



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 7

Student Name: Aditya Mehta

Branch: CSE

Semester: 6th

Subject: Java

UID: 22BCS17094

Section: IOT-642 -B

DOP: 17/03/025

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.*;
```

```
public class MySQLConnection
```

```
{
```

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

```
    {
```

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

```
        String user = "Aditya";
```



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

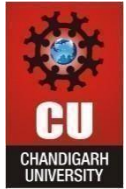
```
String password = "123456";
Connection conn = null;
Statement stmt = null;
ResultSet rs = null;

try {

    Class.forName("com.mysql.cj.jdbc.Driver");
    conn = DriverManager.getConnection(url, user, password);
    System.out.println("Connected to the database successfully!");

    stmt = conn.createStatement();
    String query = "SELECT * FROM Employee";
    rs = stmt.executeQuery(query);

    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);
    }
} catch (ClassNotFoundException e) {
    System.out.println("MySQL JDBC Driver not found!");
    e.printStackTrace();
} catch (SQLException e) {
    System.out.println("Database connection error!");
    e.printStackTrace();
} finally {
    try {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
```



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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.



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

7. Verify Output

- Ensure product records are correctly manipulated in the database.

Code :

```
import java.sql.*;

import java.util.Scanner

public class ProductCRUD {

    private static final String URL = "jdbc:mysql://localhost:3306/your_database";

    private static final String USER = "Aditya";

    private static final String PASSWORD = "123456";

    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")

            while (true) {

                System.out.println("\nProduct Management System");

                System.out.println("1. Add Product");

                System.out.println("2. View Products");

                System.out.println("3. Update Product");

                System.out.println("4. Delete Product");

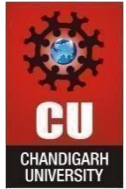
                System.out.println("5. Exit");

                System.out.print("Enter your choice: ");

                int choice = scanner.nextInt();

                switch (choice) {

                    case 1: addProduct(conn, scanner); break;
```



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        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 {

    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 sql = "INSERT INTO Product (ProductName, Price, Quantity) VALUES (?, ?, ?)";

    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(1, name);

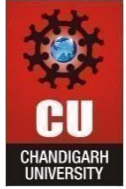
        pstmt.setDouble(2, price);

        pstmt.setInt(3, quantity);

        pstmt.executeUpdate();

        System.out.println("Product added successfully!");

    }
```



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}
```

```
private static void viewProducts(Connection conn) throws SQLException {
```

```
    String sql = "SELECT * FROM Product";
```

```
    try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {
```

```
        System.out.println("\nProductID | ProductName | Price | Quantity");
```

```
        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();
```

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

```
    double price = scanner.nextDouble();
```

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

```
    int quantity = scanner.nextInt();
```

```
    String sql = "UPDATE Product SET Price = ?, Quantity = ? WHERE ProductID = ?";
```

```
    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
```

```
        pstmt.setDouble(1, price);
```

```
        pstmt.setInt(2, quantity);
```



DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
pstmt.setInt(3, id);

int rowsUpdated = pstmt.executeUpdate();

if (rowsUpdated > 0) {

    System.out.println("Product updated successfully!");

} else {

    System.out.println("Product not found.");

}

}

}

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

    System.out.print("Enter Product ID to delete: ");

    int id = scanner.nextInt();

    String sql = "DELETE FROM Product WHERE ProductID = ?";

    try (PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setInt(1, id);

        int rowsDeleted = pstmt.executeUpdate();

        if (rowsDeleted > 0) {

            System.out.println("Product deleted successfully!");

        } else {

            System.out.println("Product not found.");

        }

    }

}

}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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!
```