To design and implement a relational database for a Library Management System (LMS), the first step is to identify the key entities, their attributes, and the relationships between them. Below is a step-by-step approach to help you design and implement such a system.

**1. Identify Entities and Relationships**

**Entities:**

* **Books**: Represents the books in the library.
* **Members**: Represents the members of the library who can borrow books.
* **Loans**: Represents the borrowing of books by members.
* **Authors**: Represents the authors of the books.
* **Categories**: Represents the genres or categories of books.
* **Staff**: Represents the staff managing the library system.

**Relationships:**

* A **Book** can have one or more **Authors**.
* A **Book** can belong to one or more **Categories**.
* A **Member** can **Borrow (Loan)** multiple books.
* A **Loan** links a **Member** with a **Book**.
* **Staff** may manage **Members**, **Books**, and **Loans**.

**2. Database Tables and Structure**

**Books Table**

* book\_id: Primary Key, unique ID for each book.
* title: The title of the book.
* isbn: International Standard Book Number (unique).
* category\_id: Foreign Key, refers to the category of the book.
* author\_id: Foreign Key, refers to the author of the book.
* copies\_available: Number of available copies in the library.

**Authors Table**

* author\_id: Primary Key, unique ID for each author.
* first\_name: First name of the author.
* last\_name: Last name of the author.

**Categories Table**

* category\_id: Primary Key, unique ID for each category.
* category\_name: Name of the book category (e.g., Fiction, Non-Fiction, Science).

**Members Table**

* member\_id: Primary Key, unique ID for each member.
* first\_name: First name of the member.
* last\_name: Last name of the member.
* phone: Contact number of the member.
* email: Email address of the member.

**Loans Table**

* loan\_id: Primary Key, unique ID for each loan.
* book\_id: Foreign Key, refers to the book being borrowed.
* member\_id: Foreign Key, refers to the member who borrowed the book.
* loan\_date: Date when the book was borrowed.
* return\_date: Date when the book is due for return.

**Staff Table**

* staff\_id: Primary Key, unique ID for each staff.
* first\_name: First name of the staff member.
* last\_name: Last name of the staff member.
* role: Role of the staff (e.g., Librarian, Assistant).

**3. ER Diagram**

**ER Diagram Entities and Relationships:**

1. **Books ↔ Authors (Many-to-Many)**:
   * A book can be written by multiple authors, and an author can write multiple books.
   * To represent this, an additional **BookAuthors** junction table is required.
2. **Books ↔ Categories (Many-to-One)**:
   * Each book belongs to one category, but each category can have multiple books.
3. **Members ↔ Loans (One-to-Many)**:
   * A member can borrow multiple books (i.e., have multiple loan records), but each loan is associated with one member.
4. **Books ↔ Loans (One-to-Many)**:
   * A book can be borrowed multiple times by different members (i.e., it can have multiple loan records), but each loan is associated with only one book.

**Enhanced Relationships:**

* **Books** ↔ **Authors**: Many-to-Many
* **Books** ↔ **Categories**: Many-to-One
* **Members** ↔ **Loans**: One-to-Many
* **Books** ↔ **Loans**: One-to-Many

**ER Diagram Visualization:**

Here’s how the entities and relationships are structured:

**Entities:**

1. **Books** (book\_id, title, isbn, category\_id, copies\_available)
2. **Authors** (author\_id, first\_name, last\_name)
3. **Categories** (category\_id, category\_name)
4. **Members** (member\_id, first\_name, last\_name, phone, email)
5. **Loans** (loan\_id, book\_id, member\_id, loan\_date, return\_date, status)
6. **BookAuthors** (book\_id, author\_id)

**Relationships:**

1. **Books ↔ Authors**: Many-to-Many (via BookAuthors table).
2. **Books ↔ Categories**: Many-to-One (Books belong to one category).
3. **Members ↔ Loans**: One-to-Many (One member can have multiple loans).
4. **Books ↔ Loans**: One-to-Many (One book can have multiple loans).
5. **SQL Table Creation (Example)**

create database liabraymgt ;

Query OK, 1 row affected (0.03 sec)

mysql> use liabraymgt;

Database changed

* **Create the Authors table**

create table authors (

author\_id int primary key auto\_increment,

first\_name varchar(100),

last\_name varchar(100)

);

Query OK, 0 rows affected (0.04 sec)

* **Create the Categories table**

create table categories(

category\_id int primary key auto\_increment,

category\_name varchar(100)

);

Query OK, 0 rows affected (0.39 sec)

* **Create the Members table**

create table members (

member\_id int primary key auto\_increment,

first\_name varchar(100),

last\_name varchar(100),

phone varchar(15),

email varchar(100)

);

Query OK, 0 rows affected (0.05 sec)

* **Create the Books table**

mysql> create table books(

book\_id int primary key auto\_increment,

title varchar(255) not null,

isbn varchar(20) unique,

category\_id int,

author\_id int,

copies\_available int default 1,

foreign key (category\_id) references categories(category\_id),

foreign key (author\_id) references authors(author\_id)

);

Query OK, 0 rows affected (0.10 sec)

* **Create the Loans table**

create table loans (

loan\_id int primary key auto\_increment,

book\_id int,

member\_id int,

loan\_date date,

return\_date date,

foreign key (book\_id) references books(book\_id),

foreign key (member\_id) references members(member\_id)

);

Query OK, 0 rows affected (0.06 sec)

* **Create the Staff table**

create table staff (

staff\_id int primary key auto\_increment,

first\_name varchar(100),

last\_name varchar(100),

role varchar(50)

);

Query OK, 0 rows affected (0.04 sec)

**5. Example Queries**

* **Add a new book:**

insert into books (title, isbn, copies\_available) values ('Yayati','924-45451211',12);

Query Oc, 1 row affected (0.01 sec)

* **Add a new member:**

insert into members (first\_name, last\_name, phone, email) values('Kiran','Patil','8956325752','kiran@gmail.com');

Query OK, 1 row affected (0.04 sec)

* **Record a book loan:**

insert into loans ( member\_id, loan\_date, return\_date)

values(1,'2024-09-17','2024-09-29');

Query OK, 1 row affected (0.01 sec)

* **Return a book:**

update Loans

set return\_date = '2024-09-20'

where loan\_id = 1;

**6. ER Diagram:**

