

1. Write a Java program to connect to a MySQL database using JDBC.

A. package JDBC_conn;

import java.sql.*;

public class Jdbc_conn {

 public static void main(String[] args) {

 String url = "jdbc:mysql://localhost:3306/mydb";

 String user = "root";

 String password = "mysql_password";

 try (Connection con = DriverManager.getConnection(url, user, password);

 Statement stmt = con.createStatement();

 ResultSet rs = stmt.executeQuery("SELECT * FROM Emp")) {

 System.out.println("ID\tName\tSalary\tDepartment");

 while (rs.next()) {

 int id = rs.getInt("ID");

 String name = rs.getString("name");

 int salary = rs.getInt("salary");

 String department = rs.getString("Department"); // adjust if needed

 System.out.println(id + "\t" + name + "\t" + salary + "\t" + department);

 }

 } catch (SQLException e) {

 System.out.println("Database error: " + e.getMessage());

 }

 }

}

2. Create a Java class to insert student records into a database table.

A. package JDBS_conn;

import java.sql.*;

```

public class Create_table {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "mysql_password";
        String sql = "create table students(rollno int, "
            + "name varchar(50), "
            + "per int, "
            + "email varchar(50))";
        String insertSQL = "insert into students values "
            + "(101,'Neeva Sharma',98,'abc@gmail.com'),"
            + "(102,'Reeva Sharma',89,'Reeva@gmail.com'),"
            + "(103,'Shiva Upadhyay',79,'shiva@gmail.com)";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user,
password);
            System.out.println("Connection created");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(sql);
            System.out.println("Students table created");
            int rowInserted = stmt.executeUpdate(insertSQL);
            if (rowInserted > 0) {
                System.out.println("New Student records inserted");
            }
            ResultSet rs = stmt.executeQuery("select * from students");
            System.out.println("rollno\t name\t per\t email");
            while (rs.next()) {
                int rollno = rs.getInt("rollno");

```

```

        String name = rs.getString("name");
        int per = rs.getInt("per");
        String email = rs.getString("email");
        System.out.println(rollno + "\t" + name + "\t" + per +
"\t" + email);
    }
    stmt.close();
    con.close();
} catch (SQLException e) {
    System.out.println(e);
} catch (ClassNotFoundException e) {
    System.out.println(e);
}
}
}

```

3. Write a JDBC program to fetch and display all student records from the database.

A. package JDBS_conn;

import java.sql.*;

public class FetchStudents {

 public static void main(String[] args) {

 String url = "jdbc:mysql://localhost:3306/mydb";

 String user = "root";

 String password = "mysql_password";

 try {

 Class.forName("com.mysql.cj.jdbc.Driver");

 Connection con = DriverManager.getConnection(url, user, password);

```

Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from students");
System.out.println("rollno\t name\t per\t email");
while (rs.next()) {
    int rollno = rs.getInt("rollno");
    String name = rs.getString("name");
    int per = rs.getInt("per");
    String email = rs.getString("email");
    System.out.println(rollno + "\t" + name + "\t" + per +
"\t" + email);
}
rs.close();
stmt.close();
con.close();
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

4. Develop a program to search a student by ID using JDBC.

```

A. package JDBS_conn;
import java.sql.*;
import java.util.Scanner;
public class SearchStudentByID {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "mysql_password";
    }
}

```

```

Scanner sc = new Scanner(System.in);

System.out.print("Enter student roll number to search: ");

int id = sc.nextInt();

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con = DriverManager.getConnection(url, user, password);
    String sql = "SELECT * FROM students WHERE rollno = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, id);
    ResultSet rs = ps.executeQuery();
    if (rs.next()) {
        System.out.println("rollno\t name\t per\t email");
        int rollno = rs.getInt("rollno");
        String name = rs.getString("name");
        int per = rs.getInt("per");
        String email = rs.getString("email");
        System.out.println(rollno + "\t" + name + "\t" + per + "\t" + email);
    } else {
        System.out.println("No student found with roll number: " + id);
    }
    rs.close();
    ps.close();
    con.close();
} catch (Exception e) {
    System.out.println(e);
}
}

```

```
}
```

5. Implement an update operation to modify student details in the database using JDBC.

A. package JDBS_conn;

```
import java.sql.*;
```

```
import java.util.Scanner;
```

```
public class UpdateStudent {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydb";
```

```
        String user = "root";
```

```
        String password = "mysql_password";
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter student roll number to update: ");
```

```
        int rollno = sc.nextInt();
```

```
        sc.nextLine();
```

```
        System.out.print("Enter new name: ");
```

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

```
        System.out.print("Enter new percentage: ");
```

```
        int per = sc.nextInt();
```

```
        sc.nextLine();
```

```
        System.out.print("Enter new email: ");
```

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

```
        try {
```

```
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            Connection con = DriverManager.getConnection(url, user, password);
```

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

```

        String sql = "UPDATE students SET name = " + name + ", per = " +
per + ", email = " + email + " WHERE rollno = " + rollno;

        int rowUpdated = stmt.executeUpdate(sql);

        if (rowUpdated > 0) {

            System.out.println("Student details updated successfully.");

        } else {

            System.out.println("No student found with roll number: " + rollno);

        }

        stmt.close();

        con.close();

    } catch (Exception e) {

        System.out.println(e);

    }

}
}

```

6. Write a Java program to delete a student record from the database using JDBC.

```

A. package JDBS_conn;

import java.sql.*;

import java.util.Scanner;

public class DeleteStudent {

    public static void main(String[] args) {

        String url = "jdbc:mysql://localhost:3306/mydb";

        String user = "root";

        String password = "mysql_password";

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter student roll number to delete: ");

        int rollno = sc.nextInt();
    }
}

```

```

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con = DriverManager.getConnection(url, user, password);
    Statement stmt = con.createStatement();
    String sql = "DELETE FROM students WHERE rollno = " + rollno;
    int rowsDeleted = stmt.executeUpdate(sql);
    if (rowsDeleted > 0) {
        System.out.println("Student record deleted successfully.");
    } else {
        System.out.println("No student found with roll number: " + rollno);
    }
    stmt.close();
    con.close();
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

7. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.

A. package JDBC_Conn;

import java.sql.*;

```

public class EmployeeCRUD {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "mysql_password";
    }
}

```



```

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con = DriverManager.getConnection(url, user, password);
    Statement stmt = con.createStatement();
    // Drop table if exists
    stmt.executeUpdate("DROP TABLE IF EXISTS Employee");
    // Create Employee table
    String createTable = "CREATE TABLE Employee (" +
        "id INT PRIMARY KEY, " +
        "name VARCHAR(50), " +
        "salary INT)";
    stmt.executeUpdate(createTable);
    System.out.println("Employee table created.");
    // Insert records (Create)
    String insertRecords = "INSERT INTO Employee VALUES " +
        "(101, 'Neeva Sharma', 50000), " +
        "(102, 'Reeva Varma', 55000), " +
        "(103, 'Shiva Upadhyay', 45000), " +
        "(104, 'Mittal Shah', 55000)";
    int rowsInserted = stmt.executeUpdate(insertRecords);
    System.out.println(rowsInserted + " employee records inserted.");
    // Read - Display all employee records
    System.out.println("\nEmployee records:");
    ResultSet rs = stmt.executeQuery("SELECT * FROM Employee");
    System.out.println("ID\tName\t\tSalary");
    while (rs.next()) {

```

```

        int id = rs.getInt("id");
        String name = rs.getString("name");
        int salary = rs.getInt("salary");
        System.out.println(id + "\t" + name + "\t" + salary);
    }
    rs.close();

    // Update - modify salary of employee with id=104
    int rowsUpdated = stmt.executeUpdate("UPDATE Employee SET salary
= 60000 WHERE id = 104");
    System.out.println("\nRows updated: " + rowsUpdated);
    // Read after update
    System.out.println("\nEmployee records after update:");
    rs = stmt.executeQuery("SELECT * FROM Employee");
    System.out.println("ID\tName\t\tSalary");
    while (rs.next()) {
        int id = rs.getInt("id");
        String name = rs.getString("name");
        int salary = rs.getInt("salary");
        System.out.println(id + "\t" + name + "\t" + salary);
    }
    rs.close();

    // Delete - delete employee with id=103
    int rowsDeleted = stmt.executeUpdate("DELETE FROM Employee
WHERE id = 103");
    System.out.println("\nRows deleted: " + rowsDeleted);
    // Read after delete
    System.out.println("\nEmployee records after deletion:");
    rs = stmt.executeQuery("SELECT * FROM Employee");

```

```

        System.out.println("ID\tName\t\tSalary");
        while (rs.next()) {
            int id = rs.getInt("id");
            String name = rs.getString("name");
            int salary = rs.getInt("salary");
            System.out.println(id + "\t" + name + "\t" + salary);
        }
        rs.close();
        // Close resources
        stmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

8. Create a JDBC-based program to count the total number of rows in a table.

A. package JDBC_CONN;

import java.sql.*;

public class CountRows {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "mysql_password";

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(url, user, password);

```

Statement stmt = con.createStatement();
String query = "SELECT COUNT(*) AS total FROM Employee";
ResultSet rs = stmt.executeQuery(query);
if (rs.next()) {
    int totalRows = rs.getInt("total");
    System.out.println("Total number of rows in Employee table: " +
totalRows);
}
rs.close();
stmt.close();
con.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

9. Develop a program to sort student data in ascending order by name using SQL in JDBC.

A. package JDBC_PRO;

import java.sql.*;

public class SortStudentsByName {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "mysql_password";

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(url, user, password);

```

System.out.println("Connection Established");

Statement stmt = con.createStatement();

// Create Student table

String createStudent = "CREATE TABLE IF NOT EXISTS Student (" +
    "rollno INT PRIMARY KEY, " +
    "name VARCHAR(50), " +
    "per FLOAT, " +
    "email VARCHAR(50), " +
    "city VARCHAR(50))";

stmt.executeUpdate(createStudent);

System.out.println("Student table created");

// Insert records

String insertStudent = "INSERT INTO Student VALUES " +
    "(101, 'Neeva Sharma', 98, 'abc@gmail.com', 'Delhi'), " +
    "(102, 'Reeva Sharma', 89, 'reeva@gmail.com', 'Delhi'), " +
    "(103, 'Shiva Upadhyay', 79, 'shiva@gmail.com', 'Noida'), " +
    "(104, 'Mittal Shah', 90, 'mittal@gmail.com', 'Noida)";

stmt.executeUpdate(insertStudent);

System.out.println("Student records inserted");

// Query to get students sorted by name ascending

ResultSet rs = stmt.executeQuery("SELECT * FROM Student ORDER
BY name ASC");

System.out.println("Students sorted by name (ascending):");

System.out.println("RollNo\tName\t\tPercentage\tEmail\t\t\tCity");

while (rs.next()) {
    int rollno = rs.getInt("rollno");
    String name = rs.getString("name");
    float per = rs.getFloat("per");

```

```

        String email = rs.getString("email");
        String city = rs.getString("city");
        System.out.println(rollno + "\t" + name + "\t" + per + "\t" + email + "\t"
+ city);
    }
    rs.close();
    stmt.close();
    con.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

10. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.

A. package JDBC_PRO;

import java.sql.*;

```

public class StudentsAbove75Percent {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "mysql_password";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("Connection Established");
            Statement stmt = con.createStatement();
            // Create Student table if not exists

```

```

String createStudent = "CREATE TABLE IF NOT EXISTS Student (" +
    "rollno INT PRIMARY KEY, " +
    "name VARCHAR(50), " +
    "per FLOAT, " +
    "email VARCHAR(50), " +
    "city VARCHAR(50))";

stmt.executeUpdate(createStudent);

// Insert sample records

String insertStudent = "INSERT INTO Student VALUES " +
    "(101, 'Neeva Sharma', 98, 'abc@gmail.com', 'Delhi'), " +
    "(102, 'Reeva Sharma', 89, 'reeva@gmail.com', 'Delhi'), " +
    "(103, 'Shiva Upadhyay', 79, 'shiva@gmail.com', 'Noida'), " +
    "(104, 'Mittal Shah', 70, 'mittal@gmail.com', 'Noida')";

stmt.executeUpdate(insertStudent);

// Query to select students with percentage > 75

ResultSet rs = stmt.executeQuery("SELECT * FROM Student WHERE
per > 75");

System.out.println("Students with percentage greater than 75:");
System.out.println("RollNo\tName\tPercentage\tEmail\tCity");
while (rs.next()) {
    int rollno = rs.getInt("rollno");
    String name = rs.getString("name");
    float per = rs.getFloat("per");
    String email = rs.getString("email");
    String city = rs.getString("city");
    System.out.println(rollno + "\t" + name + "\t" + per + "\t" + email + "\t"
+ city);
}

```

```

        rs.close();
        stmt.close();
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

11. Use **PreparedStatement** to insert multiple student records into the database.

A. package JDBS_conn;

```
import java.sql.*;
```

```
public class prepare_stmt {
```

```
    public static void main(String[] args) {
```

```
        String url = "jdbc:mysql://localhost:3306/mydb";
```

```
        String user = "root";
```

```
        String password = "mysql_password";
```

```
        try {
```

```
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            Connection con = DriverManager.getConnection(url, user, password);
```

```
            System.out.println("Connection created");
```

```
            String query = "INSERT INTO student1 (rollno, name, age) VALUES (?, ?, ?)";
```

```
            PreparedStatement pst = con.prepareStatement(query);
```

```
            // Insert first record
```

```
            pst.setInt(1, 101);
```

```
            pst.setString(2, "Neeva Sharma");
```

```
            pst.setInt(3, 20);
```

```
            pst.executeUpdate();
```



```

        // Insert second record
        pst.setInt(1, 102);
        pst.setString(2, "Reeva Sharma");
        pst.setInt(3, 21);
        pst.executeUpdate();
        // Insert third record
        pst.setInt(1, 103);
        pst.setString(2, "Seeva Sharma");
        pst.setInt(3, 22);
        pst.executeUpdate();
        System.out.println("Records inserted successfully");
        pst.close();
        con.close();
    } catch (ClassNotFoundException e) {
        System.out.println("JDBC Driver not found");
        e.printStackTrace();
    } catch (SQLException e) {
        System.out.println("SQL Exception");
        e.printStackTrace();
    }
}
}

```

12. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).

```

A. package JDBC_PRO;
import java.sql.*;

public class TransactionExample {
    public static void main(String[] args) {

```

```

String url = "jdbc:mysql://localhost:3306/mydb";
String user = "root";
String password = "mysql_password";
Connection con = null;
Statement stmt = null;
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con = DriverManager.getConnection(url, user, password);
    System.out.println("Connection established");
    con.setAutoCommit(false);
    stmt = con.createStatement();
    // Insert a record
    String insert1 = "INSERT INTO Employee VALUES (201, 'John Doe',
45000)";
    stmt.executeUpdate(insert1);
    // Insert another record
    String insert2 = "INSERT INTO Employee VALUES (202, 'Jane Smith',
55000)";
    stmt.executeUpdate(insert2);
    con.commit();
    System.out.println("Transaction committed successfully");
} catch (Exception e) {
    System.out.println("Exception occurred, rolling back transaction");
    try {
        if (con != null) {
            con.rollback();
            System.out.println("Rollback successful");
        }
    }
}

```

```

        } catch (SQLException se) {
            se.printStackTrace();
        }
        e.printStackTrace();
    } finally {
        try {
            if (stmt != null) stmt.close();
            if (con != null) con.close();
            System.out.println("Resources closed");
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
}

```

13. Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.

```

A. package JDBC_PRO;

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

public class ExceptionHandlingExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "mysql_password";
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee ID to search: ");
        int empId = sc.nextInt();
    }
}

```

```
Connection con = null;
PreparedStatement pst = null;
ResultSet rs = null;
try {
    // Load driver
    Class.forName("com.mysql.cj.jdbc.Driver");
    // Connect to DB
    con = DriverManager.getConnection(url, user, password);
    System.out.println("Connected to database successfully.");
    // Prepare statement with parameter
    String query = "SELECT * FROM employee WHERE id = ?";
    pst = con.prepareStatement(query);
    pst.setInt(1, empId);
    rs = pst.executeQuery();
    // Check if record exists
    if (rs.next()) {
        System.out.println("Employee Found:");
        System.out.println("ID: " + rs.getInt("id"));
        System.out.println("Name: " + rs.getString("name"));
        System.out.println("Salary: " + rs.getInt("salary"));
    } else {
        System.out.println("No employee found with ID: " + empId);
    }
} catch (ClassNotFoundException e) {
    System.out.println("JDBC Driver not found. Please check your setup.");
} catch (SQLException e) {
```

```

        System.out.println("Database error occurred: " + e.getMessage());
    } catch (Exception e) {
        System.out.println("An unexpected error occurred: " + e.getMessage());
    } finally {
        // Close resources in reverse order of opening
        try {
            if (rs != null) rs.close();
            if (pst != null) pst.close();
            if (con != null) con.close();
            sc.close();
        } catch (SQLException e) {
            System.out.println("Error closing resources: " + e.getMessage());
        }
    }
}
}
}

```

14. Create a login system using JDBC where user credentials are verified from the database.

```

A. package DB_connection;
import java.sql.Connection;
import java.sql.DriverManager;
public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/hospital_db";
    private static final String USER = "root";
    private static final String PASSWORD = "mysql_password";
    public static Connection getConnection() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

```

```

    }
}
package DB_connection;
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 a match is found
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
package DB_connection;
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. Exiting...");
    return;
}
while (true) {
    System.out.println("\n Hospital Management Menu");
    System.out.println("1. Register Patient");
    System.out.println("2. View Patients");
    System.out.println("3. Book Appointment");
    System.out.println("4. Generate Billing");
    System.out.println("5. Exit");
    System.out.print("Choose option: ");
    int choice = sc.nextInt();
    switch (choice) {
        case 1 -> PatientService.registerPatient(sc);
        case 2 -> PatientService.viewPatients();
        case 3 -> AppointmentService.bookAppointment(sc);
        case 4 -> BillingService.generateBill(sc);
        case 5 -> {
            System.out.println("Goodbye!");
            return;
        }
        default -> System.out.println(" Invalid option");
    }
}
}
}

```



```

        case 4 -> {
            System.out.println("Exiting...");
            con.close();
            sc.close();
            return;
        }
        default -> System.out.println("Invalid choice. Try again.");
    }
}

} catch (Exception e) {
    e.printStackTrace();
}
}

static void insertRecord(Connection con, Scanner sc) throws SQLException {
    System.out.print("Enter ID: ");
    int id = sc.nextInt();
    sc.nextLine();
    System.out.print("Enter Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Age: ");
    int age = sc.nextInt();
    String sql = "INSERT INTO student (id, name, age) VALUES (?, ?, ?)";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, id);
    ps.setString(2, name);
    ps.setInt(3, age);
    int rows = ps.executeUpdate();

```

```

        System.out.println(rows + " record(s) inserted.");
    }

    static void searchRecord(Connection con, Scanner sc) throws SQLException {
        System.out.print("Enter ID to search: ");
        int id = sc.nextInt();
        String sql = "SELECT * FROM student WHERE id = ?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            System.out.println("ID: " + rs.getInt("id"));
            System.out.println("Name: " + rs.getString("name"));
            System.out.println("Age: " + rs.getInt("age"));
        } else {
            System.out.println("Record not found.");
        }
    }

    static void updateRecord(Connection con, Scanner sc) throws SQLException
    {
        System.out.print("Enter ID to update: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter new name: ");
        String newName = sc.nextLine();
        System.out.print("Enter new age: ");
        int newAge = sc.nextInt();
        String sql = "UPDATE student SET name = ?, age = ? WHERE id = ?";
        PreparedStatement ps = con.prepareStatement(sql);
    }

```

```

        ps.setString(1, newName);
        ps.setInt(2, newAge);
        ps.setInt(3, id);
        int rows = ps.executeUpdate();
        System.out.println(rows + " record(s) updated.");
    }
}

```

16.Design the schema for a **Library Management System** and write JDBC programs for:

- Adding a book
- Viewing all books
- Issuing a book to a member
- Returning a book

A.DB CONNECTION:

```

import java.sql.*;

public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/library_db";
    private static final String USER = "root";
    private static final String PASSWORD = "root";
    public static Connection getConnection() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

```

Add Book:

```

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

public class AddBook {

```

```

public static void main(String[] args) {
    try (Connection con = DBConnection.getConnection());
        Scanner sc = new Scanner(System.in)) {
        System.out.print("Enter Book Title: ");
        String title = sc.nextLine();
        System.out.print("Enter Author: ");
        String author = sc.nextLine();
        String sql = "INSERT INTO books (title, author) VALUES (?, ?)";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, title);
        ps.setString(2, author);
        int rows = ps.executeUpdate();
        System.out.println(rows + " book added.");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

View All Books:

```

import java.sql.*;

public class ViewBooks {
    public static void main(String[] args) {
        try (Connection con = DBConnection.getConnection()) {
            String sql = "SELECT * FROM books";
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            System.out.println("ID\tTitle\tAuthor\tAvailable");

```

```

while (rs.next()) {
    int id = rs.getInt("book_id");
    String title = rs.getString("title");
    String author = rs.getString("author");
    boolean available = rs.getBoolean("available");
    System.out.println(id + "\t" + title + "\t" + author + "\t" + (available ?
"Yes" : "No"));
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

Issue Book to Member:

```

import java.sql.*;
import java.util.Scanner;
public class IssueBook {
    public static void main(String[] args) {
        try (Connection con = DBConnection.getConnection());
            Scanner sc = new Scanner(System.in)) {
                System.out.print("Enter Book ID: ");
                int bookId = sc.nextInt();
                System.out.print("Enter Member ID: ");
                int memberId = sc.nextInt();
                String check = "SELECT available FROM books WHERE book_id = ?";
                PreparedStatement checkStmt = con.prepareStatement(check);
                checkStmt.setInt(1, bookId);
                ResultSet rs = checkStmt.executeQuery();

```

```

        if (rs.next() && rs.getBoolean("available")) {
            String issue = "INSERT INTO issued_books (book_id, member_id,
issue_date) VALUES (?, ?, CURDATE())";
            PreparedStatement ps = con.prepareStatement(issue);
            ps.setInt(1, bookId);
            ps.setInt(2, memberId);
            ps.executeUpdate();
            String update = "UPDATE books SET available = FALSE WHERE
book_id = ?";
            PreparedStatement updateStmt = con.prepareStatement(update);
            updateStmt.setInt(1, bookId);
            updateStmt.executeUpdate();
            System.out.println("Book issued successfully.");
        } else {
            System.out.println("Book not available.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Return Book:

```

import java.sql.*;
import java.util.Scanner;
public class ReturnBook {
    public static void main(String[] args) {
        try (Connection con = DBConnection.getConnection());

```

```

        Scanner sc = new Scanner(System.in)) {
        System.out.print("Enter Book ID to return: ");
        int bookId = sc.nextInt();
        System.out.print("Enter Member ID: ");
        int memberId = sc.nextInt();

        String updateReturn = "UPDATE issued_books SET return_date =
CURDATE() " +
        "WHERE book_id = ? AND member_id = ? AND return_date IS
NULL";

        PreparedStatement ps = con.prepareStatement(updateReturn);
        ps.setInt(1, bookId);
        ps.setInt(2, memberId);
        int rows = ps.executeUpdate();
        if (rows > 0) {
            String updateBook = "UPDATE books SET available = TRUE WHERE
book_id = ?";

            PreparedStatement updateBookStmt =
con.prepareStatement(updateBook);
            updateBookStmt.setInt(1, bookId);
            updateBookStmt.executeUpdate();
            System.out.println("Book returned successfully.");
        } else {
            System.out.println("No active issue found for this book and member.");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

17. Create a **Hospital Management System** database. Using JDBC, implement:

- Register new patient
- Assign doctor
- Generate billing

A. DB Connection:

```
import java.sql.Connection;
import java.sql.DriverManager;
public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/hospital_db";
    private static final String USER = "root";
    private static final String PASSWORD = "root";
    public static Connection getConnection() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

Register Patient:

```
import java.sql.*;
import java.util.Scanner;
public class RegisterPatient {
    public static void main(String[] args) {
        try (Connection con = DBConnection.getConnection();
            Scanner sc = new Scanner(System.in)) {
            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 patients (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);
        int rows = ps.executeUpdate();

        System.out.println("Patient registered successfully. Rows inserted: " +
rows);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

AssignDoctor:
import java.sql.*;
import java.util.Scanner;
public class AssignDoctor {
    public static void main(String[] args) {

```

```

try (Connection con = DBConnection.getConnection());
    Scanner sc = new Scanner(System.in)) {
    System.out.print("Patient ID: ");
    int patientId = sc.nextInt();
    System.out.print("Doctor ID: ");
    int doctorId = sc.nextInt();
    sc.nextLine();
    System.out.print("Appointment Date (YYYY-MM-DD): ");
    String date = sc.nextLine();
    String sql = "INSERT INTO appointments (patient_id, doctor_id,
appointment_date) VALUES (?, ?, ?)";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, patientId);
    ps.setInt(2, doctorId);
    ps.setDate(3, Date.valueOf(date));
    int rows = ps.executeUpdate();
    System.out.println("Appointment created successfully. Rows inserted: "
+ rows);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

GenerateBilling:
import java.sql.*;
import java.util.Scanner;
public class GenerateBilling {
    public static void main(String[] args) {

```

```

try (Connection con = DBConnection.getConnection());
    Scanner sc = new Scanner(System.in)) {
    System.out.print("Patient ID: ");
    int patientId = sc.nextInt();
    System.out.print("Billing Amount: ");
    double amount = sc.nextDouble();
    sc.nextLine();
    System.out.print("Billing Description: ");
    String description = sc.nextLine();
    String sql = "INSERT INTO billing (patient_id, amount, billing_date,
description) VALUES (?, ?, CURDATE(), ?)";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setInt(1, patientId);
    ps.setDouble(2, amount);
    ps.setString(3, description);
    int rows = ps.executeUpdate();
    System.out.println("Bill generated successfully. Rows inserted: " + rows);
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

18. Write a JDBC-based report generator that exports data from a MySQL table to a text or CSV file.

```

A. import java.sql.*;
import java.io.*;
public class ExportReportToCSV {
    public static void main(String[] args) {

```

```

String jdbcURL = "jdbc:mysql://localhost:3306/hospital_db";
String username = "root";
String password = "root";
String csvFilePath = "patients_report.csv";
try (
    Connection con = DriverManager.getConnection(jdbcURL, username,
password);
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM patients");
    BufferedWriter writer = new BufferedWriter(new
FileWriter(csvFilePath));
) {
    System.out.println("Connected to database.");
    // Write header
    writer.write("patient_id,name,age,gender,phone,address");
    writer.newLine();
    // Write rows
    while (rs.next()) {
        int id = rs.getInt("patient_id");
        String name = rs.getString("name");
        int age = rs.getInt("age");
        String gender = rs.getString("gender");
        String phone = rs.getString("phone");
        String address = rs.getString("address");

        String row = id + "," + name + "," + age + "," + gender + "," + phone +
"," + address;

        writer.write(row);
        writer.newLine();
    }
}

```

```
    }  
    writer.flush();  
    System.out.println("Data exported to " + csvFilePath);  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}  
}
```