DATABASE MANAGEMENT SYSTEM - CSA0593

ASSIGNMENT 4

B.LAKSHMI ANJALI

192311344

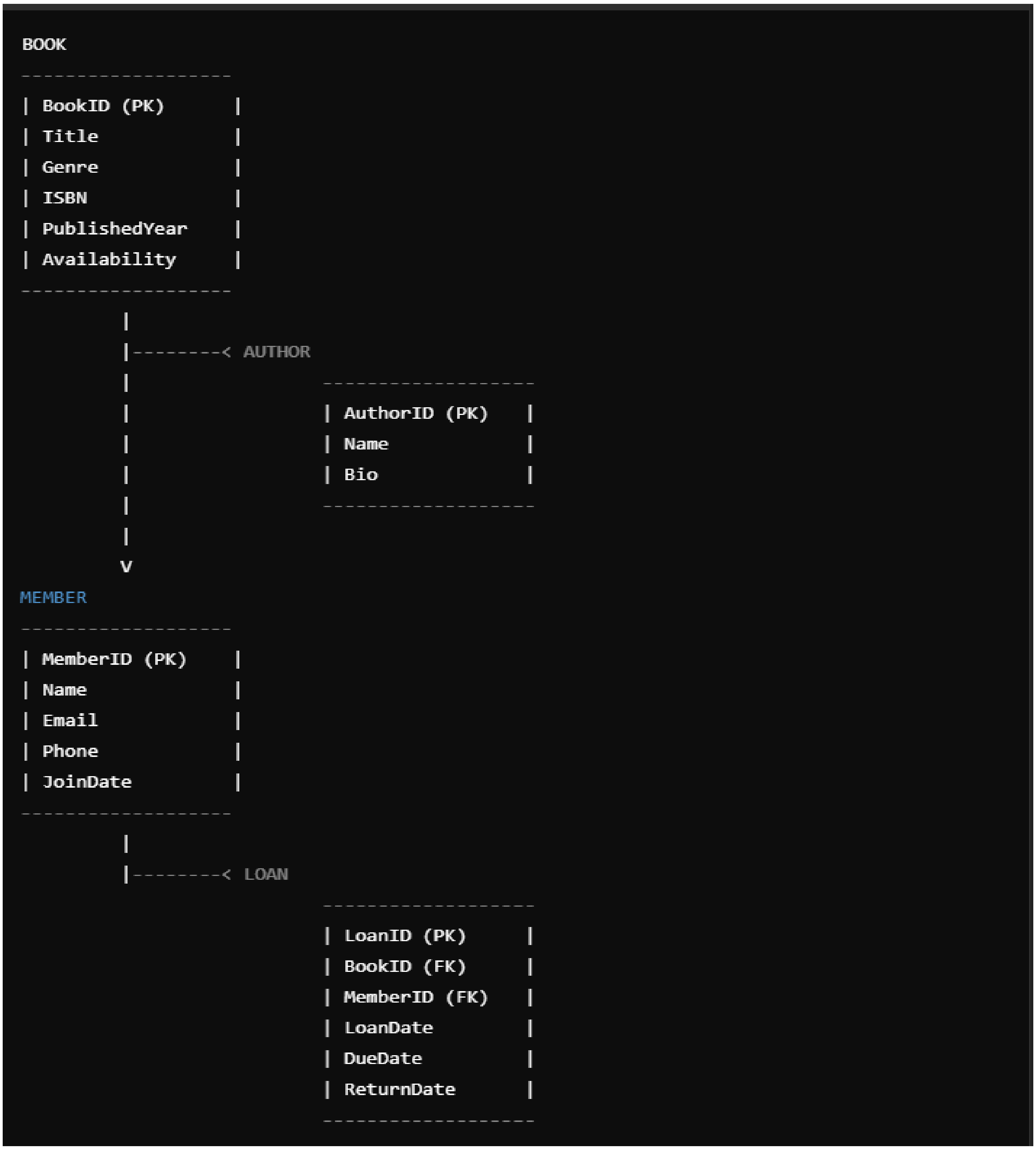
QUESTION:

Model tables for books, authors, members, and loans.

- **Write stored procedures for managing book loans and updating member borrowing history.**
- **Implement triggers to update book availability status when books are borrowed or returned.**
- **Write SQL queries to generate reports on popular books and overdue loans.**

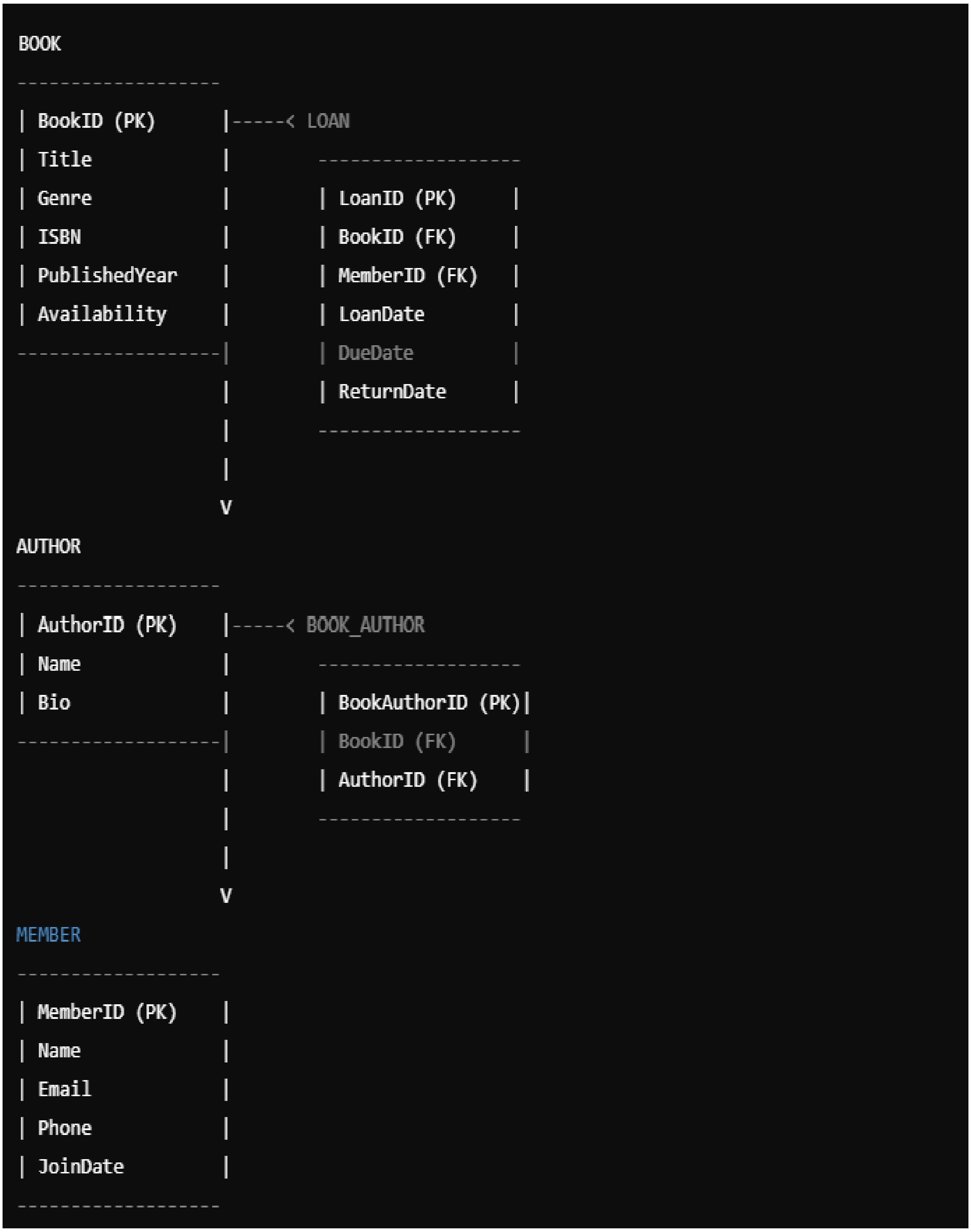
ANSWER:

CONCEPTUAL E.R.DIAGRAM:



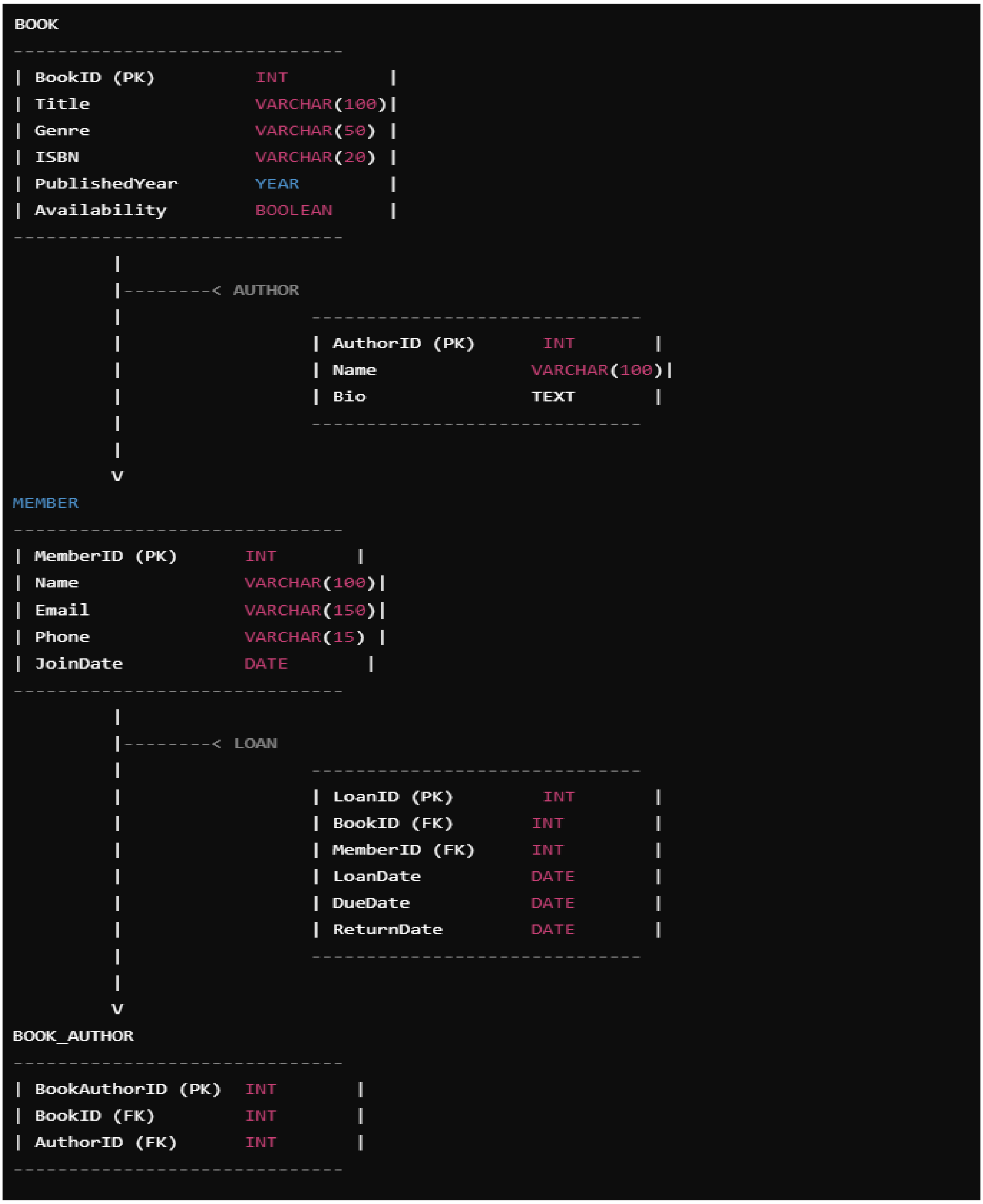
```
BOOK
-------------------
| BookID (PK)      |
| Title            |
| Genre            |
| ISBN             |
| PublishedYear    |
| Availability     |
-------------------
         |
         |--------< AUTHOR
         |                 -------------------
         |                 | AuthorID (PK)    |
         |                 | Name             |
         |                 | Bio              |
         |                 -------------------
         |
         v
MEMBER
-------------------
| MemberID (PK)    |
| Name             |
| Email            |
| Phone            |
| JoinDate         |
-------------------
         |
         |--------< LOAN
                           -------------------
                           | LoanID (PK)      |
                           | BookID (FK)      |
                           | MemberID (FK)    |
                           | LoanDate         |
                           | DueDate          |
                           | ReturnDate       |
                           -------------------
```

LOGICAL E.R DIAGRAM:



```
BOOK
-------------------
| BookID (PK)       |-----< LOAN
| Title             |        -------------------
| Genre             |        | LoanID (PK)      |
| ISBN              |        | BookID (FK)      |
| PublishedYear     |        | MemberID (FK)    |
| Availability      |        | LoanDate         |
-------------------|        | DueDate          |
                    |        | ReturnDate       |
                    |        -------------------
                    |
                    V
AUTHOR
-------------------
| AuthorID (PK)     |-----< BOOK_AUTHOR
| Name              |        -------------------
| Bio               |        | BookAuthorID (PK)|
-------------------|        | BookID (FK)      |
                    |        | AuthorID (FK)    |
                    |        -------------------
                    |
                    V
MEMBER
-------------------
| MemberID (PK)     |
| Name              |
| Email             |
| Phone             |
| JoinDate          |
-------------------
```

PHYSICAL E.R.DIAGRAM:



```
BOOK
------------------------------
| BookID (PK)          INT         |
| Title                VARCHAR(100)|
| Genre                VARCHAR(50) |
| ISBN                 VARCHAR(20) |
| PublishedYear        YEAR        |
| Availability         BOOLEAN     |
------------------------------
         |
         |--------< AUTHOR
         |
         |                 ------------------------------
         |                 | AuthorID (PK)      INT         |
         |                 | Name               VARCHAR(100)|
         |                 | Bio                TEXT        |
         |                 ------------------------------
         |
         V
MEMBER
------------------------------
| MemberID (PK)       INT         |
| Name                VARCHAR(100)|
| Email               VARCHAR(150)|
| Phone               VARCHAR(15) |
| JoinDate            DATE         |
------------------------------
         |
         |--------< LOAN
         |
         |                 ------------------------------
         |                 | LoanID (PK)         INT         |
         |                 | BookID (FK)        INT         |
         |                 | MemberID (FK)      INT         |
         |                 | LoanDate           DATE        |
         |                 | DueDate            DATE        |
         |                 | ReturnDate         DATE        |
         |                 ------------------------------
         |
         |
         V
BOOK_AUTHOR
------------------------------
| BookAuthorID (PK)  INT         |
| BookID (FK)        INT         |
| AuthorID (FK)      INT         |
------------------------------
```

MYSQL STATEMENTS:

```
mysql
CREATE DATABASE LibraryManagement;

USE LibraryManagement;

CREATE TABLE Authors (
  AuthorID INT AUTO_INCREMENT PRIMARY KEY,
  AuthorName VARCHAR(100)
);

CREATE TABLE Books (
  BookID INT AUTO_INCREMENT PRIMARY KEY,
  Title VARCHAR(100),
  AuthorID INT,
  Availability VARCHAR(20),
  FOREIGN KEY (AuthorID) REFERENCES
Authors(AuthorID)
```

```
);

CREATE TABLE Members (
  MemberID INT AUTO_INCREMENT PRIMARY KEY,
  MemberName VARCHAR(100),
  Email VARCHAR(100),
  Phone VARCHAR(20)
);

CREATE TABLE Loans (
  LoanID INT AUTO_INCREMENT PRIMARY KEY,
  BookID INT,
  MemberID INT,
  LoanDate DATE,
  ReturnDate DATE,
  Status VARCHAR(20),
  FOREIGN KEY (BookID) REFERENCES
Books(BookID),
  FOREIGN KEY (MemberID) REFERENCES
Members(MemberID)
);
```

Stored Procedures:

```
mysql
DELIMITER //

CREATE PROCEDURE sp_BorrowBook(
  IN bookID INT,
  IN memberID INT,
  IN loanDate DATE,
  IN returnDate DATE
)
BEGIN
```

```
  INSERT INTO Loans (BookID, MemberID,
LoanDate, ReturnDate, Status)
  VALUES (bookID, memberID, loanDate,
returnDate, 'Borrowed');

  UPDATE Books
  SET Availability = 'Unavailable'
  WHERE BookID = bookID;
END //

CREATE PROCEDURE sp_ReturnBook(
  IN loanID INT
)
BEGIN
  UPDATE Loans
  SET Status = 'Returned'
  WHERE LoanID = loanID;

  UPDATE Books
  SET Availability = 'Available'
  WHERE BookID = (SELECT BookID FROM Loans
WHERE LoanID = loanID);
END //

DELIMITER;
```

Triggers:

```
mysql
DELIMITER //

CREATE TRIGGER tr_UpdateBookAvailability
AFTER INSERT ON Loans
FOR EACH ROW
BEGIN
  UPDATE Books
  SET Availability = 'Unavailable'
  WHERE BookID = NEW.BookID;
END //

CREATE TRIGGER
tr_UpdateBookAvailabilityOnReturn
AFTER UPDATE ON Loans
FOR EACH ROW
BEGIN
  IF NEW.Status = 'Returned' THEN
    UPDATE Books
    SET Availability = 'Available'
    WHERE BookID = NEW.BookID;
  END IF;
END //

DELIMITER;
```

SQL Queries:

```
mysql
-- Popular Books
SELECT
  Title,
  COUNT(*) AS TotalLoans
FROM
  Books
  JOIN Loans ON Books.BookID = Loans.BookID
GROUP BY
  Title
ORDER BY
  TotalLoans DESC;

-- Overdue Loans
SELECT
  MemberName,
  Title,
  LoanDate,
  ReturnDate
FROM
  Members
  JOIN Loans ON Members.MemberID =
Loans.MemberID
  JOIN Books ON Loans.BookID = Books.BookID
WHERE
  ReturnDate < CURDATE();
```

Conclusion:

This database design provides a comprehensive foundation for managing books, authors, members, and loans. The stored procedures simplify book borrowing and returning, while the triggers ensure data consistency and accuracy. The SQL queries enable reporting on popular books and overdue loans.