**1. Diagrama Entidad-Relación (E/R)**

**Entidades**:

* **Libro**: título, género, fecha\_publicación, ISBN
* **Autor**: nombre, apellido, nacionalidad
* **Usuario**: nombre, apellido, teléfono, email
* **Préstamo**: fecha\_préstamo, fecha\_devolución

**Relaciones**:

* Un **autor** puede escribir muchos **libros**. (1 a N)
* Un **libro** puede estar en varios **préstamos**, pero solo un **usuario** puede tomarlo prestado en un momento dado. (1 a N)
* Un **usuario** puede hacer varios **préstamos**. (1 a N)

**2. Modelo Relacional**

1. **Libros** (ISBN, título, género, fecha\_publicación)
   * ISBN (PK)
2. **Autores** (autor\_id, nombre, apellido, nacionalidad)
   * autor\_id (PK)
3. **Usuarios** (usuario\_id, nombre, apellido, teléfono, email)
   * usuario\_id (PK)
4. **Préstamos** (prestamo\_id, libro\_id, usuario\_id, fecha\_préstamo, fecha\_devolución)
   * prestamo\_id (PK)
   * libro\_id (FK hacia ISBN en Libros)
   * usuario\_id (FK hacia usuario\_id en Usuarios)
5. **Autores\_Libros** (autor\_id, libro\_id)
   * autor\_id (FK hacia autor\_id en Autores)
   * libro\_id (FK hacia ISBN en Libros)

**3. Modelo Físico (SQL en PostgreSQL)**

CREATE TABLE Autores (autor\_id SERIAL PRIMARY KEY, nombre VARCHAR(50) NOT NULL, apellido VARCHAR(100) NOT NULL, pais\_id VARCHAR(3) REFERENCES Naciones(pais\_id));

CREATE TABLE Naciones (pais\_id VARCHAR(3) PRIMARY KEY, nombre VARCHAR(50) NOT NULL);

CREATE TABLE Libros (ISBN VARCHAR(13) PRIMARY KEY, titulo VARCHAR(150) NOT NULL, genero\_id INT REFERENCES genero(genero\_id), fecha\_publicacion DATE);

CREATE TABLE Genero (genero\_id SERIAL PRIMARY KEY, nombre VARCHAR(50) NOT NULL);

CREATE TABLE Usuarios (usuario\_id SERIAL PRIMARY KEY, nombre VARCHAR(50) NOT NULL, apellido VARCHAR(50) NOT NULL, telefono VARCHAR(15), email VARCHAR(50));

CREATE TABLE Prestamos (libro\_id VARCHAR(13) REFERENCES Libros(ISBN), usuario\_id INTEGER REFERENCES Usuarios(usuario\_id), fecha\_prestamo DATE PRIMARY KEY, fecha\_devolucion DATE );

**4. Inserción de Datos**

1. Naciones

INSERT INTO Naciones (pais\_id, nombre) VALUES

('ARG', 'Argentina'),

('ESP', 'España'),

('USA', 'Estados Unidos'),

('MEX', 'México'),

('COL', 'Colombia');

2. Autores

INSERT INTO Autores (nombre, apellido, pais\_id) VALUES

('Gabriel', 'García Márquez', 'COL'),

('Jorge Luis', 'Borges', 'ARG'),

('Isabel', 'Allende', 'ESP'),

('Carlos', 'Fuentes', 'MEX'),

('Ernest', 'Hemingway', 'USA');

3. Genero

INSERT INTO Genero (nombre) VALUES

('Novela'),

('Poesía'),

('Cuento'),

('Ensayo'),

('Fantasía');

4. Libros

INSERT INTO Libros (ISBN, titulo, genero\_id, fecha\_publicacion) VALUES

('9780060883287', 'Cien años de soledad', 1, '1967-05-30'),

('9788481095872', 'Ficciones', 3, '1944-01-01'),

('9781501117430', 'La casa de los espíritus', 1, '1982-01-01'),

('9780679732266', 'Gringo viejo', 1, '1985-01-01'),

('9780684830483', 'El viejo y el mar', 1, '1952-09-01');

5. Usuarios

INSERT INTO Usuarios (nombre, apellido, telefono, email) VALUES

('Juan', 'Pérez', '123456789', 'juan.perez@gmail.com'),

('Ana', 'Martínez', '987654321', 'ana.martinez@yahoo.com'),

('Luis', 'Gómez', '456123789', 'luis.gomez@hotmail.com'),

('Marta', 'López', '654321987', 'marta.lopez@gmail.com'),

('Carlos', 'Hernández', '321654987', 'carlos.hernandez@outlook.com');

6. Prestamos

INSERT INTO Prestamos (libro\_id, usuario\_id, fecha\_prestamo, fecha\_devolucion) VALUES

('9780060883287', 1, '2024-09-01', '2024-09-15'),

('9788481095872', 2, '2024-09-05', '2024-09-20'),

('9781501117430', 3, '2024-09-10', '2024-09-25'),

('9780679732266', 4, '2024-09-15', NULL), -- Aún no devuelto

('9780684830483', 5, '2024-09-20', NULL); -- Aún no devuelto

**4. Inserción de Datos**

**5. Consultas SQL**

* **Listar todos los libros disponibles en la biblioteca (no prestados):**

sql

Copiar código

SELECT \* FROM Libros

WHERE ISBN NOT IN (SELECT libro\_id FROM Prestamos WHERE fecha\_devolucion IS NULL);

* **Listar todos los autores y los libros que han escrito:**

sql

Copiar código

SELECT A.nombre, A.apellido, L.titulo

FROM Autores A

JOIN Autores\_Libros AL ON A.autor\_id = AL.autor\_id

JOIN Libros L ON AL.libro\_id = L.ISBN;

* **Listar todos los usuarios y los libros que han tomado prestados:**

sql

Copiar código

SELECT U.nombre, U.apellido, L.titulo

FROM Usuarios U

JOIN Prestamos P ON U.usuario\_id = P.usuario\_id

JOIN Libros L ON P.libro\_id = L.ISBN;

* **Consultar los libros que están actualmente prestados y a qué usuarios:**

sql

Copiar código

SELECT U.nombre, U.apellido, L.titulo, P.fecha\_prestamo

FROM Usuarios U

JOIN Prestamos P ON U.usuario\_id = P.usuario\_id

JOIN Libros L ON P.libro\_id = L.ISBN

WHERE P.fecha\_devolucion IS NULL;

Esto te permitirá gestionar toda la información de la biblioteca y realizar las consultas solicitadas.

1. **Listar todos los libros disponibles en la biblioteca:** Esta consulta busca todos los libros que no están en la tabla de préstamos o que ya han sido devueltos.

SELECT l.ISBN, l.titulo, l.fecha\_publicacion

FROM Libros l

WHERE l.ISBN NOT IN (SELECT p.libro\_id FROM Prestamos p WHERE p.fecha\_devolucion IS NULL);

1. **Listar todos los autores y los libros que han escrito:** Aquí necesitas hacer un JOIN entre la tabla de autores y la tabla de libros (considerando que habría una relación entre ambas, como una tabla intermedia "Libros\_Autores").

SELECT a.nombre, a.apellido, l.titulo

FROM Autores a

JOIN Libros\_Autores la ON a.autor\_id = la.autor\_id

JOIN Libros l ON la.ISBN = l.ISBN;

**Nota:** Esto asume la existencia de una tabla intermedia Libros\_Autores que conecta los autores con los libros.

1. **Listar todos los miembros y los libros que han tomado prestados:**

SELECT u.nombre, u.apellido, l.titulo, p.fecha\_prestamo

FROM Usuarios u

JOIN Prestamos p ON u.usuario\_id = p.usuario\_id

JOIN Libros l ON p.libro\_id = l.ISBN;

1. **Consultar los libros que están actualmente prestados y a qué miembros:** Aquí se buscan los libros que aún no han sido devueltos (suponiendo que la tabla Prestamos tiene una columna fecha\_devolucion para gestionar devoluciones).

SELECT u.nombre, u.apellido, l.titulo, p.fecha\_prestamo

FROM Usuarios u

JOIN Prestamos p ON u.usuario\_id = p.usuario\_id

JOIN Libros l ON p.libro\_id = l.ISBN

WHERE p.fecha\_devolucion IS NULL;

**Explicaciones:**

* **Libros disponibles:** Se consulta por libros cuyo ISBN no aparece en préstamos no devueltos.
* **Autores y sus libros:** Se realiza un JOIN entre autores y libros a través de una tabla intermedia, si existe.
* **Miembros y libros prestados:** Se hace un JOIN entre usuarios, préstamos y libros para ver qué libros ha tomado prestados cada usuario.
* **Libros actualmente prestados:** Se filtra la tabla de préstamos para mostrar solo aquellos que aún no han sido devueltos.

Si tienes más consultas o deseas agregar detalles a las tablas, no dudes en preguntar.

4o

Principio del formulario

Final del formulario