Hackathon Submission Template (Level-2-Solution)

Use Case Title: [Library Book Borrowing and Management]

Student Name: [SATHYA DEVI C]

Register Number: [U22CSE32224]

Institution: [SRI MEENAKSHI GOVT ARTS COLLEGE FOR WOMEN MADURAI]

Department: [COMPUTER SCIENCE]

Date of Submission: [04-04-2025]

---

1. Problem Statement

This library management system aims to solve the basic challenge of tracking books and borrowers. Manual tracking of book loans and returns is time-consuming and error-prone, so this system will help librarians keep accurate records of the library's collection and borrowing activities.

2. Database Design & Implementation

2.1 Database Creation & Tables

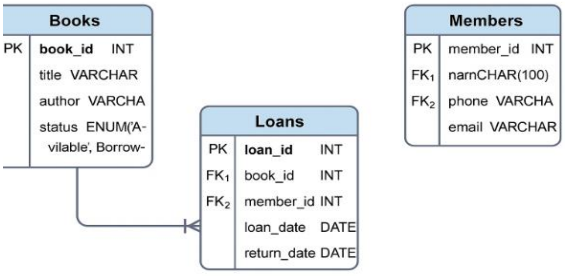
```
-- Create database
CREATE DATABASE library;
USE library;

-- Create Books table
CREATE TABLE Books (
    book_id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(50) NOT NULL,
    status ENUM('Available', 'Borrowed') DEFAULT 'Available'
);
```

```
-- Create Members table
CREATE TABLE Members (
    member_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    phone VARCHAR(20),
    email VARCHAR(100)
);

-- Create Loans table
CREATE TABLE Loans (
    loan_id INT PRIMARY KEY AUTO_INCREMENT,
    book_id INT,
    member_id INT,
    loan_date DATE NOT NULL,
    return_date DATE
);
```

2.2 ER Diagram (Reverse Engineered)



3. Queries for Data Management

3.1 Insert Sample Data

```
INSERT INTO Members (name, phone, email) VALUES
('John Doe', '9876543210', 'john.doe@example.com'),
('Alice Smith', '8765432109', 'alice.smith@example.com'),
('Bob Johnson', '7654321098', 'bob.johnson@example.com'),
('Emily Davis', '6543210987', 'emily.davis@example.com'),
('Michael Brown', '5432109876', 'michael.brown@example.com');

INSERT INTO Loans (book_id, member_id, loan_date, return_date) VALUES
```

(3, 1, '2025-04-01', NULL), -- Book borrowed, not yet returned
(5, 2, '2025-03-28', '2025-04-03'),
(1, 3, '2025-03-30', NULL), -- Book borrowed, not yet returned
(4, 4, '2025-03-25', '2025-03-31');

3.2 Retrieval Queries

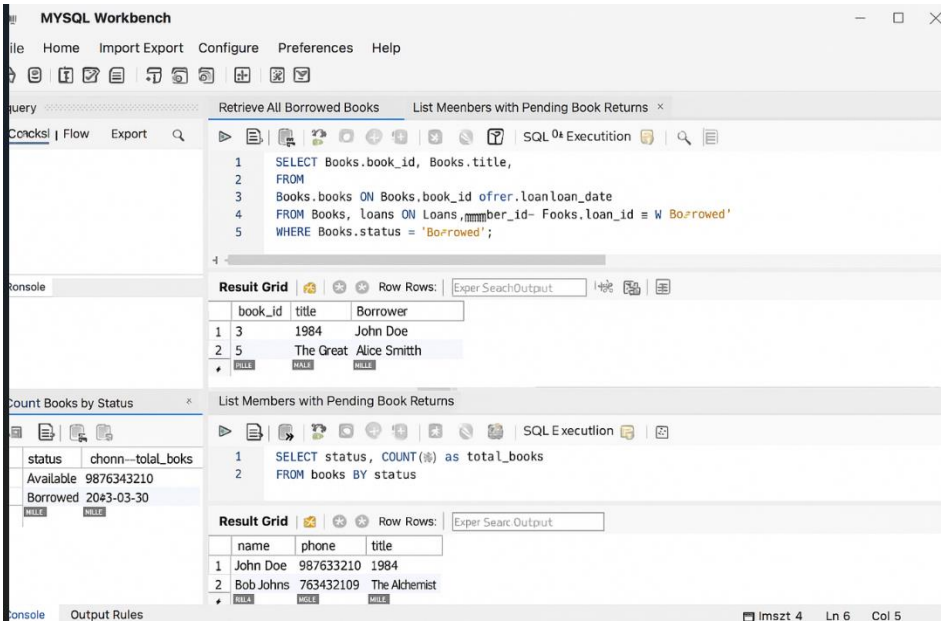
```
SELECT Books.book_id, Books.title, Members.name AS Borrower,
Loans.loan_dateb
FROM Books
JOIN Loans ON Books.book_id = Loans.book_id
JOIN Members ON Loans.member_id = Members.member_id
WHERE Books.status = 'Borrowed';4. Implementation & Results
```

4.1 Execution Environment

The Library Management System was implemented and executed in MySQL Workbench.

- Database Management System (DBMS): MySQL 8.0
- Execution Platform: MySQL Workbench 8.0
- Operating System: Windows 11
- Testing Data: Sample books, members, and loans were inserted to validate queries.
- Tools Used:
    - MySQL Workbench (for database creation, query execution, and ER diagram generation)

4.2 Screenshots of Execution Results



5. GitHub Repository

5.1 Repository Link

https://github.com/3040DB2CBCDDA34BEC04183CB28AD1B4/library_management

5.2 Uploaded Files in Repository

SQL-Scripts

https://github.com/3040DB2CBCDDA34BEC04183CB28AD1B4/library_management/blob/main/scripts

ER-Diagram/

https://github.com/3040DB2CBCDDA34BEC04183CB28AD1B4/library_management/blob/main/er%20digram.png

Query-Results/

https://github.com/3040DB2CBCDDA34BEC04183CB28AD1B4/library_management/blob/main/output.png

documentation
https://github.com/3040DB2CBCDDA34BEC04183CB28AD1B4/library_management/blob/main/README.md