HACKATHON SUBMISSION (LEVEL-2-SOLUTION)

Use Case Title: Library Management System.

Student Name: M. Kalaiyarasi.

Register Number: U22CSE32237.

Institution: Sri Meenakshi Govt Arts college for Women (Autonomous), Madurai. Department: Computer Science.

1. PROBLEM STATEMENT

Traditional library systems focus primarily on book lending and catalog management, which often leads to inefficiencies such as:

- ➢ Difficulty for users to find books matching their interests.
- ➢ Overwhelming book choices without personalized guidance.
- ➢ Lack of user engagement with available resources.
- ➢ No intelligent system to analyze reading patterns or preferences.

A **Library Management System** solves these issues by:

- ➢ Analyzing users reading history, preferences, and behavior.
- ➢ Recommending books based on genres, authors, and past borrowing trends.
- ➢ Helping users discover new books aligned with their interests.
- ➢ Enhancing user satisfaction and library engagement through data-driven suggestions.
- ➢ Supporting librarians with insights into popular books and reading trends.

2. DATABASE DESIGN & IMPLEMENTATION

2.1 DATABASE CREATION & TABLES:

Database creation:

```
CREATE DATABASE libraryDB;
```

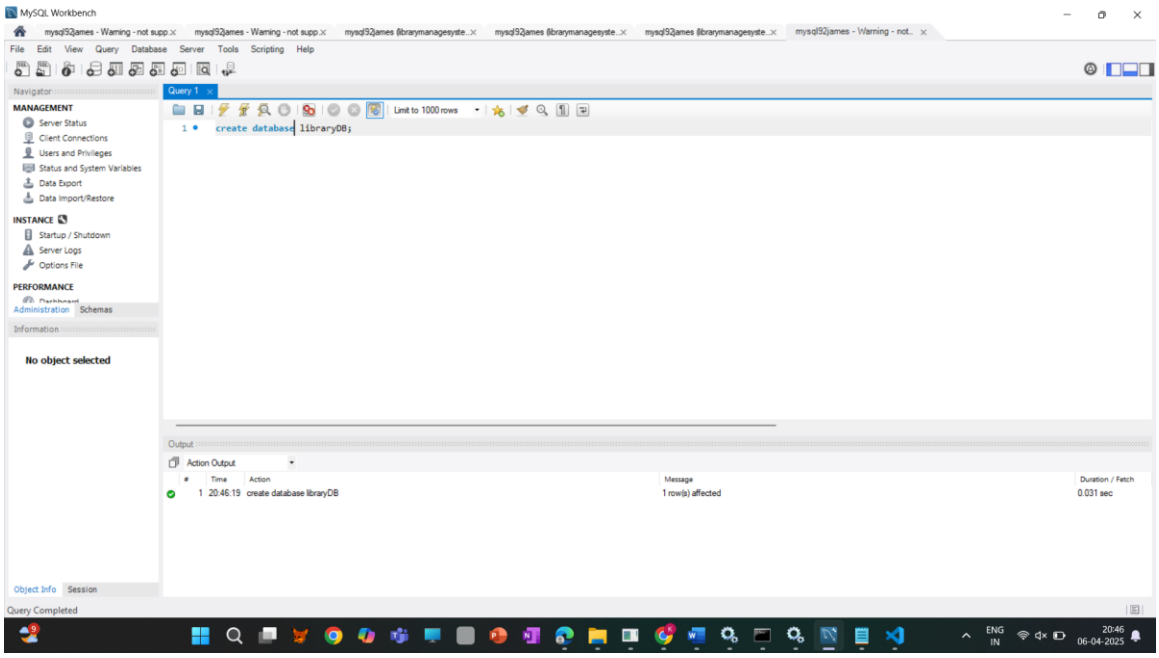


TABLE CREATION:

Genres table:

```
CREATE TABLE Genres (
    genre_id INT AUTO_INCREMENT PRIMARY KEY,
    genre_name VARCHAR(50) NOT NULL,
    description TEXT
);
```

Users table:

```
CREATE TABLE Users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    membership_date DATE,
    status VARCHAR(20)
);
```

Books table:

```
CREATE TABLE Books (
    book_id INT AU TO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author VARCHAR(100),
    genre_id INT,
```

```
    year_published INT,
    isbn VARCHAR(20) UNIQUE,
    copies_available INT DEFAULT 1,
    FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)
);
```

Borrowing_History table:

```
CREATE TABLE Borrowing_History (
    borrow_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    book_id INT,
    borrow_date DATE,
    due_date DATE,
    return_date DATE,
    fine_amount DECIMAL(5,2) DEFAULT 0.00,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

Book_Reviews table:

```
CREATE TABLE Book_Reviews (
    review_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    book_id INT,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    review_text TEXT,
    review_date DATE,
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

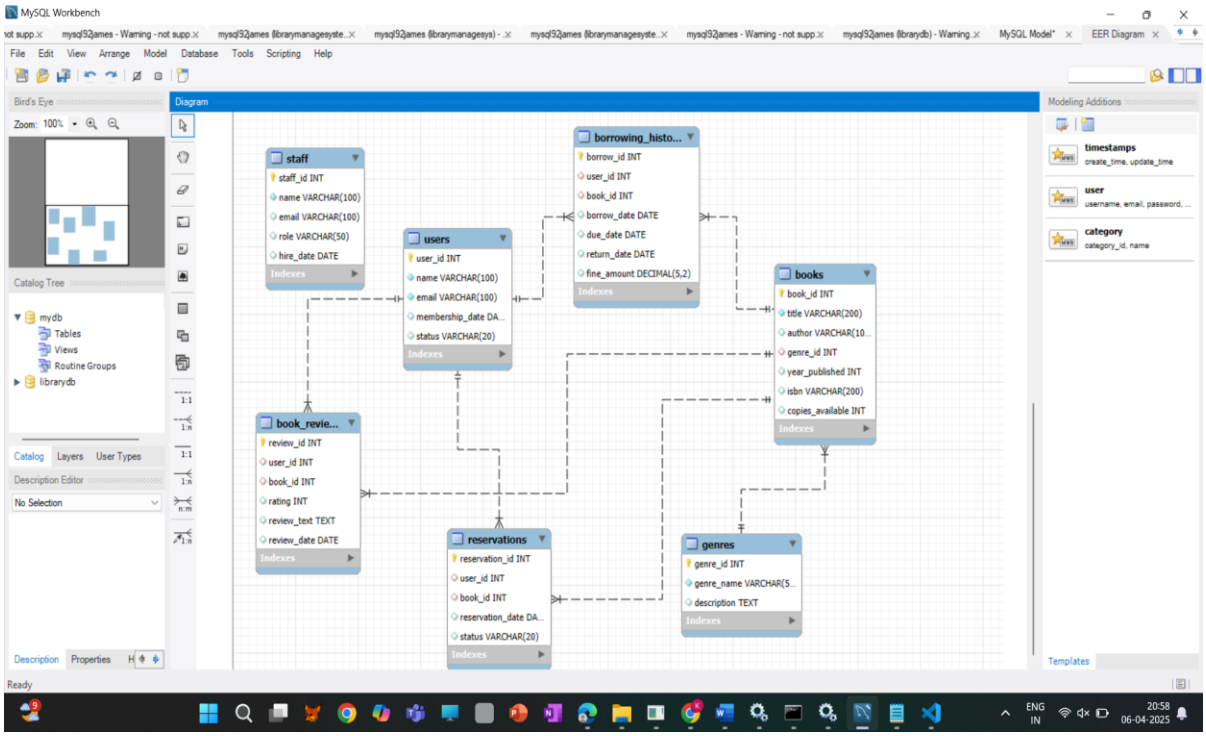
Staff table:

```
CREATE TABLE Staff (
    staff_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    role VARCHAR(50),
    hire_date DATE
);
```

Reservations table:

```
CREATE TABLE Reservations (
    reservation_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    book_id INT,
    reservation_date DATE,
    status VARCHAR(20),
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

2.2 ER DIAGRAM (REVERSE ENGINEERED)



3. QUERIES FOR DATA MANAGEMENT

3.1 INSERT SAMPLE DATA:

GENRES TABLE:

```
INSERT INTO Genres (genre_name, description) VALUES
('Science Fiction', 'Fictional books based on futuristic science and technology'),
```

('Mystery', 'Books involving crime or puzzle-solving'),
('Romance', 'Books focusing on love stories'),
('History', 'Historical events and biographies'),
('Fantasy', 'Books involving magical or imaginary worlds');

USERS TABLE:

INSERT INTO Users (name, email, membership_date, status) VALUES
('Alice Johnson', 'alice@gmail.com', '2023-01-15', 'Active'),
('Bob Smith', 'bob@yahoo.com', '2022-11-22', 'Active'),
('Clara White', 'clara@hotmail.com', '2023-05-03', 'Inactive'),
('David Brown', 'david@gmail.com', '2021-08-30', 'Active'),
('Ella Green', 'ella@rediffmail.com', '2024-02-14', 'Active');

BOOKS TABLE:

INSERT INTO Books (title, author, genre_id, year_published, isbn, copies_available)
VALUES ('The Time Machine', 'H.G. Wells', 1, 1895, 'ISBN1234567890', 3),
('Sherlock Holmes', 'Arthur Conan Doyle', 2, 1887, 'ISBN2345678901', 2),
('Pride and Prejudice', 'Jane Austen', 3, 1813, 'ISBN3456789012', 4),
('World War II', 'Winston Churchill', 4, 1948, 'ISBN4567890123', 1),
('Harry Potter', 'J.K. Rowling', 5, 1997, 'ISBN5678901234', 5);

BORROWING_HISTORY TABLE:

INSERT INTO Borrowing_History (user_id, book_id, borrow_date, due_date,
return_date, fine_amount) VALUES
(1, 1, '2024-01-01', '2024-01-15', '2024-01-14', 0.00),
(2, 2, '2024-02-01', '2024-02-14', '2024-02-20', 15.00),
(3, 3, '2024-03-05', '2024-03-19', NULL, 0.00),
(4, 5, '2024-04-10', '2024-04-24', '2024-04-25', 5.00),
(5, 4, '2024-03-01', '2024-03-15', '2024-03-12', 0.00);

BOOK_REVIEWS TABLE:

```
INSERT INTO Book_Reviews (user_id, book_id, rating, review_text, review_date)
VALUES
(1, 1, 5, 'Amazing sci-fi classic!', '2024-01-16'),
(2, 2, 4, 'Intriguing mystery and great deduction.', '2024-02-21'),
(3, 3, 3, 'Romantic but a bit slow.', '2024-03-20'),
(4, 5, 5, 'Magical and thrilling.', '2024-04-26'),
(5, 4, 4, 'Informative and detailed.', '2024-03-13');
```

STAFF TABLE:

```
INSERT INTO Staff (name, email, role, hire_date) VALUES
('Ravi Kumar', 'ravi@library.com', 'Librarian', '2020-05-10'),
('Sneha Sharma', 'sneha@library.com', 'Assistant Librarian', '2021-07-15'),
('Anil Mehta', 'anil@library.com', 'Technician', '2022-01-20'),
('Priya Das', 'priya@library.com', 'Data Entry', '2023-03-01'),
('Mohit Verma', 'mohit@library.com', 'Clerk', '2024-01-12');
```

RESERVATIONS TABLE:

```
INSERT INTO Reservations (user_id, book_id, reservation_date, status) VALUES
(1, 2, '2024-04-01', 'Pending'),
(2, 1, '2024-04-02', 'Completed'),
(3, 5, '2024-04-03', 'Cancelled'),
(4, 3, '2024-04-04', 'Pending'),
(5, 4, '2024-04-05', 'Completed');
```

3.2 RETRIEVAL QUERIES:

1. Books by Genre

```
SELECT b.title, b.author, g.genre_name
FROM Books b
JOIN Genres g ON b.genre_id = g.genre_id
WHERE g.genre_name = 'Science Fiction';
```

2. Books Borrowed by a Specific User

```
SELECT u.name, b.title, bh.borrow_date, bh.due_date
FROM Borrowing_History bh
JOIN Users u ON bh.user_id = u.user_id
JOIN Books b ON bh.book_id = b.book_id
WHERE u.name = 'Alice Johnson';
```

3. List All Available Books

```
SELECT title, author, copies_available
FROM Books
WHERE copies_available > 0;
```

4. Books with Average Rating

```
SELECT b.title, AVG(br.rating) AS average_rating
FROM Books b
JOIN Book_Reviews br ON b.book_id = br.book_id
GROUP BY b.title
ORDER BY average_rating DESC;
```

5. Reservations for a Book

```
SELECT r.reservation_id, u.name AS user_name, b.title,
r.reservation_date, r.status
FROM Reservations r
JOIN Users u ON r.user_id = u.user_id
JOIN Books b ON r.book_id = b.book_id
WHERE b.title = 'The Time Machine';
```

6. Overdue Books

```
SELECT u.name, b.title, bh.due_date, bh.return_date
FROM Borrowing_History bh
JOIN Users u ON bh.user_id = u.user_id
```

```
JOIN Books b ON bh.book_id = b.book_id
WHERE bh.return_date IS NULL AND bh.due_date < CURDATE();
```

7. Users with Outstanding Fines

```
SELECT u.name, SUM(bh.fine_amount) AS total_fines
FROM Borrowing_History bh
JOIN Users u ON bh.user_id = u.user_id
WHERE bh.fine_amount > 0
GROUP BY u.user_id;
```

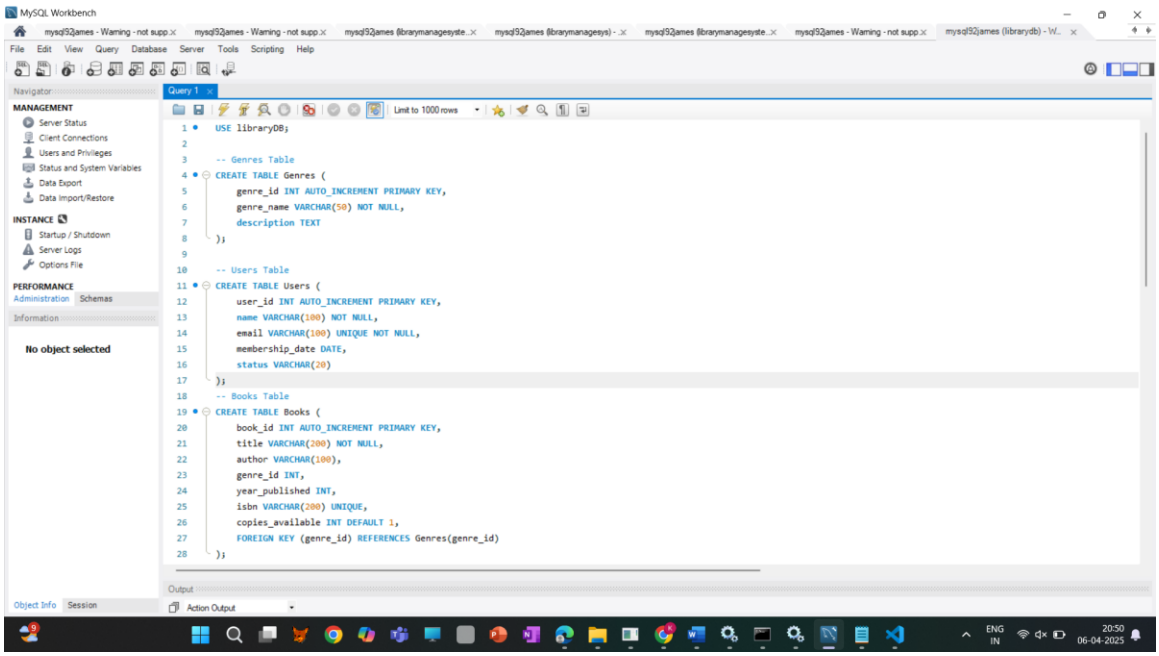
4. IMPLEMENTATION & RESULTS

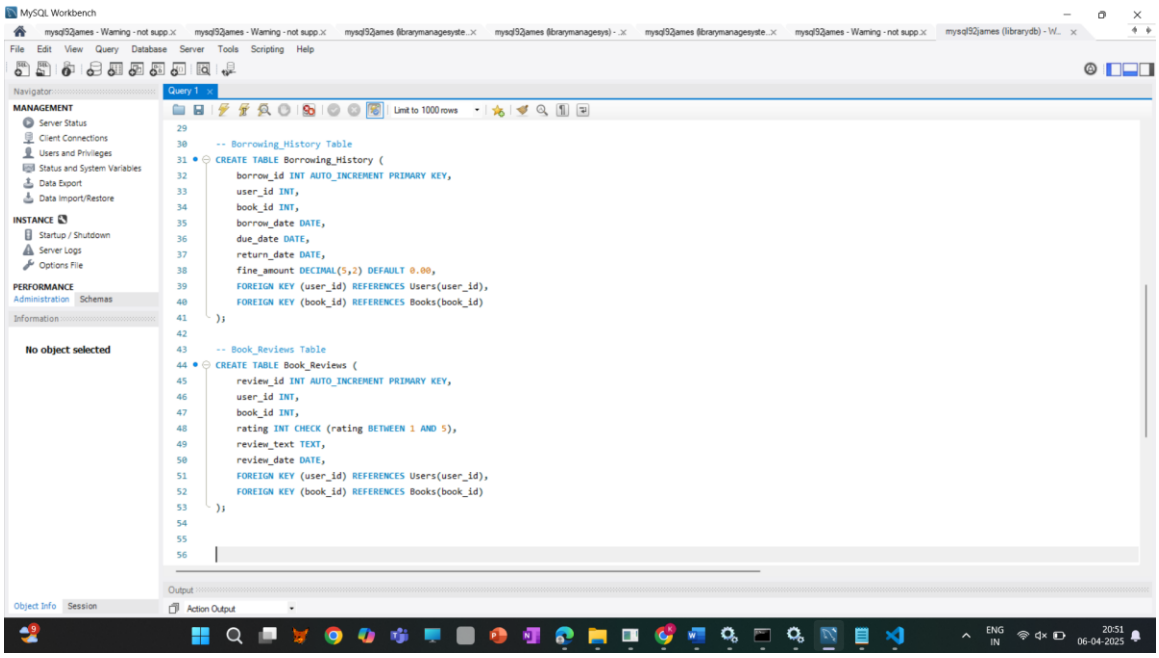
4.1 EXECUTION ENVIRONMENT

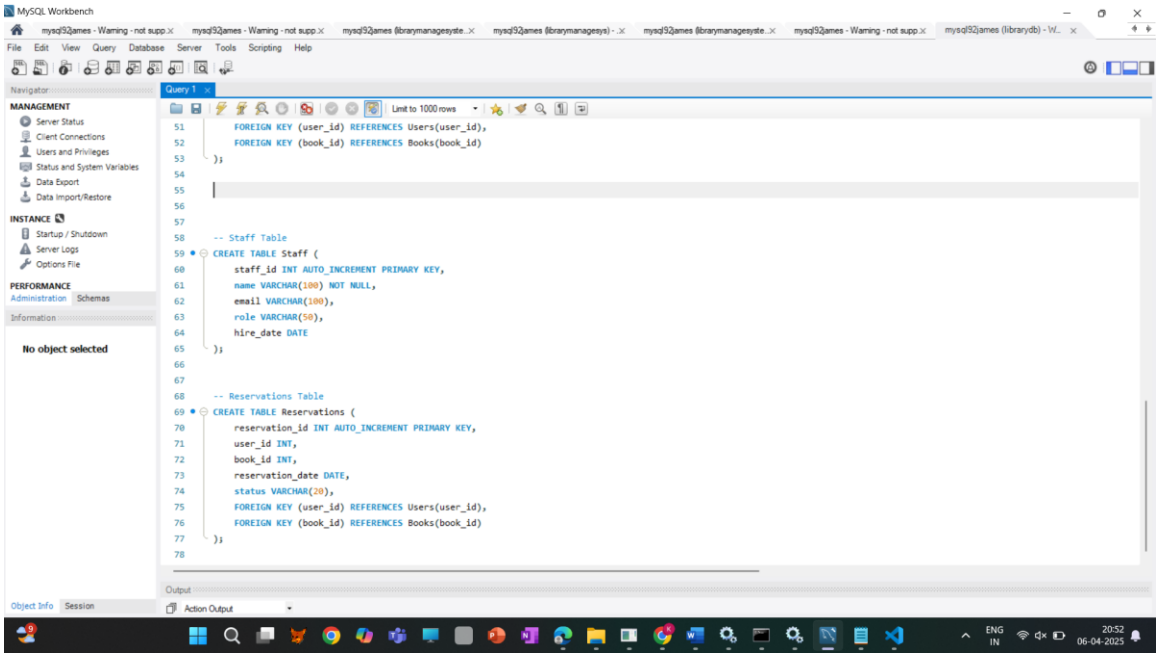
The implementation was executed using MySQL Workbench 9.2 on Windows 11.

4.2 SCREENSHOTS OF EXECUTION RESULTS

1.Table Creation:





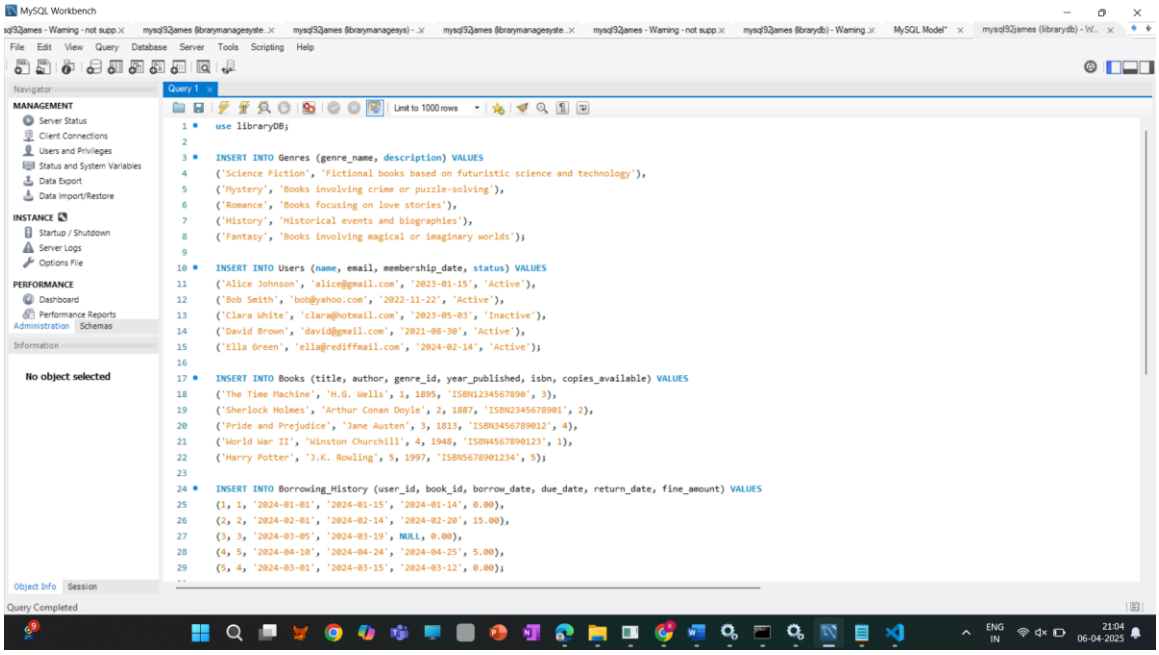


Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	20:49:57	USE libraryDB	0 row(s) affected	0.000 sec
2	20:49:57	CREATE TABLE Genres (genre_id INT AUTO_INCREMENT PRIMARY KEY, genre_name VARCHAR(50) ...	0 row(s) affected	0.047 sec
3	20:49:57	CREATE TABLE Users (user_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NU...	0 row(s) affected	0.047 sec
4	20:49:57	CREATE TABLE Books (book_id INT AUTO_INCREMENT PRIMARY KEY, title VARCHAR(200) NOT NUL...	0 row(s) affected	0.047 sec
5	20:49:57	CREATE TABLE Borrowing_History (borrow_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, bo...	0 row(s) affected	0.062 sec
6	20:49:57	CREATE TABLE Book_Reviews (review_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, book...	0 row(s) affected	0.047 sec
7	20:49:57	CREATE TABLE Staff (staff_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NUL...	0 row(s) affected	0.047 sec
8	20:49:58	CREATE TABLE Reservations (reservation_id INT AUTO_INCREMENT PRIMARY KEY, user_id INT, bo...	0 row(s) affected	0.047 sec

2.Sample Data Insertion:



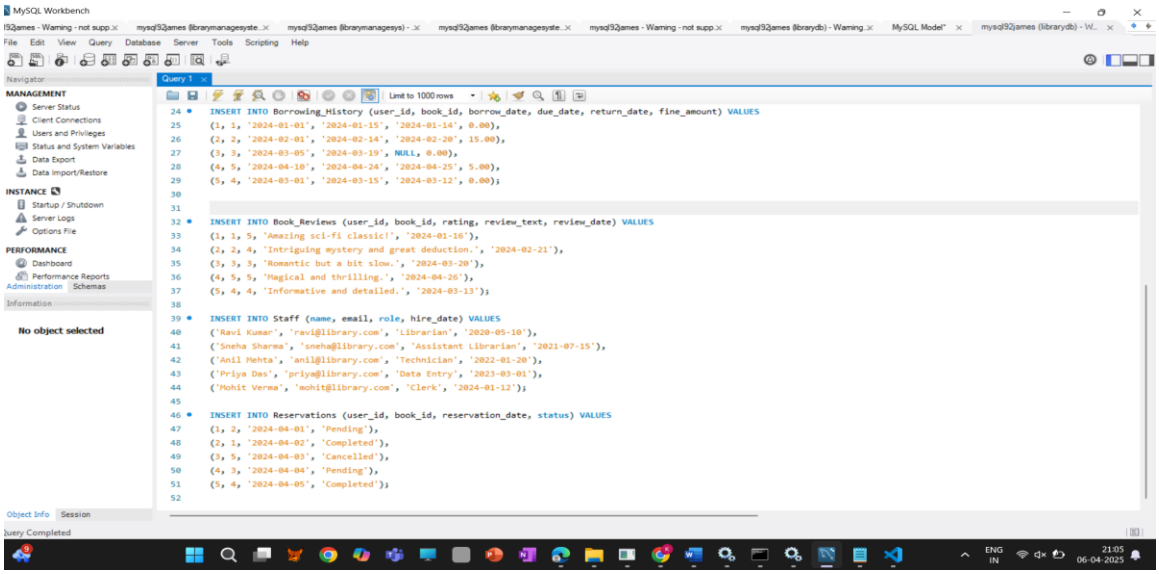
```
use libraryDB;

INSERT INTO Genres (genre_name, description) VALUES
('Science Fiction', 'Fictional books based on futuristic science and technology'),
('Mystery', 'Books involving crime or puzzle-solving'),
('Romance', 'Books focusing on love stories'),
('History', 'Historical events and biographies'),
('Fantasy', 'Books involving magical or imaginary worlds');

INSERT INTO Users (name, email, membership_date, status) VALUES
('Alice Johnson', 'alice@gmail.com', '2023-01-15', 'Active'),
('Bob Smith', 'bob@yahoo.com', '2022-11-22', 'Active'),
('Clara White', 'clara@hotmail.com', '2023-05-03', 'Inactive'),
('David Brown', 'david@gmail.com', '2021-08-30', 'Active'),
('Ella Green', 'ella@rediffmail.com', '2024-02-14', 'Active');

INSERT INTO Books (title, author, genre_id, year_published, isbn, copies_available) VALUES
('The Time Machine', 'H.G. Wells', 1, 1895, 'ISBN1234567890', 3),
('Sherlock Holmes', 'Arthur Conan Doyle', 2, 1887, 'ISBN2345678901', 2),
('Pride and Prejudice', 'Jane Austen', 3, 1813, 'ISBN3456789012', 4),
('World War II', 'Winston Churchill', 4, 1948, 'ISBN4567890123', 1),
('Harry Potter', 'J.K. Rowling', 5, 1997, 'ISBN5678901234', 5);

INSERT INTO Borrowing_History (user_id, book_id, borrow_date, due_date, return_date, fine_amount) VALUES
(1, 1, '2024-01-01', '2024-01-15', '2024-01-14', 0.00),
(2, 2, '2024-02-01', '2024-02-14', '2024-02-20', 15.00),
(3, 3, '2024-03-05', '2024-03-19', NULL, 0.00),
(4, 5, '2024-04-10', '2024-04-24', '2024-04-25', 5.00),
(5, 4, '2024-03-01', '2024-03-15', '2024-03-12', 0.00);
```



```
INSERT INTO Borrowing_History (user_id, book_id, borrow_date, due_date, return_date, fine_amount) VALUES
(1, 1, '2024-01-01', '2024-01-15', '2024-01-14', 0.00),
(2, 2, '2024-02-01', '2024-02-14', '2024-02-20', 15.00),
(3, 3, '2024-03-05', '2024-03-19', NULL, 0.00),
(4, 5, '2024-04-10', '2024-04-24', '2024-04-25', 5.00),
(5, 4, '2024-03-01', '2024-03-15', '2024-03-12', 0.00);

INSERT INTO Book_Reviews (user_id, book_id, rating, review_text, review_date) VALUES
(1, 1, 5, 'Amazing sci-fi classic!', '2024-01-16'),
(2, 2, 4, 'Intriguing mystery and great deduction.', '2024-02-21'),
(3, 3, 3, 'Romantic but a bit slow.', '2024-03-20'),
(4, 5, 5, 'Magical and thrilling.', '2024-04-26'),
(5, 4, 4, 'Informative and detailed.', '2024-03-13');

INSERT INTO Staff (name, email, role, hire_date) VALUES
('Ravi Kumar', 'ravi@library.com', 'Librarian', '2020-05-10'),
('Sneha Sharma', 'sneha@library.com', 'Assistant Librarian', '2021-07-15'),
('Anil Mehta', 'anil@library.com', 'Technician', '2022-01-20'),
('Priya Das', 'priya@library.com', 'Data Entry', '2023-03-01'),
('Mohit Verma', 'mohit@library.com', 'Clerk', '2024-01-12');

INSERT INTO Reservations (user_id, book_id, reservation_date, status) VALUES
(1, 2, '2024-04-01', 'Pending'),
(2, 1, '2024-04-02', 'Completed'),
(3, 5, '2024-04-03', 'Cancelled'),
(4, 3, '2024-04-04', 'Pending'),
(5, 4, '2024-04-05', 'Completed');
```

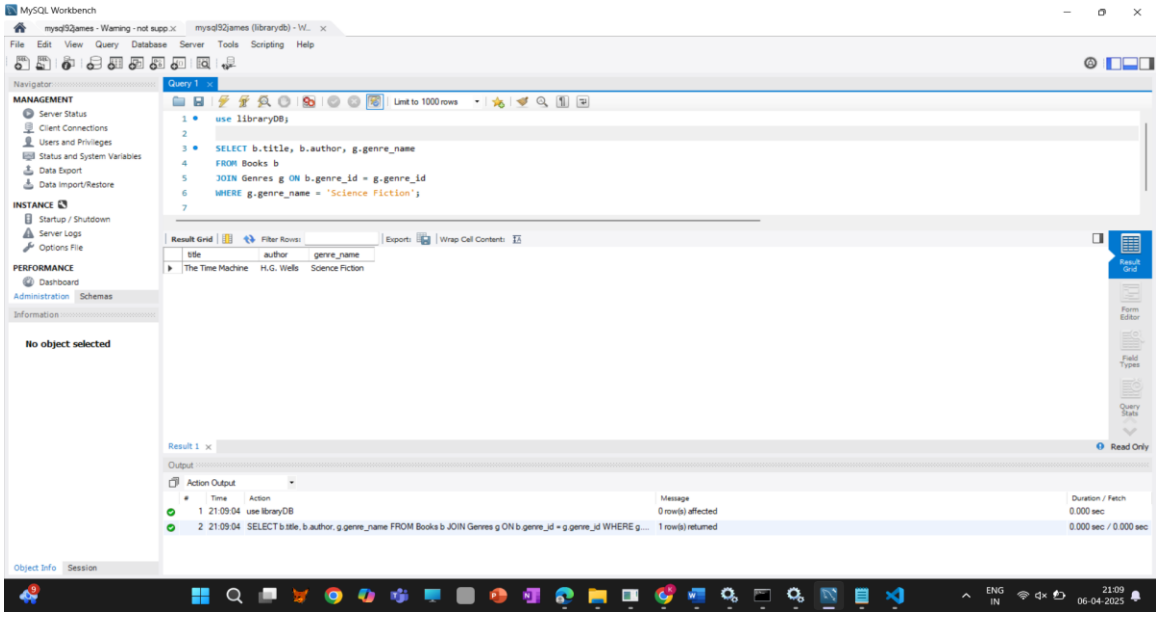
Output

Action Output

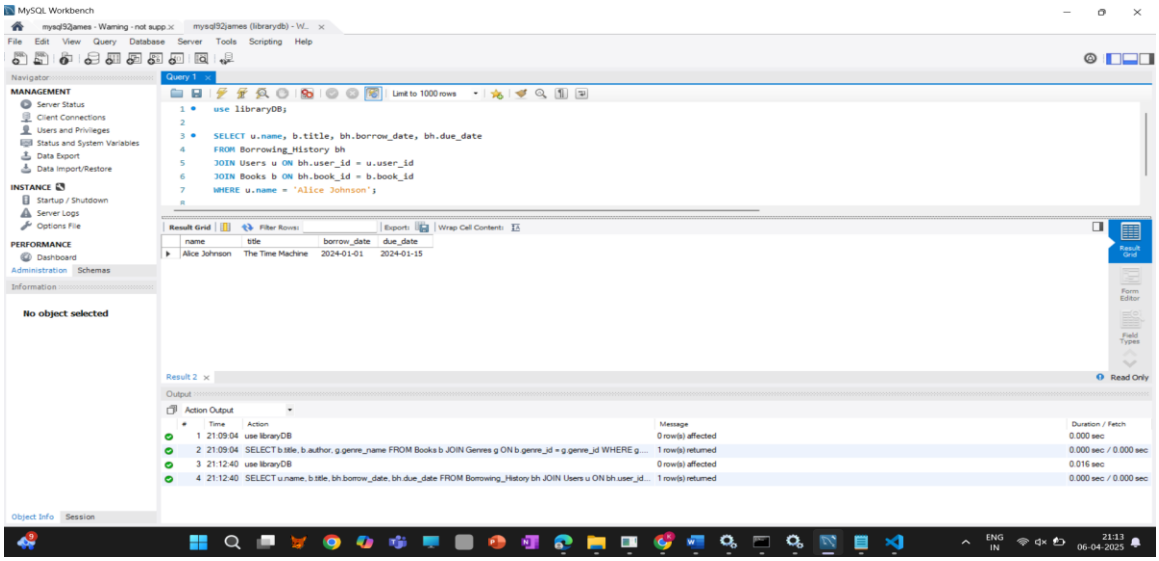
#	Time	Action	Message	Duration / Fetch
1	21:04:05	use libraryDB	0 row(s) affected	0.000 sec
2	21:04:05	INSERT INTO Genres (genre_name, description) VALUES ('Science Fiction', 'Fictional books based on futuristic scie...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
3	21:04:05	INSERT INTO Users (name, email, membership_date, status) VALUES ('Alice Johnson', 'alice@gmail.com', '2023-01-...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
4	21:04:05	INSERT INTO Books (title, author, genre_id, year_published, isbn, copies_available) VALUES ('The Time Machine', '...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
5	21:04:05	INSERT INTO Borrowing_History (user_id, book_id, borrow_date, due_date, return_date, fine_amount) VALUES (1, ...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
6	21:04:05	INSERT INTO Book_Reviews (user_id, book_id, rating, review_text, review_date) VALUES (1, 1, 5, 'Amazing sci-fi cl...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
7	21:04:05	INSERT INTO Staff (name, email, role, hire_date) VALUES ('Ravi Kumar', 'ravi@library.com', 'Librarian', '2020-05-10'),...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
8	21:04:05	INSERT INTO Reservations (user_id, book_id, reservation_date, status) VALUES (1, 2, '2024-04-01', 'Pending'), (2, 1...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec

3.Data Retrieval Queries:

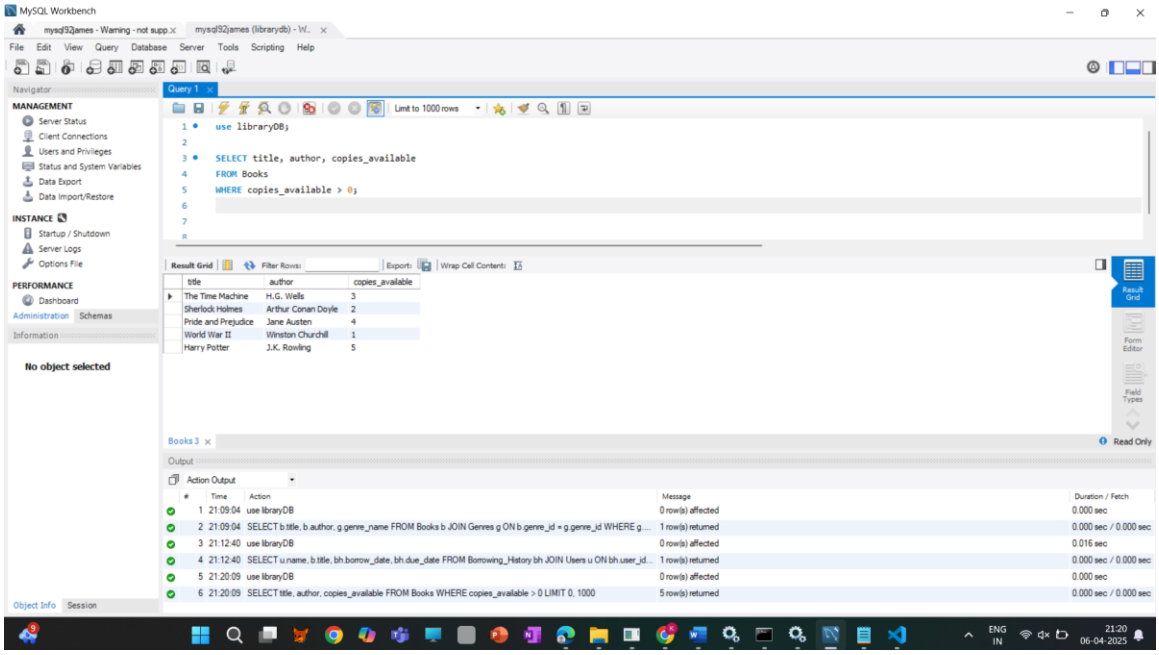
1.Books by genre:



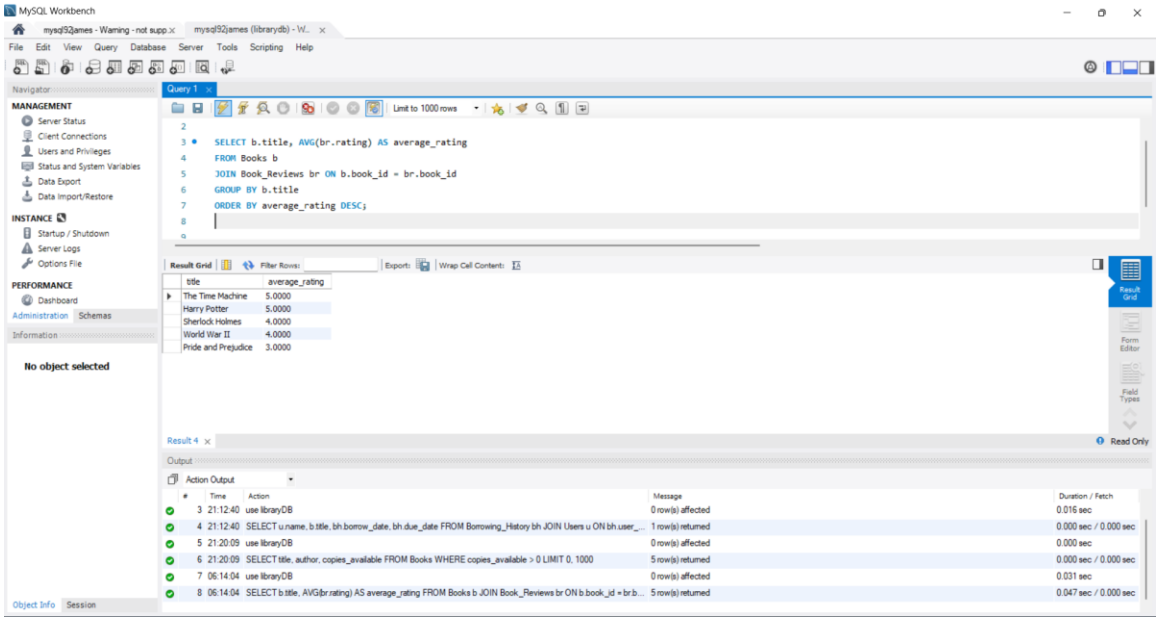
2. Books Borrowed by a Specific User:



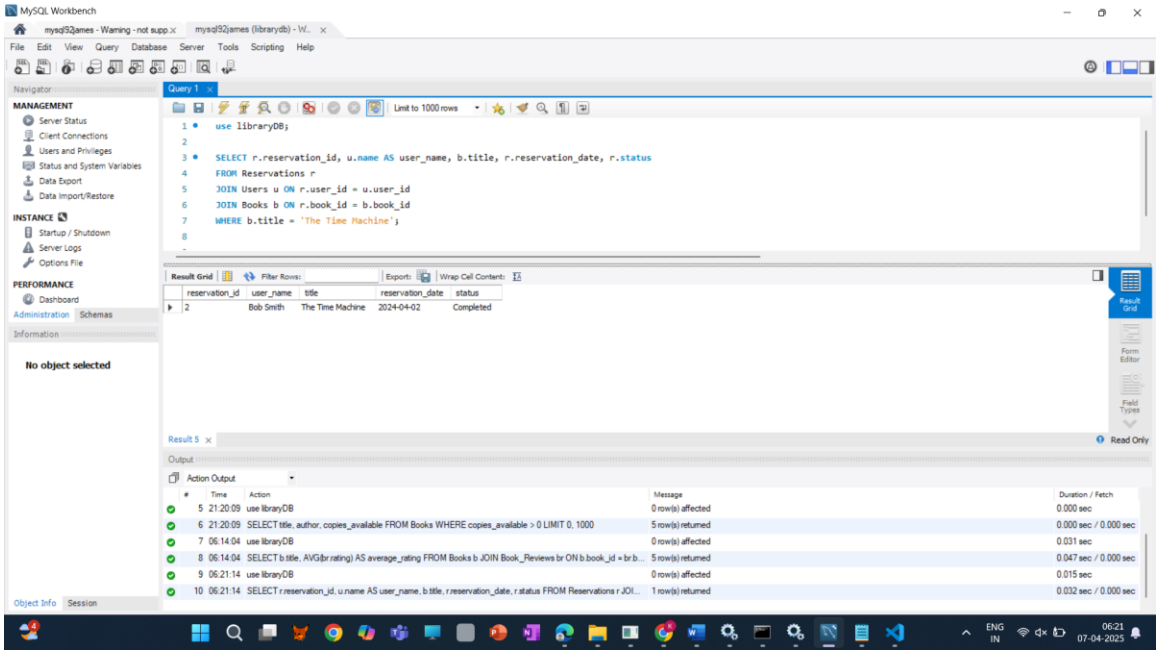
3. List all Available books:



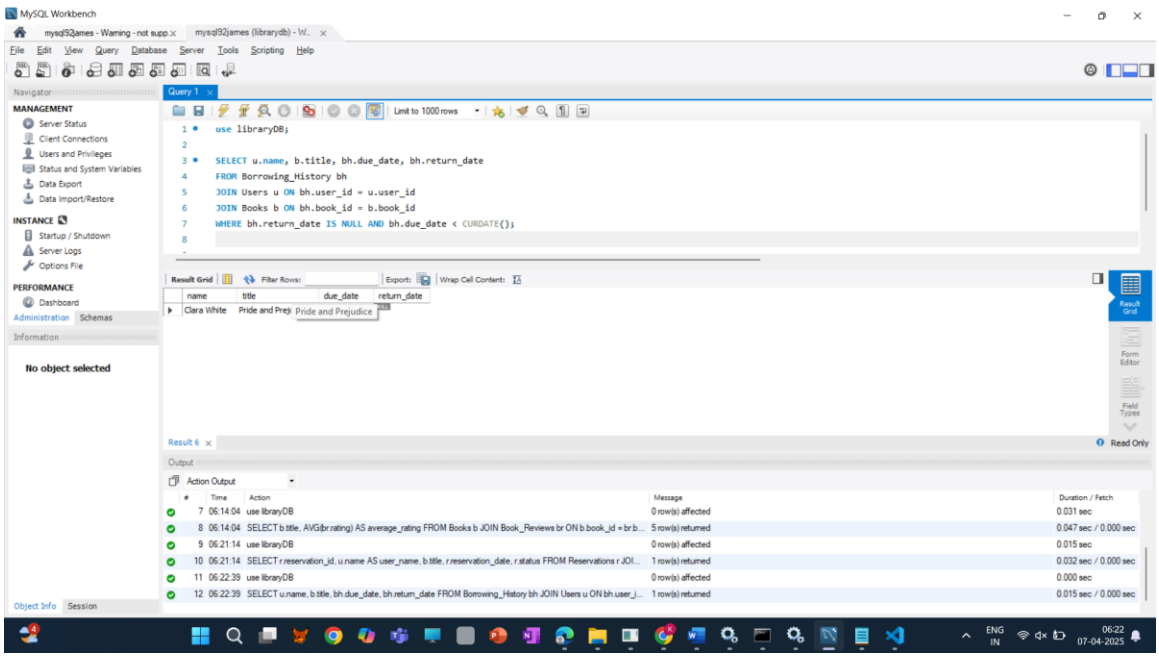
4.Books with Average Rating



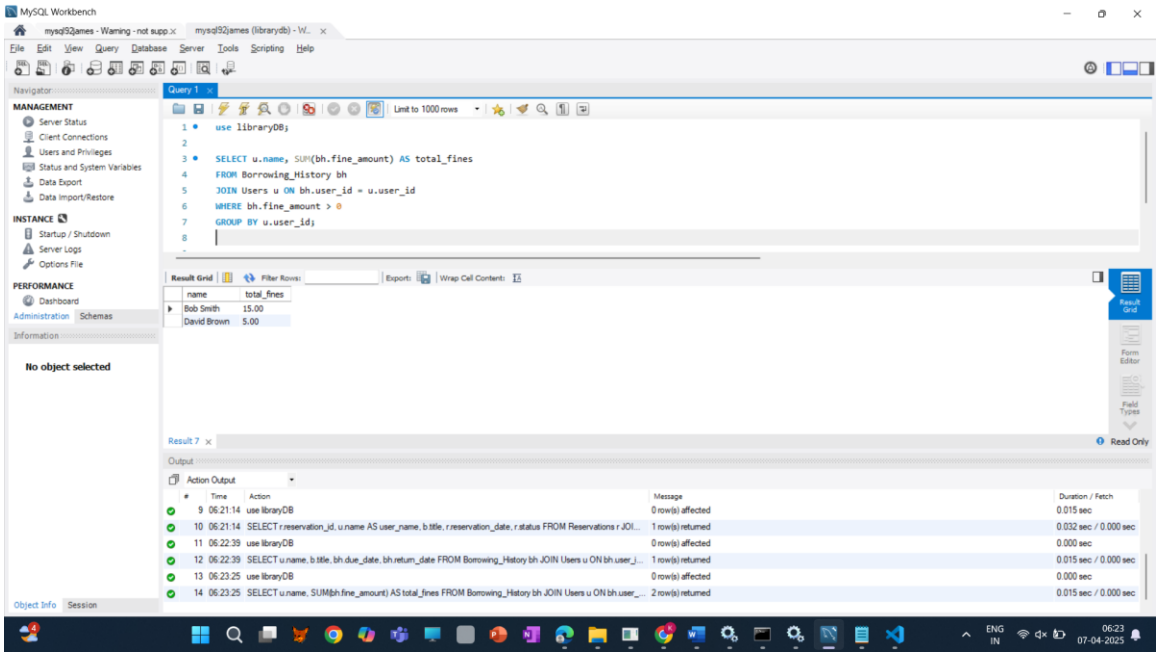
5. Reservations for a Book:



6. Overdue Books:

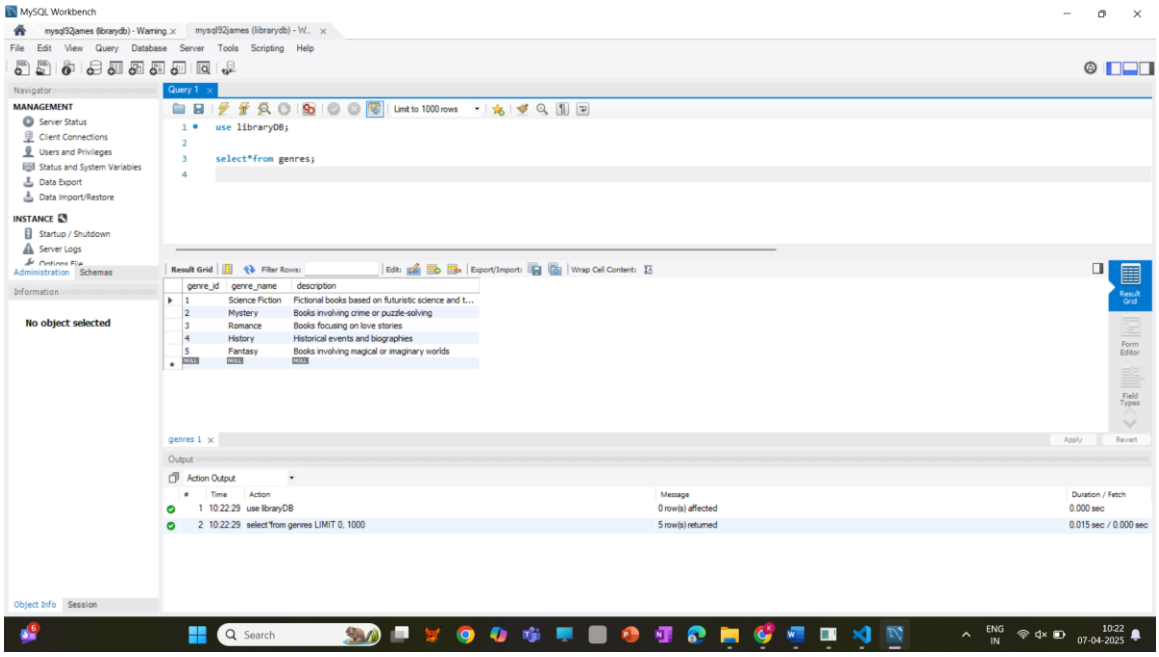


7. User with outstanding fines:

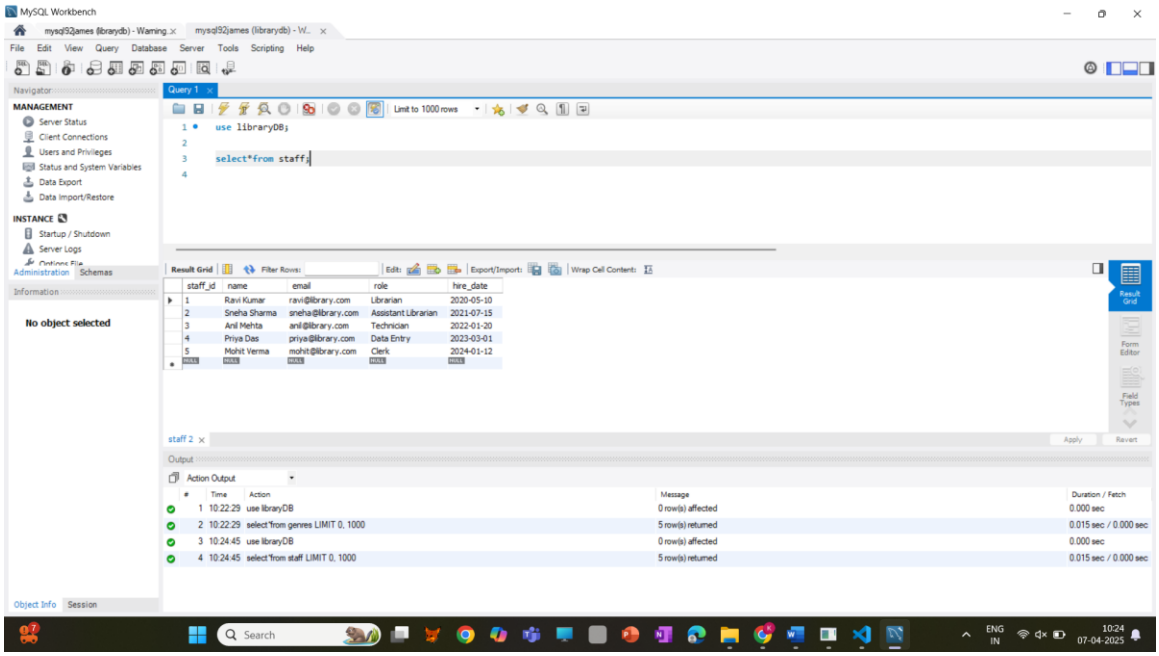


4.Select Queries:

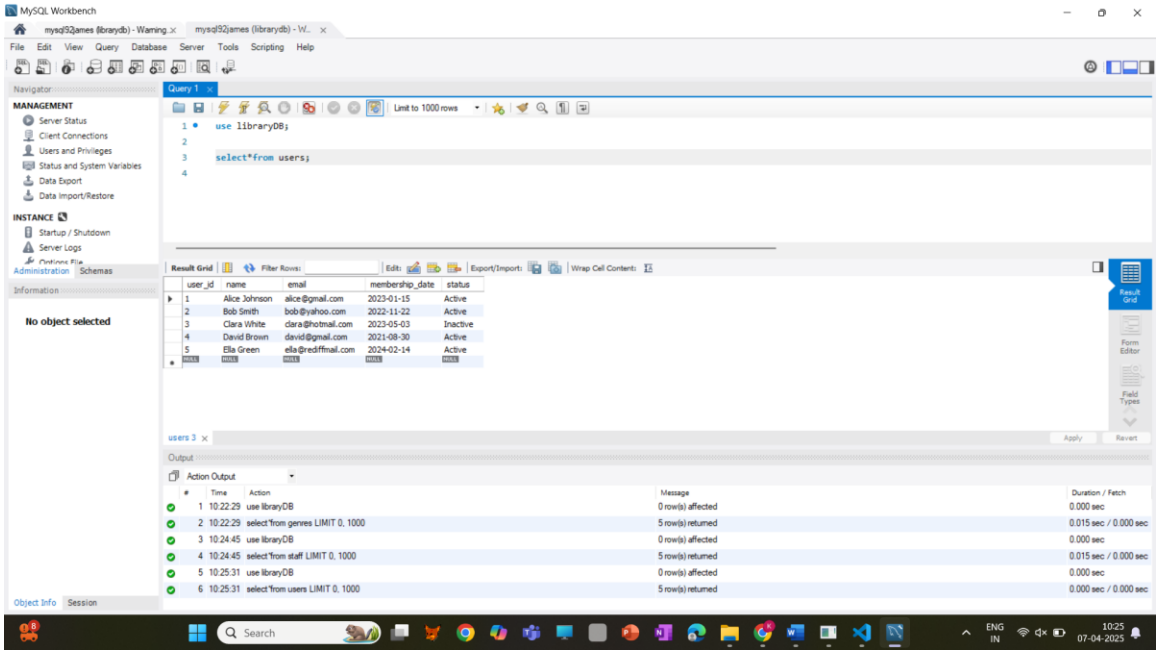
GENRES TABLE:



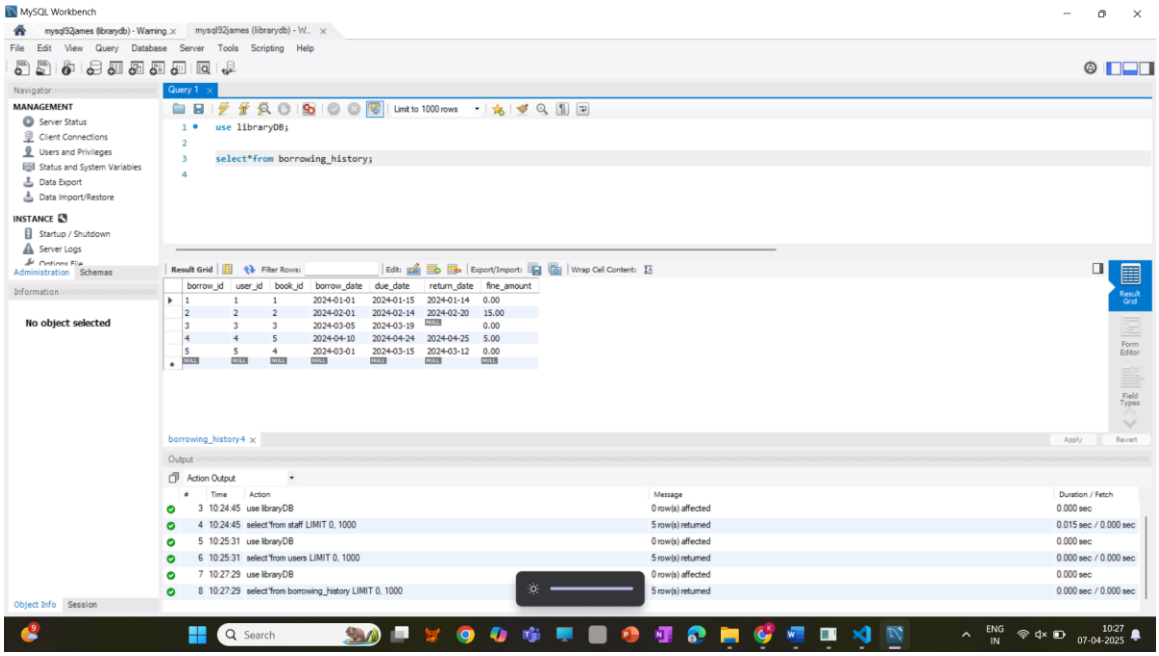
STAFF TABLE:



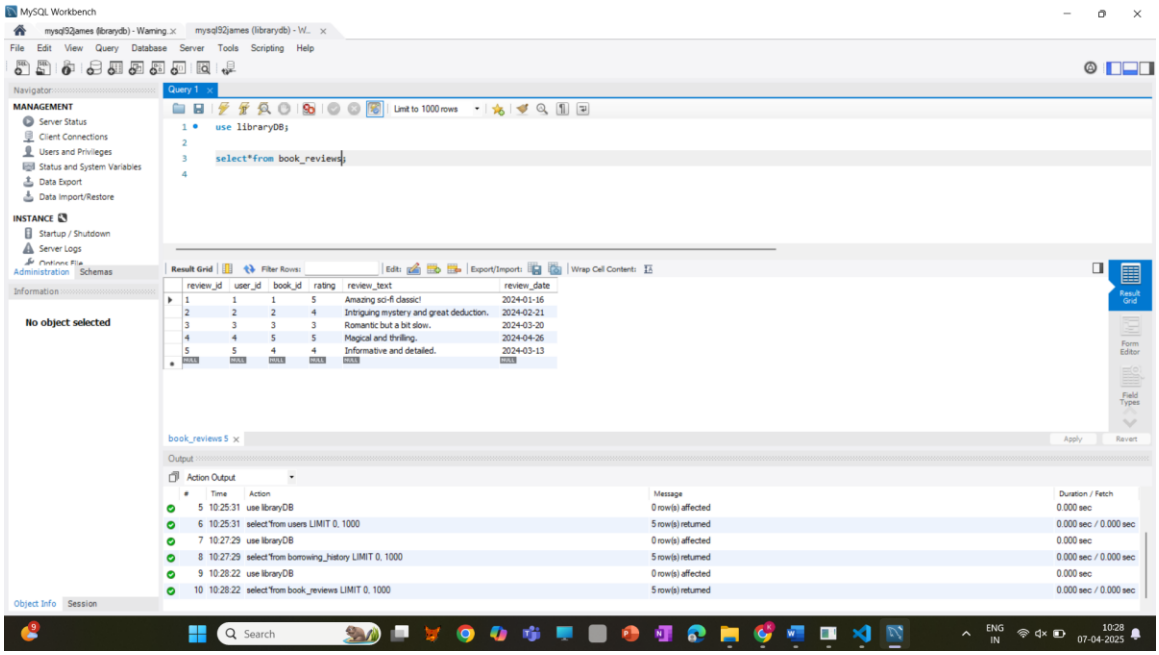
USERS TABLE:



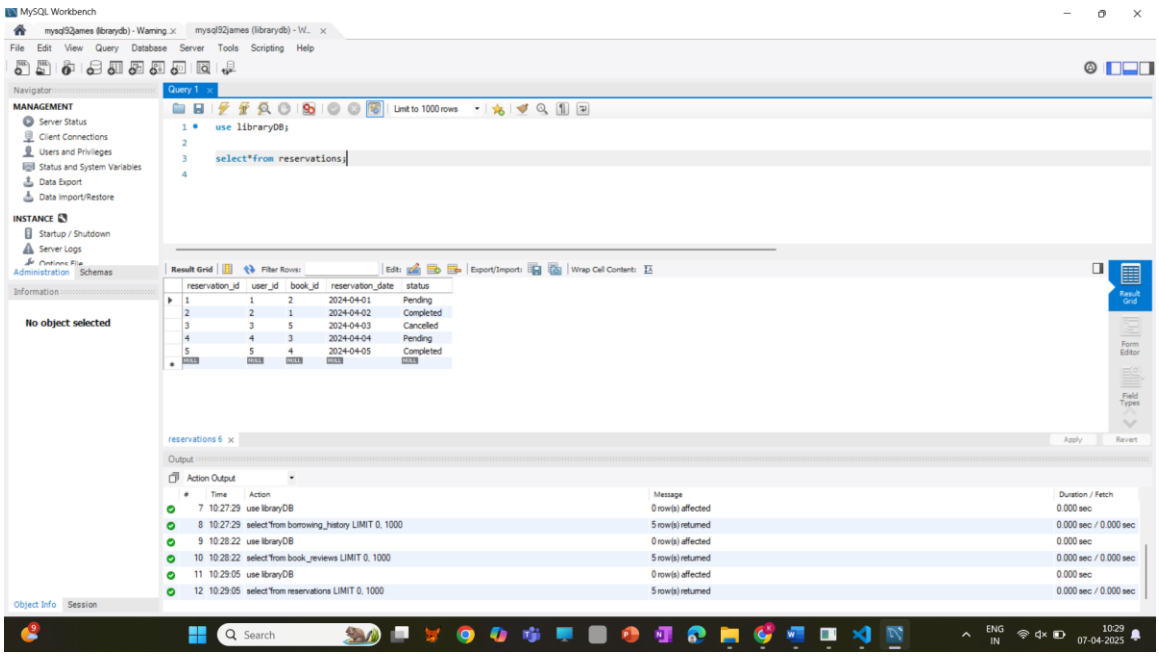
BORROWING_HISTORY TABLE:



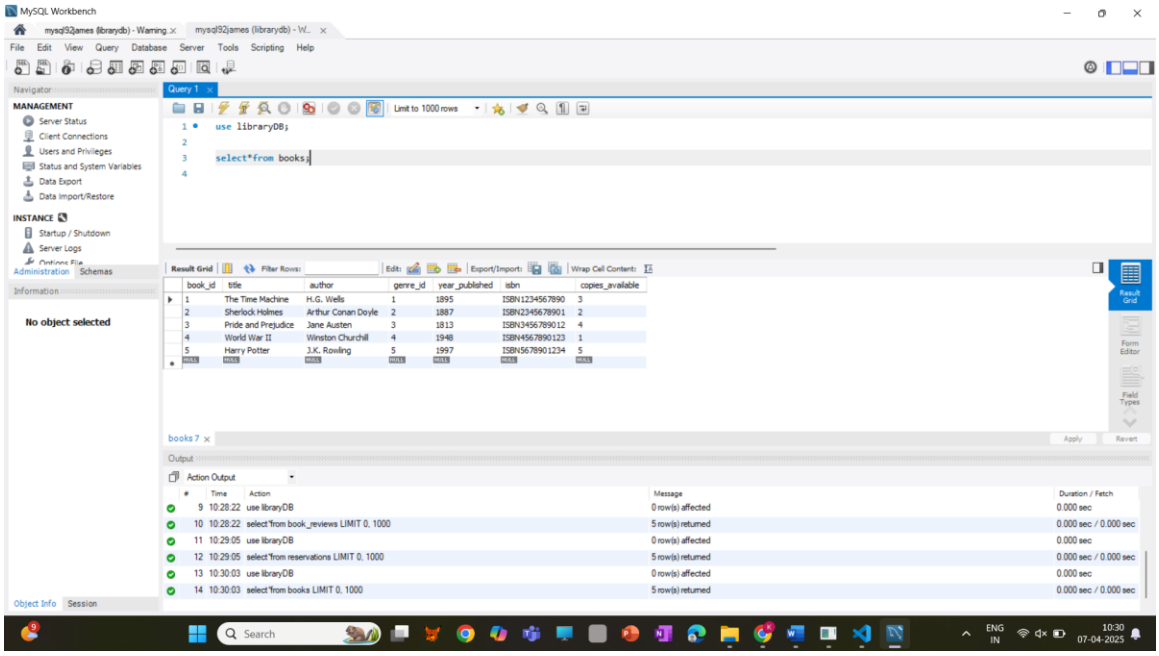
BORROWING_REVIEWS TABLE:



RESERVATIONS TABLE:



BOOKS TABLE:



5. GITHUB REPOSITORY

5.1 REPOSITORY LINK

https://github.com/kalai02082005/LibraryManagementsystem.git

5.2 UPLOADED FILES IN REPOSITORY

The following files are included in the repository:

- **database_schema.sql** – SQL scripts for creating all required tables (Users, Books, Genres, Borrowing_History, Book_Reviews, Staff, Reservations).
- **insert_sample_data.sql** – SQL scripts for inserting sample records into each table.
- **retrieval_queries.sql** – SQL scripts for retrieving data such as books by genre, available books, borrowed books, top-rated books, etc.
- **ER_Diagram.png** – Entity-Relationship Diagram showing relationships between tables.
- **project_report.docx** – Final formatted documentation report (can be in .docx or .pdf).
- **screenshots/** – Folder containing screenshots of executed SQL queries and output from MySQL Workbench or any DBMS used.

6. Conclusion

The Library Management System simplifies library operations and enhances user engagement through personalized book suggestions. With a well-structured database and efficient retrieval queries, it ensures smooth management of users, books, and transactions. This project demonstrates effective use of SQL and ER modeling, laying the groundwork for future enhancements like smart recommendations and web integration.