Case Study

Designing an Efficient Database for a Library Management System

AKILANDESHWARI SRINIVASAN

Table Of contents:

- ❖ Introduction
- ❖ Mission Statement
- ❖ Mission Objective
- ❖ Entity-Relationship Diagram
- ❖ Identified Subjects
- ❖ Database Tables
- ❖ Importance of Good Database Design
- ❖ Appendix A
    - ◆ Data Dictionary for Tables
- ❖ Appendix B
    - ◆ Screenshots

Introduction:

The Library Management System is designed to provide an efficient and user-friendly platform that enhances library operations and user experience. It helps in cataloging, circulation, tracking of books and resources, managing member accounts, and generating reports. By streamlining processes, the system improves efficiency, reduces manual work, and enhances the overall user.

Mission Statement:

To streamline library operations with improving the integrity and facilitate easier access to library resources.

Mission Objectives:

1. Efficiently manage the borrowing, returning, and reserving of library resources.

2. Automate library operations to reduce manual work and improve efficiency.

3. Track the availability and status of library materials in real-time.

4. Provide detailed reports and analytics on library usage to support decision-making.

Entity Relationship Diagram:
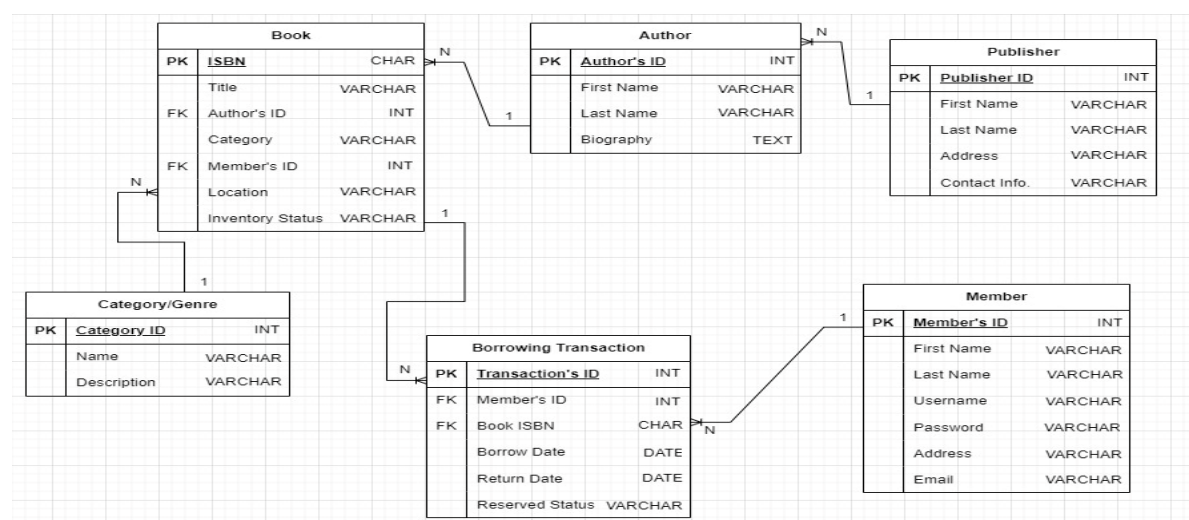
Identified Subjects:

When designing a library management system, it is essential to identify the key **subjects** that the database will manage. These subjects represent different types of information or entities that the library needs to track. The subjects identified for the Library Management System are:

1. **Book** – Details about every book available in the library, including its title, ISBN, and other related information.

2. **Author** – Information about the authors of books, such as their names and biographies.

3. **Member** – Records of library members, including their personal details and borrowing history.

4. **Reservation** – Data on the books reserved by members, including the reservation date and status.

5. **Publisher** – Information about the publishers of books, including their contact information.

6. **Category/Genre** – Categories or genres that group books based on their content.

Database Tables:

Once the subjects are identified, they are translated into **tables** in the database. Each table stores specific data about a subject, and relationships between tables are created using **foreign keys**. Below are the primary tables used in this LMS:

1. **Book Table**:

   - Stores information about each book, including its ISBN, title, author ID, category, location in the library, and its availability status.

2. **Author Table**:

   o Holds details about book authors, including their unique author ID, names, and biographical information.

3. **Member Table**:

   o Tracks library members and their details, such as their names, addresses, emails, and login credentials for the system.

4. **Publisher Table**:

   o Contains records about book publishers, including contact details and address.

5. **Borrowing Transaction Table**:

   o Stores information about book borrowings, including the transaction ID, member ID, book ISBN, borrow and return dates, and the status of the transaction (e.g., returned or overdue).

6. **Category/Genre Table**:

   o Lists the different categories or genres available in the library, allowing books to be classified for easier searching and organization.

7. **Reservation Table**:

   o Manages the reservations made by library members, keeping track of who reserved which book and when.

Importance of Good Database Design:

A well-designed database is the backbone of an effective Library Management System. A good database design ensures:

- **Efficiency**: By organizing data effectively and eliminating redundancy, the system runs more smoothly, reducing the time needed to complete tasks.

- **Data Integrity**: With clear relationships between tables, the system can enforce rules, such as ensuring that a book cannot be borrowed if it's not in the library's catalog.

- **Scalability**: A well-structured database can grow with the library, accommodating more users, books, and services without breaking down or slowing down.

Relationships:

- **Book → Author**:
  Relationship: **Many-to-One**
  Explanation: Each book can be written by one author, but an author can write multiple books. This relationship is managed using the author_id foreign key in the **Book** table, referencing the **Author** table.

- **Book → Category/Genre**:
  Relationship: **Many-to-One**
  Explanation: Each book belongs to one category or genre, but a category can have many books. The category_id in the **Book** table will reference the **Category/Genre** table.

- **Book → Publisher**:
  Relationship: **Many-to-One**
  Explanation: Each book has one publisher, but a publisher can publish many books. The **Book** table will have a publisher_id that references the **Publisher** table.

- **Book → BorrowingTransaction**:
  Relationship: **One-to-Many**
  Explanation: A book can be involved in multiple borrowing transactions, but a transaction can only reference one book.

The **BorrowingTransaction** table has a foreign key book_isbn referencing the **Book** table.

- **Member → BorrowingTransaction**:
  Relationship: **One-to-Many**
  Explanation: A member can borrow multiple books, but each transaction references one member. The **BorrowingTransaction** table has a member_id that references the **Member** table.

- **Reservation → Member**:
  Relationship: **Many-to-One**
  Explanation: A member can reserve multiple books, but a reservation references one member. The **Reservation** table will have a foreign key member_id that references the **Member** table.

- **Reservation → Book**:
  Relationship: **Many-to-One**
  Explanation: A book can be reserved by multiple members, but each reservation is for one book. The **Reservation** table will have a book_isbn foreign key referencing the **Book** table.

Conclusion:

The database design for the Library Management System (LMS) effectively supports the objective of streamlining library operations and improving access to resources. By organizing key tables such as Book, Author, Member, and BorrowingTransaction, and defining relationships between them, the system enables efficient management of library resources, automates transactions, and ensures data integrity. This design provides a solid foundation for enhancing user experience and supporting future growth.

Appendix A:

Data Dictionary – Book Table:

Field Name	Data Type	Description
ISBN (PK)	VARCHAR	International Standard Book Number
Title	VARCHAR	Title of the Book
Author's ID (FK)	INT	Links to the Author of the book
Category	VARCHAR	Links to the book's category/genre
Member's ID (FK)	INT	Unique identifier for each library member
Location	VARCHAR	Shelf location of the book in the library
Inventory Status	VARCHAR	Availability status of the book (Available/Checked Out)

Data Dictionary –Author Table:

Field Name	Data Type	Description
ISBN (PK)	VARCHAR	International Standard Book Number
Title	VARCHAR	Title of the Book
Author's ID (FK)	INT	Links to the Author of the book

Category	VARCHAR	Links to the book's category/genre
Member's ID (FK)	INT	Unique identifier for each library member
Location	VARCHAR	Shelf location of the book in the library
Inventory Status	VARCHAR	Availability status of the book (Available/Checked Out)

Data Dictionary –Member Table:

Field Name	Data Type	Description
ISBN (PK)	VARCHAR	International Standard Book Number
Title	VARCHAR	Title of the Book
Author's ID (FK)	INT	Links to the Author of the book
Category	VARCHAR	Links to the book's category/genre
Member's ID (FK)	INT	Unique identifier for each library member
Location	VARCHAR	Shelf location of the book in the library
Inventory Status	VARCHAR	Availability status of the book (Available/Checked Out)

Data Dictionary –Publisher Table:

Field Name	Data Type	Description
Publisher ID (PK)	INT	Unique identifier for each publisher
First Name	VARCHAR	First name of the publisher
Last Name	VARCHAR	Last name of the publisher

|	VARCHAR	Publisher's address
Address	VARCHAR	Publisher's address
Contact Information	VARCHAR	Contact details for the publisher

Data Dictionary – Borrowing Transactions Table

Field Name	Data Type	Description
Transaction's ID (PK)	INT	Unique identifier for each borrowing transaction
Member's ID (FK)	INT	Links to the borrowing member
Book ISBN (FK)	VARCHAR	Links to the borrowed book
Borrow Date	DATETIME	Date when the book was borrowed
Return Date	DATETIME	Date when the book is expected or was returned
Status	VARCHAR	Status of the transaction (Borrowed/Returned)

Appendix B:
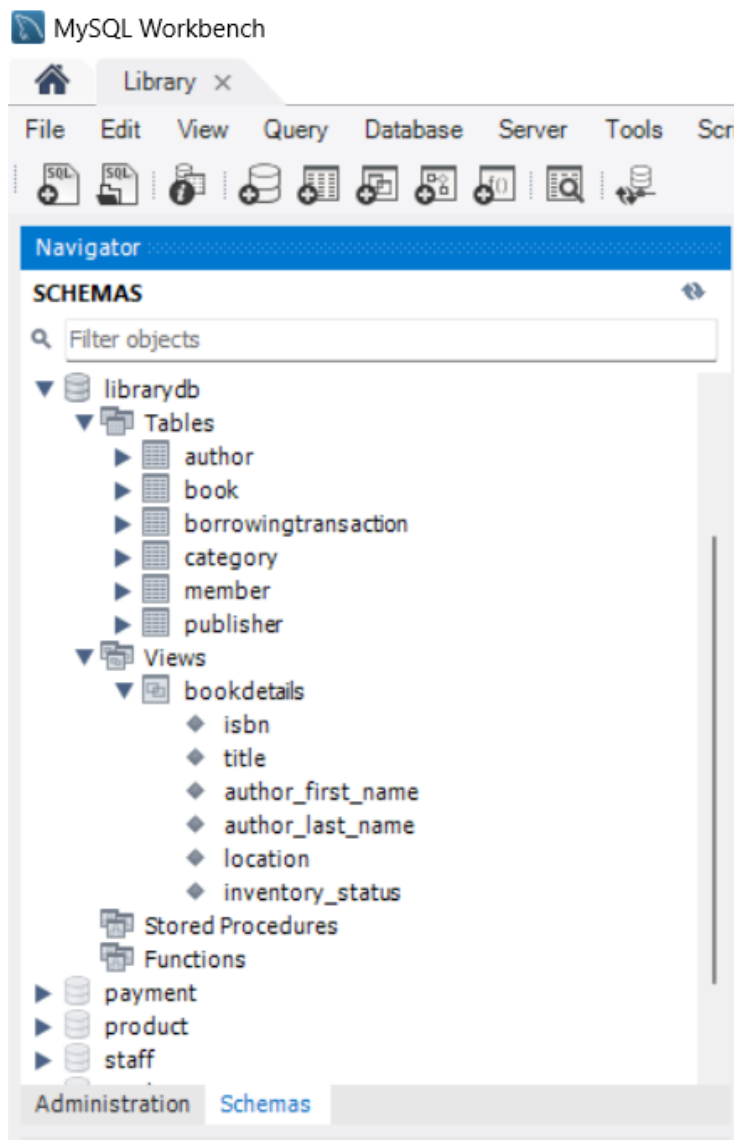
ScreenShots :

```
88  ●   SELECT * FROM Book;
89  ●   SELECT * FROM Author;
90
91  ●   SELECT * FROM BorrowingTransaction;
92
93
94  ●   SELECT
95          B.title,
96          A.firstName AS author_first_name,
97          A.lastName AS author_last_name
98      FROM
99          Book B
100     INNER JOIN
101         Author A ON B.author_id = A.author_id;
102
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

title	author_first_name	author_last_name
The Great Novel	John	Doe
A Sci-Fi Adventure	Jane	Smith

Result 1 · Read Only

```
102
103
104 ●   CREATE VIEW BookDetails AS
105     SELECT
106         B.isbn,
107         B.title,
108         A.firstName AS author_first_name,
109         A.lastName AS author_last_name,
110         B.location,
111         B.inventory_status
112     FROM
113         Book B
114     INNER JOIN
115         Author A ON B.author_id = A.author_id;
116
117 ●   SELECT * FROM BookDetails;
118
119
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

isbn	title	author_first_name	author_last_name	location	inventory_status
978-3-16-148410-0	The Great Novel	John	Doe	Shelf A1	Available
978-1-40-289462-6	A Sci-Fi Adventure	Jane	Smith	Shelf B2	Borrowed