Draft Assignment: Library Management Database

Mahmoud Hussein Shee

C110156202

University of the People

CS 2203-01 - AY2026-T1

Stella Chibuike-Ezike (Instructor)

Date: Sep 25, 2025

Introduction

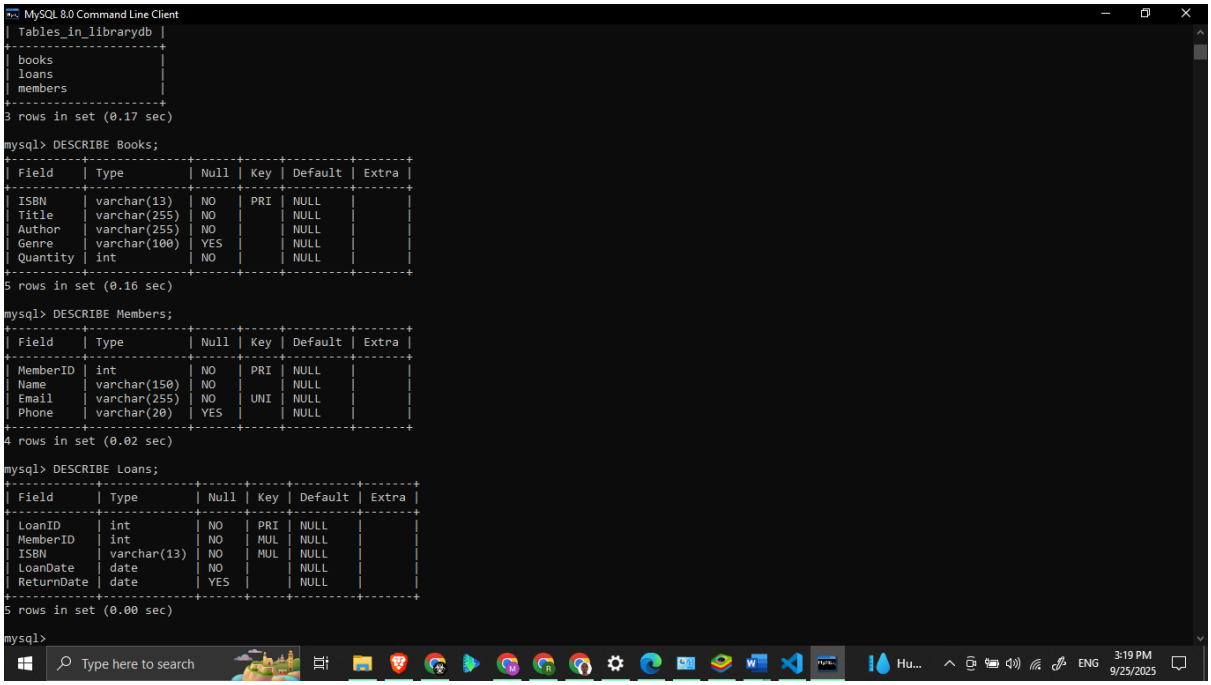
Efficient library management relies on robust database systems that maintain accuracy, integrity, and ease of data retrieval. Structured Query Language (SQL) provides constructs for designing, implementing, and manipulating relational databases effectively. This assignment demonstrates the creation of a normalized library database schema with three interconnected entities: **Books**, **Members**, and **Loans**. It also implements SQL queries to insert, update, delete, and retrieve information, ensuring both functionality and integrity of the system.

Database Schema Creation

The database schema includes the following entities and attributes:

```
-- Creating Books table
CREATE TABLE Books (
    ISBN VARCHAR(13) PRIMARY KEY,
```

```
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255) NOT NULL,
    Genre VARCHAR(100),
    Quantity INT NOT NULL CHECK (Quantity >= 0)
);

-- Creating Members table
CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    Name VARCHAR(150) NOT NULL,
    Email VARCHAR(255) UNIQUE NOT NULL,
    Phone VARCHAR(20)
);

-- Creating Loans table
CREATE TABLE Loans (
    LoanID INT PRIMARY KEY,
    MemberID INT NOT NULL,
    ISBN VARCHAR(13) NOT NULL,
    LoanDate DATE NOT NULL,
    ReturnDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (ISBN) REFERENCES Books(ISBN)
);
```



Justification of Schema Design

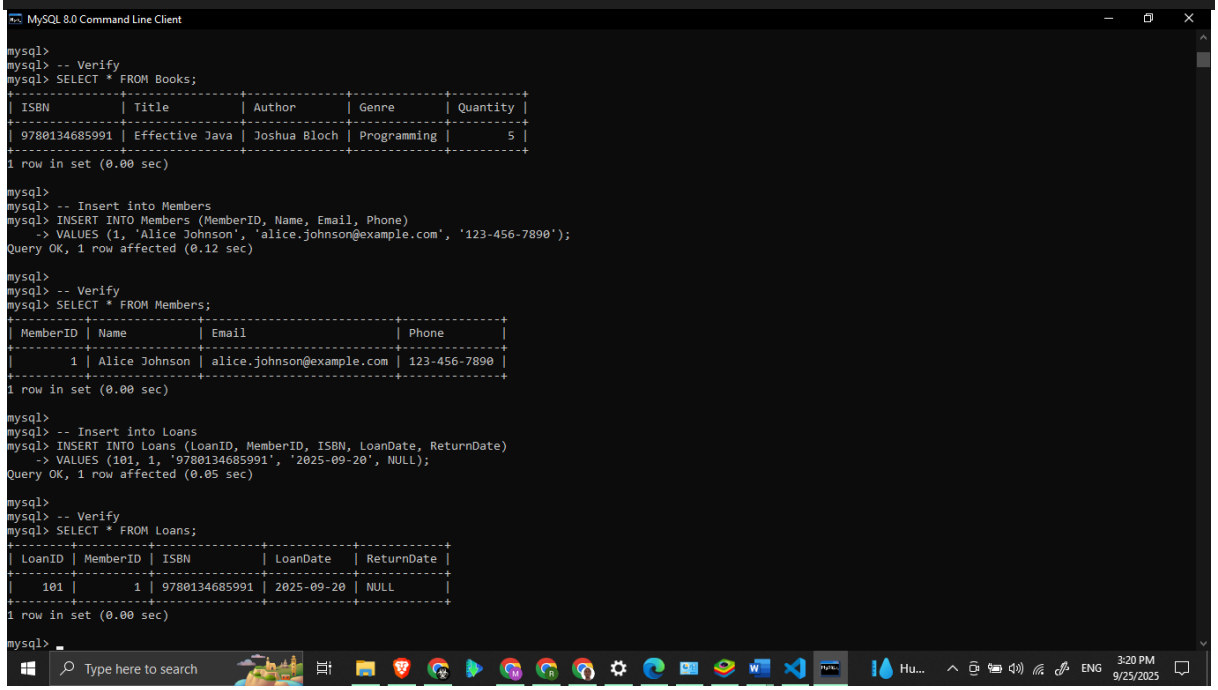
- **Books** table uses VARCHAR(13) for ISBN since international ISBN codes are 13 characters.
- **Members** table enforces uniqueness in Email for integrity.
- **Loans** table connects members and books through **foreign keys**, ensuring relational consistency.
- Data types are chosen for efficiency: INT for numerical attributes, VARCHAR for text, and DATE for loan records.

This structure adheres to **third normal form (3NF)**, eliminating redundancy and supporting efficient queries (Coronel & Morris, 2015).

SQL Queries with Documentation

1. Insert Records

```
-- Inserting sample books

INSERT INTO Books (ISBN, Title, Author, Genre, Quantity)

VALUES ('9780134685991', 'Effective Java', 'Joshua Bloch', 'Programming', 5);



-- Inserting sample members

INSERT INTO Members (MemberID, Name, Email, Phone)

VALUES (1, 'Alice Johnson', 'alice.johnson@example.com', '123-456-7890');



-- Inserting sample loans

INSERT INTO Loans (LoanID, MemberID, ISBN, LoanDate, ReturnDate)

VALUES (101, 1, '9780134685991', '2025-09-20', NULL);
```



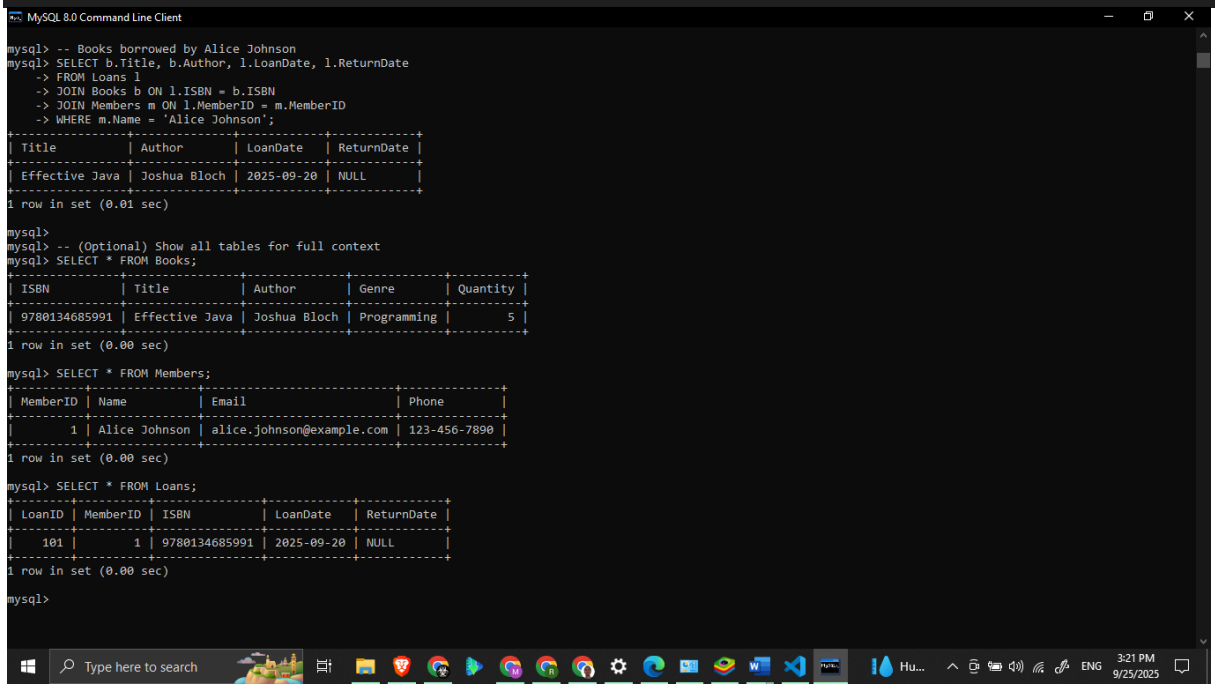
Comments:

- Records demonstrate population of each entity.
- NULL in ReturnDate indicates the book is still on loan.

---

2. Retrieve Books Borrowed by a Specific Member

```
-- Retrieving all books borrowed by a member

SELECT b.Title, b.Author, l.LoanDate, l.ReturnDate

FROM Loans l

JOIN Books b ON l.ISBN = b.ISBN

JOIN Members m ON l.MemberID = m.MemberID

WHERE m.Name = 'Alice Johnson';
```



Comments:

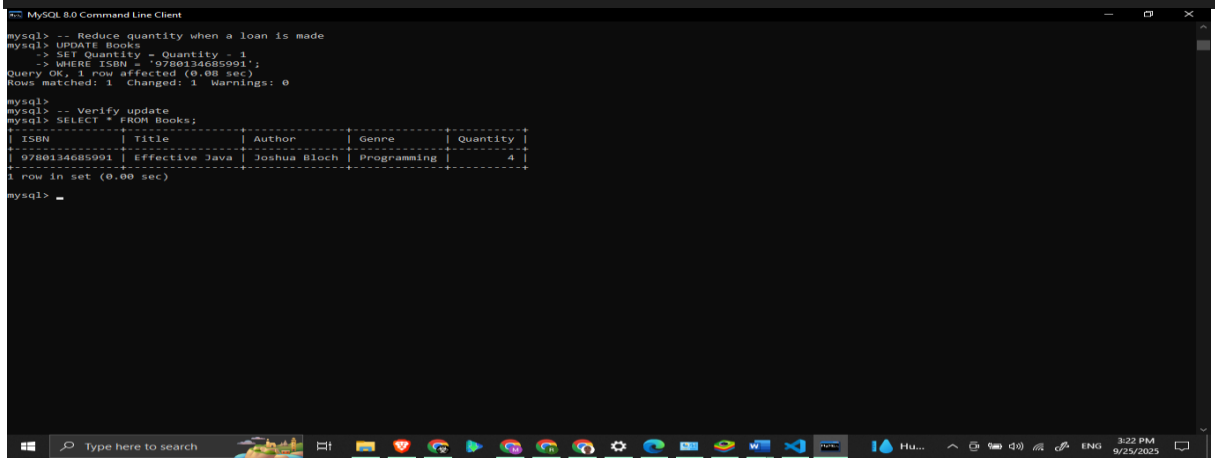
- JOIN connects all three tables.
- WHERE filters results to a specific member.

---

3. Update Book Quantity

```
-- Updating the number of copies of a book
```

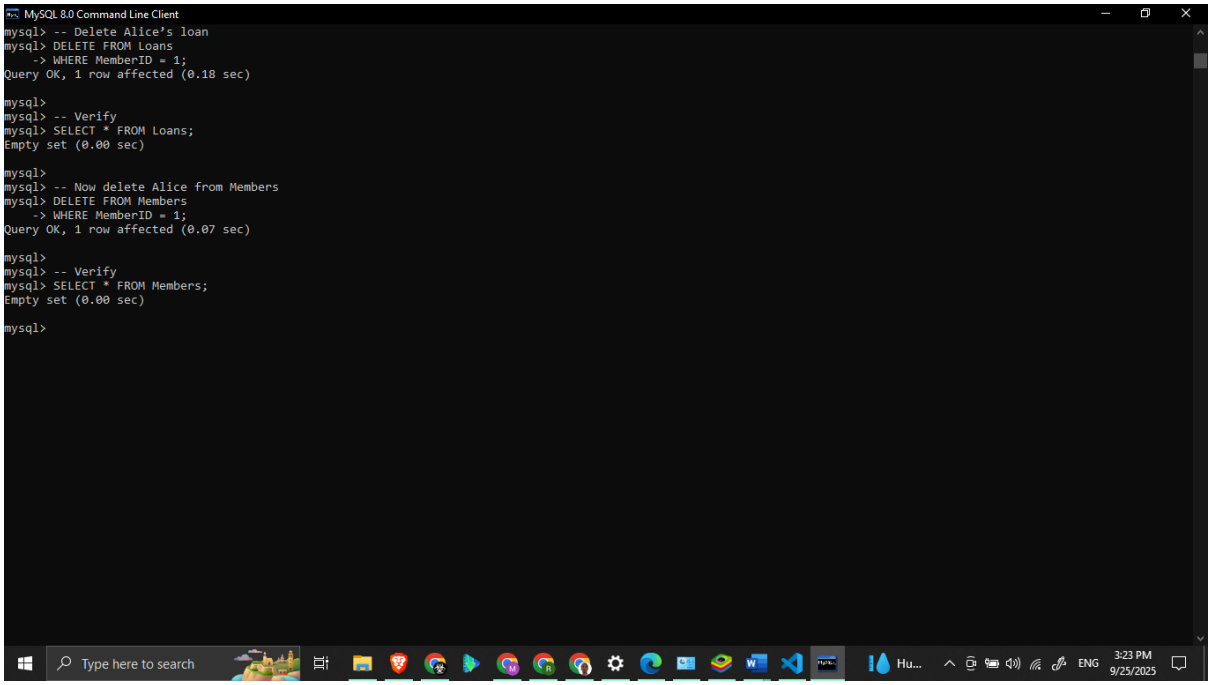
```
UPDATE Books

SET Quantity = Quantity - 1

WHERE ISBN = '9780134685991';
```



Comments:

- Decreases book quantity when a loan is made.
- CHECK constraint prevents negative values.

```
4. Delete a Member Record

-- Deleting a member record

DELETE FROM Members

WHERE MemberID = 1;
```



Comments:

- Removes a member entirely.
- FOREIGN KEY constraints ensure integrity; loans must be handled first.

Overview of Database Schema

This schema ensures smooth operation of the library by linking members to books through loans, while preventing anomalies such as duplicate records or inconsistent data. Proper use of SQL data types and constraints enhances **data integrity, consistency, and efficiency** (Elmasri & Navathe, 2016). The queries demonstrate fundamental operations: adding, retrieving, updating, and deleting records.

Conclusion

The developed schema and SQL queries illustrate practical application of relational database principles in managing a library system. By structuring entities clearly and applying relational constraints, the system ensures reliable data handling. The use of SQL queries further enables librarians to efficiently manage everyday operations, demonstrating the flexibility and power of database systems (Harrington, J. L. (2016)).

Discussion Question

If this library system were to be scaled to handle **thousands of members and books**, what indexing strategies would improve query performance, particularly for frequently searched attributes like ISBN and MemberID?

References

Coronel, C., & Morris, S. (2015). *Database systems: Design, implementation, & management* (11th ed.). Cengage Learning.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson.

Harrington, J. L. (2016). *Relational database design and implementation* (4th ed.). Morgan Kaufmann.

Peterson, R. (2023, December 16). SQL cheat sheet with commands & description (2024). GURU99. https://www.guru99.com/sql-cheat-sheet.html

Ramakrishnan, R., & Gehrke, J. (2020). *Database management systems* (3rd ed.). McGraw-Hill.

Vidhya, V., Jeyaram, G., & Ishwarya, K. (2016). *Database management systems*. Alpha Science International.