Create and Manage a Library Database System

Database Script:

1. Create the Database-

```
CREATE DATABASE LibraryDB;
USE LibraryDB;
```

2. Create Tables-

Authors Table-

```
CREATE TABLE Authors (
AuthorID INT AUTO_INCREMENT PRIMARY KEY,
Name VARCHAR(100) NOT NULL
 );
```

Books Table-

```
CREATE TABLE Books (
BookID INT AUTO_INCREMENT PRIMARY KEY,
 Title VARCHAR(200) NOT NULL,
 AuthorID INT NOT NULL,
 PublicationYear INT,
 FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID) ON DELETE CASCADE
 );
```

Members Table-

```
CREATE TABLE Members (
MemberID INT AUTO_INCREMENT PRIMARY KEY,
Name VARCHAR(100) NOT NULL,
MembershipDate DATE NOT NULL
 );
```

Loans Table-

```
CREATE TABLE Loans (
LoanID INT AUTO_INCREMENT PRIMARY KEY,
BookID INT NOT NULL,
MemberID INT NOT NULL,
LoanDate DATE NOT NULL,
ReturnDate DATE,
FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE,
FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE    CASCADE
);
```

Book Audit-

```
CREATE TABLE Books_Audit (
  AuditID INT AUTO_INCREMENT PRIMARY KEY,
  ActionType VARCHAR(10),
  BookID INT,
  Title VARCHAR(200),
  AuthorID INT,
  PublicationYear INT,
  ChangedBy VARCHAR(50),
  ActionDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

3. Insert Data-

Authors-

```
INSERT INTO Authors (Name) VALUES
('J.K. Rowling'),
('George R.R. Martin'),
('J.R.R. Tolkien');
```

Books-

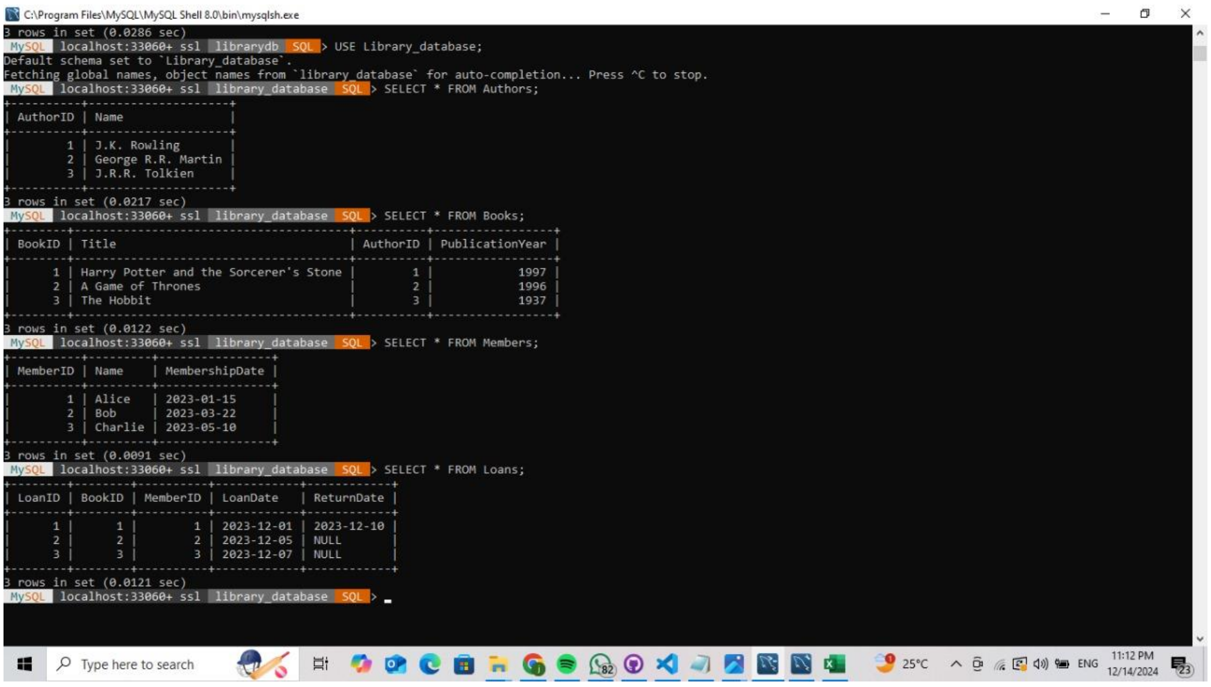
```
INSERT INTO Books (Title, AuthorID, PublicationYear) VALUES
('Harry Potter and the Sorcerer\'s Stone', 1, 1997),
('A Game of Thrones', 2, 1996),
('The Hobbit', 3, 1937);
```

Members-

```
INSERT INTO Members (Name, MembershipDate) VALUES
('Alice', '2023-01-15'),
('Bob', '2023-03-22'),
('Charlie', '2023-05-10');
```

Loans-

```
INSERT INTO Loans (BookID, MemberID, LoanDate, ReturnDate) VALUES
(1, 1, '2023-12-01', '2023-12-10'),
(2, 2, '2023-12-05', NULL),
(3, 3, '2023-12-07', NULL);
```



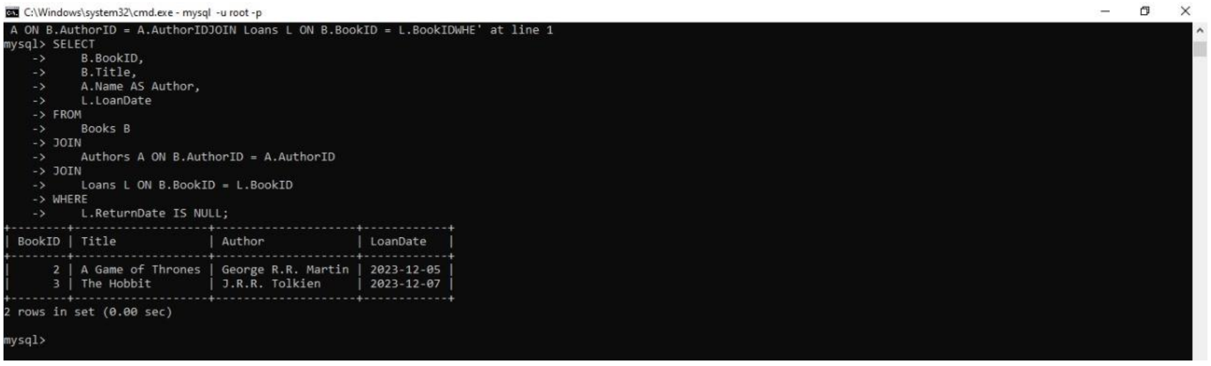
C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe

```
3 rows in set (0.0286 sec)
 MySQL  localhost:33060+ ssl  librarydb  SQL > USE Library_database;
Default schema set to `Library_database`.
Fetching global names, object names from `library_database` for auto-completion... Press ^C to stop.
 MySQL  localhost:33060+ ssl  library_database  SQL > SELECT * FROM Authors;
+----------+------------------+
| AuthorID | Name             |
+----------+------------------+
|        1 | J.K. Rowling     |
|        2 | George R.R. Martin |
|        3 | J.R.R. Tolkien   |
+----------+------------------+
3 rows in set (0.0217 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > SELECT * FROM Books;
+--------+----------------------------------------+----------+-----------------+
| BookID | Title                                  | AuthorID | PublicationYear |
+--------+----------------------------------------+----------+-----------------+
|      1 | Harry Potter and the Sorcerer's Stone |        1 |            1997 |
|      2 | A Game of Thrones                      |        2 |            1996 |
|      3 | The Hobbit                             |        3 |            1937 |
+--------+----------------------------------------+----------+-----------------+
3 rows in set (0.0122 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > SELECT * FROM Members;
+----------+---------+----------------+
| MemberID | Name    | MembershipDate |
+----------+---------+----------------+
|        1 | Alice   | 2023-01-15     |
|        2 | Bob     | 2023-03-22     |
|        3 | Charlie | 2023-05-10     |
+----------+---------+----------------+
3 rows in set (0.0091 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > SELECT * FROM Loans;
+--------+--------+----------+------------+------------+
| LoanID | BookID | MemberID | LoanDate   | ReturnDate |
+--------+--------+----------+------------+------------+
|      1 |      1 |        1 | 2023-12-01 | 2023-12-10 |
|      2 |      2 |        2 | 2023-12-05 | NULL       |
|      3 |      3 |        3 | 2023-12-07 | NULL       |
+--------+--------+----------+------------+------------+
3 rows in set (0.0121 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > _
```

SQL Query Requirements

1. List all books currently on loan (i.e., ReturnDate is NULL):

```
SELECT B.BookID, B.Title, A.Name AS Author, L.LoanDate
FROM Books B
JOIN Authors A ON B.AuthorID = A.AuthorID
JOIN Loans L ON B.BookID = L.BookID
WHERE L.ReturnDate IS NULL;
```

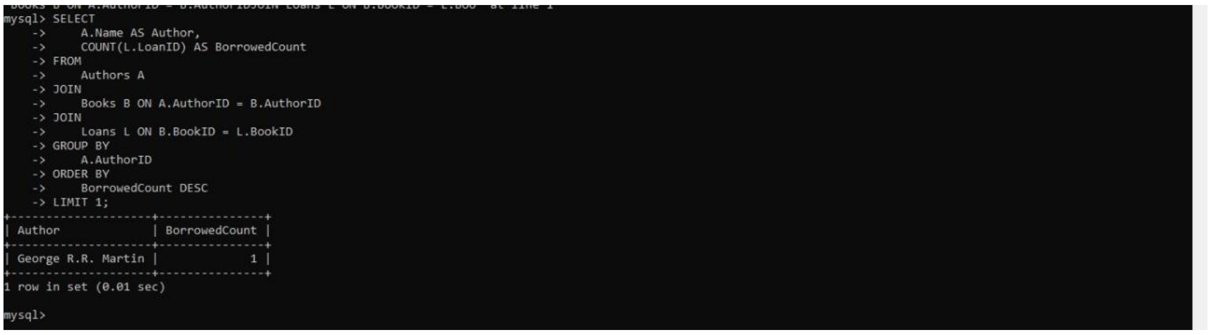


C:\Windows\system32\cmd.exe - mysql -u root -p

```
 A ON B.AuthorID = A.AuthorIDJOIN Loans L ON B.BookID = L.BookIDWHE' at line 1
mysql> SELECT
    ->     B.BookID,
    ->     B.Title,
    ->     A.Name AS Author,
    ->     L.LoanDate
    -> FROM
    ->     Books B
    -> JOIN
    ->     Authors A ON B.AuthorID = A.AuthorID
    -> JOIN
    ->     Loans L ON B.BookID = L.BookID
    -> WHERE
    ->     L.ReturnDate IS NULL;
+--------+-------------------+--------------------+------------+
| BookID | Title             | Author             | LoanDate   |
+--------+-------------------+--------------------+------------+
|      2 | A Game of Thrones | George R.R. Martin | 2023-12-05 |
|      3 | The Hobbit        | J.R.R. Tolkien     | 2023-12-07 |
+--------+-------------------+--------------------+------------+
2 rows in set (0.00 sec)

mysql>
```

2. Find the most borrowed author (author whose books have been borrowed the most):

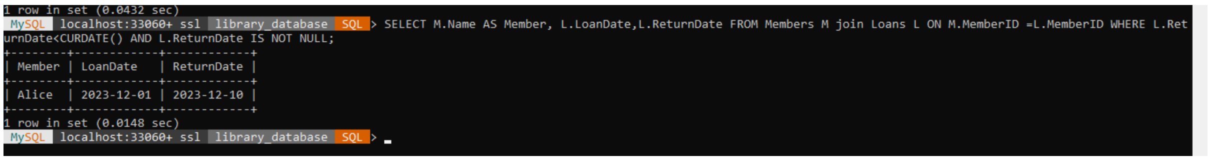
```
SELECT A.Name AS Author, COUNT(L.LoanID) AS BorrowedCount
FROM Authors A
JOIN Books B ON A.AuthorID = B.AuthorID
JOIN Loans L ON B.BookID = L.BookID
GROUP BY A.AuthorID
ORDER BY BorrowedCount DESC
 LIMIT 1;
```



```
Books B ON A.AuthorID = B.AuthorIDJOIN Loans L ON B.BookID = L.Boo' at line 1
mysql> SELECT
    ->     A.Name AS Author,
    ->     COUNT(L.LoanID) AS BorrowedCount
    -> FROM
    ->     Authors A
    -> JOIN
    ->     Books B ON A.AuthorID = B.AuthorID
    -> JOIN
    ->     Loans L ON B.BookID = L.BookID
    -> GROUP BY
    ->     A.AuthorID
    -> ORDER BY
    ->     BorrowedCount DESC
    -> LIMIT 1;
+--------------------+---------------+
| Author             | BorrowedCount |
+--------------------+---------------+
| George R.R. Martin |             1 |
+--------------------+---------------+
1 row in set (0.01 sec)

mysql>
```

3. Retrieve members with overdue books (based on ReturnDate being in the past):

```
SELECT M.Name AS Member, L.LoanDate, L.ReturnDate
FROM Members M
JOIN Loans L ON M.MemberID = L.MemberID
WHERE L.ReturnDate < CURDATE() AND L.ReturnDate IS NOT NULL;
```



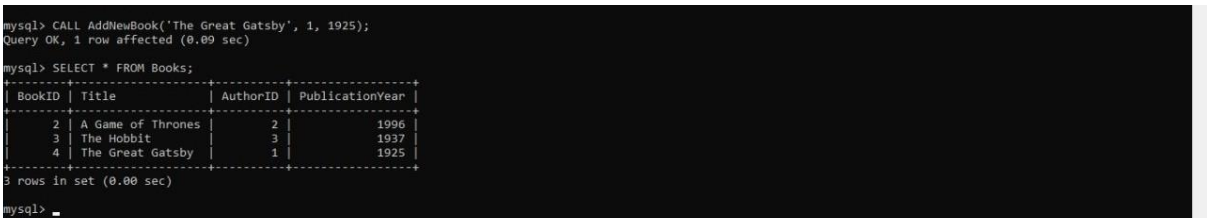
```
1 row in set (0.0432 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > SELECT M.Name AS Member, L.LoanDate,L.ReturnDate FROM Members M join Loans L ON M.MemberID =L.MemberID WHERE L.Ret
urnDate<CURDATE() AND L.ReturnDate IS NOT NULL;
+--------+------------+------------+
| Member | LoanDate   | ReturnDate |
+--------+------------+------------+
| Alice  | 2023-12-01 | 2023-12-10 |
+--------+------------+------------+
1 row in set (0.0148 sec)
 MySQL  localhost:33060+ ssl  library_database  SQL > _
```

Stored Procedures:

1. Adding new books:

```
DELIMITER $$
CREATE PROCEDURE AddNewBook(
 IN bookTitle VARCHAR (255),
 IN authorID INT,
 IN publicationYear INT
 )
 BEGIN
 INSERT INTO Books (Title, AuthorID, PublicationYear)
 VALUES (bookTitle, authorID, publicationYear);
 END$$
 DELIMITER;
```

Usage - CALL AddNewBook('The Great Gatsby', 1, 1925);

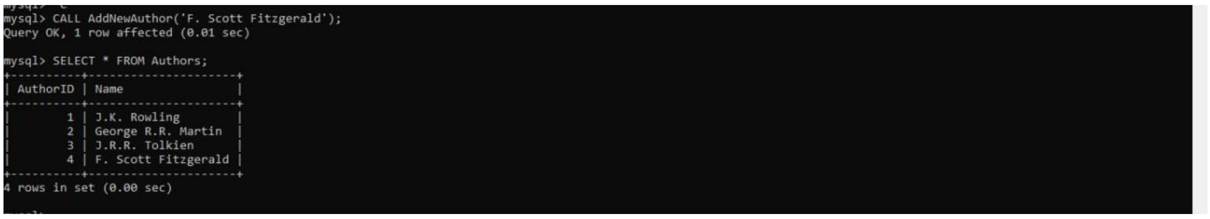


```
mysql> CALL AddNewBook('The Great Gatsby', 1, 1925);
Query OK, 1 row affected (0.09 sec)

mysql> SELECT * FROM Books;
+--------+-------------------+----------+-----------------+
| BookID | Title             | AuthorID | PublicationYear |
+--------+-------------------+----------+-----------------+
|      2 | A Game of Thrones |        2 |            1996 |
|      3 | The Hobbit        |        3 |            1937 |
|      4 | The Great Gatsby  |        1 |            1925 |
+--------+-------------------+----------+-----------------+
3 rows in set (0.00 sec)

mysql> _
```

2.Add a New Author

```
DELIMITER $$
CREATE PROCEDURE AddNewAuthor(IN authorName VARCHAR(255))
BEGIN
INSERT INTO Authors (Name) VALUES (authorName);
END$$
DELIMITER ;
```

Usage - CALL AddAuthor('F. Scott Fitzgerald');

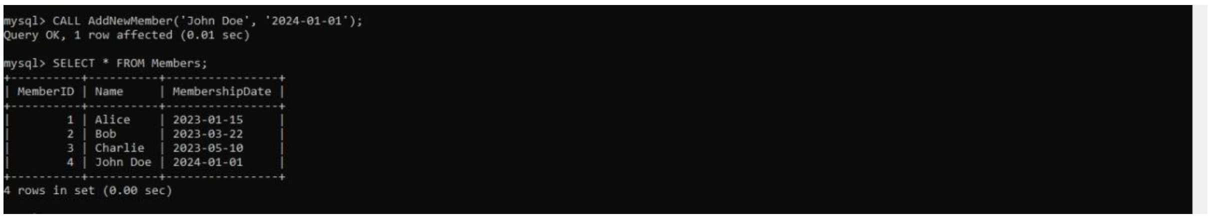


```
mysql> CALL AddNewAuthor('F. Scott Fitzgerald');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Authors;
+----------+---------------------+
| AuthorID | Name                |
+----------+---------------------+
|        1 | J.K. Rowling        |
|        2 | George R.R. Martin  |
|        3 | J.R.R. Tolkien      |
|        4 | F. Scott Fitzgerald |
+----------+---------------------+
4 rows in set (0.00 sec)
```

3.Add a New Member

```
DELIMITER $$
CREATE PROCEDURE AddNewMember(
 IN memberName VARCHAR(255),
 IN membershipDate DATE
 )
 BEGIN
 INSERT INTO Members (Name, MembershipDate)
  VALUES (memberName, membershipDate);
  END$$
  DELIMITER ;
```

Usage - CALL AddNewMember(‘john Doe’, ‘2024-01-01’);



```
mysql> CALL AddNewMember('John Doe', '2024-01-01');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Members;
+----------+----------+----------------+
| MemberID | Name     | MembershipDate |
+----------+----------+----------------+
|        1 | Alice    | 2023-01-15     |
|        2 | Bob      | 2023-03-22     |
|        3 | Charlie  | 2023-05-10     |
|        4 | John Doe | 2024-01-01     |
+----------+----------+----------------+
4 rows in set (0.00 sec)
```

4. Add loan

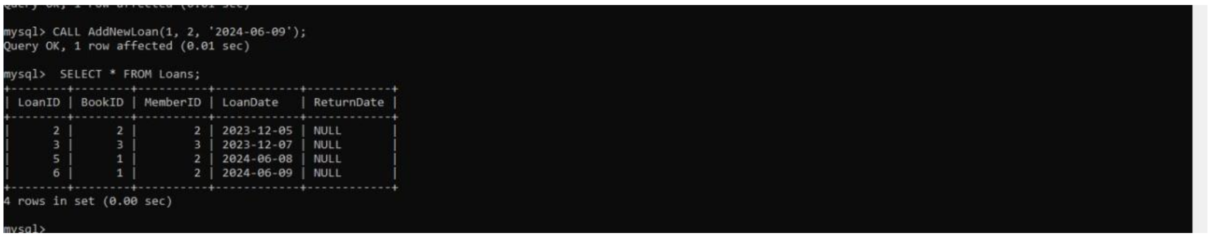
```
 DELIMITER $$
 CREATE PROCEDURE AddNewLoan(
 IN bookID INT,
 IN memberID INT,
 IN loanDate DATE
)
BEGIN
INSERT INTO Loans (BookID, MemberID, LoanDate)
```

VALUES (bookID, memberID, loanDate);

END$$

DELIMITER ;

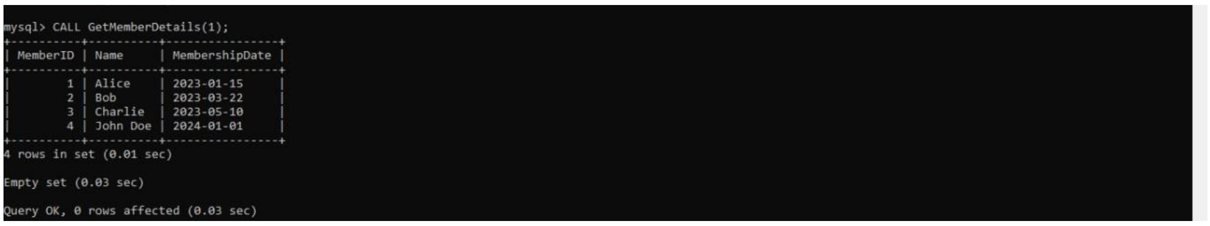
Usage - CALL AddNewLoan(1,2, ‘2024-06-09’);



```
mysql> CALL AddNewLoan(1, 2, '2024-06-09');
Query OK, 1 row affected (0.01 sec)

mysql>  SELECT * FROM Loans;
+--------+--------+----------+------------+------------+
| LoanID | BookID | MemberID | LoanDate   | ReturnDate |
+--------+--------+----------+------------+------------+
|      2 |      2 |        2 | 2023-12-05 | NULL       |
|      3 |      3 |        3 | 2023-12-07 | NULL       |
|      5 |      1 |        2 | 2024-06-08 | NULL       |
|      6 |      1 |        2 | 2024-06-09 | NULL       |
+--------+--------+----------+------------+------------+
4 rows in set (0.00 sec)

mysql>
```

4. Get Member Details

```
DELIMITER $$

CREATE PROCEDURE GetMemberDetails(IN memberID INT)

BEGIN

SELECT * FROM Members WHERE MemberID = memberID;

SELECT B.BookID, B.Title, L.LoanDate, L.ReturnDate

FROM Loans L

JOIN Books B ON L.BookID = B.BookID

WHERE L.MemberID = memberID;

 END$$

 DELIMITER ;
```

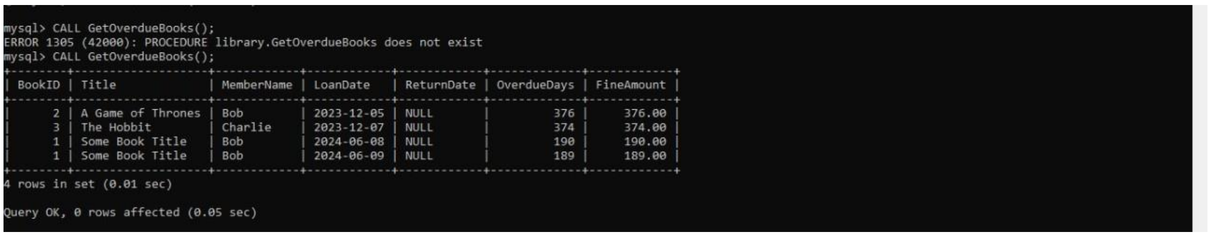


```
mysql> CALL GetMemberDetails(1);
+----------+----------+----------------+
| MemberID | Name     | MembershipDate |
+----------+----------+----------------+
|        1 | Alice    | 2023-01-15     |
|        2 | Bob      | 2023-03-22     |
|        3 | Charlie  | 2023-05-10     |
|        4 | John Doe | 2024-01-01     |
+----------+----------+----------------+
4 rows in set (0.01 sec)

Empty set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)
```

5. Get Overdue Books

```
DELIMITER $$

CREATE PROCEDURE GetOverdueBooks()
```

```
BEGIN

 -- Retrieve overdue books and calculate fines

SELECT B.BookID, B.Title, M.Name AS MemberName, L.LoanDate, L.ReturnDate,

DATEDIFF(CURDATE(), L.LoanDate) AS OverdueDays,

DATEDIFF(CURDATE(), L.LoanDate) * 1.00 AS FineAmount

FROM Loans L

JOIN Books B ON L.BookID = B.BookID

JOIN Members M ON L.MemberID = M.MemberID

WHERE L.ReturnDate IS NULL AND L.LoanDate < CURDATE();

END$$

 DELIMITER ;
```



```
mysql> CALL GetOverdueBooks();
ERROR 1305 (42000): PROCEDURE library.GetOverdueBooks does not exist
mysql> CALL GetOverdueBooks();
+--------+-------------------+------------+------------+------------+-------------+------------+
| BookID | Title             | MemberName | LoanDate   | ReturnDate | OverdueDays | FineAmount |
+--------+-------------------+------------+------------+------------+-------------+------------+
|      2 | A Game of Thrones | Bob        | 2023-12-05 | NULL       |         376 |     376.00 |
|      3 | The Hobbit        | Charlie    | 2023-12-07 | NULL       |         374 |     374.00 |
|      1 | Some Book Title   | Bob        | 2024-06-08 | NULL       |         190 |     190.00 |
|      1 | Some Book Title   | Bob        | 2024-06-09 | NULL       |         189 |     189.00 |
+--------+-------------------+------------+------------+------------+-------------+------------+
4 rows in set (0.01 sec)

Query OK, 0 rows affected (0.05 sec)
```

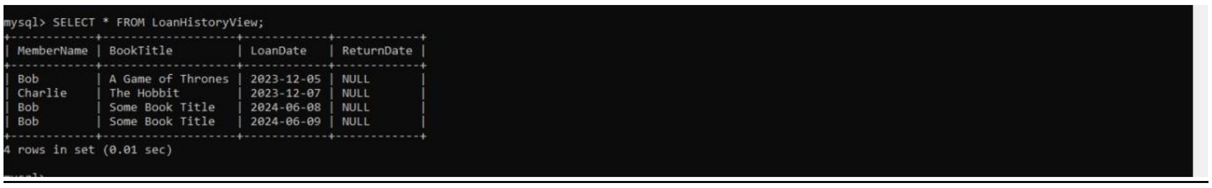
Views

1. Loan History View:

```
CREATE VIEW LoanHistoryView AS

SELECT

M.Name AS MemberName,

B.Title AS BookTitle,

L.LoanDate,

L.ReturnDate

FROM Loans L

JOIN Members M ON L.MemberID = M.MemberID
```

```sql
JOIN Books B ON L.BookID = B.BookID;
```

View- SELECT * FROM LoanHistoryView;

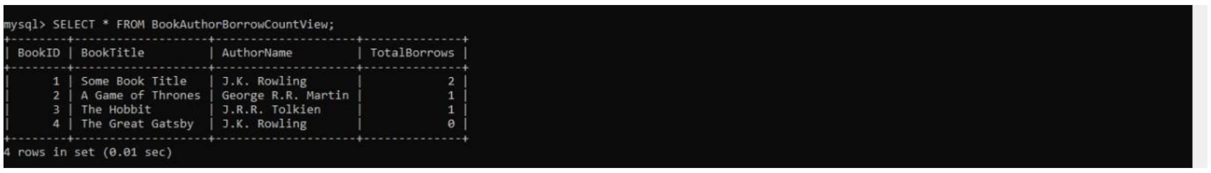


```
mysql> SELECT * FROM LoanHistoryView;
+------------+--------------------+------------+------------+
| MemberName | BookTitle          | LoanDate   | ReturnDate |
+------------+--------------------+------------+------------+
| Bob        | A Game of Thrones  | 2023-12-05 | NULL       |
| Charlie    | The Hobbit         | 2023-12-07 | NULL       |
| Bob        | Some Book Title    | 2024-06-08 | NULL       |
| Bob        | Some Book Title    | 2024-06-09 | NULL       |
+------------+--------------------+------------+------------+
4 rows in set (0.01 sec)
```

2. Book Author Borrow Count View

```sql
CREATE VIEW BookAuthorBorrowCountView AS
 SELECT
 B.BookID,
 B.Title AS BookTitle,
 A.Name AS AuthorName,
 COUNT(L.LoanID) AS TotalBorrows
  FROM Books B
  JOIN Authors A ON B.AuthorID = A.AuthorID
  LEFT JOIN Loans L ON B.BookID = L.BookID
  GROUP BY B.BookID, A.Name;
```

View - SELECT * FROM BookAuthorBorrowCountView;



```
mysql> SELECT * FROM BookAuthorBorrowCountView;
+--------+--------------------+--------------------+--------------+
| BookID | BookTitle          | AuthorName         | TotalBorrows |
+--------+--------------------+--------------------+--------------+
|      1 | Some Book Title    | J.K. Rowling       |            2 |
|      2 | A Game of Thrones  | George R.R. Martin |            1 |
|      3 | The Hobbit         | J.R.R. Tolkien     |            1 |
|      4 | The Great Gatsby   | J.K. Rowling       |            0 |
+--------+--------------------+--------------------+--------------+
4 rows in set (0.01 sec)
```

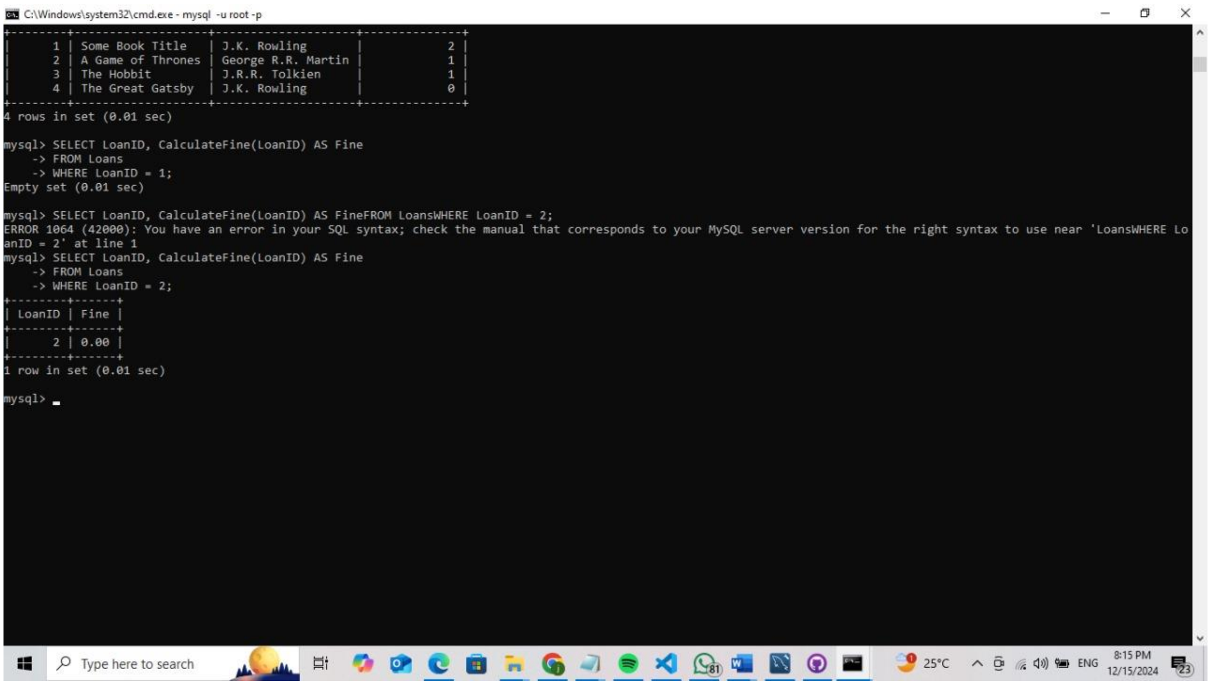
FUNCTIONS

1. Calculate Fine:

```
DELIMITER $$

CREATE FUNCTION CalculateFine(loanID INT)

RETURNS DECIMAL(10, 2)

DETERMINISTIC

 BEGIN

 DECLARE fine DECIMAL(10, 2);

 DECLARE returnDate DATE;

 DECLARE loanDate DATE;

 SELECT LoanDate, ReturnDate INTO loanDate, returnDate

  FROM Loans

   WHERE LoanID = loanID

LIMIT 1;

IF returnDate < CURDATE() THEN

SET fine = DATEDIFF(CURDATE(), returnDate) * 1.00;

ELSE

SET fine = 0;

END IF;

RETURN fine;

END$$

 DELIMITER ;
```

View -

```
SELECT LoanID, CalculateFine(LoanID) AS Fine

FROM Loans

 WHERE LoanID = 2;
```



```
C:\Windows\system32\cmd.exe - mysql  -u root -p
+--------+-------------------+-------------------+-------------+
|      1 | Some Book Title   | J.K. Rowling      |           2 |
|      2 | A Game of Thrones | George R.R. Martin |          1 |
|      3 | The Hobbit        | J.R.R. Tolkien    |           1 |
|      4 | The Great Gatsby  | J.K. Rowling      |           0 |
+--------+-------------------+-------------------+-------------+
4 rows in set (0.01 sec)

mysql> SELECT LoanID, CalculateFine(LoanID) AS Fine
    -> FROM Loans
    -> WHERE LoanID = 1;
Empty set (0.01 sec)

mysql> SELECT LoanID, CalculateFine(LoanID) AS FineFROM LoansWHERE LoanID = 2;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'LoansWHERE Lo
anID = 2' at line 1
mysql> SELECT LoanID, CalculateFine(LoanID) AS Fine
    -> FROM Loans
    -> WHERE LoanID = 2;
+--------+------+
| LoanID | Fine |
+--------+------+
|      2 | 0.00 |
+--------+------+
1 row in set (0.01 sec)

mysql> _
```

Triggers

1. After Update on Books:

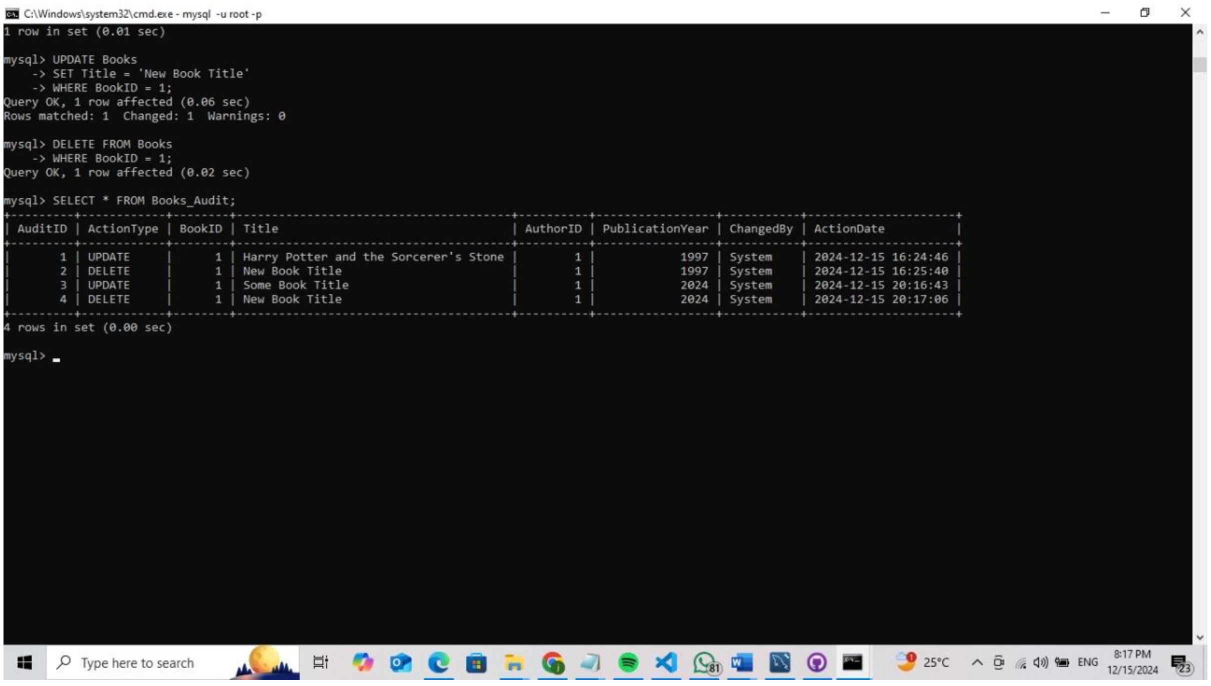
```
DELIMITER $$

CREATE TRIGGER after_books_update

AFTER UPDATE ON Books

FOR EACH ROW

BEGIN

INSERT INTO Books_Audit (ActionType, BookID, Title, AuthorID, PublicationYear, ChangedBy)

VALUES ('UPDATE', OLD.BookID, OLD.Title, OLD.AuthorID, OLD.PublicationYear, 'System');

END$$

DELIMITER ;
```

2. After Delete on Books:

```
DELIMITER $$

CREATE TRIGGER after_books_delete

CREATE TRIGGER after_books_delete

AFTER DELETE ON Books
```

```
FOR EACH ROW

BEGIN

INSERT INTO Books_Audit (ActionType, BookID, Title, AuthorID, PublicationYear, ChangedBy)

VALUES ('DELETE', OLD.BookID, OLD.Title, OLD.AuthorID, OLD.PublicationYear, 'System');

END$$

DELIMITER ;
```

```
UPDATE Books

SET Title = 'New Book Title'

WHERE BookID = 1;


DELETE FROM Books

WHERE BookID = 1;


SELECT * FROM Books_Audit;
```



```
C:\Windows\system32\cmd.exe - mysql  -u root -p
1 row in set (0.01 sec)

mysql> UPDATE Books
    -> SET Title = 'New Book Title'
    -> WHERE BookID = 1;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM Books
    -> WHERE BookID = 1;
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM Books_Audit;
+---------+------------+--------+----------------------------------------+----------+-----------------+-----------+---------------------+
| AuditID | ActionType | BookID | Title                                  | AuthorID | PublicationYear | ChangedBy | ActionDate          |
+---------+------------+--------+----------------------------------------+----------+-----------------+-----------+---------------------+
|       1 | UPDATE     |      1 | Harry Potter and the Sorcerer's Stone |        1 |            1997 | System    | 2024-12-15 16:24:46 |
|       2 | DELETE     |      1 | New Book Title                         |        1 |            1997 | System    | 2024-12-15 16:25:40 |
|       3 | UPDATE     |      1 | Some Book Title                        |        1 |            2024 | System    | 2024-12-15 20:16:43 |
|       4 | DELETE     |      1 | New Book Title                         |        1 |            2024 | System    | 2024-12-15 20:17:06 |
+---------+------------+--------+----------------------------------------+----------+-----------------+-----------+---------------------+
4 rows in set (0.00 sec)

mysql> _
```