

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;
```

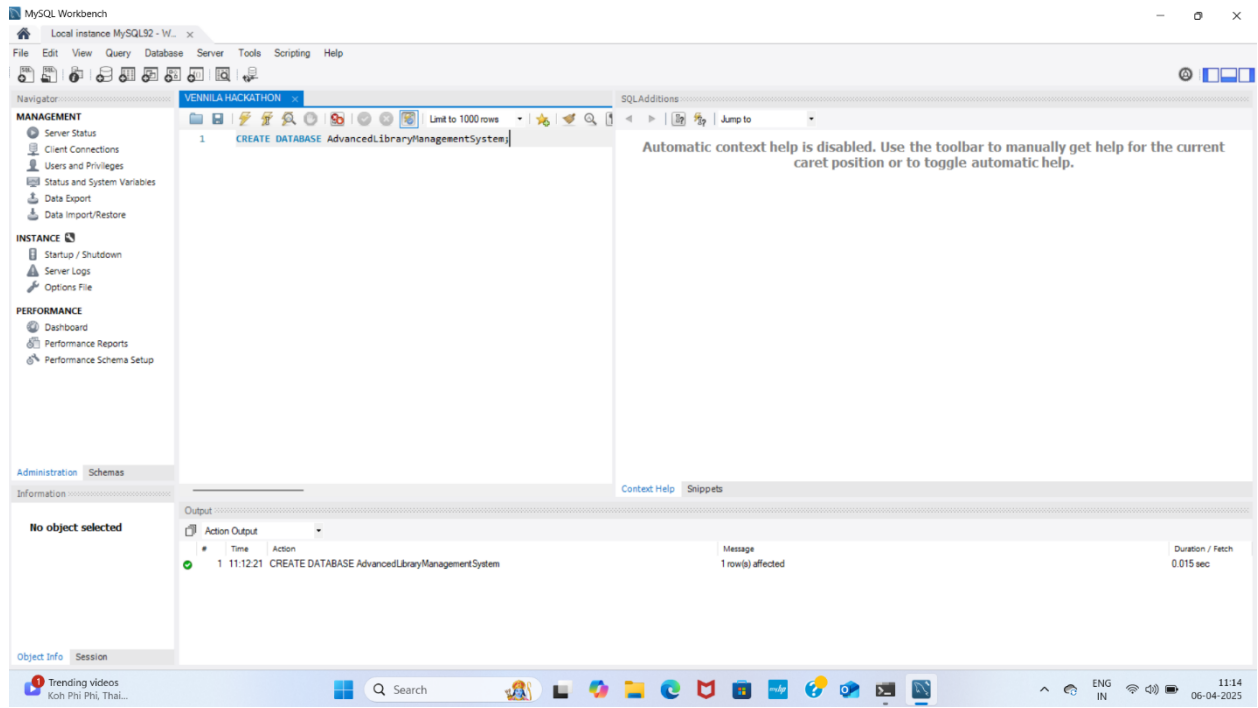


TABLE CREATION:

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:

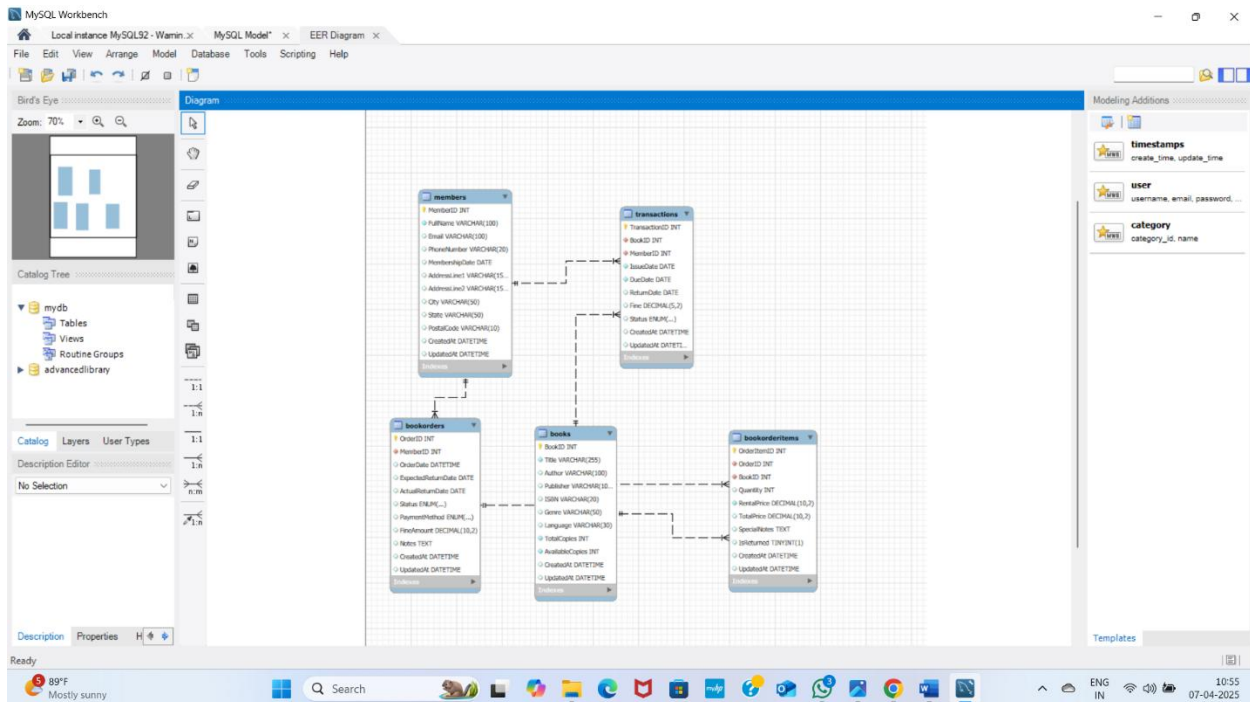
```

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 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;
```

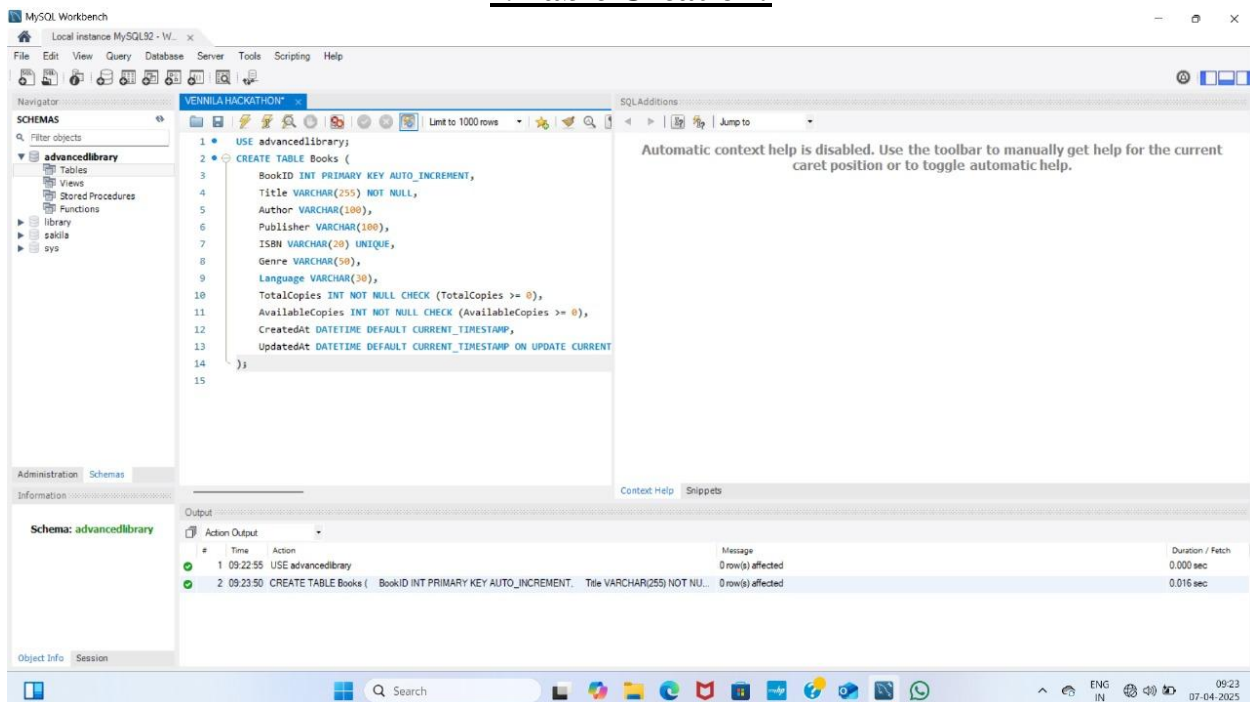
4. IMPLEMENTATION & RESULTS

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 MySQL82 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

advancedlibrary

Tables

Views

Stored Procedures

Functions

library

skilla

sys

VENNILA HACKATHON

```
11 AvailableCopies INT NOT NULL CHECK (AvailableCopies >= 0),
12 C Execute the statement under the keyboard cursor
13 UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
14 );
15 CREATE TABLE Members (
16 MemberID INT NOT NULL AUTO_INCREMENT,
17 FullName VARCHAR(100) NOT NULL,
18 Email VARCHAR(100) UNIQUE,
19 PhoneNumber VARCHAR(20),
20 MembershipDate DATE DEFAULT (CURRENT_DATE),
21 AddressLine1 VARCHAR(150),
22 AddressLine2 VARCHAR(150),
23 City VARCHAR(50),
24 State VARCHAR(50),
25 PostalCode VARCHAR(10),
26 CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
27 UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
28 PRIMARY KEY (MemberID)
29 );
30
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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

Object Info Session

Search

ENG IN

09:26 07-04-2025

MySQL Workbench

Local instance MySQL82 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

advancedlibrary

Tables

Views

Stored Procedures

Functions

library

skilla

sys

VENNILA HACKATHON

```
26 CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
27 UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
28 PRIMARY KEY (MemberID)
29 );
30 CREATE TABLE Transactions (
31 TransactionID INT NOT NULL AUTO_INCREMENT,
32 BookID INT NOT NULL,
33 MemberID INT NOT NULL,
34 IssueDate DATE NOT NULL,
35 DueDate DATE NOT NULL,
36 ReturnDate DATE,
37 Fine DECIMAL(5,2) DEFAULT 0.00,
38 Status ENUM('Issued', 'Returned', 'Overdue') DEFAULT 'Issued',
39 CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
40 UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT
41 PRIMARY KEY (TransactionID),
42 FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE
43 FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE
44 );
45
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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

Object Info Session

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

VENNILA HACKATHON

```
42 FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE
43
44
45 CREATE TABLE BookOrders (
46   OrderID INT NOT NULL AUTO_INCREMENT,
47   MemberID INT NOT NULL,
48   OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
49   ExpectedReturnDate DATE,
50   ActualReturnDate DATE,
51   Status ENUM('Issued', 'Returned', 'Overdue', 'Cancelled') DEFAULT 'Issued',
52   PaymentMethod ENUM('Cash', 'Card', 'UPI', 'Wallet') DEFAULT 'Cash',
53   FineAmount DECIMAL(10,2) DEFAULT 0.00,
54   Notes TEXT,
55   CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
56   UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
57   PRIMARY KEY (OrderID),
58   FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE CASCADE,
59   CHECK (FineAmount >= 0)
60 )
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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 NULL, ...	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, ...	0 row(s) affected	0.016 sec

Object Info Session

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

VENNILA HACKATHON

```
76 CHECK (RentalPrice >= 0)
77
78 CREATE TABLE BookOrderItems (
79   OrderItemID INT NOT NULL AUTO_INCREMENT,
80   OrderID INT NOT NULL,
81   BookID INT NOT NULL,
82   Quantity INT DEFAULT 1,
83   RentalPrice DECIMAL(10,2) NOT NULL,
84   TotalPrice DECIMAL(10,2) GENERATED ALWAYS AS (Quantity * RentalPrice) STORED,
85   SpecialNotes TEXT,
86   IsReturned BOOLEAN DEFAULT FALSE,
87   CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
88   UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
89   PRIMARY KEY (OrderItemID),
90   FOREIGN KEY (OrderID) REFERENCES BookOrders(OrderID) ON DELETE CASCADE,
91   FOREIGN KEY (BookID) REFERENCES Books(BookID) ON DELETE CASCADE,
92   CHECK (Quantity > 0),
93   CHECK (RentalPrice >= 0)
94 )
95
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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 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 NULL, ...	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, ...	0 row(s) affected	0.016 sec
6	09:30:04	CREATE TABLE BookOrderItems (OrderItemID INT NOT NULL AUTO_INCREMENT, OrderID INT NOT NULL, ...	0 row(s) affected	0.015 sec

Object Info Session

09:30 07-04-2025

2. Sample Data Insertion:

The screenshot shows the MySQL Workbench interface with a SQL editor window titled 'VENNILA HACKATHON'. The editor contains SQL statements to create tables and insert sample data into a database named 'advancedlibrary'.

```
1 USE advancedlibrary;
2 INSERT INTO Books (Title, Author, Publisher, ISBN, Genre, Language
3 VALUES
4 ('The Alchemist', 'Paulo Coelho', 'HarperCollins', '9780061122415'
5 ('Atomic Habits', 'James Clear', 'Penguin', '9780735211292', 'Self
6 ('Wings of Fire', 'A.P.J. Abdul Kalam', 'Universities Press', '978
7 ('Zero to One', 'Peter Thiel', 'Crown Business', '9780804139298',
8 ('Think and Grow Rich', 'Napoleon Hill', 'The Ralston Society', '9
9 INSERT INTO Members (FullName, Email, PhoneNumber, AddressLine1, C
10 VALUES
11 ('Anjali Sharma', 'anjali@example.com', '9876543210', '12 MG Road'
12 ('Rahul Verma', 'rahul@example.com', '9123456780', '23 Park Avenu
13 ('Priya Kumar', 'priya@example.com', '9988776655', '45 Gandhi Str
14 ('Karan Mehta', 'karan@example.com', '9345678912', '67 Ring Road'
15 ('Sneha Iyer', 'sneha@example.com', '9871234567', '89 Lotus Enclav
16
17
```

The Output window at the bottom shows the execution results of the SQL statements:

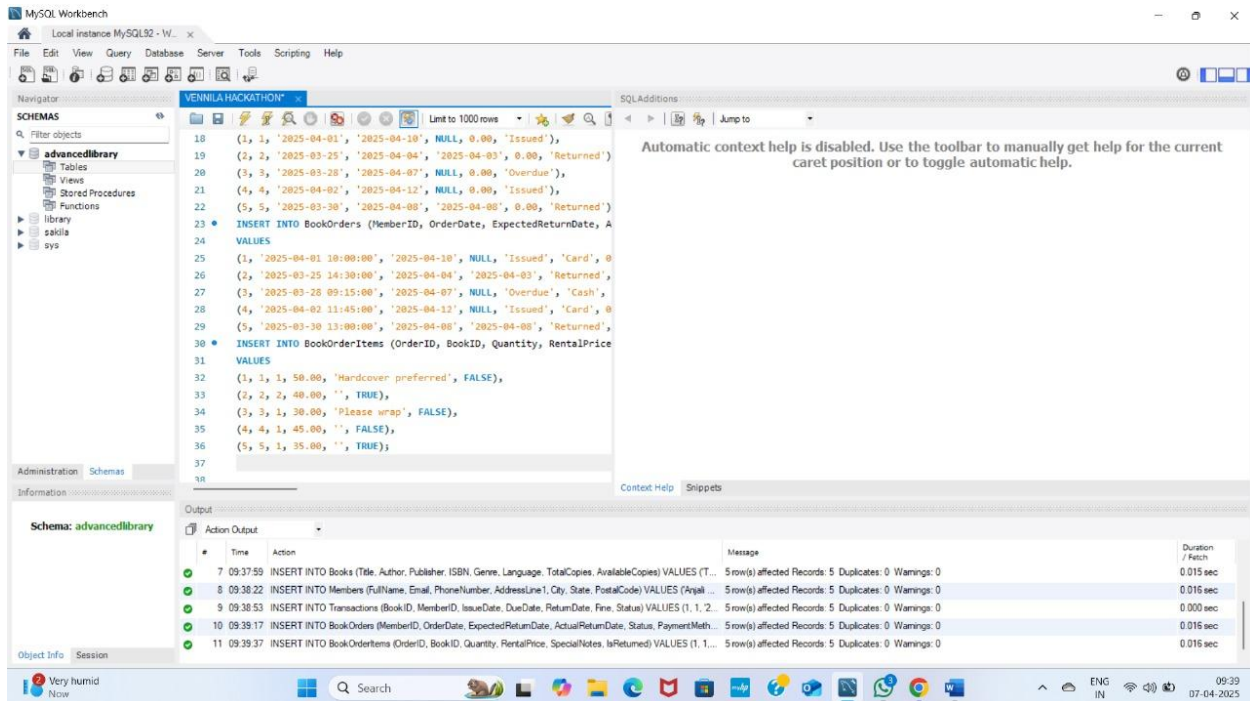
#	Time	Action	Message	Duration / Fetch
4	09:27:32	CREATE TABLE Transactions (TransactionID INT NOT NULL AUTO_INCREMENT, BookID INT NOT NULL, MemberID INT NOT NULL, IssueDate DATE, DueDate DATE, ReturnDate DATE, Fine DECIMAL(10,2), Status VARCHAR(20), PaymentMethod VARCHAR(20))	0 row(s) affected	0.031 sec
5	09:28:37	CREATE TABLE BookOrders (OrderID INT NOT NULL AUTO_INCREMENT, MemberID INT NOT NULL, OrderDate DATE, ExpectedReturnDate DATE, ActualReturnDate DATE, Status VARCHAR(20), PaymentMethod VARCHAR(20))	0 row(s) affected	0.016 sec
6	09:30:04	CREATE TABLE BookOrderItems (OrderItemID INT NOT NULL AUTO_INCREMENT, OrderID INT NOT NULL, BookID INT NOT NULL, IssueDate DATE, DueDate DATE, ReturnDate DATE, Fine DECIMAL(10,2), Status VARCHAR(20), PaymentMethod VARCHAR(20))	0 row(s) affected	0.015 sec
7	09:37:59	INSERT INTO Books (Title, Author, Publisher, ISBN, Genre, Language, TotalCopies, AvailableCopies) VALUES (1, 'The Alchemist', 'Paulo Coelho', '9780061122415', 'Fiction', 'English', 10, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
8	09:38:22	INSERT INTO Members (FullName, Email, PhoneNumber, AddressLine1, City, State, PostalCode) VALUES ('Anjali Sharma', 'anjali@example.com', '9876543210', '12 MG Road', 'Bangalore', 'Karnataka', '560001')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec

The screenshot shows the MySQL Workbench interface with a SQL editor window titled 'VENNILA HACKATHON'. The editor contains SQL statements to create tables and insert sample data into a database named 'advancedlibrary'.

```
11 ('Anjali Sharma', 'anjali@example.com', '9876543210', '12 MG Road'
12 ('Rahul Verma', 'rahul@example.com', '9123456780', '23 Park Avenu
13 ('Priya Kumar', 'priya@example.com', '9988776655', '45 Gandhi Str
14 ('Karan Mehta', 'karan@example.com', '9345678912', '67 Ring Road'
15 ('Sneha Iyer', 'sneha@example.com', '9871234567', '89 Lotus Enclav
16 INSERT INTO Transactions (BookID, MemberID, IssueDate, DueDate, Re
17 VALUES
18 (1, 1, '2025-04-01', '2025-04-10', NULL, 0.00, 'Issued'),
19 (2, 2, '2025-03-25', '2025-04-04', '2025-04-03', 0.00, 'Returned')
20 (3, 3, '2025-03-28', '2025-04-07', NULL, 0.00, 'Overdue'),
21 (4, 4, '2025-04-02', '2025-04-12', NULL, 0.00, 'Issued'),
22 (5, 5, '2025-03-30', '2025-04-08', '2025-04-08', 0.00, 'Returned')
23 INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, A
24 VALUES
25 (1, '2025-04-01 10:00:00', '2025-04-10', NULL, 'Issued', 'Card', 0
26 (2, '2025-03-25 14:30:00', '2025-04-04', '2025-04-03', 'Returned',
27 (3, '2025-03-28 09:15:00', '2025-04-07', NULL, 'Overdue', 'Cash', 0
28 (4, '2025-04-02 11:45:00', '2025-04-12', NULL, 'Issued', 'Card', 0
29 (5, '2025-03-30 13:00:00', '2025-04-08', '2025-04-08', 'Returned',
30
31
```

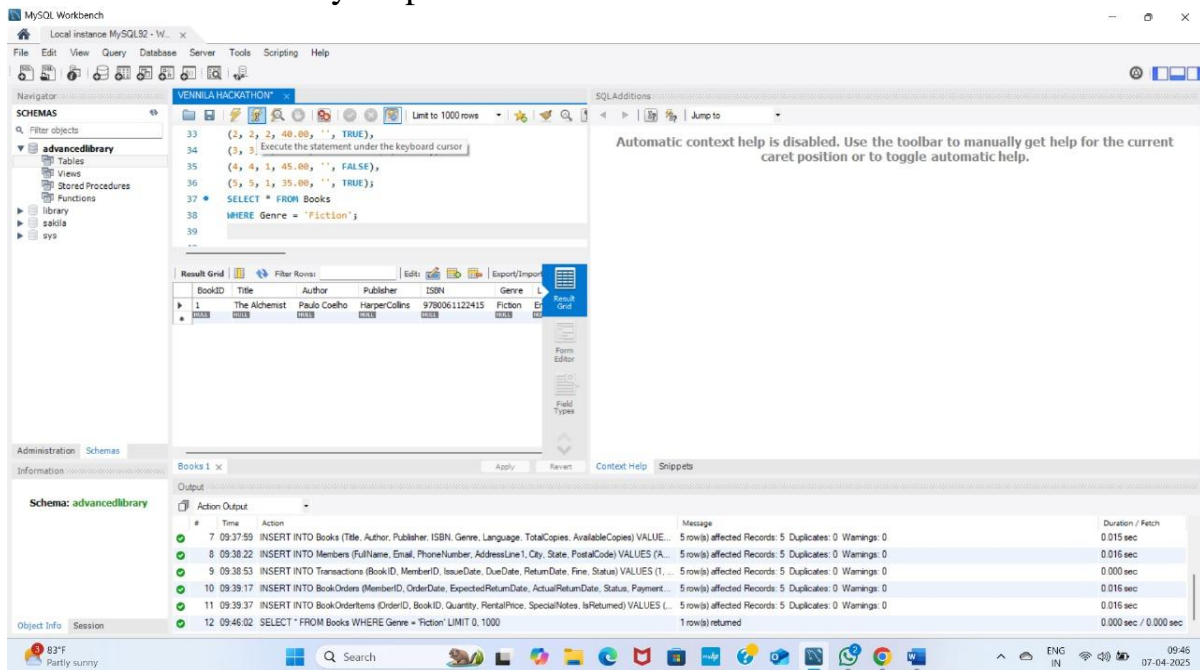
The Output window at the bottom shows the execution results of the SQL statements:

#	Time	Action	Message	Duration / Fetch
6	09:30:04	CREATE TABLE BookOrderItems (OrderItemID INT NOT NULL AUTO_INCREMENT, OrderID INT NOT NULL, BookID INT NOT NULL, IssueDate DATE, DueDate DATE, ReturnDate DATE, Fine DECIMAL(10,2), Status VARCHAR(20), PaymentMethod VARCHAR(20))	0 row(s) affected	0.015 sec
7	09:37:59	INSERT INTO Books (Title, Author, Publisher, ISBN, Genre, Language, TotalCopies, AvailableCopies) VALUES (1, 'The Alchemist', 'Paulo Coelho', '9780061122415', 'Fiction', 'English', 10, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
8	09:38:22	INSERT INTO Members (FullName, Email, PhoneNumber, AddressLine1, City, State, PostalCode) VALUES ('Anjali Sharma', 'anjali@example.com', '9876543210', '12 MG Road', 'Bangalore', 'Karnataka', '560001')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
9	09:38:53	INSERT INTO Transactions (BookID, MemberID, IssueDate, DueDate, ReturnDate, Fine, Status) VALUES (1, 1, 2025-04-01, 2025-04-10, NULL, 0.00, 'Issued')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
10	09:39:17	INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, ActualReturnDate, Status, PaymentMethod) VALUES (1, 2025-04-01 10:00:00, 2025-04-10, NULL, 'Issued', 'Card')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec



3. Data Retrieval Queries:

1. List All Books by a Specific Genre



2. Count of Books by Each Genre

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
36 (5, 5, 1, 35.00, '', TRUE);  
37 SELECT * FROM Books  
38 WHERE Genre = 'Fiction';  
39 SELECT Genre, COUNT(*) AS TotalBooks  
40 FROM Books  
41 GROUP BY Genre;  
42
```

The Results grid displays the following data:

Genre	TotalBooks
Fiction	1
Self-Help	1
Autobiography	1
Business	1
Motivation	1

The Output tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
8	09:38:22	INSERT INTO Members (FullName, Email, PhoneNumber, AddressLine1, City, State, PostalCode) VALUES (A...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
9	09:38:53	INSERT INTO Transactions (BookID, MemberID, IssueDate, DueDate, ReturnDate, Fine, Status) VALUES (1, ...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
10	09:39:17	INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, ActualReturnDate, Status, Payment...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
11	09:39:37	INSERT INTO BookOrderItems (OrderID, BookID, Quantity, RentalPrice, SpecialNotes, IsReturned) VALUES (...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
12	09:46:02	SELECT * FROM Books WHERE Genre = 'Fiction' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	09:46:35	SELECT Genre, COUNT(*) AS TotalBooks FROM Books GROUP BY Genre LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec

3. List Books of a Genre That Are Currently Available

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
39 SELECT Genre, COUNT(*) AS TotalBooks  
40 FROM Books  
41 GROUP BY Genre;  
42 SELECT *  
43 FROM Books  
44 WHERE Genre = 'Self-help' AND AvailableCopies > 0;  
45
```

The Results grid displays the following data:

BookID	Title	Author	Publisher	ISBN	Genre	Lang
2	Atomic Habits	James Clear	Penguin	9780735211292	Self-help	Engl

The Output tab shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
9	09:38:53	INSERT INTO Transactions (BookID, MemberID, IssueDate, DueDate, ReturnDate, Fine, Status) VALUES (1, ...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
10	09:39:17	INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, ActualReturnDate, Status, Payment...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
11	09:39:37	INSERT INTO BookOrderItems (OrderID, BookID, Quantity, RentalPrice, SpecialNotes, IsReturned) VALUES (...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
12	09:46:02	SELECT * FROM Books WHERE Genre = 'Fiction' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	09:46:35	SELECT Genre, COUNT(*) AS TotalBooks FROM Books GROUP BY Genre LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
14	09:46:57	SELECT * FROM Books WHERE Genre = 'Self-help' AND AvailableCopies > 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

4. Get All Issued Books by Genre

The screenshot shows the MySQL Workbench interface with a query window titled "VENNILA HACKATHON". The query is as follows:

```
43 FROM Books
44 WHERE Genre = 'Self-help' AND AvailableCopies > 0;
45 SELECT b.Title, b.Genre, t.Status
46 FROM Transactions t
47 JOIN Books b ON t.BookID = b.BookID
48 WHERE t.Status = 'Issued' AND b.Genre = 'Business';
49
```

The query results are displayed in a table with columns: Title, Genre, Status. The results show books in the 'Business' genre that are 'Issued'.

The Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
10	09:39:17	INSERT INTO BookOrders (MemberID, OrderDate, ExpectedReturnDate, ActualReturnDate, Status, Payment...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
11	09:39:37	INSERT INTO BookOrderItems (OrderID, BookID, Quantity, RentalPrice, SpecialNotes, IsReturned) VALUES (...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
12	09:46:02	SELECT * FROM Books WHERE Genre = 'Fiction' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	09:46:35	SELECT Genre, COUNT(*) AS TotalBooks FROM Books GROUP BY Genre LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
14	09:46:57	SELECT * FROM Books WHERE Genre = 'Self-help' AND AvailableCopies > 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
15	09:47:18	SELECT b.Title, b.Genre, t.Status FROM Transactions t JOIN Books b ON t.BookID = b.BookID WHERE t.Sta...	1 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with a query window titled "VENNILA HACKATHON". The query is as follows:

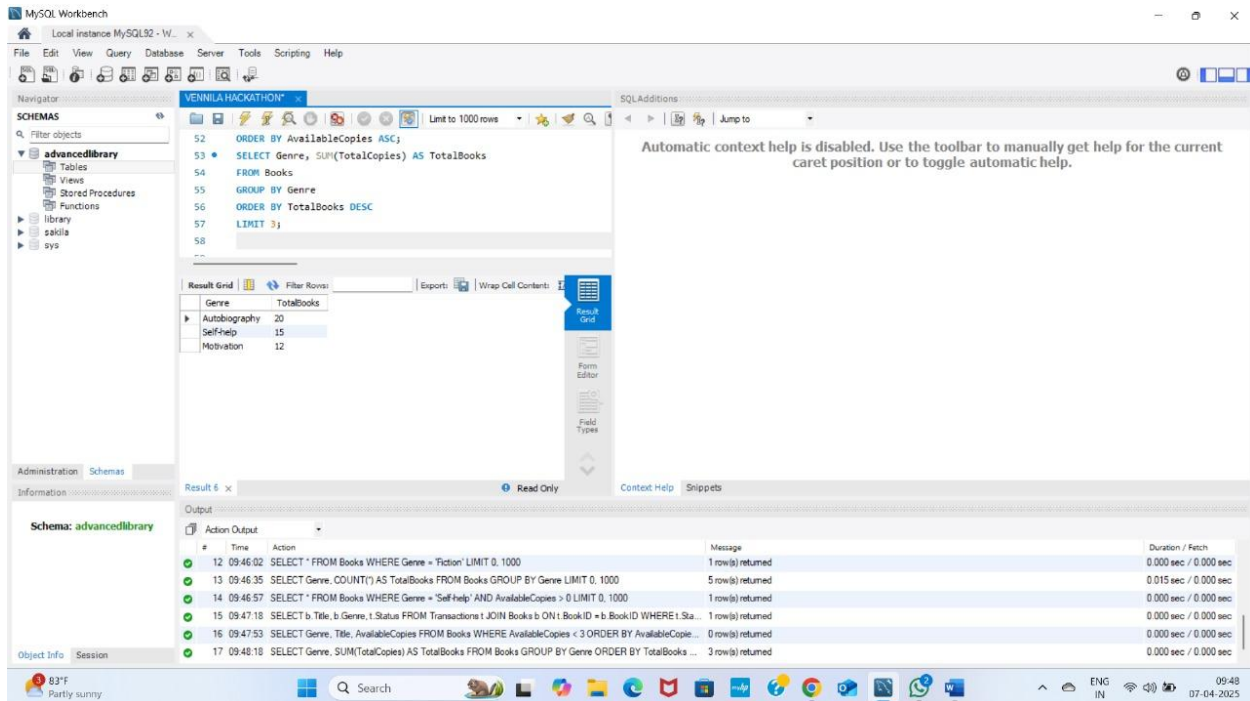
```
47 JOIN Books b ON t.BookID = b.BookID
48 WHERE t.Status = 'Issued' AND b.Genre = 'Business';
49 SELECT Genre, Title, AvailableCopies
50 FROM Books
51 WHERE AvailableCopies < 3
52 ORDER BY AvailableCopies ASC;
53
```

The query results are displayed in a table with columns: Genre, Title, AvailableCopies. The results show books in the 'Business' genre that have low availability (less than 3 copies left).

The Output window shows the following results:

#	Time	Action	Message	Duration / Fetch
11	09:39:37	INSERT INTO BookOrderItems (OrderID, BookID, Quantity, RentalPrice, SpecialNotes, IsReturned) VALUES (...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
12	09:46:02	SELECT * FROM Books WHERE Genre = 'Fiction' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	09:46:35	SELECT Genre, COUNT(*) AS TotalBooks FROM Books GROUP BY Genre LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
14	09:46:57	SELECT * FROM Books WHERE Genre = 'Self-help' AND AvailableCopies > 0 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
15	09:47:18	SELECT b.Title, b.Genre, t.Status FROM Transactions t JOIN Books b ON t.BookID = b.BookID WHERE t.Sta...	1 row(s) returned	0.000 sec / 0.000 sec
16	09:47:53	SELECT Genre, Title, AvailableCopies FROM Books WHERE AvailableCopies < 3 ORDER BY AvailableCopie...	0 row(s) returned	0.000 sec / 0.000 sec

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.