Experiment 7

Student Name: Aashish Pant **UID: 22BCS12862**
Branch: CSE **Section: IOT-642 -B**
Semester: 6th **DOP: 17/03/025**
Subject: Java **Subject Code:22CSH-359**

Problem - 7.1

Aim:

1. Setup MySQL Database
   - Ensure MySQL is installed and running.
   - Create a database and an `Employee` table with columns `EmpID`, `Name`, and `Salary`.

2. Update Database Credentials
   - Replace `your_database`, `your_username`, and `your_password` in the code with actual database credentials.

3. Add MySQL JDBC Driver
   - Download and add `mysql-connector-java.jar` to your project’s classpath.

4. Compile and Run the Program
   - Compile: `javac MySQLConnection.java`
   - Run: `java MySQLConnection`

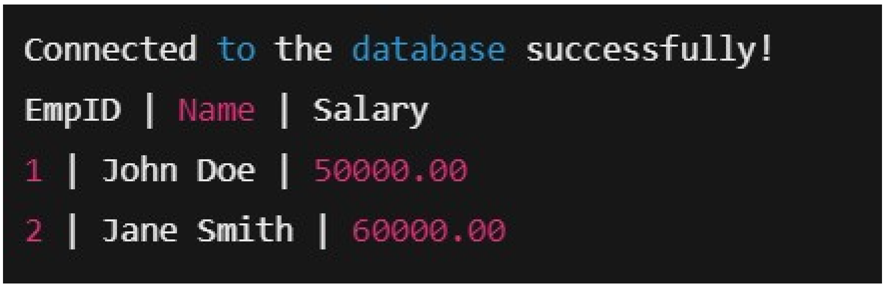
5. Verify Output
   - Ensure that employee records are displayed correctly from the database.

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class MySQLConnection {
    public static void main(String[] args) {
        // Database credentials
        String url = "jdbc:mysql://localhost:3306/your_database";
```

```
            String user = "your_username";
        String password = "your_password";

        // SQL query to fetch data
        String query = "SELECT * FROM Employee";

        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 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 (ClassNotFoundException e) {
            System.out.println("MySQL JDBC Driver not found.");
            e.printStackTrace();
        } catch (SQLException e) {
            System.out.println("Database connection error.");
            e.printStackTrace();
        }
    }
}
```

Output:



```
Connected to the database successfully!
EmpID | Name | Salary
1 | John Doe | 50000.00
2 | Jane Smith | 60000.00
```

Problem - 7.2

Aim : Instructions to Run the Java CRUD Program

1. Setup MySQL Database

- Ensure MySQL is installed and running.

- Create a database and a `Product` table with columns `ProductID`, `ProductName`, `Price`, and `Quantity`

2. Update Database Credentials

- Replace `your_database`, `your_username`, and `your_password` in the code with actual database credentials

3. Add MySQL JDBC Driver

- Download and add `mysql-connector-java.jar` to your project's classpath.

4. Compile and Run the Progra

- Compile: `javac ProductCRUD.java`

- Run: `java ProductCRUD`

5. Menu-Driven Operations

- Select options to **Create**, **Read**, **Update**, or **Delete** products.

- Input values as prompted.

6. Transaction Handling

- Transactions ensure data integrity.

- If an error occurs, changes are rolled back.

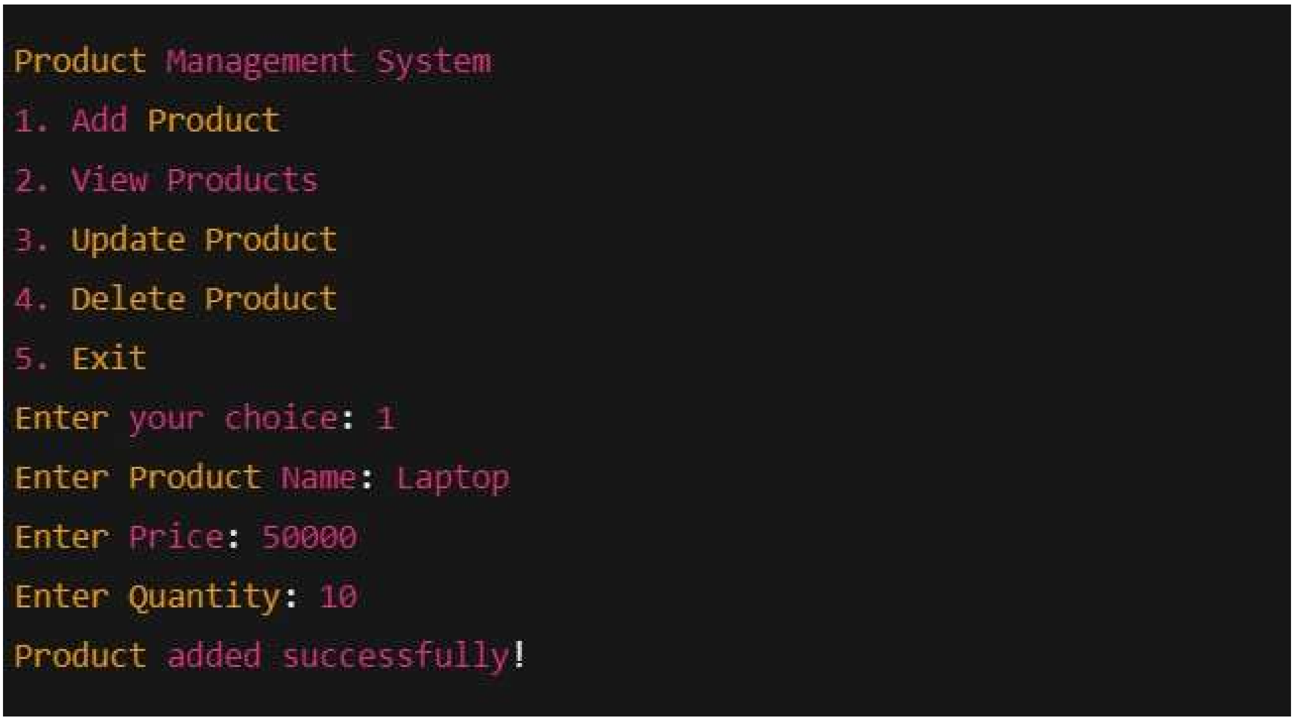
7. Verify Output

- Ensure product records are correctly manipulated in the database.

Code :

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

public class ProductCRUD {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/your_database";
        String user = "your_username";
        String password = "your_password";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user, password);
            Scanner scanner = new Scanner(System.in);

            while (true) {
                System.out.println("1. Add Product\n2. View Products\n3. Update Product\n4. Delete
Product\n5. Exit");
                System.out.print("Enter choice: ");
                int choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        System.out.print("Enter Product Name: ");
                        String name = scanner.next();
                        System.out.print("Enter Price: ");
                        double price = scanner.nextDouble();
                        System.out.print("Enter Quantity: ");
                        int quantity = scanner.nextInt();
                        String insertQuery = "INSERT INTO Product (ProductName, Price, Quantity) VALUES
(?, ?, ?)";
                        PreparedStatement pstmt = conn.prepareStatement(insertQuery);
                        pstmt.setString(1, name);
                        pstmt.setDouble(2, price);
                        pstmt.setInt(3, quantity);
                        pstmt.executeUpdate();
                        System.out.println("Product Added Successfully!");
                        pstmt.close();
                        break;

                    case 2:
                        Statement stmt = conn.createStatement();
                        ResultSet rs = stmt.executeQuery("SELECT * FROM Product");
                        System.out.println("ProductID | ProductName | Price | Quantity");
                        while (rs.next()) {
                            System.out.println(rs.getInt("ProductID") + " | " + rs.getString("ProductName") + " | "
+ rs.getDouble("Price") + " | " + rs.getInt("Quantity"));
                        }
                        rs.close();
                        stmt.close();
                        break;

                    case 3:
                        System.out.print("Enter ProductID to Update: ");
                        int updateId = scanner.nextInt();
                        System.out.print("Enter New Price: ");
                        double newPrice = scanner.nextDouble();
                        System.out.print("Enter New Quantity: ");
                        int newQuantity = scanner.nextInt();
                        String updateQuery = "UPDATE Product SET Price = ?, Quantity = ? WHERE
ProductID = ?";
                        pstmt = conn.prepareStatement(updateQuery);
                        pstmt.setDouble(1, newPrice);
                        pstmt.setInt(2, newQuantity);
                        pstmt.setInt(3, updateId);
                        pstmt.executeUpdate();
                        System.out.println("Product Updated Successfully!");
                        pstmt.close();
                        break;

                    case 4:
                        System.out.print("Enter ProductID to Delete: ");
                        int deleteId = scanner.nextInt();
                        String deleteQuery = "DELETE FROM Product WHERE ProductID = ?";
                        pstmt = conn.prepareStatement(deleteQuery);
                        pstmt.setInt(1, deleteId);
                        pstmt.executeUpdate();
                        System.out.println("Product Deleted Successfully!");
                        pstmt.close();
                        break;

                    case 5:
                        System.out.println("Exiting...");
                        conn.close();
                        scanner.close();
                        return;

                    default:
                        System.out.println("Invalid Choice!");
                }
            }
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        }
}
```

Output :



```
Product Management System
1. Add Product
2. View Products
3. Update Product
4. Delete Product
5. Exit
Enter your choice: 1
Enter Product Name: Laptop
Enter Price: 50000
Enter Quantity: 10
Product added successfully!
```

Problem - 7.3

1. Setup MySQL Database:
   - Install and start MySQL.
   - Create a database (e.g., `StudentDB`).
   - Create a table:
     ```sql
     CREATE TABLE Student (
         StudentID INT PRIMARY KEY,
         Name VARCHAR(100),
         Department VARCHAR(50),
         Marks DOUBLE
     );
     ```

2. Update Database Credentials:
   - Modify `URL`, `USER`, and `PASSWORD` in the code to match your MySQL database credentials.

3. Add MySQL JDBC Driver:
   - Download and add `mysql-connector-java.jar` to your project's classpath.

4. Compile and Run the Program:

- Compile: `javac StudentManagementApp.java`
- Run: `java StudentManagementApp`

5. Menu-Driven Operations:
- Add Student: Enter StudentID, Name, Department, and Marks.
- View Students: Displays all students in the table.
- Update Student: Modify Name, Department, or Marks using StudentID.
- Delete Student: Remove a student using StudentID.
- Exit: Quit the program.

6. Transaction Handling:
- Ensures data integrity by using `conn.setAutoCommit(false)` and `conn.commit()`.
- Rolls back changes in case of errors.

7. Verify Database Changes:
- Use `SELECT * FROM Student;` in MySQL to confirm modifications.

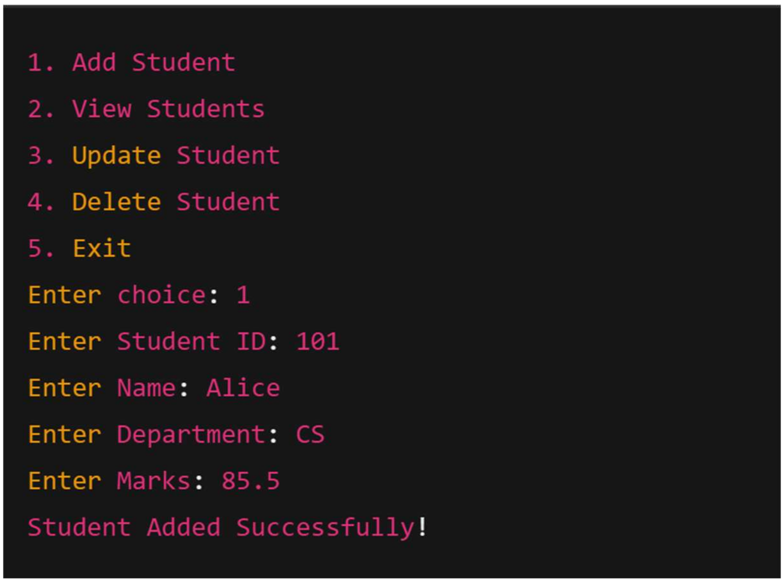
Code :

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

public class StudentManagementApp {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/StudentDB";
        String user = "your_username";
        String password = "your_password";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user, password);
            conn.setAutoCommit(false);
            Scanner scanner = new Scanner(System.in);

            while (true) {
                System.out.println("1. Add Student\n2. View Students\n3. Update Student\n4. Delete Student\n5.
Exit");
                System.out.print("Enter choice: ");
                int choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        System.out.print("Enter Student ID: ");
                        int studentId = scanner.nextInt();
                        System.out.print("Enter Name: ");
                        String name = scanner.next();
                        System.out.print("Enter Department: ");
                        String department = scanner.next();
```

```
              System.out.print("Enter Marks: ");
              double marks = scanner.nextDouble();
              String insertQuery = "INSERT INTO Student (StudentID, Name, Department, Marks)
VALUES (?, ?, ?, ?)";
              PreparedStatement pstmt = conn.prepareStatement(insertQuery);
              pstmt.setInt(1, studentId);
              pstmt.setString(2, name);
              pstmt.setString(3, department);
              pstmt.setDouble(4, marks);
              pstmt.executeUpdate();
              conn.commit();
              System.out.println("Student Added Successfully!");
              pstmt.close();
              break;

            case 2:
              Statement stmt = conn.createStatement();
              ResultSet rs = stmt.executeQuery("SELECT * FROM Student");
              System.out.println("StudentID | Name | Department | Marks");
              while (rs.next()) {
                System.out.println(rs.getInt("StudentID") + " | " + rs.getString("Name") + " | " +
rs.getString("Department") + " | " + rs.getDouble("Marks"));
              }
              rs.close();
              stmt.close();
              break;

            case 3:
              System.out.print("Enter Student ID to Update: ");
              int updateId = scanner.nextInt();
              System.out.print("Enter New Marks: ");
              double newMarks = scanner.nextDouble();
              String updateQuery = "UPDATE Student SET Marks = ? WHERE StudentID = ?";
              pstmt = conn.prepareStatement(updateQuery);
              pstmt.setDouble(1, newMarks);
              pstmt.setInt(2, updateId);
              pstmt.executeUpdate();
              conn.commit();
              System.out.println("Student Updated Successfully!");
              pstmt.close();
              break;

            case 4:
              System.out.print("Enter Student ID to Delete: ");
              int deleteId = scanner.nextInt();
              String deleteQuery = "DELETE FROM Student WHERE StudentID = ?";
              pstmt = conn.prepareStatement(deleteQuery);
              pstmt.setInt(1, deleteId);
              pstmt.executeUpdate();
              conn.commit();
              System.out.println("Student Deleted Successfully!");
```

```
                        pstmt.close();
                        break;

                    case 5:
                        System.out.println("Exiting...");
                        conn.close();
                        scanner.close();
                        return;

                    default:
                        System.out.println("Invalid Choice!");
                }
            }
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output :



```
1. Add Student
2. View Students
3. Update Student
4. Delete Student
5. Exit
Enter choice: 1
Enter Student ID: 101
Enter Name: Alice
Enter Department: CS
Enter Marks: 85.5
Student Added Successfully!
```