \*\*prestamo\*\*

- \*\*id\_prestamo\*\* (PK) (int, auto incrementable)

- fecha\_entrega (date)

- \*\*id\_empleado\*\* (FK hacia `personal.id\_personal`)

- \*\*id\_cliente\*\* (FK hacia `cliente.id\_cliente`)

-fecha\_inicio (date)

3. \*\*multa\*\*

- \*\*id\_multa\*\* (PK) (int, auto incrementable)

- monto(doble)

- \*\*id\_cliente\*\* (FK hacia `cliente.id\_cliente`)

- status(bool, para pagada o no pagada)

4. \*\*cliente\*\*

- \*\*id\_cliente\*\* (PK) (int, auto incrementable)

- teléfono(int[10])

- correo(mail)

- vigencia(date)

- inicio (date

5. \*\*genero\*\*

- \*\*id\_genero\*\* (PK) (int, auto incrementable)

- nombre(string)

6. \*\*libros\*\*

- \*\*id\_libro\*\* (PK) (int, auto incrementable)

- titulo(string)

- autor(string)

- edición(int)

- precio(doble)

- stock(int)

- \*\*id\_genero\*\* (FK hacia `genero.id\_genero`)

- \*\*id\_editorial\*\* (FK hacia `proveedor.id\_editorial`)

7. \*\*venta\*\*

- \*\*folio\_venta\*\* (PK) (int, auto incrementable)

- total(doble)

- fecha(date)

- metodo\_pago(string)

- \*\*id\_cliente\*\* (FK hacia `cliente.id\_cliente`)que pueda ser null

8. \*\*proveedor\*\*

- \*\*id\_editorial\*\* (PK) (int, auto incrementable)

- teléfono(int [10])

- dirección (string)

9. \*\*compra\*\*

- \*\*id\_compra\*\* (PK) (int, auto incrementable)

- total (doble)

- metodo\_pago (string)

- fecha (date)

- \*\*id\_editorial\*\* (FK hacia `proveedor.id\_editorial`)

- \*\*id\_empleado\*\* (FK hacia `personal.id\_personal`)

10. \*\*personal\*\*

- \*\*id\_personal\*\* (PK)

- nombre (string)

- teléfono (int [10)

- cargo (string)

- salario (doble)

### Tablas intermedias:

11. \*\*compra\_libro\*\* (tabla intermedia para la relación N:M entre `compra` y `libros`)

- \*\*id\_compra\*\* (FK hacia `compra.id\_compra`)

- \*\*id\_libro\*\* (FK hacia `libros.id\_libro`)

- cantidad (número de copias del libro en la compra)(int)

- PRIMARY KEY (id\_compra, id\_libro)

12. \*\*prestamo\_libro\*\* (tabla intermedia para la relación N:M entre `prestamo` y `libros`)

- \*\*id\_prestamo\*\* (FK hacia `prestamo.id\_prestamo`)

- \*\*id\_libro\*\* (FK hacia `libros.id\_libro`)

- cantidad (número de copias del libro en el préstamo) (int)

- PRIMARY KEY (id\_prestamo, id\_libro)

13. \*\*venta\_libro\*\* (tabla intermedia para la relación N:M entre `venta` y `libros`)

- \*\*folio\_venta\*\* (FK hacia `venta.folio\_venta`)

- \*\*id\_libro\*\* (FK hacia `libros.id\_libro`)

- cantidad (número de copias del libro en la venta)(int)

- PRIMARY KEY (folio\_venta, id\_libro)

14. \*\*preferencias\_cliente\*\* (tabla para la relación 1:M entre `cliente` y `genero`)

- \*\*id\_cliente\*\* (FK hacia `cliente.id\_cliente`)

- \*\*id\_genero\*\* (FK hacia `genero.id\_genero`)

- PRIMARY KEY (id\_cliente, id\_genero)

### Relaciones:

- \*\*compra\*\* ↔ \*\*compra\_libro\*\* ↔ \*\*libros\*\*: Una compra puede incluir múltiples libros, y un libro puede estar en múltiples compras.

- \*\*prestamo\*\* ↔ \*\*prestamo\_libro\*\* ↔ \*\*libros\*\*: Un préstamo puede incluir múltiples libros, y un libro puede estar en múltiples préstamos.

- \*\*venta\*\* ↔ \*\*venta\_libro\*\* ↔ \*\*libros\*\*: Una venta puede incluir múltiples libros, y un libro puede estar en múltiples ventas.

- \*\*cliente\*\* ↔ \*\*preferencias\_cliente\*\* ↔ \*\*genero\*\*: Un cliente puede tener múltiples géneros como preferencias, y un género puede ser preferido por múltiples clientes.

### Detalles de las claves foráneas y primarias:

- \*\*compra\_libro\*\*:

- \*\*id\_compra\*\* y \*\*id\_libro\*\* forman una clave primaria compuesta.

- \*\*id\_compra\*\* es una clave foránea que referencia a `compra.id\_compra`.

- \*\*id\_libro\*\* es una clave foránea que referencia a `libros.id\_libro`.

- \*\*prestamo\_libro\*\*:

- \*\*id\_prestamo\*\* y \*\*id\_libro\*\* forman una clave primaria compuesta.

- \*\*id\_prestamo\*\* es una clave foránea que referencia a `prestamo.id\_prestamo`.

- \*\*id\_libro\*\* es una clave foránea que referencia a `libros.id\_libro`.

- \*\*venta\_libro\*\*:

- \*\*folio\_venta\*\* y \*\*id\_libro\*\* forman una clave primaria compuesta.

- \*\*folio\_venta\*\* es una clave foránea que referencia a `venta.folio\_venta`.

- \*\*id\_libro\*\* es una clave foránea que referencia a `libros.id\_libro`.

- \*\*preferencias\_cliente\*\*:

- \*\*id\_cliente\*\* y \*\*id\_genero\*\* forman una clave primaria compuesta.

- \*\*id\_cliente\*\* es una clave foránea que referencia a `cliente.id\_cliente`.

- \*\*id\_genero\*\* es una clave foránea que referencia a `genero.id\_genero`.

### Revisión final:

- Verifica que todas las claves foráneas estén correctamente definidas con restricciones de integridad referencial (ON DELETE, ON UPDATE).

- Implementa índices en las claves foráneas para mejorar el rendimiento de las consultas.

Con estas modificaciones, tu diseño de base de datos debería reflejar con precisión las relaciones, mejorar la integridad y eficiencia del sistema, y estar bien normalizado.

-- Tabla mobiliario

CREATE TABLE IF NOT EXISTS mobiliario (

id\_mobiliario INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

estatus BOOLEAN

);

-- Tabla prestamo

CREATE TABLE IF NOT EXISTS prestamo (

id\_prestamo INT AUTO\_INCREMENT PRIMARY KEY,

fecha\_entrega DATE,

fecha\_inicio DATE,

id\_empleado INT,

id\_cliente INT,

FOREIGN KEY (id\_empleado) REFERENCES personal(id\_personal),

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente)

);

-- Tabla multa

CREATE TABLE IF NOT EXISTS multa (

id\_multa INT AUTO\_INCREMENT PRIMARY KEY,

monto DOUBLE,

id\_cliente INT,

status BOOLEAN,

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente)

);

-- Tabla cliente

CREATE TABLE IF NOT EXISTS cliente (

id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

telefono VARCHAR(20), -- Ajuste en la longitud del campo

correo VARCHAR(255),

vigencia DATE,

fecha\_inicio DATE, -- Corrección del nombre de la columna

CONSTRAINT fk\_cliente FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente) ON DELETE CASCADE -- Añadir restricción de clave externa para la auto-referencia

);

-- Tabla genero

CREATE TABLE IF NOT EXISTS genero (

id\_genero INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255)

);

-- Tabla libros

CREATE TABLE IF NOT EXISTS libros (

id\_libro INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(255),

autor VARCHAR(255),

edicion INT,

precio DOUBLE,

stock INT,

id\_genero INT,

id\_editorial INT,

FOREIGN KEY (id\_genero) REFERENCES genero(id\_genero),

FOREIGN KEY (id\_editorial) REFERENCES proveedor(id\_editorial)

);

-- Tabla venta

CREATE TABLE IF NOT EXISTS venta (

folio\_venta INT AUTO\_INCREMENT PRIMARY KEY,

total DOUBLE,

fecha DATE,

metodo\_pago VARCHAR(255),

id\_cliente INT, -- Permitir valores nulos para id\_cliente

CONSTRAINT fk\_venta\_cliente FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente) ON DELETE SET NULL -- Añadir restricción de clave externa y permitir valores nulos

);

-- Tabla proveedor

CREATE TABLE IF NOT EXISTS proveedor (

id\_editorial INT AUTO\_INCREMENT PRIMARY KEY,

telefono VARCHAR(20), -- Cambiar la longitud del campo telefono

direccion VARCHAR(255)

);

-- Tabla compra

CREATE TABLE IF NOT EXISTS compra (

id\_compra INT AUTO\_INCREMENT PRIMARY KEY,

total DOUBLE,

metodo\_pago VARCHAR(255),

fecha DATE,

id\_editorial INT,

id\_empleado INT,

FOREIGN KEY (id\_editorial) REFERENCES proveedor(id\_editorial),

FOREIGN KEY (id\_empleado) REFERENCES personal(id\_personal)

);

-- Tabla personal

CREATE TABLE IF NOT EXISTS personal (

id\_personal INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

telefono VARCHAR(20), -- Cambiar la longitud del campo telefono

cargo VARCHAR(255),

salario DOUBLE

);

-- Tabla compras\_libros

CREATE TABLE IF NOT EXISTS compras\_libros (

id\_compra INT,

id\_libro INT,

cantidad INT,

PRIMARY KEY (id\_compra, id\_libro),

FOREIGN KEY (id\_compra) REFERENCES compra(id\_compra),

FOREIGN KEY (id\_libro) REFERENCES libros(id\_libro)

);

-- Tabla prestamo\_libro

CREATE TABLE IF NOT EXISTS prestamo\_libro (

id\_prestamo INT,

id\_libro INT,

cantidad INT,

PRIMARY KEY (id\_prestamo, id\_libro),

FOREIGN KEY (id\_prestamo) REFERENCES prestamo(id\_prestamo),

FOREIGN KEY (id\_libro) REFERENCES libros(id\_libro)

);

-- Tabla venta\_libro

CREATE TABLE IF NOT EXISTS venta\_libro (

folio\_venta INT,

id\_libro INT,

cantidad INT,

PRIMARY KEY (folio\_venta, id\_libro),

FOREIGN KEY (folio\_venta) REFERENCES venta(folio\_venta),

FOREIGN KEY (id\_libro) REFERENCES libros(id\_libro)

);

-- Tabla clientes\_generos

CREATE TABLE IF NOT EXISTS clientes\_generos (

id\_cliente INT,

id\_genero INT,

PRIMARY KEY (id\_cliente, id\_genero),

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente),

FOREIGN KEY (id\_genero) REFERENCES genero(id\_genero)

);

//

-- Tabla prestamo

CREATE TABLE IF NOT EXISTS prestamo (

id\_prestamo INT AUTO\_INCREMENT PRIMARY KEY,

fecha\_entrega DATE,

fecha\_inicio DATE,

id\_empleado INT,

id\_cliente INT,

FOREIGN KEY (id\_empleado) REFERENCES personal(id\_personal),

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente)

);

-- Tabla cliente

CREATE TABLE IF NOT EXISTS cliente (

id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

telefono VARCHAR(20), -- Ajuste en la longitud del campo

correo VARCHAR(255),

vigencia DATE,

fecha\_inicio DATE, -- Corrección del nombre de la columna

CONSTRAINT fk\_cliente FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente) ON DELETE CASCADE -- Añadir restricción de clave externa para la auto-referencia

);

-- Tabla libros

CREATE TABLE IF NOT EXISTS libros (

id\_libro INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(255),

autor VARCHAR(255),

edicion INT,

precio DOUBLE,

stock INT,

id\_genero INT,

id\_editorial INT,

FOREIGN KEY (id\_genero) REFERENCES genero(id\_genero),

FOREIGN KEY (id\_editorial) REFERENCES proveedor(id\_editorial)

);

-- Tabla prestamo\_libro

CREATE TABLE IF NOT EXISTS prestamo\_libro (

id\_prestamo INT,

id\_libro INT,

status bool,

PRIMARY KEY (id\_prestamo, id\_libro),

FOREIGN KEY (id\_prestamo) REFERENCES prestamo(id\_prestamo),

FOREIGN KEY (id\_libro) REFERENCES libros(id\_libro)

);

Tengo estas tablas en una base de datos, quiero saber cuales son los los libros que un cliente tiene prestados a partir de su id\_cliente, dame el sql necesario para la consulta

DELIMITER //

CREATE TRIGGER restar\_stock\_libro

AFTER INSERT ON venta\_libro

FOR EACH ROW

BEGIN

UPDATE libros

SET stock = stock - NEW.cantidad

WHERE id\_libro = NEW.id\_libro;

END//

DELIMITER ;