Name : RAMYA R

Company: CODTECH IT SOLUTIONS

Intern ID : CT08DS7025

Domain : SQL

Duration: Augest to September 2024

Mentor : Muzammil Ahmed

Library Management System

Introduction:

The Library Management System (LMS) is a sophisticated database-driven application designed to facilitate the efficient management of a library's book inventory, member records, and borrow/return transactions. This project aims to provide a practical learning experience in SQL commands, database design principles, and data manipulation techniques. The LMS will help librarians streamline their operations, improve user experience, and maintain accurate records.

Objectives:

- **Database Design**: To design a relational database that effectively captures the relationships between books, members, and transactions.
- **Data Manipulation**: To implement SQL queries for inserting, updating, deleting, and retrieving data.
- **User Management**: To manage library members and their borrowing privileges.
- **Reporting**: To generate reports and statistics that provide insights into library usage and inventory status.
- **Future Enhancements**: To outline potential improvements and features for the LMS.

Database Design:

The database consists of five primary tables:

- Books
- Members
- Transactions
- BookCategories
- MembershipTypes

SQL Queries:

1) Create a Database or Schema:

```
CREATE DATABASE Library_Management_System;
```

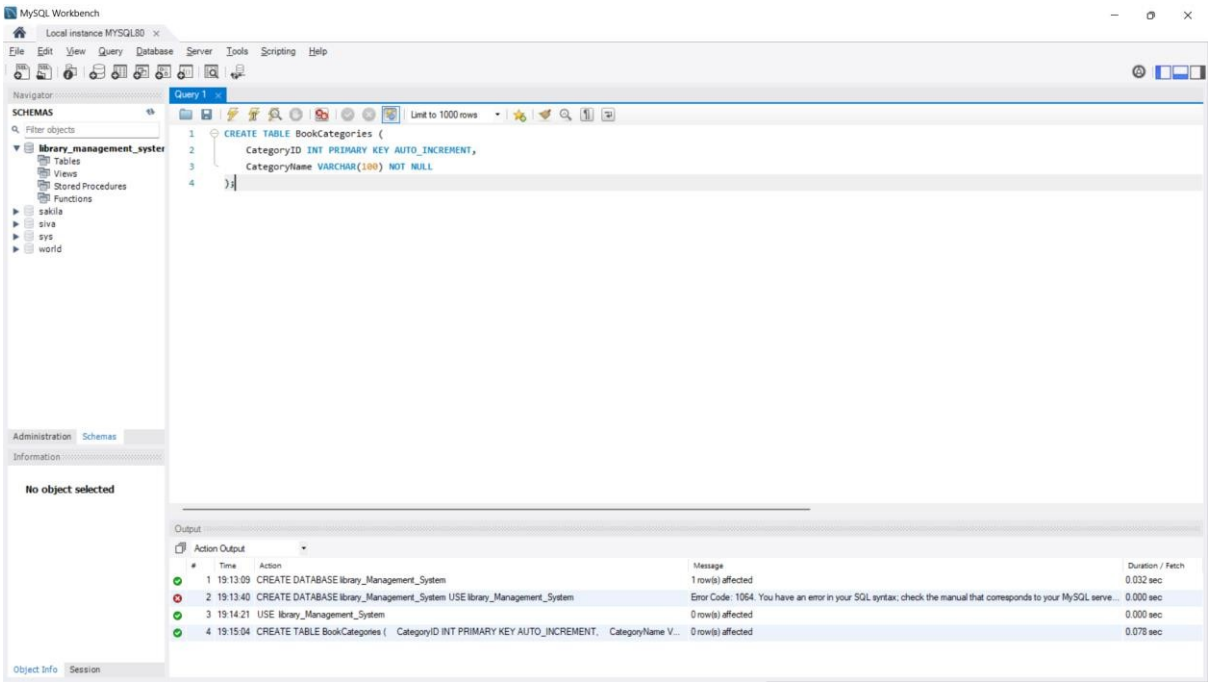
2) Use Database for Data Collection:

```
USE Library_Management_System;
```

3) Create Table:

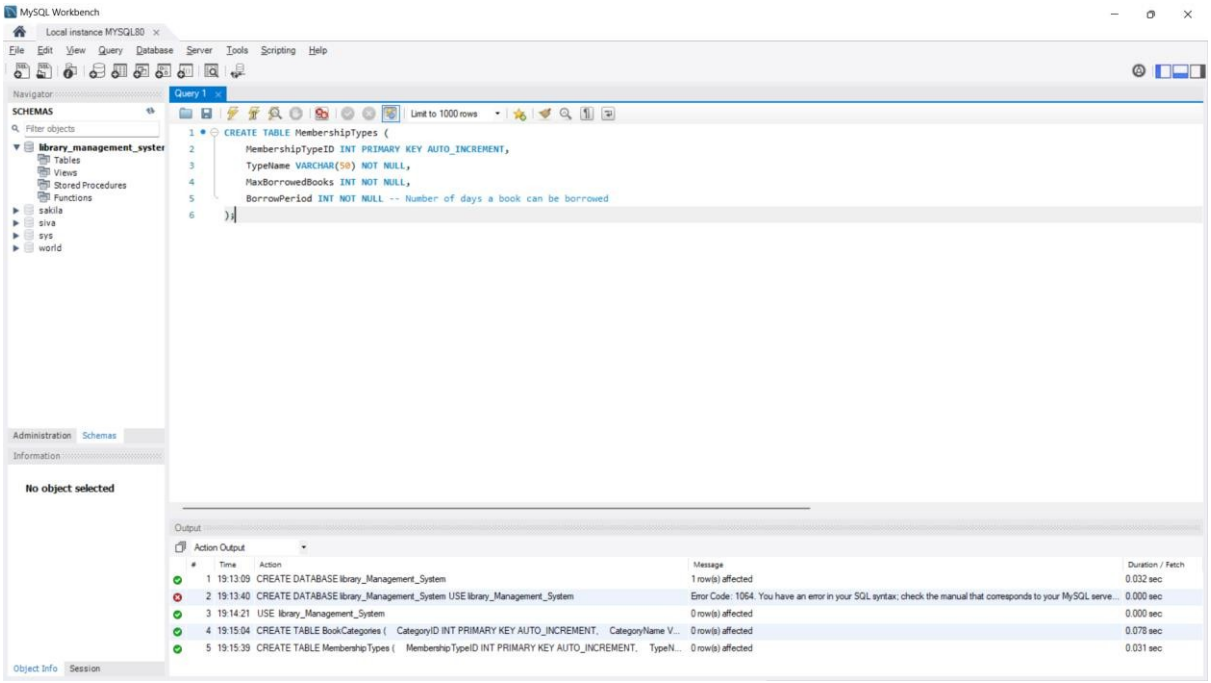
1. Create the BookCategories Table

```
CREATE TABLE BookCategories (
CategoryID INT PRIMARY KEY AUTO_INCREMENT,
CategoryName VARCHAR(100) NOT NULL
);
```



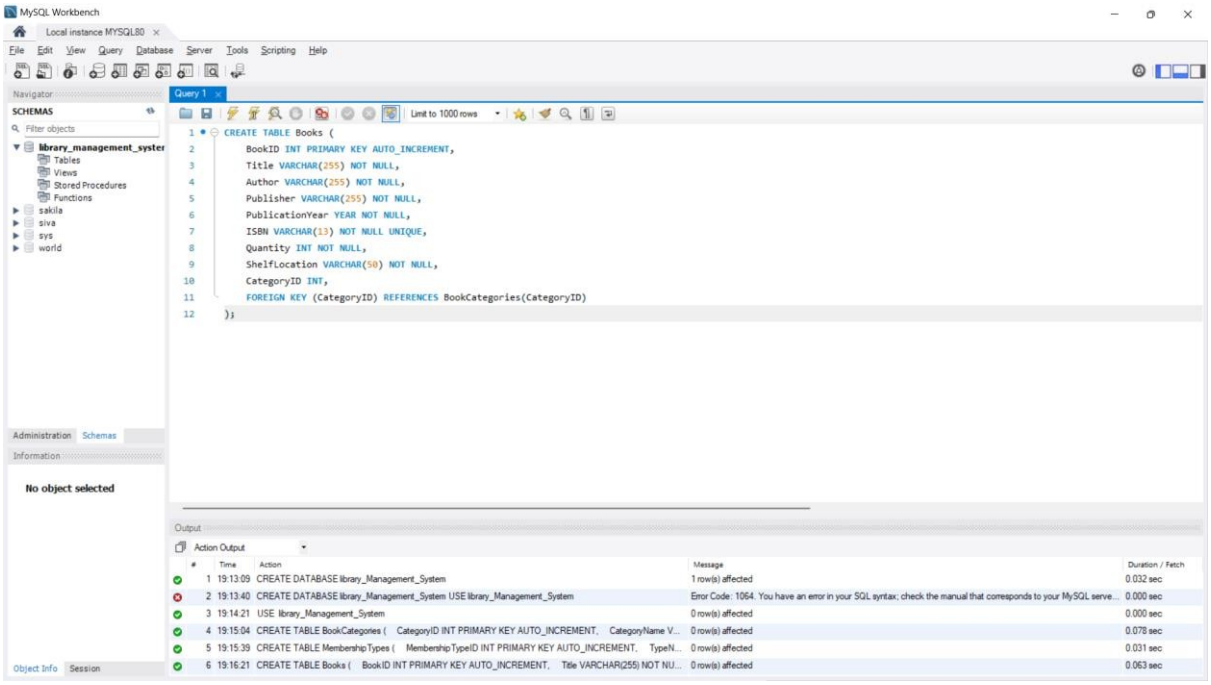
2. Create the MembershipTypes Table:

```
CREATE TABLE MembershipTypes (
MembershipTypeID INT PRIMARY KEY AUTO_INCREMENT,
TypeName VARCHAR(50) NOT NULL,
MaxBorrowedBooks INT NOT NULL,
BorrowPeriod INT NOT NULL -- Number of days a book can be borrowed );
```



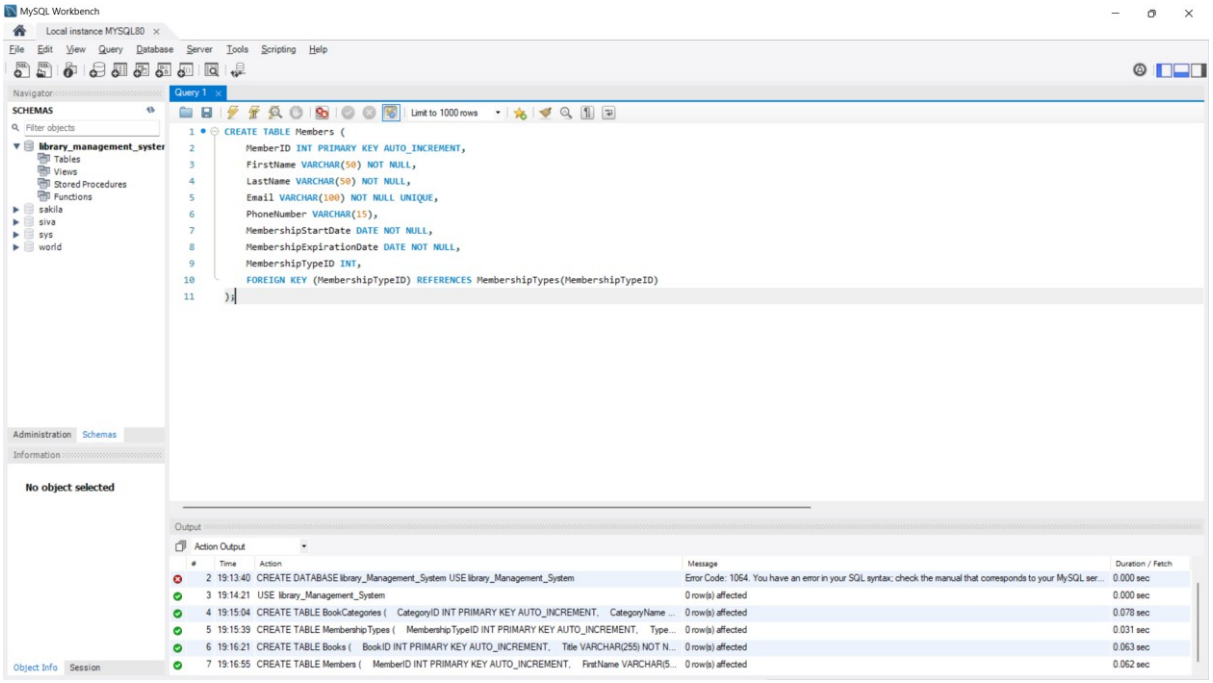
3. Create the Books Table:

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255) NOT NULL,
    Publisher VARCHAR(255) NOT NULL,
    PublicationYear YEAR NOT NULL,
    ISBN VARCHAR(13) NOT NULL UNIQUE,
    Quantity INT NOT NULL,
    ShelfLocation VARCHAR(50) NOT NULL,
    CategoryID INT,
    FOREIGN KEY (CategoryID) REFERENCES BookCategories(CategoryID)
);
```



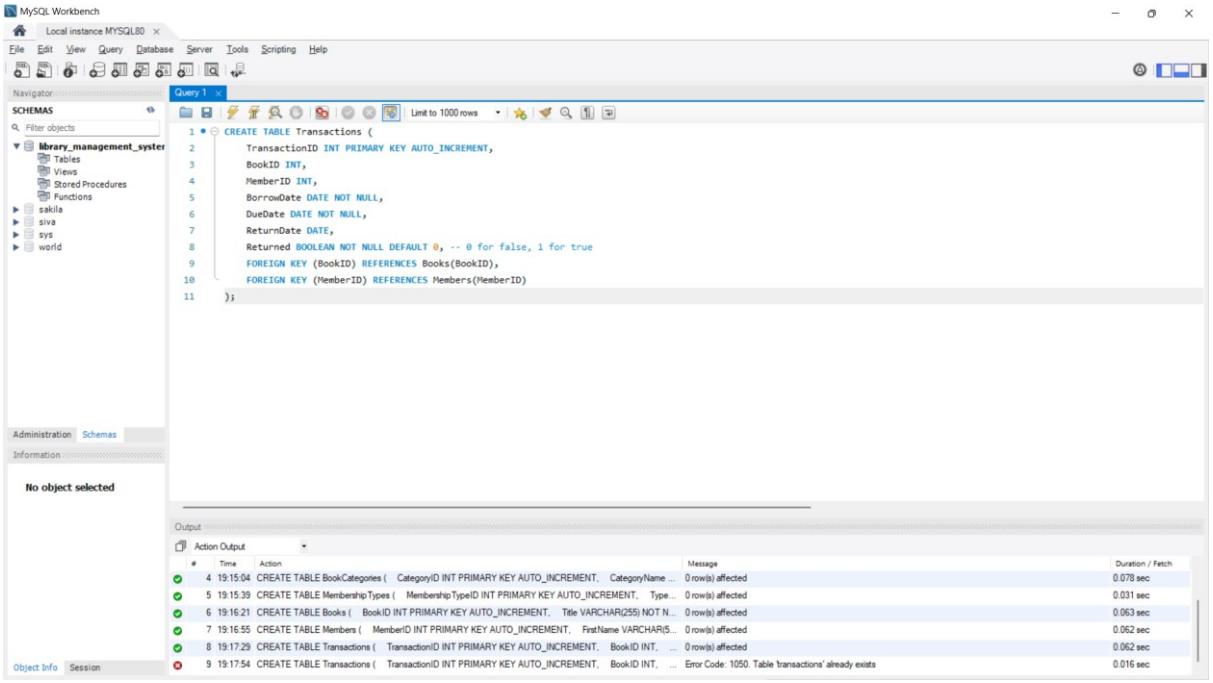
4. Create the Members Table

```
CREATE TABLE Members (
    MemberID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    PhoneNumber VARCHAR(15),
    MembershipStartDate DATE NOT NULL,
    MembershipExpirationDate DATE NOT NULL,
    MembershipTypeID INT,
    FOREIGN KEY (MembershipTypeID) REFERENCES
MembershipTypes(MembershipTypeID)
);
```



5. Create the Transactions Table

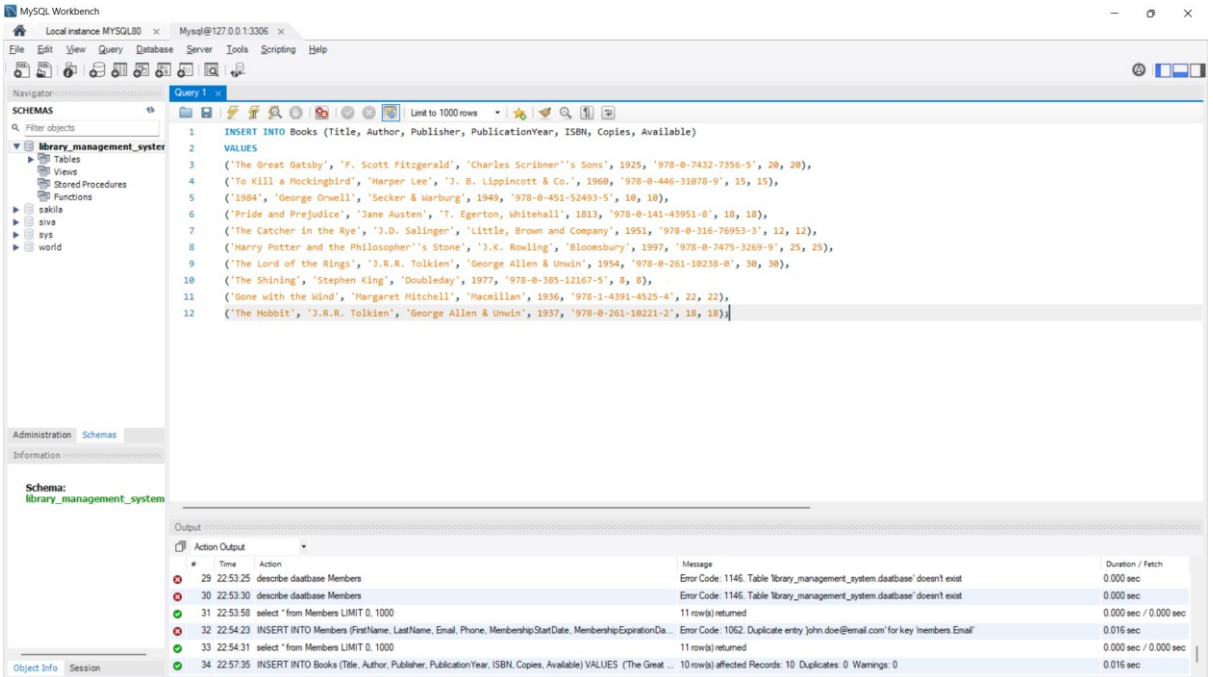
```
CREATE TABLE Transactions (
    TransactionID INT PRIMARY KEY AUTO_INCREMENT,
    BookID INT,
    MemberID INT,
    BorrowDate DATE NOT NULL,
    DueDate DATE NOT NULL,
    ReturnDate DATE,
    Returned BOOLEAN NOT NULL DEFAULT 0, -- 0 for false, 1 for true
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```



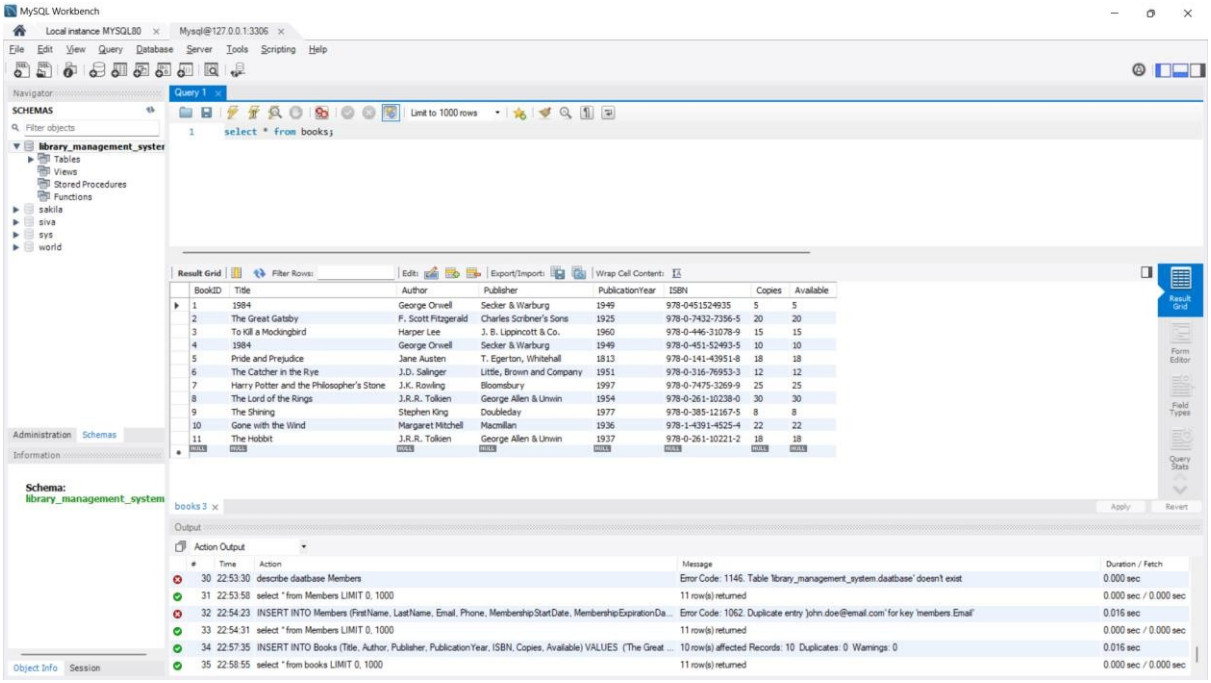
SQL Queries:

1. Inserting Data

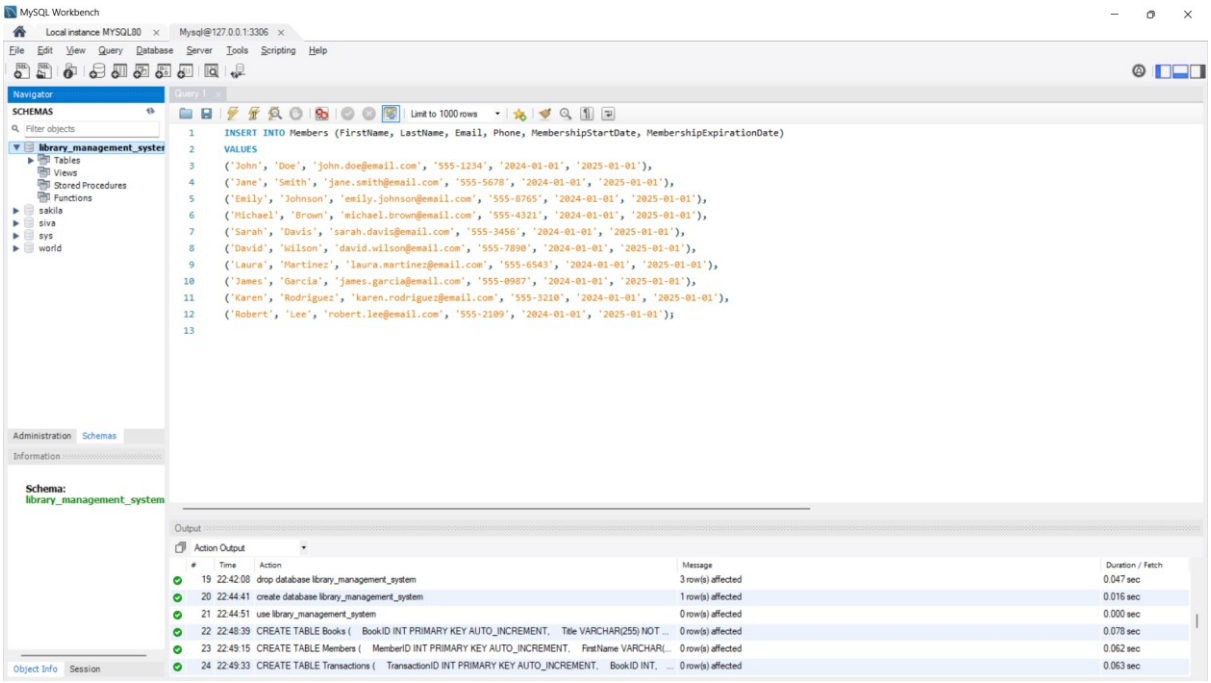
Insert a New Book



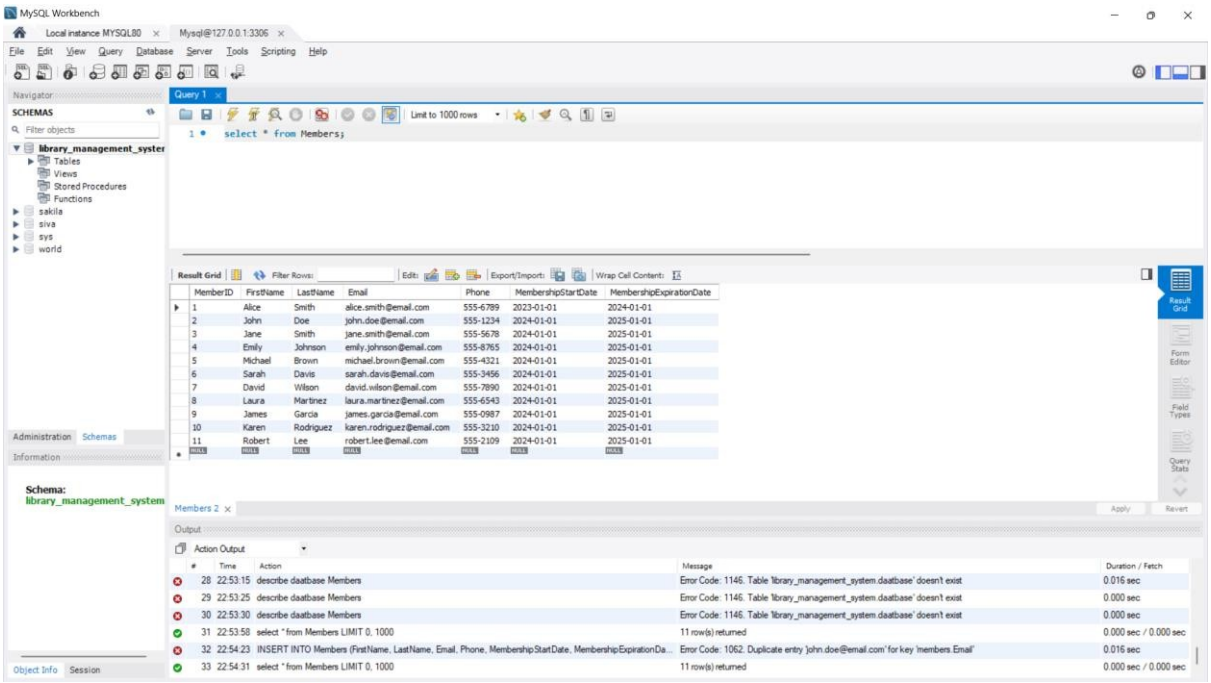
SELECT * FROM books;



3) Insert a New Member

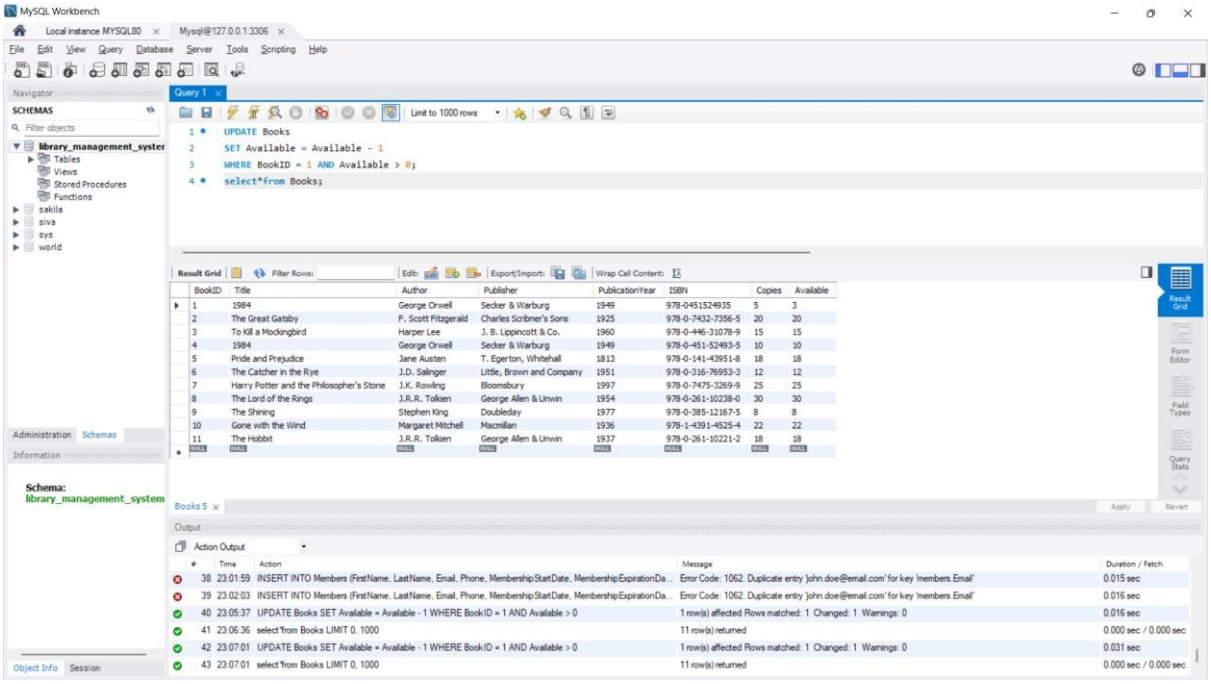


SELECT * FROM Members;

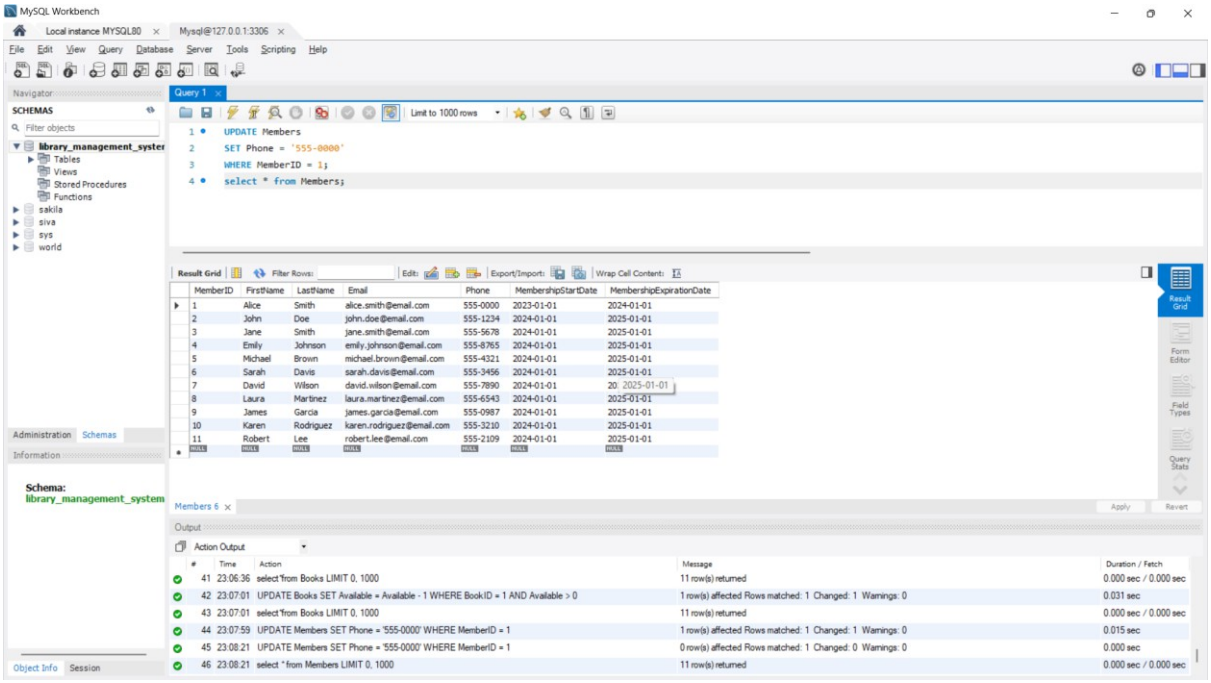


4) Update Data

Update Book Information:

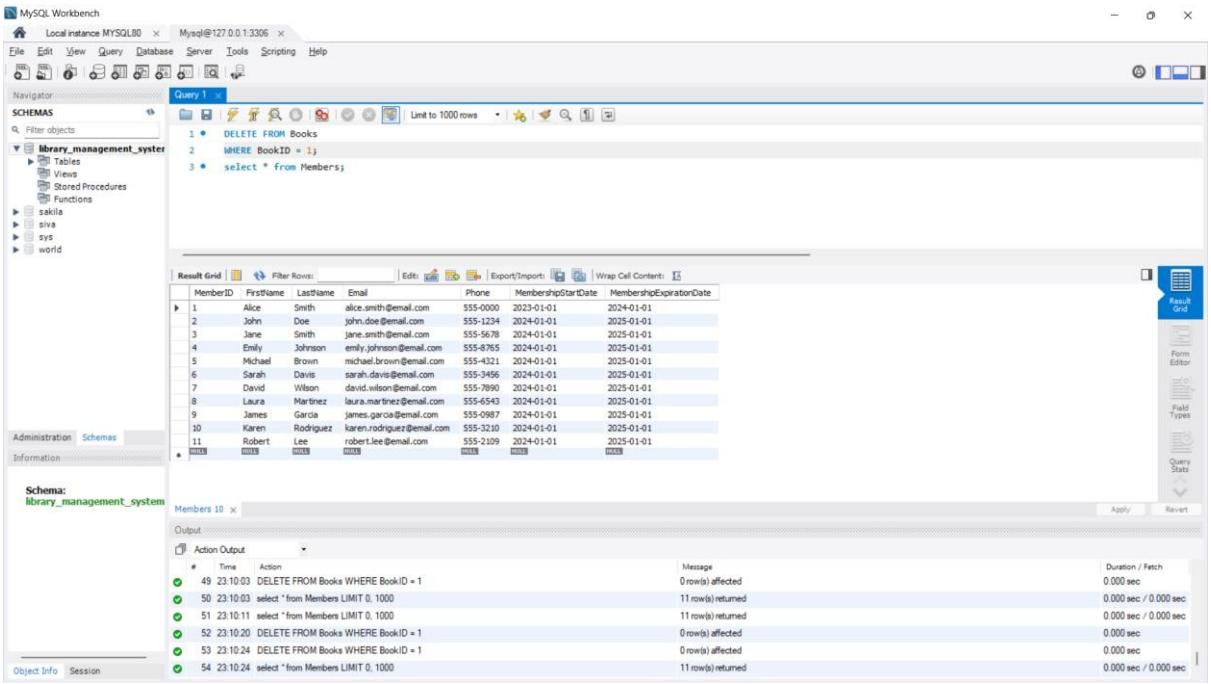


Update Member Information :

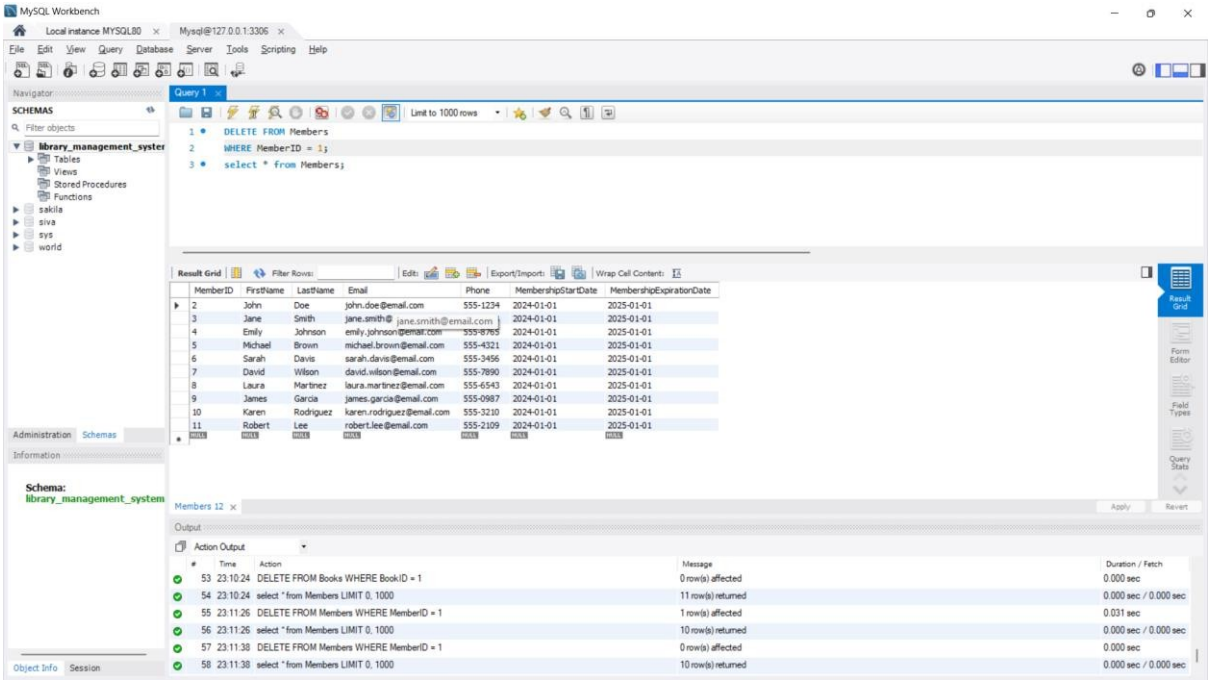


5) Delete Data

Delete a Book :

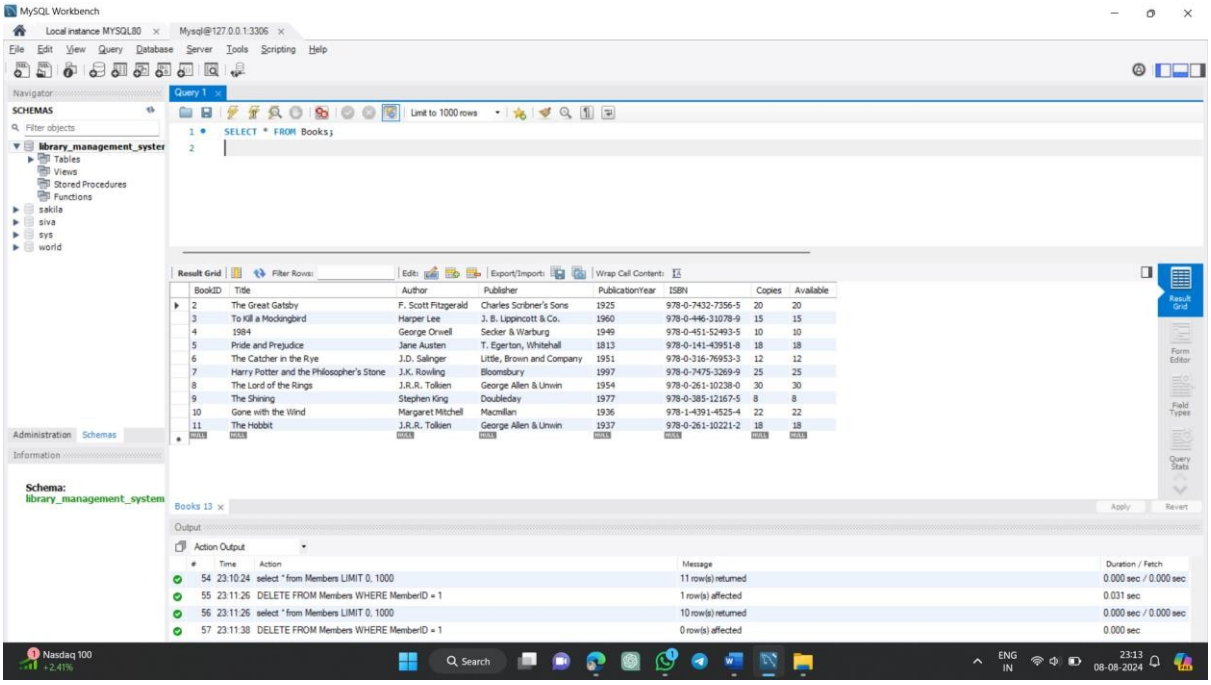


Delete a Member :

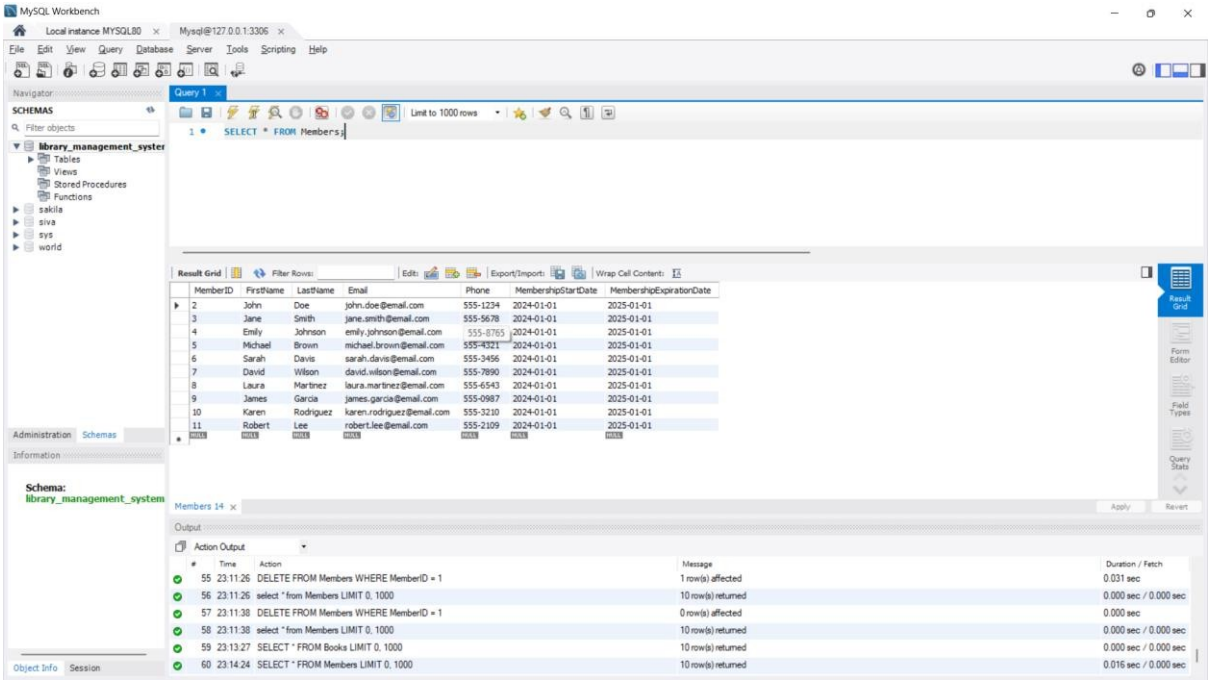


5) Retrieve Data

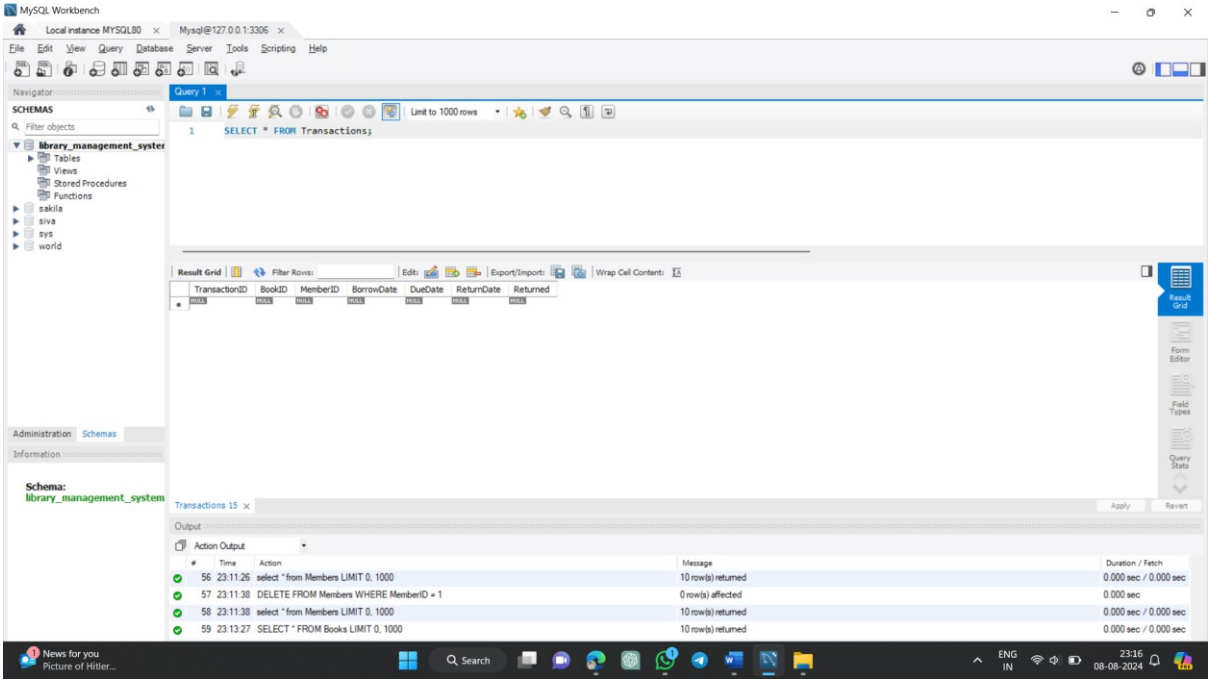
Get All Books :



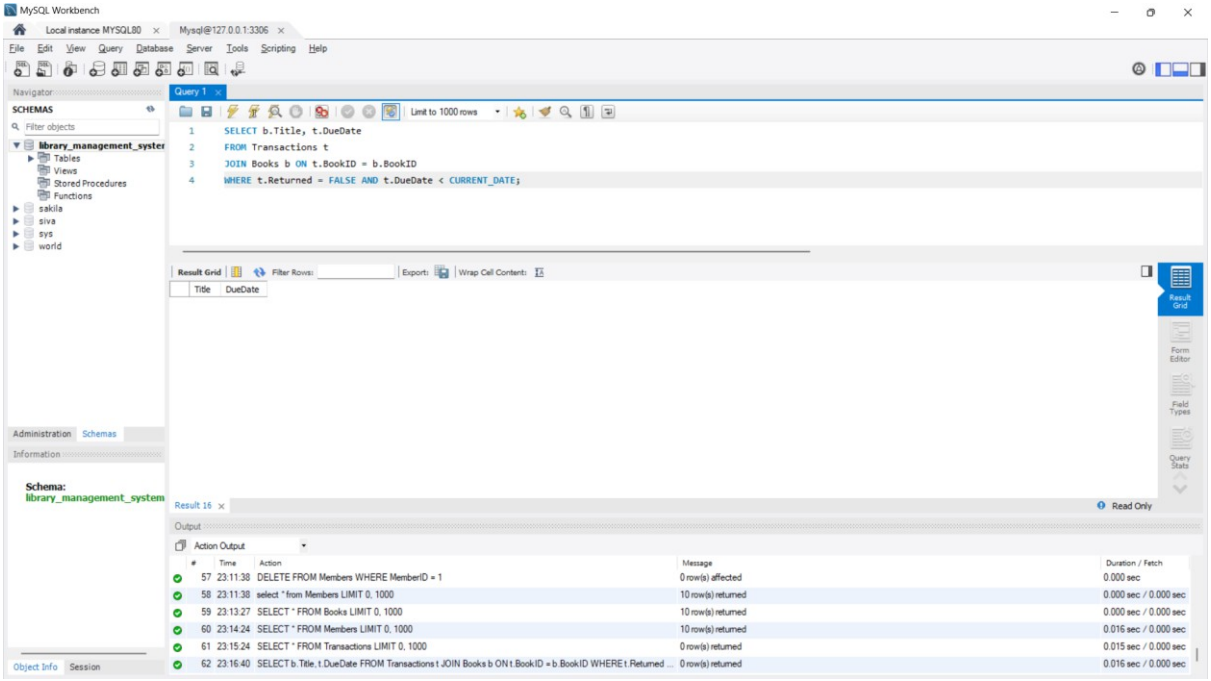
Get All Members



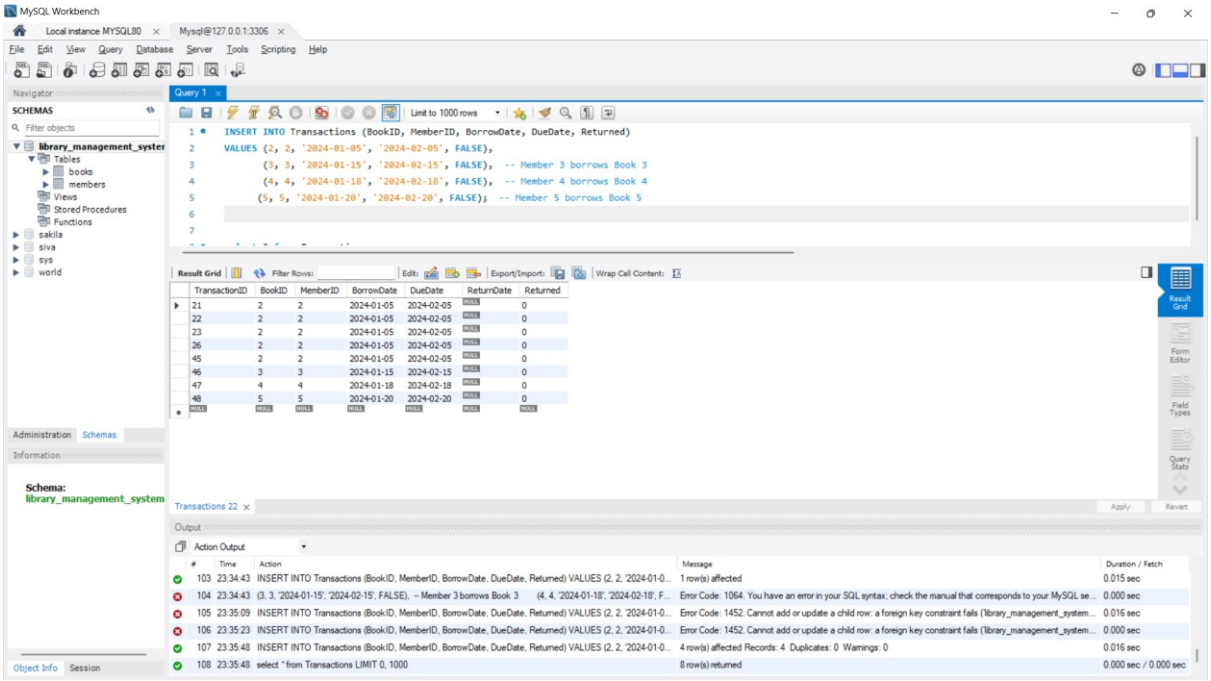
Get All Transactions



Get Overdue Books



6) Borrow and Return Transactions :



Library Database Management System has been Successfully Created.