**SQL-based Library Management System project**, covering:

1. **Database & Table Creation**
2. **Relationships (ER model)**
3. **Inserted Data (what and why)**
4. **SQL Queries (how to access, update, delete, filter)**
5. **Practice tips**

**🧱 1. DATABASE & TABLE CREATION**

**🔹 CREATE DATABASE LibraryDB;**

Creates a new database named LibraryDB to hold all the tables and data.

**🔹 USE LibraryDB;**

Switches to the LibraryDB so further commands affect this database.

**📂 TABLE STRUCTURES**

**🧑‍💼 Author**

sql

CopyEdit

CREATE TABLE Author (

author\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

bio TEXT

);

* **author\_id**: Unique ID for each author (auto-incremented).
* **name**: Author’s full name.
* **bio**: Optional biography or description of the author.

**📚 Book**

sql

CopyEdit

CREATE TABLE Book (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(150) NOT NULL,

category VARCHAR(100),

available\_copies INT DEFAULT 1,

author\_id INT,

FOREIGN KEY (author\_id) REFERENCES Author(author\_id)

);

* **book\_id**: Unique ID for each book.
* **title**: Name of the book.
* **category**: Genre/type like Mystery, Fantasy, etc.
* **available\_copies**: Number of copies available (default is 1).
* **author\_id**: Links to Author(author\_id) as foreign key.

➡️ **Relation:** Many books can be written by one author.

**🎓 Student**

sql

CopyEdit

CREATE TABLE Student (

student\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE,

phone VARCHAR(15)

);

* **student\_id**: Unique ID for each student.
* **name**, **email**, **phone**: Contact details.

➡️ Students can borrow books from the library.

**🔁 Loan**

sql

CopyEdit

CREATE TABLE Loan (

loan\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT,

book\_id INT,

issue\_date DATE,

return\_date DATE,

FOREIGN KEY (student\_id) REFERENCES Student(student\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id)

);

* **loan\_id**: Unique ID for the loan transaction.
* **student\_id**: Who borrowed the book (foreign key to Student).
* **book\_id**: Which book was borrowed.
* **issue\_date / return\_date**: When the book was taken/returned.

➡️ Tracks each borrowing event — useful for history and overdue tracking.

**👨‍🏫 Librarian**

sql

CopyEdit

CREATE TABLE Librarian (

librarian\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100),

phone VARCHAR(15)

);

* **Librarian** table holds info about library staff.

**🧾 2. SAMPLE DATA EXPLAINED**

**📘 Authors (5 entries)**

sql

CopyEdit

INSERT INTO Author (name, bio) VALUES

('Jane Austen', 'English novelist...'),

('Mark Twain', 'American writer...'),

('J.K. Rowling', 'Harry Potter series...'),

('George Orwell', '1984 and Animal Farm...'),

('Agatha Christie', 'Queen of Crime...');

➡️ These authors are famous and widely used in libraries.

**📕 Books (10 entries)**

sql

CopyEdit

INSERT INTO Book (title, category, available\_copies, author\_id) VALUES

('Pride and Prejudice', 'Classic', 3, 1),

...

('And Then There Were None', 'Mystery', 2, 5);

* Each book belongs to one author.
* Different genres/categories like "Fantasy", "Mystery", etc.
* available\_copies helps track how many can be borrowed.

**🎓 Students (10 entries)**

sql

CopyEdit

INSERT INTO Student (name, email, phone) VALUES

('Arjun Rai', 'arjun.rai@example.com', '9876543201'),

...

('Jayant Mehta', 'jayant.mehta@example.com', '9876543210');

➡️ Represent students or members using the library.

**🔁 Loans (5 entries)**

sql

CopyEdit

INSERT INTO Loan (student\_id, book\_id, issue\_date, return\_date) VALUES

(1, 1, '2025-06-01', '2025-06-15'),

...

(8, 3, '2025-06-09', NULL);

* Tracks who borrowed what and when.
* NULL in return\_date means book is still borrowed.

**👩‍🏫 Librarians (2 entries)**

sql

CopyEdit

INSERT INTO Librarian (name, email, phone) VALUES

('Kavita Menon', ...),

('Rahul Bose', ...);

➡️ Library staff records.

**🔍 3. SQL QUERIES EXPLAINED**

**📖 READ data (SELECT)**

sql

CopyEdit

SELECT \* FROM Book;

SELECT title FROM Book WHERE category = 'Mystery';

**🔗 JOIN Tables**

sql

CopyEdit

SELECT Book.title, Author.name

FROM Book

JOIN Author ON Book.author\_id = Author.author\_id;

➡️ Join shows which author wrote which book.

**✏️ UPDATE data**

sql

CopyEdit

UPDATE Book SET available\_copies = 5 WHERE book\_id = 2;

**❌ DELETE data**

sql

CopyEdit

DELETE FROM Loan WHERE loan\_id = 4;

**🎯 FILTERING**

sql

CopyEdit

SELECT \* FROM Loan WHERE return\_date IS NULL;

➡️ Shows books currently on loan.

**🔄 4. RELATIONSHIPS / ER MODEL (Quick Recap)**

| **Table** | **Related To** | **Type** |
| --- | --- | --- |
| Book | Author | Many-to-One |
| Loan | Book, Student | Many-to-One |
| Student | Loan | One-to-Many |
| Librarian | None | Independent |

**💪 5. PRACTICE TIPS**

✅ Create this database in your MySQL or SQLite environment.  
✅ Try inserting your own books/students.  
✅ Write queries for:

* Students who borrowed Fantasy books
* Books with less than 2 copies
* Authors who have more than 1 book (use GROUP BY)
* Find overdue books (return\_date < CURRENT\_DATE)