Query 1 | SQL File 1* | SQL File 2* | SQL File 3* | SQL File 6* | SQL File 7* | SQL File 8*

Limit to 1000 rows

```sql
CREATE DATABASE LibraryDB;

USE LibraryDB;

CREATE TABLE Books (
    BookID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(255) NOT NULL,
    PublishYear INT,
    ISBN VARCHAR(20) UNIQUE,
    Author VARCHAR(255) NOT NULL
);

CREATE TABLE Members (
    MemberID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Address VARCHAR(255),
    Email VARCHAR(100) UNIQUE,
    MembershipYear INT,
    PhoneNo VARCHAR(15)
);
```

```sql
CREATE TABLE BorrowedBooks (
    BorrowID INT PRIMARY KEY AUTO_INCREMENT,
    BookID INT,
    MemberID INT,
    BorrowDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID)
);
```

Database connection

Source | History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package librarymanagementsys;

/**
 *
 * @author bbhuv
 */
import java.sql.*;
import java.util.Scanner;

public class LibraryManagementsys {
    static final String URL = "jdbc:mysql://localhost:3306/LibraryDB";
    static final String USER = "root";
    static final String PASSWORD = "bhuvan@123";

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
            while (true) {
                System.out.println("\nLibrary Management System");
                System.out.println("1. Add Book");
                System.out.println("2. View Books");
                System.out.println("3. Add Member");
                System.out.println("4. View Members");
                System.out.println("5. Borrow Book");
                System.out.println("6. Return Book");
                System.out.println("7. Exit");
                System.out.print("Enter choice: ");
                int choice = sc.nextInt();
                sc.nextLine();
```

```
                switch (choice) {
                    case 1:
                        addBook(con, sc);
                        break;
                    case 2:
                        viewBooks(con);
                        break;
                    case 3:
                        addMember(con, sc);
                        break;
                    case 4:
                        viewMembers(con);
                        break;
                    case 5:
                        borrowBook(con, sc);
                        break;
                    case 6:
                        returnBook(con, sc);
                        break;
                    case 7:
                        System.out.println("Exiting...");
                        return;
                    default:
                        System.out.println("Invalid choice! Try again.");
                }
            }
        } catch (Exception e) {
            System.out.println("Database Error: " + e.getMessage());
        }
    }
```

```
    private static void addBook(Connection con, Scanner sc) throws SQLException {
        System.out.print("Enter title: ");
        String title = sc.nextLine();
        System.out.print("Enter publish year: ");
        int year = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter ISBN: ");
        String isbn = sc.nextLine();
        System.out.print("Enter author: ");
        String author = sc.nextLine();

        String query = "INSERT INTO Books (Title, PublishYear, ISBN, Author) VALUES (?, ?, ?, ?)";
        try (PreparedStatement pst = con.prepareStatement(query)) {
            pst.setString(1, title);
            pst.setInt(2, year);
            pst.setString(3, isbn);
            pst.setString(4, author);
            pst.executeUpdate();
            System.out.println("Book added successfully!");
        }
    }

```

```

    private static void viewBooks(Connection con) throws SQLException {
        String query = "SELECT * FROM Books";
        try (Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                System.out.println("BookID: " + rs.getInt("BookID") + ", Title: " + rs.getString("Title") +
                        ", Publish Year: " + rs.getInt("PublishYear") + ", ISBN: " + rs.getString("ISBN") +
                        ", Author: " + rs.getString("Author"));
            }
        }
    }
```

```
    private static void addMember(Connection con, Scanner sc) throws SQLException {
        System.out.print("Enter name: ");
        String name = sc.nextLine();
        System.out.print("Enter address: ");
        String address = sc.nextLine();
        System.out.print("Enter email: ");
        String email = sc.nextLine();
        System.out.print("Enter membership year: ");
        int year = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter phone number: ");
        String phone = sc.nextLine();

        String query = "INSERT INTO Members (Name, Address, Email, MembershipYear, PhoneNo) VALUES (?, ?, ?, ?, ?)";
        try (PreparedStatement pst = con.prepareStatement(query)) {
            pst.setString(1, name);
            pst.setString(2, address);
            pst.setString(3, email);
            pst.setInt(4, year);
            pst.setString(5, phone);
            pst.executeUpdate();
            System.out.println("Member added successfully!");
        }
    }
```

```
    private static void viewMembers(Connection con) throws SQLException {
        String query = "SELECT * FROM Members";
        try (Statement stmt = con.createStatement(); ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                System.out.println("MemberID: " + rs.getInt("MemberID") + ", Name: " + rs.getString("Name") +
                        ", Address: " + rs.getString("Address") + ", Email: " + rs.getString("Email") +
                        ", Membership Year: " + rs.getInt("MembershipYear") + ", Phone No: " + rs.getString("PhoneNo"));
            }
        }
    }
```

```
    private static void borrowBook(Connection con, Scanner sc) throws SQLException {
        System.out.print("Enter Book ID: ");
        int bookID = sc.nextInt();
        System.out.print("Enter Member ID: ");
        int memberID = sc.nextInt();
        sc.nextLine();

        String query = "INSERT INTO BorrowedBooks (BookID, MemberID, BorrowDate) VALUES (?, ?, CURDATE())";
        try (PreparedStatement pst = con.prepareStatement(query)) {
            pst.setInt(1, bookID);
            pst.setInt(2, memberID);
            pst.executeUpdate();
            System.out.println("Book borrowed successfully!");
        }
    }
```

```
    private static void returnBook(Connection con, Scanner sc) throws SQLException {
        System.out.print("Enter Borrow ID: ");
        int borrowID = sc.nextInt();
        sc.nextLine();

        String query = "UPDATE BorrowedBooks SET ReturnDate = CURDATE() WHERE BorrowID = ?";
        try (PreparedStatement pst = con.prepareStatement(query)) {
            pst.setInt(1, borrowID);
            pst.executeUpdate();
            System.out.println("Book returned successfully!");
        }
    }
}
```