Student Project

SQL Queries to run in MySQL:

```
-- Create database
CREATE DATABASE IF NOT EXISTS student_db;
USE student_db;

-- Staff Table (for login)
CREATE TABLE staff (
    staff_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    role ENUM('Admin', 'Staff') NOT NULL
);

-- Students Table
CREATE TABLE students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT,
    gender ENUM('Male', 'Female', 'Other'),
    phone VARCHAR(15),
    address VARCHAR(255)
);
```

```
-- Fees Table
CREATE TABLE fees (
    fee_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    amount DECIMAL(10,2),
    payment_date DATE,
    status ENUM('Paid', 'Pending'),
    FOREIGN KEY (student_id) REFERENCES students(student_id)
);

-- Library Table
CREATE TABLE library (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100),
    author VARCHAR(100),
    total_copies INT
);

-- Borrowed Books Table
CREATE TABLE borrowed_books (
    borrow_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    book_id INT,
    borrow_date DATE,
    return_date DATE,
    FOREIGN KEY (student_id) REFERENCES students(student_id),
```

```
    FOREIGN KEY (book_id) REFERENCES library(book_id)
);
--Add admin user
INSERT INTO staff (username, password, role) VALUES ('admin', 'admin123', 'Admin');
```

DBConnection.java

```
import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/student_db";
    private static final String USER = "root";
    private static final String PASSWORD = "root";  // change if needed

    public static Connection getConnection() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

LoginService.java

```
import java.sql.*;
import java.util.Scanner;

public class LoginService {
    public static boolean login(String username, String password) {
        try (Connection con = DBConnection.getConnection()) {
```

```
            String query = "SELECT * FROM staff WHERE username = ? AND password = ?";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setString(1, username);
            ps.setString(2, password);

            ResultSet rs = ps.executeQuery();
            return rs.next(); // login success if match found
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

StudentService.java

```
import java.sql.*;
import java.util.Scanner;

public class StudentService {

    public static void registerStudent(Scanner sc) {
        try (Connection con = DBConnection.getConnection()) {
            System.out.print("Name: ");
            String name = sc.nextLine();
            System.out.print("Age: ");
            int age = sc.nextInt();
```

```
            sc.nextLine();
            System.out.print("Gender: ");
            String gender = sc.nextLine();
            System.out.print("Phone: ");
            String phone = sc.nextLine();
            System.out.print("Address: ");
            String address = sc.nextLine();

            String sql = "INSERT INTO students (name, age, gender, phone, address) VALUES (?, ?, ?,
?, ?)";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, name);
            ps.setInt(2, age);
            ps.setString(3, gender);
            ps.setString(4, phone);
            ps.setString(5, address);
            ps.executeUpdate();

            System.out.println("Student registered.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void viewStudents() {
        try (Connection con = DBConnection.getConnection();
```

```
         Statement stmt = con.createStatement()) {


        ResultSet rs = stmt.executeQuery("SELECT * FROM students");
        System.out.printf("\n%-5s %-20s %-5s %-10s %-15s %-30s\n", "ID", "Name", "Age",
"Gender", "Phone", "Address");
        while (rs.next()) {
           System.out.printf("%-5d %-20s %-5d %-10s %-15s %-30s\n",
                rs.getInt("student_id"),
                rs.getString("name"),
                rs.getInt("age"),
                rs.getString("gender"),
                rs.getString("phone"),
                rs.getString("address"));
        }
      } catch (Exception e) {
        e.printStackTrace();
      }
   }
}
```

FeesService.java

```
import java.sql.*;
import java.util.Scanner;

public class FeesService {
```

```
    public static void payFees(Scanner sc) {
        try (Connection con = DBConnection.getConnection()) {
            System.out.print("Student ID: ");
            int studentId = sc.nextInt();
            System.out.print("Amount: ");
            double amount = sc.nextDouble();
            sc.nextLine(); // consume newline
            System.out.print("Payment Date (YYYY-MM-DD): ");
            String date = sc.nextLine();
            System.out.print("Status (Paid/Pending): ");
            String status = sc.nextLine();

            String sql = "INSERT INTO fees (student_id, amount, payment_date, status) VALUES (?, ?,
?, ?)";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setInt(1, studentId);
            ps.setDouble(2, amount);
            ps.setDate(3, Date.valueOf(date));
            ps.setString(4, status);
            ps.executeUpdate();

            System.out.println("Fees record added.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
    public static void viewFees() {
        try (Connection con = DBConnection.getConnection();
             Statement stmt = con.createStatement()) {
            ResultSet rs = stmt.executeQuery(
                "SELECT f.fee_id, s.name, f.amount, f.payment_date, f.status " +
                "FROM fees f JOIN students s ON f.student_id = s.student_id");

            System.out.printf("\n%-5s %-20s %-10s %-12s %-10s\n", "FeeID", "Student Name",
"Amount", "Date", "Status");
            while (rs.next()) {
                System.out.printf("%-5d %-20s %-10.2f %-12s %-10s\n",
                    rs.getInt("fee_id"),
                    rs.getString("name"),
                    rs.getDouble("amount"),
                    rs.getDate("payment_date").toString(),
                    rs.getString("status"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

LibraryService.java

```
import java.sql.*;
import java.util.Scanner;
```

```
public class LibraryService {

    public static void addBook(Scanner sc) {
        try (Connection con = DBConnection.getConnection()) {
            System.out.print("Title: ");
            String title = sc.nextLine();
            System.out.print("Author: ");
            String author = sc.nextLine();
            System.out.print("Total Copies: ");
            int totalCopies = sc.nextInt();
            sc.nextLine();

            String sql = "INSERT INTO library (title, author, total_copies) VALUES (?, ?, ?)";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, title);
            ps.setString(2, author);
            ps.setInt(3, totalCopies);
            ps.executeUpdate();

            System.out.println("Book added.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
public static void viewBooks() {
    try (Connection con = DBConnection.getConnection();
         Statement stmt = con.createStatement()) {
        ResultSet rs = stmt.executeQuery("SELECT * FROM library");
        System.out.printf("\n%-5s %-30s %-25s %-10s\n", "ID", "Title", "Author", "Copies");
        while (rs.next()) {
            System.out.printf("%-5d %-30s %-25s %-10d\n",
                    rs.getInt("book_id"),
                    rs.getString("title"),
                    rs.getString("author"),
                    rs.getInt("total_copies"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void borrowBook(Scanner sc) {
    try (Connection con = DBConnection.getConnection()) {
        System.out.print("Student ID: ");
        int studentId = sc.nextInt();
        System.out.print("Book ID: ");
        int bookId = sc.nextInt();
        sc.nextLine();
        System.out.print("Borrow Date (YYYY-MM-DD): ");
        String borrowDate = sc.nextLine();
```

```
        String sql = "INSERT INTO borrowed_books (student_id, book_id, borrow_date) VALUES (?, ?, ?)";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setInt(1, studentId);
        ps.setInt(2, bookId);
        ps.setDate(3, Date.valueOf(borrowDate));
        ps.executeUpdate();

        System.out.println("Book borrowed.");
      } catch (Exception e) {
        e.printStackTrace();
      }
    }

    public static void viewBorrowedBooks() {
      try (Connection con = DBConnection.getConnection();
           Statement stmt = con.createStatement()) {
        String sql = "SELECT bb.borrow_id, s.name, l.title, bb.borrow_date, bb.return_date " +
                     "FROM borrowed_books bb " +
                     "JOIN students s ON bb.student_id = s.student_id " +
                     "JOIN library l ON bb.book_id = l.book_id";
        ResultSet rs = stmt.executeQuery(sql);

        System.out.printf("\n%-5s %-20s %-30s %-12s %-12s\n", "ID", "Student", "Book Title", "Borrow Date", "Return Date");
        while (rs.next()) {
```

```
            String returnDate = rs.getDate("return_date") != null ?
rs.getDate("return_date").toString() : "Not Returned";
            System.out.printf("%-5d %-20s %-30s %-12s %-12s\n",
                rs.getInt("borrow_id"),
                rs.getString("name"),
                rs.getString("title"),
                rs.getDate("borrow_date").toString(),
                returnDate);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
  }
}
```

Main.java

```
import java.util.Scanner;

public class Main {

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Username: ");
    String username = sc.nextLine();
    System.out.print("Password: ");
```

```
String password = sc.nextLine();

if (!LoginService.login(username, password)) {
    System.out.println("Login failed.");
    sc.close();
    return;
}

while (true) {
    System.out.println("\nStudent Management Menu");
    System.out.println("1. Register Student");
    System.out.println("2. View Students");
    System.out.println("3. Pay Fees");
    System.out.println("4. View Fees");
    System.out.println("5. Add Book");
    System.out.println("6. View Books");
    System.out.println("7. Borrow Book");
    System.out.println("8. View Borrowed Books");
    System.out.println("9. Exit");
    System.out.print("Choose option: ");

    int choice = sc.nextInt();
    sc.nextLine();

    switch (choice) {
        case 1 -> StudentService.registerStudent(sc);
```

```
            case 2 -> StudentService.viewStudents();
            case 3 -> FeesService.payFees(sc);
            case 4 -> FeesService.viewFees();
            case 5 -> LibraryService.addBook(sc);
            case 6 -> LibraryService.viewBooks();
            case 7 -> LibraryService.borrowBook(sc);
            case 8 -> LibraryService.viewBorrowedBooks();
            case 9 -> {
                System.out.println("Goodbye!");
                sc.close();
                return;
            }
            default -> System.out.println("Invalid option");
        }
    }
  }
}
```