HACKATHON SUBMISSION (LEVEL-2-SOLUTION)

Use Case Title: Library Management System.

Student Name: N. VENNILA.

Register Number: U22CSE32259.

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    AdvancedLibraryManagementSystem;
```

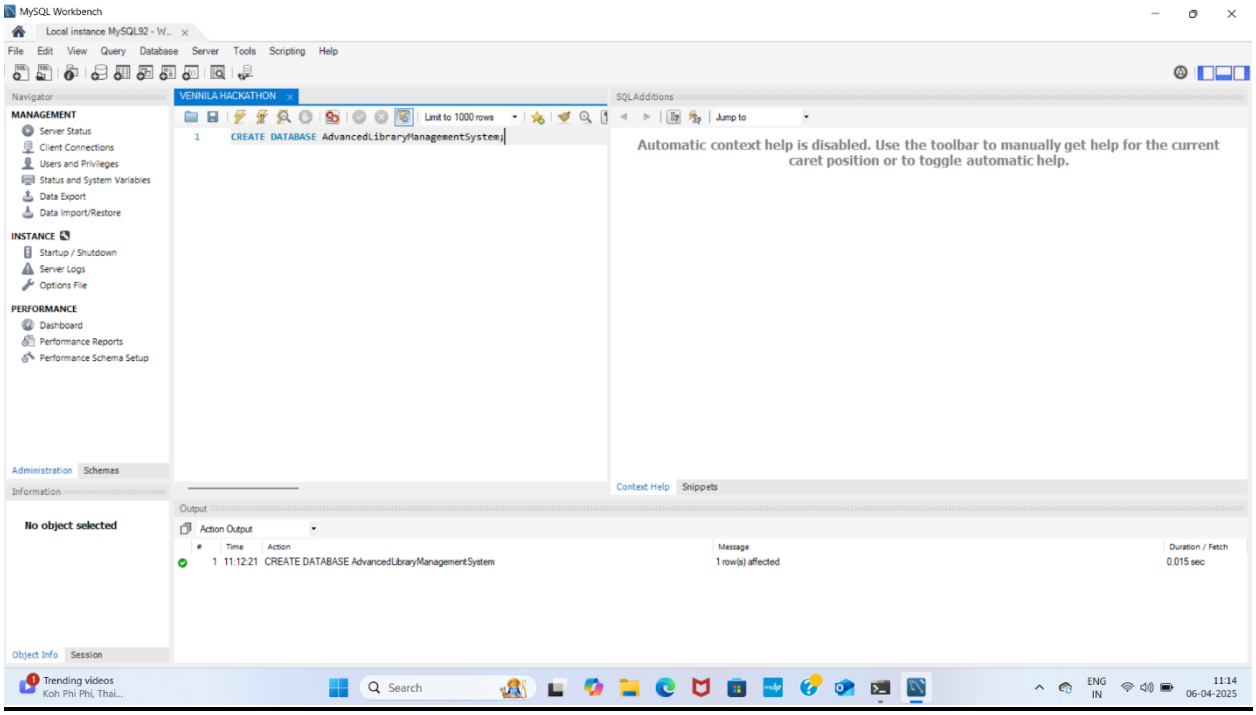


TABLE CREATION:

Books table:

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(100),
    Publisher VARCHAR(100),
    ISBN VARCHAR(20) UNIQUE,
    Genre VARCHAR(50),
    Language VARCHAR(30),
    TotalCopies INT NOT NULL CHECK (TotalCopies >= 0),
    AvailableCopies INT NOT NULL CHECK (AvailableCopies >= 0),
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

Members table:

```
CREATE TABLE Members (
    MemberID INT NOT NULL AUTO_INCREMENT,
    FullName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE,
    PhoneNumber VARCHAR(20),
    MembershipDate DATE DEFAULT (CURRENT_DATE),
    AddressLine1 VARCHAR(150),
    AddressLine2 VARCHAR(150),
    City VARCHAR(50),
    State VARCHAR(50),
    PostalCode VARCHAR(10),
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (MemberID)
);
```

Transactions table:

```
CREATE TABLE Transactions (
    TransactionID INT NOT NULL AUTO_INCREMENT,
    BookID INT NOT NULL,
    MemberID INT NOT NULL,
    IssueDate DATE NOT NULL,
    DueDate DATE NOT NULL,
```

```
 ReturnDate DATE,

    Fine DECIMAL(5,2) DEFAULT 0.00,

    Status ENUM('Issued', 'Returned', 'Overdue') DEFAULT 'Issued',

    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,

    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

    PRIMARY KEY (TransactionID),

    FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE,

    FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE
CASCADE

);
```

BookOrders table:

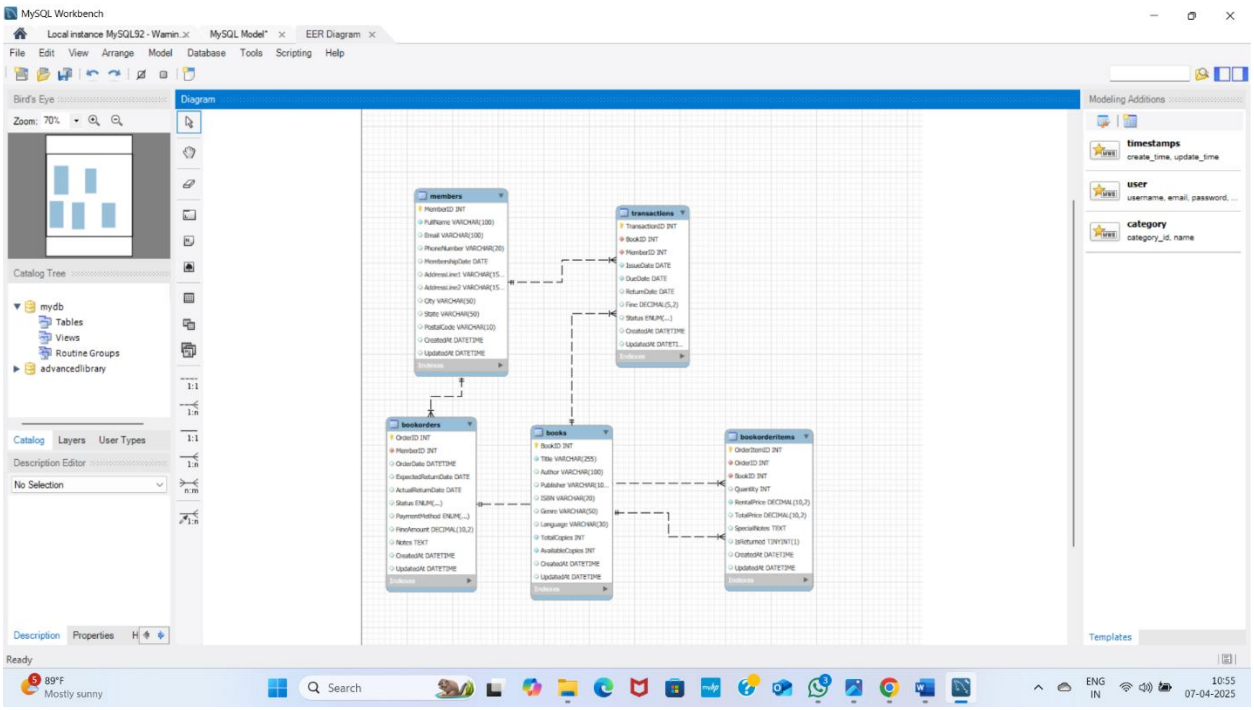
```
    CREATE TABLE BookOrders (
    OrderID INT NOT NULL AUTO_INCREMENT,
    MemberID INT NOT NULL,
    OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    ExpectedReturnDate DATE,
    ActualReturnDate DATE,
    Status ENUM('Issued', 'Returned', 'Overdue', 'Cancelled') DEFAULT 'Issued',
    PaymentMethod ENUM('Cash', 'Card', 'UPI', 'Wallet') DEFAULT 'Cash',
    FineAmount DECIMAL(10,2) DEFAULT 0.00,
    Notes TEXT,
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE
CASCADE,
    CHECK (FineAmount >= 0)
);
```

BookOrderItems TABLE:

```
 CREATE TABLE BookOrderItems (
    OrderItemID INT NOT NULL AUTO_INCREMENT,
    OrderID INT NOT NULL,
```

```
BookID INT NOT NULL,
    Quantity INT DEFAULT 1,
    RentalPrice DECIMAL(10,2) NOT NULL,
    TotalPrice DECIMAL(10,2) GENERATED ALWAYS AS (Quantity * RentalPrice)
STORED,
    SpecialNotes TEXT,
    IsReturned BOOLEAN DEFAULT FALSE,
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (OrderItemID),
    FOREIGN KEY (OrderID) REFERENCES BookOrders(OrderID) ON DELETE CASCADE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE,
    CHECK (Quantity > 0),
    CHECK (RentalPrice >= 0)
);
```

2.2 ER DIAGRAM (REVERSE ENGINEERED)



3. QUERIES FOR DATA MANAGEMENT

3.1 INSERT SAMPLE DATA:

Books TABLE:

```
INSERT INTO Books (Title, Author, Publisher, ISBN, Genre, Language, TotalCopies,
AvailableCopies)
VALUES
('The Alchemist', 'Paulo Coelho', 'HarperCollins', '9780061122415', 'Fiction', 'English', 10, 8),
('Atomic Habits', 'James Clear', 'Penguin', '9780735211292', 'Self-help', 'English', 15, 12),
('Wings of Fire', 'A.P.J. Abdul Kalam', 'Universities Press', '9788173711466',
'Autobiography', 'English', 20, 17),
('Zero to One', 'Peter Thiel', 'Crown Business', '9780804139298', 'Business', 'English', 8, 6),
('Think and Grow Rich', 'Napoleon Hill', 'The Ralston Society', '9781585424337',
'Motivation', 'English', 12, 10);
```

Members TABLE:

```
INSERT INTO INSERT INTO Members (FullName, Email, PhoneNumber,
AddressLine1, City, State, PostalCode)

VALUES

('Anjali Sharma', 'anjali@example.com', '9876543210', '12 MG Road', 'Delhi',
'Delhi', '110001'),

('Rahul Verma', 'rahulv@example.com', '9123456780', '23 Park Avenue',
'Mumbai', 'Maharashtra', '400001'),

('Priya Kumar', 'priyak@example.com', '9988776655', '45 Gandhi Street',
'Chennai', 'Tamil Nadu', '600001'),

('Karan Mehta', 'karanm@example.com', '9345678912', '67 Ring Road',
'Bangalore', 'Karnataka', '560001'),

('Sneha Iyer', 'sneha@example.com', '9871234567', '89 Lotus Enclave',
'Hyderabad', 'Telangana', '500001');
```

Transactions TABLE:

INSERT INTO Transactions (BookID, MemberID, IssueDate, DueDate, ReturnDate, Fine, Status)

VALUES

(1, 1, '2025-04-01', '2025-04-10', NULL, 0.00, 'Issued'),

(2, 2, '2025-03-25', '2025-04-04', '2025-04-03', 0.00, 'Returned'),

(3, 3, '2025-03-28', '2025-04-07', NULL, 0.00, 'Overdue'),

(4, 4, '2025-04-02', '2025-04-12', NULL, 0.00, 'Issued'),

(5, 5, '2025-03-30', '2025-04-08', '2025-04-08', 0.00, 'Returned');

BookOrders TABLE:

INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, ActualReturnDate, Status, PaymentMethod, FineAmount, Notes)

VALUES

(1, '2025-04-01 10:00:00', '2025-04-10', NULL, 'Issued', 'Card', 0.00, 'Urgent delivery requested'),

(2, '2025-03-25 14:30:00', '2025-04-04', '2025-04-03', 'Returned', 'UPI', 0.00, ''),

(3, '2025-03-28 09:15:00', '2025-04-07', NULL, 'Overdue', 'Cash', 20.00, 'Returned late'),

(4, '2025-04-02 11:45:00', '2025-04-12', NULL, 'Issued', 'Card', 0.00, 'First-time borrower'),

(5, '2025-03-30 13:00:00', '2025-04-08', '2025-04-08', 'Returned', 'Wallet', 0.00, '');

BOOKOrderItems TABLE:

INSERT INTO BookOrderItems (OrderID, BookID, Quantity, RentalPrice, SpecialNotes, IsReturned)

VALUES

(1, 1, 1, 50.00, 'Hardcover preferred', FALSE),

(2, 2, 2, 40.00, '', TRUE),

(3, 3, 1, 30.00, 'Please wrap', FALSE),

(4, 4, 1, 45.00, '', FALSE),

(5, 5, 1, 35.00, '', TRUE);

3.2 RETRIEVAL QUERIES:

1. List All Books by a Specific Genre

```
SELECT * FROM Books

WHERE Genre = 'Fiction';
```

2. Count of Books by Each Genre

```
SELECT Genre, COUNT(*) AS TotalBooks
FROM Books
GROUP BY Genre;
```

3. List Books of a Genre That Are Currently Available

```
SELECT *
FROM Books
WHERE Genre = 'Self-help' AND AvailableCopies > 0;
```

4. Get All Issued Books by Genre

```
SELECT b.Title, b.Genre, t.Status
FROM Transactions t
JOIN Books b ON t.BookID = b.BookID
WHERE t.Status = 'Issued' AND b.Genre = 'Business';
```

5. Genres With Low Availability (e.g., < 3 Copies Left)

```
SELECT Genre, Title, AvailableCopies
FROM Books
WHERE AvailableCopies < 3
ORDER BY AvailableCopies ASC;
```

6. Top 3 Genres by Total Number of Books

```
SELECT Genre, SUM(TotalCopies) AS TotalBooks
FROM Books
GROUP BY Genre
ORDER BY TotalBooks DESC
LIMIT 3;
```

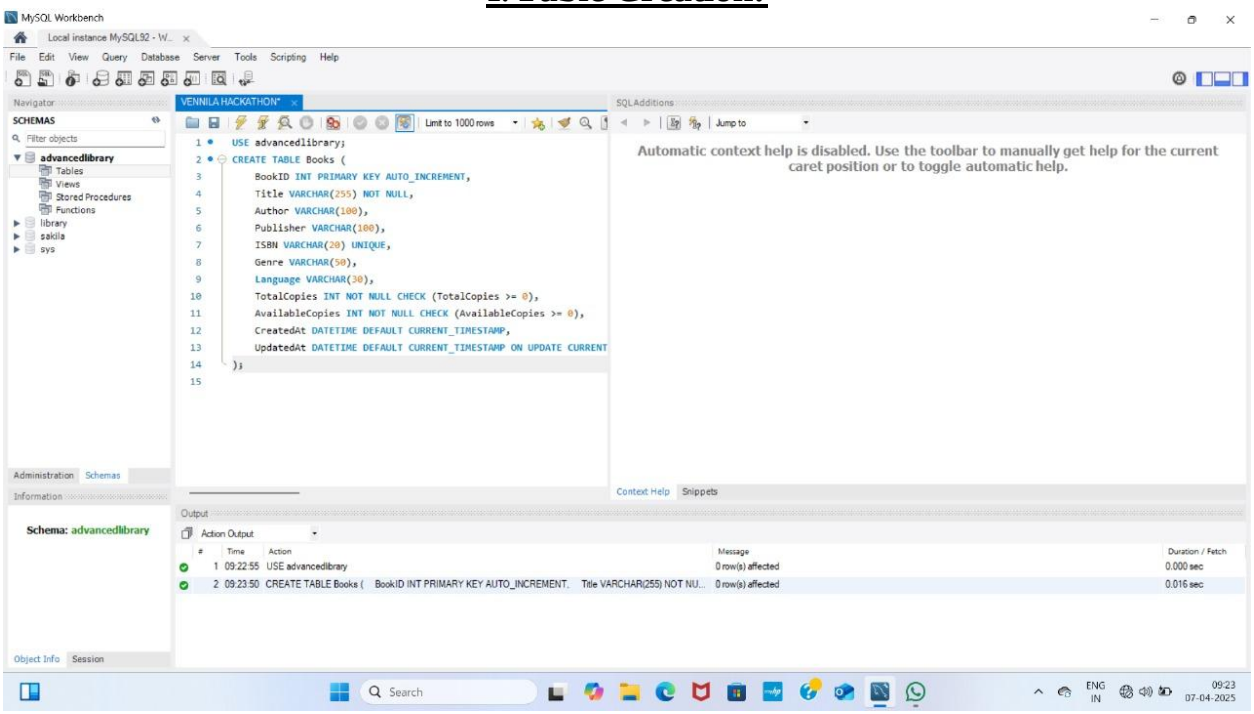
4. IMPLEMENTATION & RESULTS

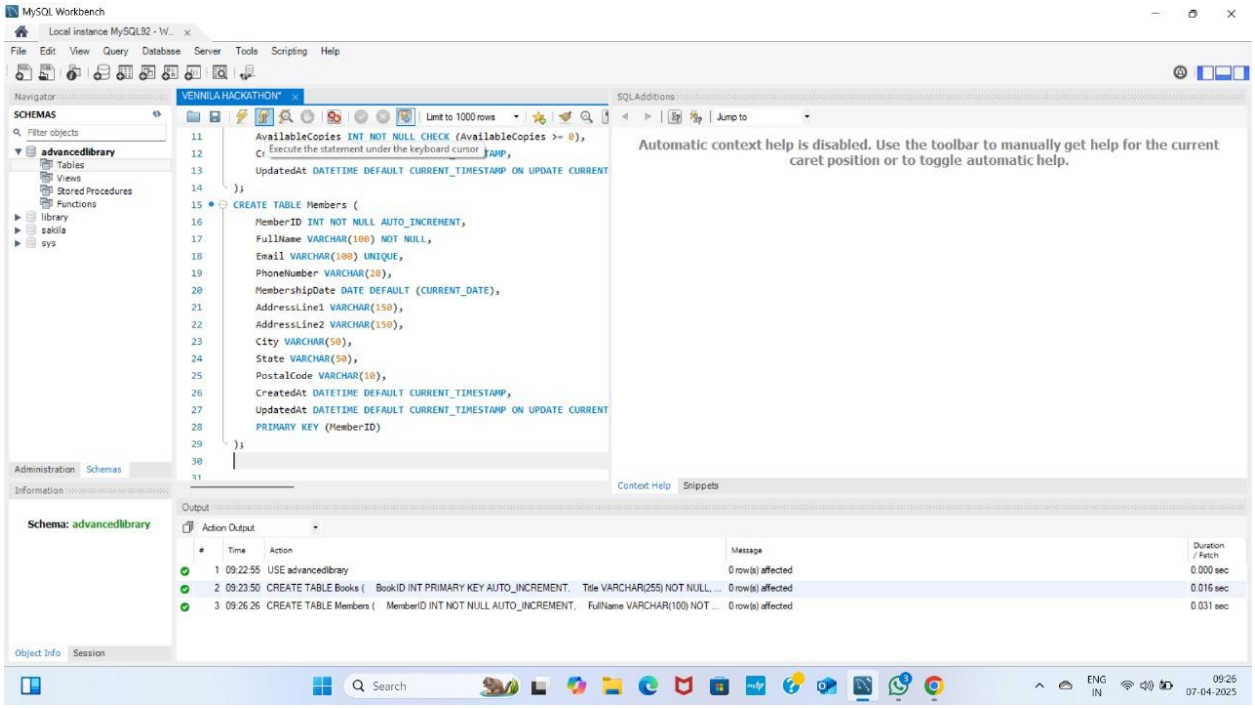
4.1 EXECUTION ENVIRONMENT

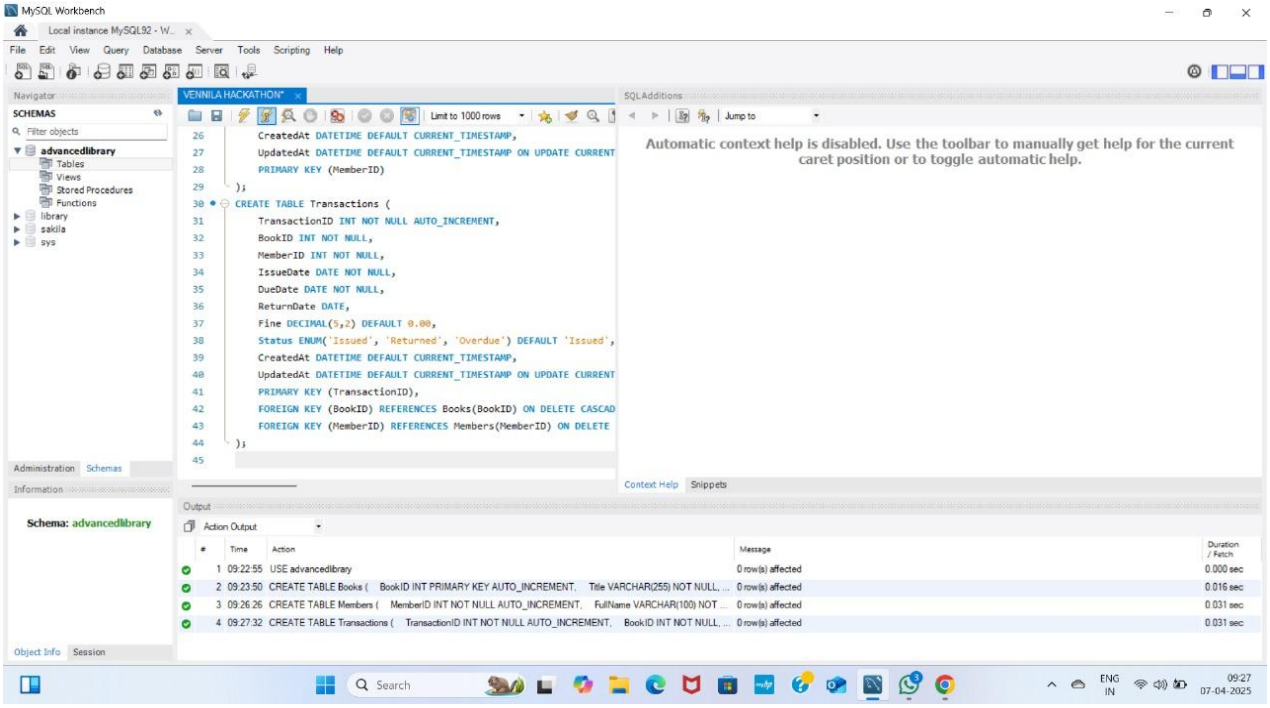
The implementation was executed using MySQL Workbench 9.2 on Windows 11.

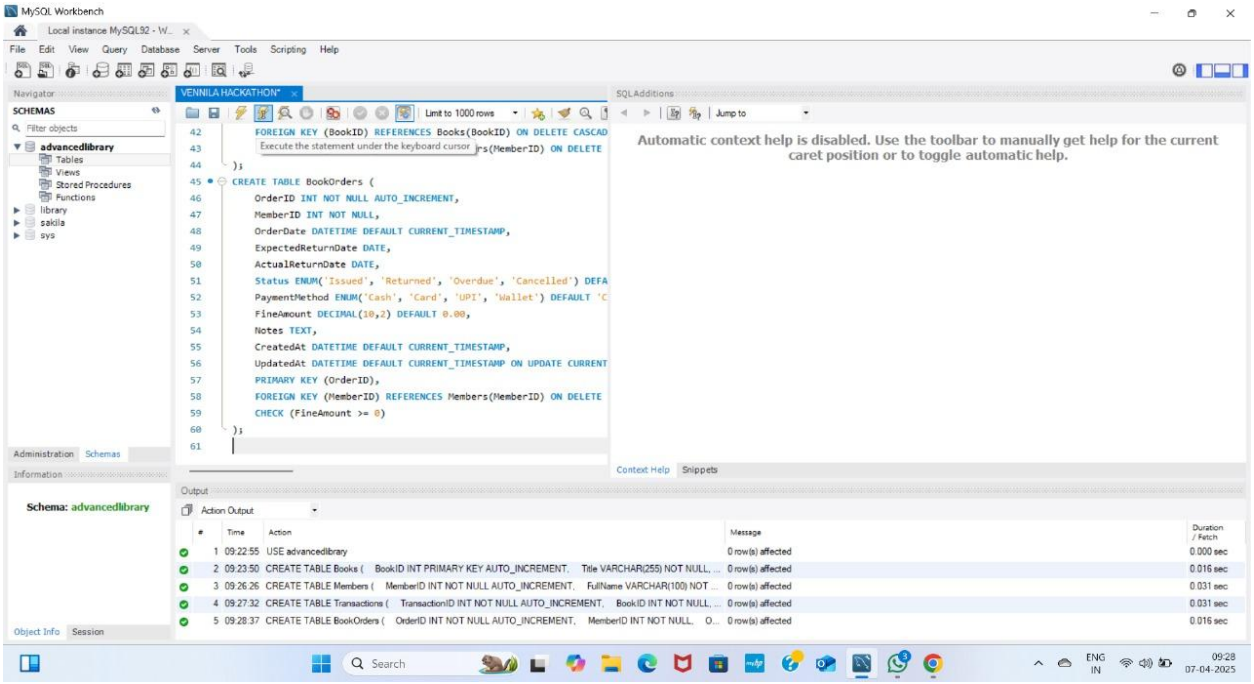
4.2 SCREENSHOTS OF EXECUTION RESULTS

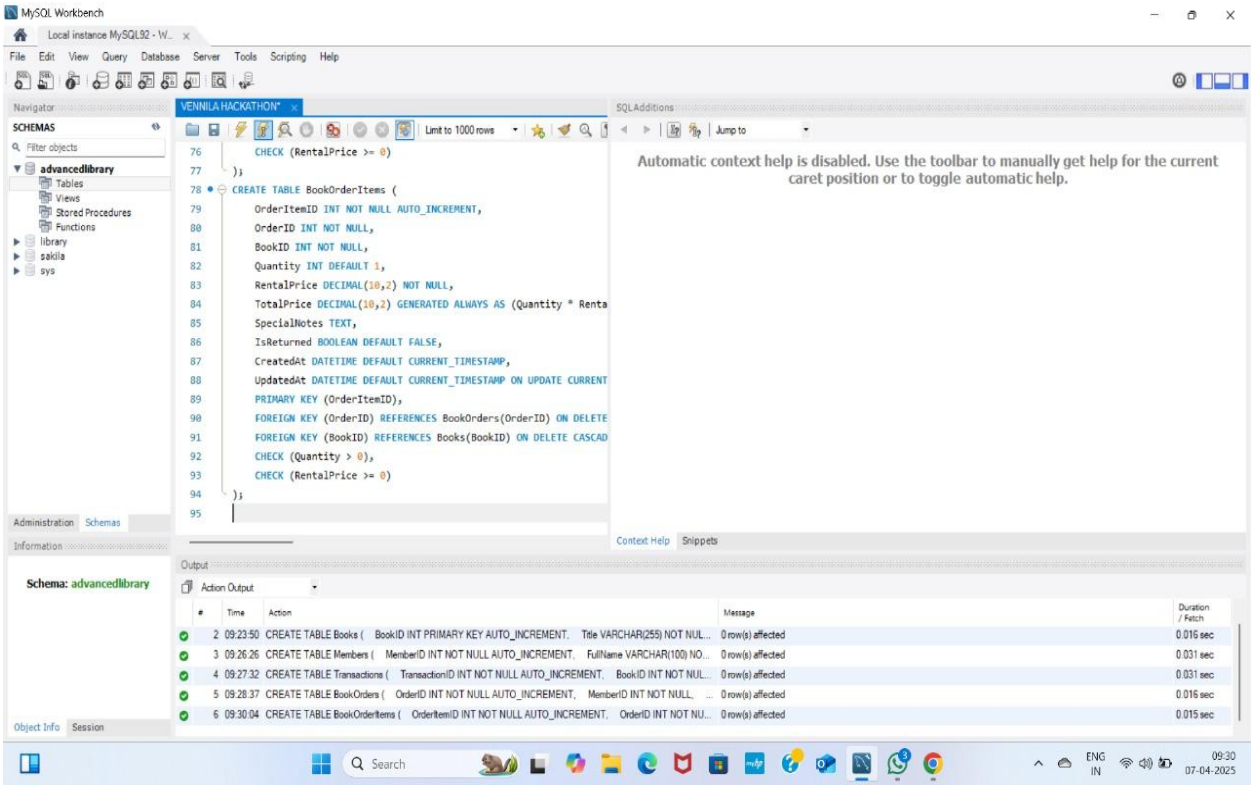
1. Table Creation:



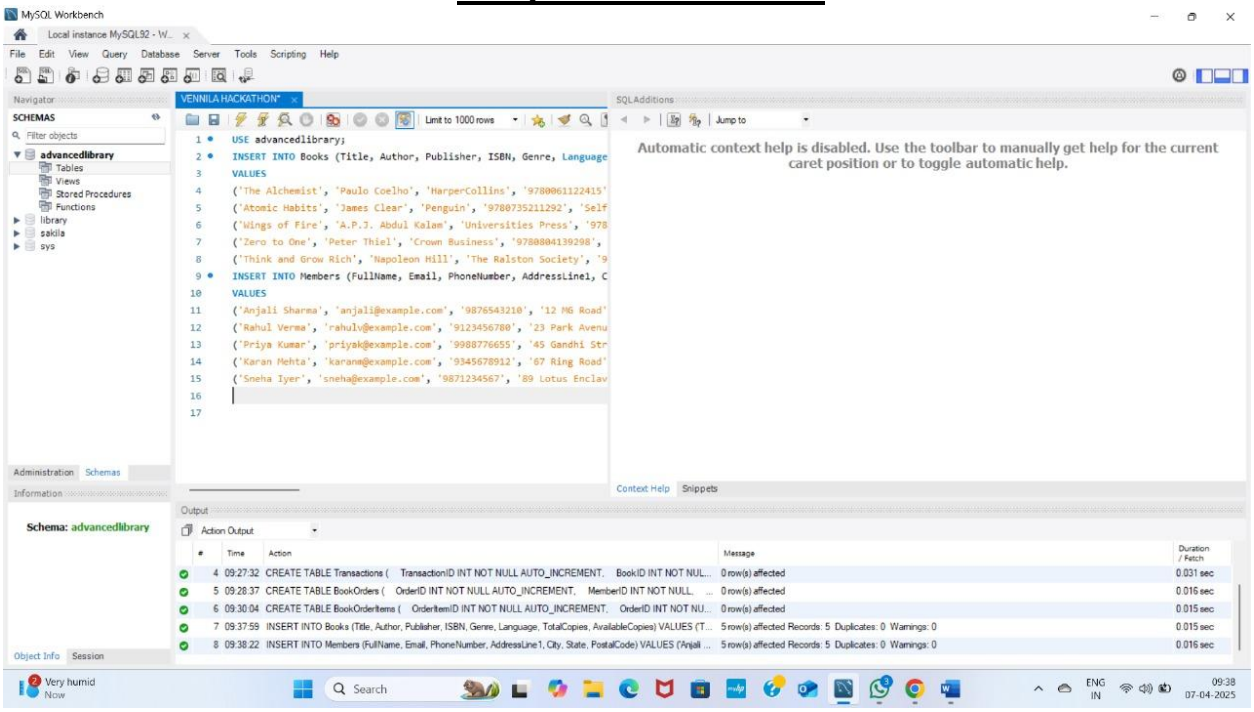

MySQL Workbench
Local instance MySQL92 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
advancedlibrary
Tables
Views
Stored Procedures
Functions
library
sakila
sys
Administration Schemas
Information
Schema: advancedlibrary
Object Info Session
VENNILA HACKATHON*
Limit to 1000 rows
AvailableCopies INT NOT NULL CHECK (AvailableCopies >= 0),
Cr Execute the statement under the keyboard cursor TAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
);
CREATE TABLE Members (
MemberID INT NOT NULL AUTO_INCREMENT,
FullName VARCHAR(100) NOT NULL,
Email VARCHAR(100) UNIQUE,
PhoneNumber VARCHAR(20),
MembershipDate DATE DEFAULT (CURRENT_DATE),
AddressLine1 VARCHAR(150),
AddressLine2 VARCHAR(150),
City VARCHAR(50),
State VARCHAR(50),
PostalCode VARCHAR(10),
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
PRIMARY KEY (MemberID)
);
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 09:22:55 USE advancedlibrary 0 row(s) affected 0.000 sec
2 09:23:50 CREATE TABLE Books (BookID INT PRIMARY KEY AUTO_INCREMENT, Title VARCHAR(255) NOT NULL, ... 0 row(s) affected 0.016 sec
3 09:26:26 CREATE TABLE Members (MemberID INT NOT NULL AUTO_INCREMENT, FullName VARCHAR(100) NOT ... 0 row(s) affected 0.031 sec
Search
ENG IN
09:26 07-04-2025

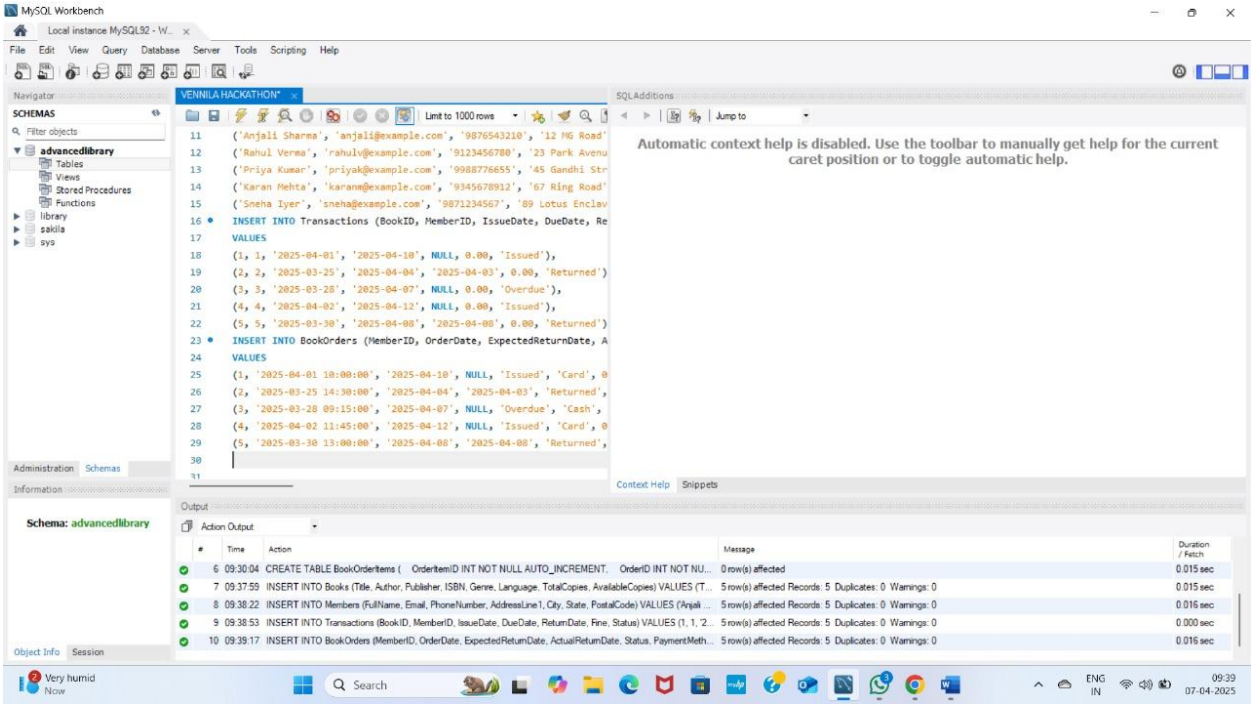

MySQL Workbench
Local instance MySQL92 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
advancedlibrary
Tables
Views
Stored Procedures
Functions
library
sakila
sys
Administration Schemas
Information
Schema: advancedlibrary
Object Info Session
VENNILA HACKATHON*
Limit to 1000 rows
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
PRIMARY KEY (MemberID)
);
CREATE TABLE Transactions (
TransactionID INT NOT NULL AUTO_INCREMENT,
BookID INT NOT NULL,
MemberID INT NOT NULL,
IssueDate DATE NOT NULL,
DueDate DATE NOT NULL,
ReturnDate DATE,
Fine DECIMAL(5,2) DEFAULT 0.00,
Status ENUM('Issued', 'Returned', 'Overdue') DEFAULT 'Issued',
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
PRIMARY KEY (TransactionID),
FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCAD
FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE
);
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 09:22:55 USE advancedlibrary 0 row(s) affected 0.000 sec
2 09:23:50 CREATE TABLE Books (BookID INT PRIMARY KEY AUTO_INCREMENT, Title VARCHAR(255) NOT NULL, ... 0 row(s) affected 0.016 sec
3 09:26:26 CREATE TABLE Members (MemberID INT NOT NULL AUTO_INCREMENT, FullName VARCHAR(100) NOT ... 0 row(s) affected 0.031 sec
4 09:27:32 CREATE TABLE Transactions (TransactionID INT NOT NULL AUTO_INCREMENT, BookID INT NOT NULL, ... 0 row(s) affected 0.031 sec
Search
ENG IN
09:27 07-04-2025

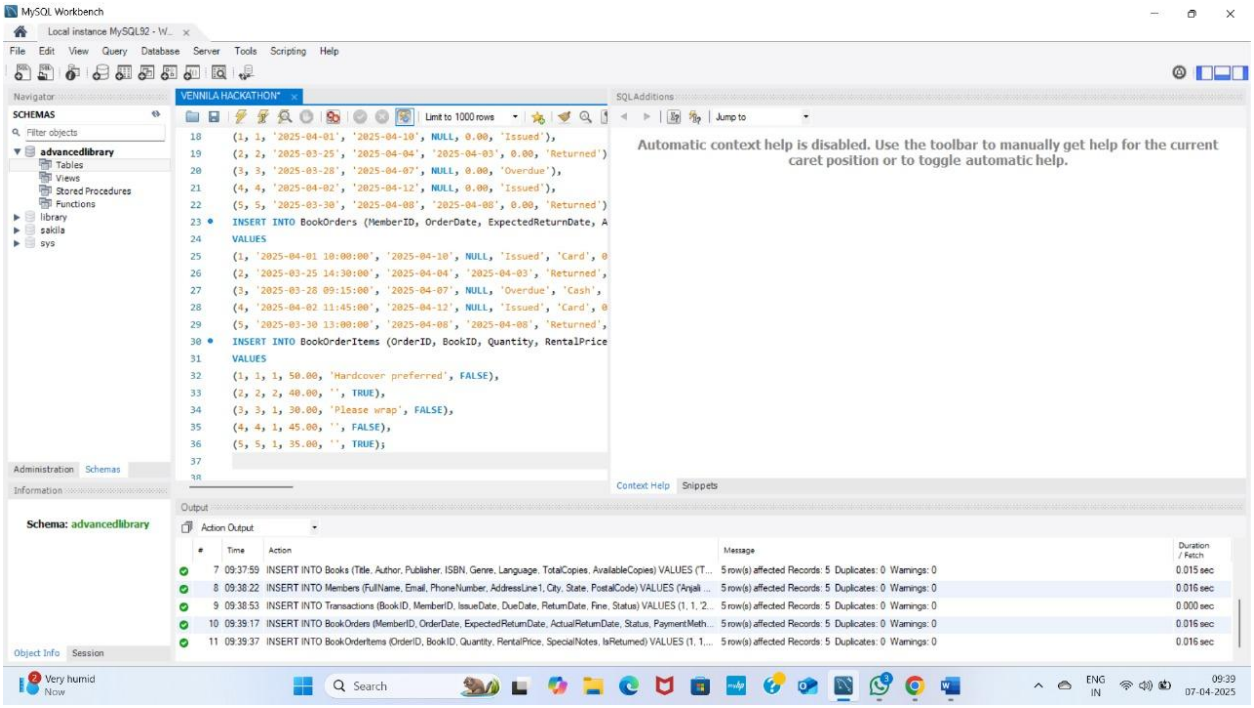

MySQL Workbench
Local instance MySQL92 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
advancedlibrary
Tables
Views
Stored Procedures
Functions
library
sakila
sys
Administration Schemas
Information
Schema: advancedlibrary
Object Info Session
VENNILA HACKATHON*
Limit to 1000 rows
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 09:22:55 USE advancedlibrary 0 row(s) affected 0.000 sec
2 09:23:50 CREATE TABLE Books (BookID INT PRIMARY KEY AUTO_INCREMENT, Title VARCHAR(255) NOT NULL, ... 0 row(s) affected 0.016 sec
3 09:26:26 CREATE TABLE Members (MemberID INT NOT NULL AUTO_INCREMENT, FullName VARCHAR(100) NOT ... 0 row(s) affected 0.031 sec
4 09:27:32 CREATE TABLE Transactions (TransactionID INT NOT NULL AUTO_INCREMENT, BookID INT NOT NULL, ... 0 row(s) affected 0.031 sec
5 09:28:37 CREATE TABLE BookOrders (OrderID INT NOT NULL AUTO_INCREMENT, MemberID INT NOT NULL, O... 0 row(s) affected 0.016 sec
Search
ENG IN
09:28 07-04-2025


MySQL Workbench
Local instance MySQL92 - W...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
advancedlibrary
Tables
Views
Stored Procedures
Functions
library
sakila
sys
Administration Schemas
Information
Schema: advancedlibrary
Object Info Session
VENNILA HACKATHON*
Limit to 1000 rows
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
2 09:23:50 CREATE TABLE Books (BookID INT PRIMARY KEY AUTO_INCREMENT, Title VARCHAR(255) NOT NUL... 0 row(s) affected 0.016 sec
3 09:26:26 CREATE TABLE Members (MemberID INT NOT NULL AUTO_INCREMENT, FullName VARCHAR(100) NO... 0 row(s) affected 0.031 sec
4 09:27:32 CREATE TABLE Transactions (TransactionID INT NOT NULL AUTO_INCREMENT, BookID INT NOT NUL... 0 row(s) affected 0.031 sec
5 09:28:37 CREATE TABLE BookOrders (OrderID INT NOT NULL AUTO_INCREMENT, MemberID INT NOT NULL, ... 0 row(s) affected 0.016 sec
6 09:30:04 CREATE TABLE BookOrderItems (OrderItemID INT NOT NULL AUTO_INCREMENT, OrderID INT NOT NU... 0 row(s) affected 0.015 sec
Search
ENG IN
09:30 07-04-2025

2. Sample Data Insertion:

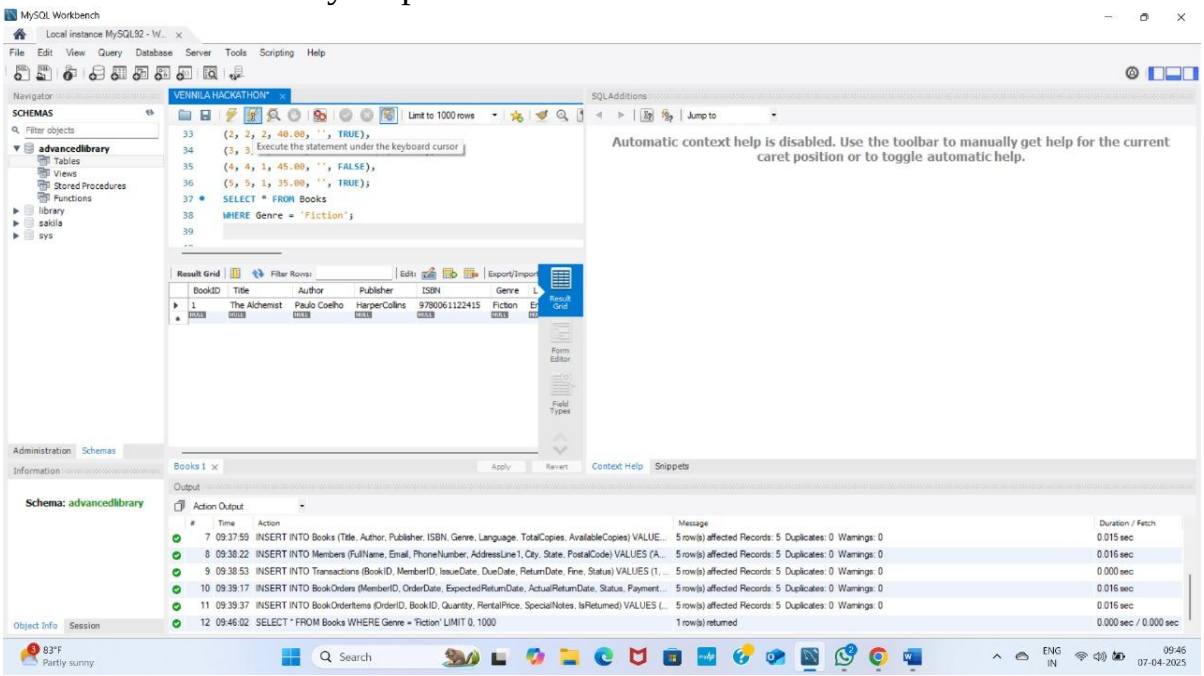




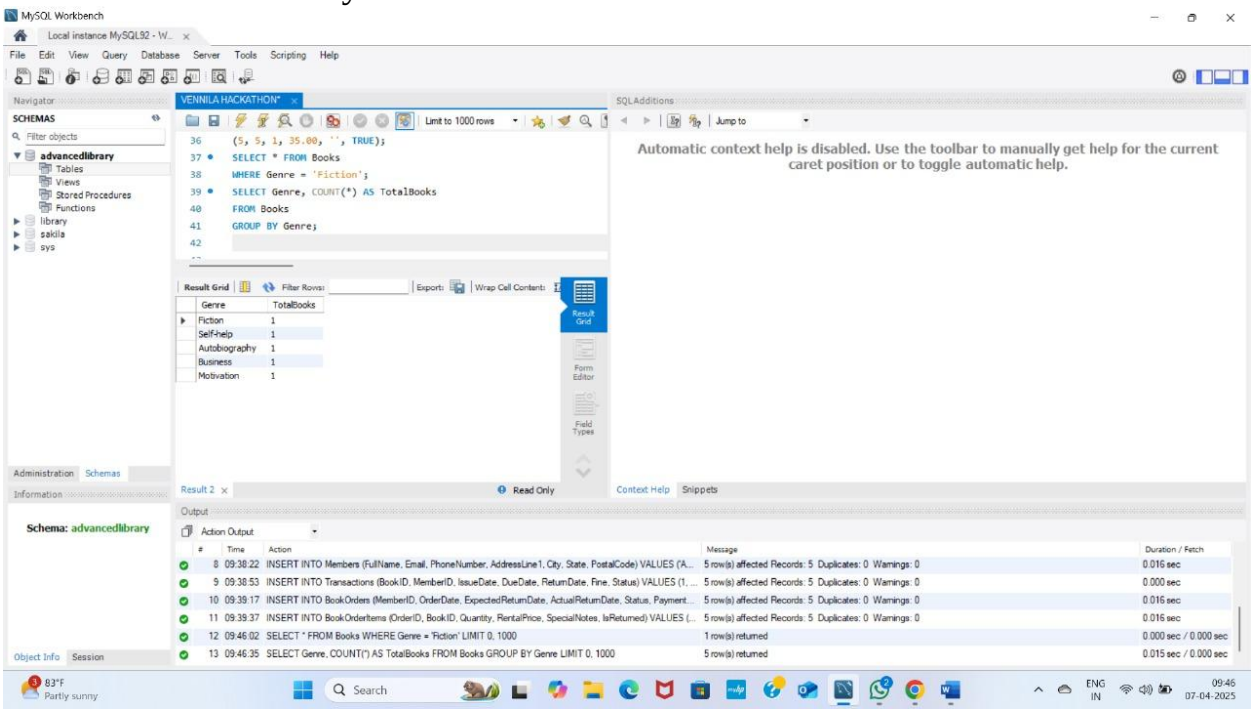


3. Data Retrieval Queries:

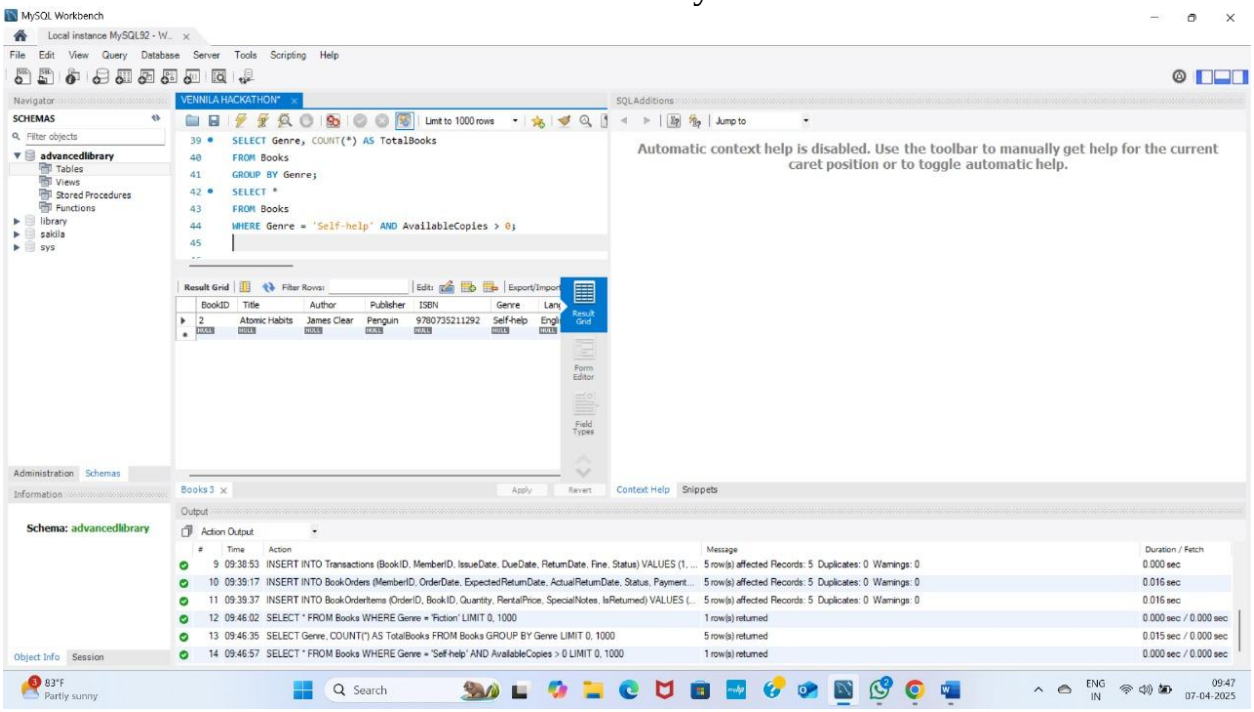
1. List All Books by a Specific Genre



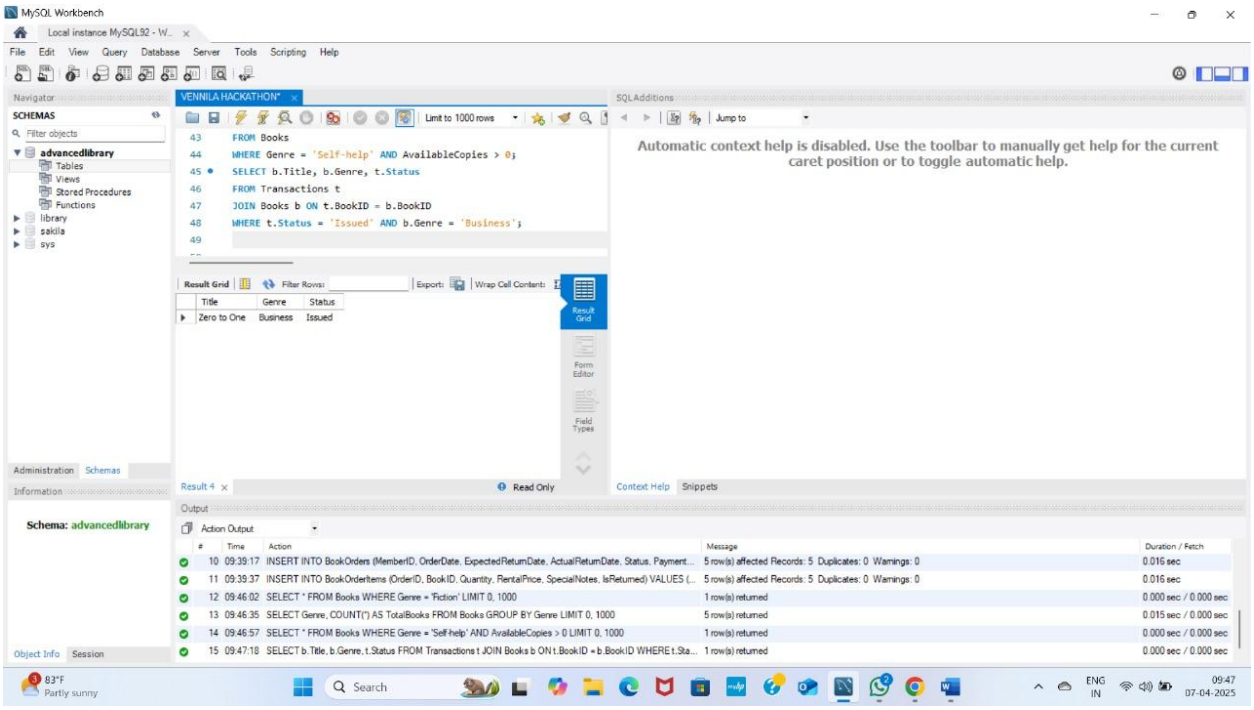
2. Count of Books by Each Genre



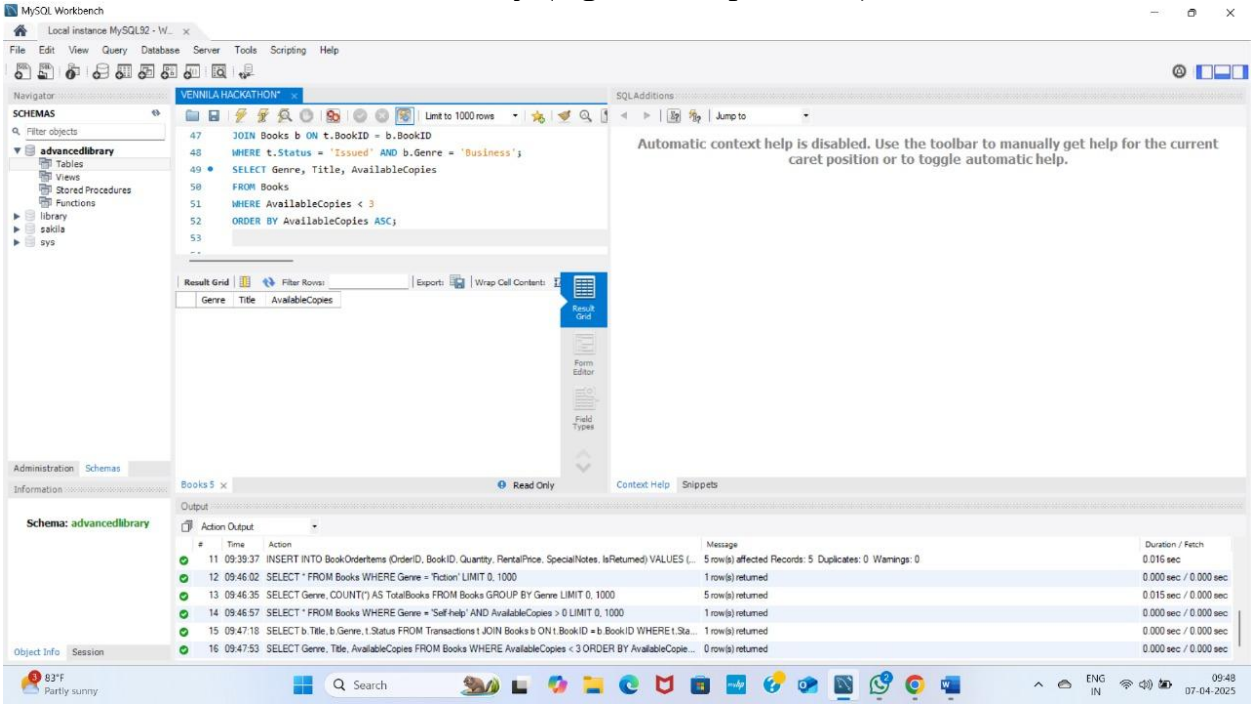
3. List Books of a Genre That Are Currently Available



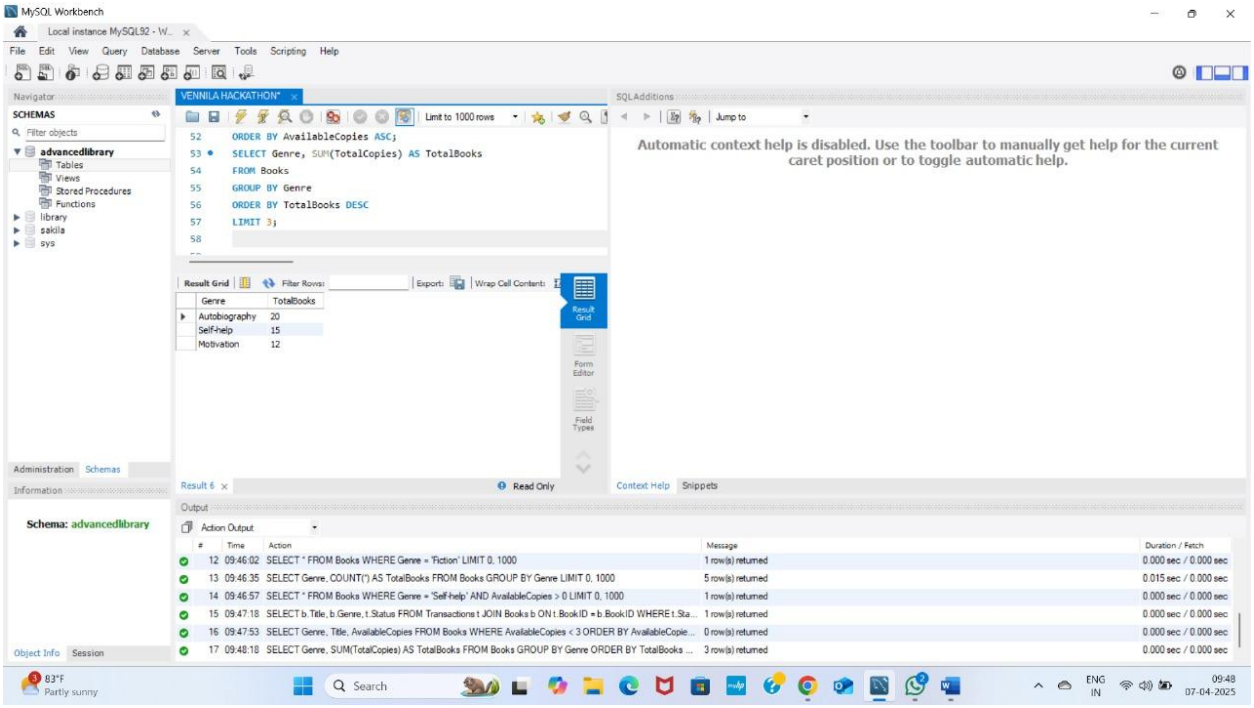
4. Get All Issued Books by Genre



5. Genres With Low Availability (e.g., < 3 Copies Left)



6. Top 3 Genres by Total Number of Books



5. GITHUB REPOSITORY

5.1 REPOSITORY LINK

https://github.com/vennilaaaa/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.

- **README.md** – Project overview, objective, and setup instructions.
- **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.

- **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 helps to manage books, users, and library activities more easily and quickly. It reduces manual work, saves time, and keeps records safe and organized. This system makes it easier for both students and librarians to use the library effectively.