Experiment-7

Student Name: Neha Kumari   **UID:** 22BCS10009
Branch: CSE   **Section/Group:** IOT-618/A
Semester: 7th   **Date of Performance:** 21/03/25
Subject Name: Java Lab   **Subject Code:** 22CSH-359

Problem-1 (Easy)

1. **Aim:**

   Create a Java program to connect to a MySQL database and fetch data from a single table. The program should: Use DriverManager and Connection objects. Retrieve and display all records from a table named Employee with columns EmpID, Name, and Salary.

2. **Implementation/Code:**

```java
import java.sql.*; public class

MySQLConnection { public static

void main(String[] args) {

 // Database credentials

  String url = "jdbc:mysql://localhost:3306/your_database"; // Replace with your
database name

  String user = "your_username"; // Replace with your username

  String password = "your_password"; // Replace with your password

     // SQL query to fetch employee records

     String query = "SELECT * FROM Employee";

// Establish connection and retrieve data       try {

       // Load MySQL JDBC Driver

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

       // Establish connection

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

       Statement stmt = conn.createStatement();
       ResultSet rs = stmt.executeQuery(query);
```

```
        // Display employee records
        System.out.println("EmpID | Name | Salary");
while (rs.next()) {              int id = rs.getInt("EmpID");
String name = rs.getString("Name");             double salary
= rs.getDouble("Salary");            System.out.println(id + "
| " + name + " | " + salary);
        }

        // Close resources
rs.close();
stmt.close();
conn.close();         } catch
(Exception e)
{ e.printStackTrace();
     }
   }
}
```

3. Output:

```
                                                                  input
java.lang.ClassNotFoundException: com.mysql.cj.jdbc.Driver
        at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:641)
        at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:188)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:528)
        at java.base/java.lang.Class.forName0(Native Method)
        at java.base/java.lang.Class.forName(Class.java:462)
        at java.base/java.lang.Class.forName(Class.java:453)
        at MySQLConnection.main(MySQLConnection.java:16)


...Program finished with exit code 0
Press ENTER to exit console.
```

Problem-2 (Medium)

1. **Aim:**

   Build a program to perform CRUD operations (Create, Read, Update, Delete) on a database table Product with columns: ProductID, ProductName, Price, and Quantity. The program should include: Menu-driven options for each operation. Transaction handling to ensure data integrity.

2. **Implementation/Code:**   import java.sql.*;
   import java.util.Scanner;

```java
public class ProductCRUD {
    static final String URL = "jdbc:mysql://localhost:3306/your_database";
    static final String USER = "your_username";     static final String
    PASSWORD = "your_password";

    public static void main(String[] args) { try (Connection conn =
        DriverManager.getConnection(URL, USER, PASSWORD);          Scanner
        scanner = new Scanner(System.in)) {
```

3

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

        while (true) {
            System.out.println("\nProduct CRUD Menu:");
            System.out.println("1. Add Product");
            System.out.println("2. View Products");
            System.out.println("3. Update Product");
            System.out.println("4. Delete Product");
            System.out.println("5. Exit");
System.out.print("Enter choice: ");                int
choice = scanner.nextInt();
scanner.nextLine();

            switch (choice) {
case 1:
                addProduct(conn, scanner);
                break;
case 2:
                viewProducts(conn);
                break;
case 3:
                updateProduct(conn, scanner);
break;            case 4:
                deleteProduct(conn, scanner);
                break;
case 5:
                System.out.println("Exiting...");
                return;
default:
                System.out.println("Invalid choice, try again.");
            }
        }
    } catch (Exception e)
{ e.printStackTrace();
    }
  }

    private static void addProduct(Connection conn, Scanner scanner) throws
SQLException {
```

```java
        System.out.print("Enter Product Name: ");
        String name = scanner.nextLine();
System.out.print("Enter Price: ");        double
price = scanner.nextDouble();
System.out.print("Enter Quantity: ");
        int quantity = scanner.nextInt();

        String query = "INSERT INTO Product (ProductName, Price, Quantity) VALUES
(?, ?, ?)";
        try (PreparedStatement pstmt = conn.prepareStatement(query))
{ pstmt.setString(1, name);        pstmt.setDouble(2, price);
pstmt.setInt(3, quantity);        pstmt.executeUpdate();
        System.out.println("Product added successfully.");
    }
  }

  private static void viewProducts(Connection conn) throws SQLException
{ String query = "SELECT * FROM Product";
    try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery(query)) {
        System.out.println("\nProduct List:");
        while (rs.next()) {
          System.out.println(rs.getInt("ProductID") + " | " +
rs.getString("ProductName") + " | " + rs.getDouble("Price") + " | " +
rs.getInt("Quantity"));
        }
    }
  }

  private static void updateProduct(Connection conn, Scanner scanner) throws
SQLException {
    System.out.print("Enter Product ID to update: ");
    int id = scanner.nextInt(); scanner.nextLine();
    System.out.print("Enter new Product Name:
    ");
    String name = scanner.nextLine();
System.out.print("Enter new Price: ");
```

```
        double price = scanner.nextDouble();
        System.out.print("Enter new Quantity: ");        int
        quantity = scanner.nextInt();

            String query = "UPDATE Product SET ProductName=?, Price=?, Quantity=?
        WHERE ProductID=?";
            try (PreparedStatement pstmt = conn.prepareStatement(query))
        { pstmt.setString(1, name);          pstmt.setDouble(2, price);
        pstmt.setInt(3, quantity);           pstmt.setInt(4, id);
        pstmt.executeUpdate();
                System.out.println("Product updated successfully.");
            }
        }

        private static void deleteProduct(Connection conn, Scanner scanner) throws
        SQLException {
            System.out.print("Enter Product ID to delete: ");
            int id = scanner.nextInt();

            String query = "DELETE FROM Product WHERE ProductID=?";
        try (PreparedStatement pstmt = conn.prepareStatement(query))
        { pstmt.setInt(1, id);          pstmt.executeUpdate();
                System.out.println("Product deleted successfully.");
            }
        }
        }
```

3. Output:

```
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/your_database
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:707)
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:230)
        at ProductCRUD.main(ProductCRUD.java:10)


...Program finished with exit code 0
Press ENTER to exit console.
```

Problem-3 (Hard)

1. **Aim:**

Develop a Java application using JDBC and MVC architecture to manage student data. The application should: Use a Student class as the model with fields like StudentID, Name, Department, and Marks. Include a database table to store student data. Allow the user to perform CRUD operations through a simple menu-driven view. Implement database operations in a separate controller class.

2. **Implementation/Code:**   import java.sql.*;
   import java.util.Scanner;

```java
public class StudentManagementApp {
    static final String URL = "jdbc:mysql://localhost:3306/StudentDB";
    static final String USER = "your_username";     static final String
    PASSWORD = "your_password";

    public static void main(String[] args) { try (Connection conn =
        DriverManager.getConnection(URL, USER, PASSWORD);          Scanner
        scanner = new Scanner(System.in))
        { Class.forName("com.mysql.cj.jdbc.Driver");          conn.setAutoCommit(false);

            while (true) {
```

```java
System.out.println("\nStudent Management System:");
System.out.println("1. Add Student");
System.out.println("2. View Students");
System.out.println("3. Update Student");
System.out.println("4. Delete Student");
System.out.println("5. Exit");
System.out.print("Enter choice: ");             int
choice = scanner.nextInt();
scanner.nextLine();

switch (choice) {
case 1:
        addStudent(conn, scanner);
        break;
case 2:
        viewStudents(conn);
        break;
case 3:
        updateStudent(conn, scanner);
break;          case 4:
        deleteStudent(conn, scanner);
        break;
case 5:
        System.out.println("Exiting...");
        return;
default:
        System.out.println("Invalid choice, try again.");
    }
  }
} catch (Exception e)
{ e.printStackTrace();
  }
}

private static void addStudent(Connection conn, Scanner scanner) throws
SQLException {
    System.out.print("Enter Student ID: ");
    int id = scanner.nextInt();
scanner.nextLine();
```

```java
        System.out.print("Enter Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Department: ");
        String dept = scanner.nextLine();
System.out.print("Enter Marks: ");        double
marks = scanner.nextDouble();

        String query = "INSERT INTO Student (StudentID, Name, Department, Marks)
VALUES (?, ?, ?, ?)";
        try (PreparedStatement pstmt = conn.prepareStatement(query))
{ pstmt.setInt(1, id);        pstmt.setString(2, name);
pstmt.setString(3, dept);          pstmt.setDouble(4, marks);
pstmt.executeUpdate();          conn.commit();
            System.out.println("Student added successfully.");
        } catch (SQLException e)
            { conn.rollback();
throw e;
        }
    }

    private static void viewStudents(Connection conn) throws SQLException
{ String query = "SELECT * FROM Student";
        try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery(query)) {
        System.out.println("\nStudent List:");
        while (rs.next()) {
            System.out.println(rs.getInt("StudentID") + " | " + rs.getString("Name") + " | "
+ rs.getString("Department") + " | " + rs.getDouble("Marks"));
        }
        }
    }

    private static void updateStudent(Connection conn, Scanner scanner) throws
SQLException {
        System.out.print("Enter Student ID to update: ");
        int id = scanner.nextInt();
scanner.nextLine();
```

```java
        System.out.print("Enter new Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter new Department: ");
        String dept = scanner.nextLine();
System.out.print("Enter new Marks: ");
        double marks = scanner.nextDouble();

        String query = "UPDATE Student SET Name=?, Department=?, Marks=? WHERE
StudentID=?";
        try (PreparedStatement pstmt = conn.prepareStatement(query))
{ pstmt.setString(1, name);          pstmt.setString(2, dept);
pstmt.setDouble(3, marks);              pstmt.setInt(4, id);
pstmt.executeUpdate();          conn.commit();
            System.out.println("Student updated successfully.");
        } catch (SQLException e)
            { conn.rollback();
throw e;
        }
    }

    private static void deleteStudent(Connection conn, Scanner scanner) throws
SQLException {
        System.out.print("Enter Student ID to delete: ");
int id = scanner.nextInt();

        String query = "DELETE FROM Student WHERE StudentID=?";
        try (PreparedStatement pstmt = conn.prepareStatement(query)) {
pstmt.setInt(1, id);            pstmt.executeUpdate();
conn.commit();
            System.out.println("Student deleted successfully.");
        } catch (SQLException e)
            { conn.rollback();
            throw e;
        }
    }
}
```

3. Output:

```
java.sql.SQLException: No suitable driver found for jdbc:mysql://localhost:3306/StudentDB
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:707)
        at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:230)
        at StudentManagementApp.main(StudentManagementApp.java:10)


...Program finished with exit code 0
Press ENTER to exit console.
```