

Experiment - 7

Student Name: Shashwat Chalana

Branch: BE-CSE

Semester: 5th

Subject Name: PBLJ

UID: 23BCS10511

Section/Group: KRG-2B

Date of Performance: 15/10/25

Subject Code: 23CSH-304

Aim:

To build a Java program that performs CRUD (Create, Read, Update, Delete) operations on a Product table using JDBC with transaction handling.

Objective:

To learn how to implement CRUD operations using JDBC, apply transaction handling, and use a menu-driven program for database operations.

Apparatus / Input Used:

- Java (JDK 8 or above)
- MySQL Database
- JDBC API
- MySQL Table: **Product(ProductID, ProductName, Price, Quantity)**
- IDE: Eclipse / IntelliJ / VS Code

Procedure:

1. Create a MySQL table **Product(ProductID, ProductName, Price, Quantity)**
2. Load the MySQL JDBC Driver using `Class.forName()`
3. Establish a connection using `DriverManager.getConnection()`
4. Create a menu-driven program with options: Add, View, Update, Delete
5. Use **PreparedStatement** for secure queries
6. Use `connection.setAutoCommit(false)` for manual transaction mode
7. Use `commit()` on successful operations
8. Use `rollback()` on errors
9. Close all JDBC resources properly (Connection, Statement, ResultSet)

Code

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

public class EXPERIMENT7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            Class.forName("org.postgresql.Driver");

            Connection con = DriverManager.getConnection(
                "jdbc:postgresql://localhost:5432/testdb",
                "postgres",
                "root"
            );

            con.setAutoCommit(false);

            while (true) {
                System.out.println("\n--- Product Management Menu ---");
                System.out.println("1. Add Product");
                System.out.println("2. View All 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 = sc.nextInt();

                if (choice == 1) {
                    PreparedStatement ps = con.prepareStatement(
                        "INSERT INTO Product (ProductID, ProductName, Price, Quantity)
VALUES (?, ?, ?, ?)"
                    );
                    System.out.print("Enter Product ID: ");
                    ps.setInt(1, sc.nextInt());
                    System.out.print("Enter Product Name: ");
                    ps.setString(2, sc.next());
                    System.out.print("Enter Price: ");
                    ps.setDouble(3, sc.nextDouble());
                    System.out.print("Enter Quantity: ");
                    ps.setInt(4, sc.nextInt());

                    ps.executeUpdate();
                    con.commit();
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("Product Added Successfully!");
    }

    else if (choice == 2) {
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery("SELECT * FROM Product");

        System.out.println("\nProductID | Name | Price | Quantity");
        System.out.println("-----");

        while (rs.next()) {
            System.out.println(
                rs.getInt("ProductID") + " | " +
                rs.getString("ProductName") + " | " +
                rs.getDouble("Price") + " | " +
                rs.getInt("Quantity")
            );
        }
    }

    else if (choice == 3) {
        PreparedStatement ps = con.prepareStatement(
            "UPDATE Product SET Price=?, Quantity=? WHERE ProductID=?"
        );
        System.out.print("Enter Product ID: ");
        ps.setInt(3, sc.nextInt());
        System.out.print("Enter New Price: ");
        ps.setDouble(1, sc.nextDouble());
        System.out.print("Enter New Quantity: ");
        ps.setInt(2, sc.nextInt());

        ps.executeUpdate();
        con.commit();
        System.out.println("Product Updated Successfully!");
    }

    else if (choice == 4) {
        PreparedStatement ps = con.prepareStatement(
            "DELETE FROM Product WHERE ProductID=?"
        );
        System.out.print("Enter Product ID: ");
        ps.setInt(1, sc.nextInt());

        ps.executeUpdate();
        con.commit();
        System.out.println("Product Deleted Successfully!");
    }
}
```

```
        else if (choice == 5) {
            System.out.println("Exiting...");
            break;
        }

        else {
            System.out.println("Invalid Choice!");
        }
    }

    con.close();
    sc.close();
} catch (Exception e) {
    System.out.println("Error! Rolling Back...");
    e.printStackTrace();
}
}
```

OUTPUT:

```
PS C:\Users\ASUS\Desktop\Sem 5\PBLJ_23B
>>

--- Product Management Menu ---
1. Add Product
2. View All Products
3. Update Product
4. Delete Product
5. Exit
Enter choice: 1
Enter Product ID: 1
Enter Product Name: Phone
Enter Price: 10000
Enter Quantity: 5
Product Added Successfully!
```

```
--- Product Management Menu ---
```

1. Add Product
2. View All Products
3. Update Product
4. Delete Product
5. Exit

```
Enter choice: 2
```

```
ProductID | Name | Price | Quantity
```

```
-----
```

```
1 | Phone | 10000.0 | 5
```

```
--- Product Management Menu ---
```

1. Add Product
2. View All Products
3. Update Product
4. Delete Product
5. Exit

```
Enter choice: 3
```

```
Enter Product ID: 1
```

```
Enter New Price: 100000
```

```
Enter New Quantity: 40
```

```
Product Updated Successfully!
```

```
--- Product Management Menu ---
```

1. Add Product
2. View All Products
3. Update Product
4. Delete Product
5. Exit

```
Enter choice: 5
```

```
Exiting...
```

```
PS C:\Users\ASUS\Desktop\Sem 5\PBLJ_23BCS10889_KRG-2B\Experiment_Codes>
```