Java Full Stack Lab Manual - JDBC Experiment

Aim

To write a JDBC application that interacts with the database using PreparedStatement to:

- Create a Student table.

- Insert, display, update, and delete records.

Technologies Used

Java, JDBC, MySQL Database, MySQL JDBC Driver

Database Table Structure

Table Name: Student

Columns:

- RollNo (INT)

- Name (VARCHAR)

- Address (VARCHAR)

Procedure

1. Load JDBC driver.

2. Establish a connection to MySQL.

3. Use PreparedStatement for the following operations:

   - Create Table

   - Insert Initial Records

   - Display Records

   - Insert 2 More Records

   - Update 1 Record

   - Delete 1 Record

   - Display Final Records

Java Full Stack Lab Manual - JDBC Experiment

Java Program

```
import java.sql.*;

public class StudentPreparedStatementApp {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/college";
        String username = "root";
        String password = "your_password";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, username, password);

            Statement stmt = con.createStatement();
            stmt.executeUpdate("CREATE TABLE IF NOT EXISTS Student (" +
                               "RollNo INT PRIMARY KEY, Name VARCHAR(50), Address VARCHAR(100))");

            String insertSQL = "INSERT INTO Student VALUES (?, ?, ?)";
            PreparedStatement psInsert = con.prepareStatement(insertSQL);

            psInsert.setInt(1, 1);
            psInsert.setString(2, "John");
            psInsert.setString(3, "Hyderabad");
            psInsert.executeUpdate();

            psInsert.setInt(1, 2);
            psInsert.setString(2, "Alice");
            psInsert.setString(3, "Mumbai");
            psInsert.executeUpdate();

            psInsert.setInt(1, 3);
            psInsert.setString(2, "Bob");
            psInsert.setString(3, "Chennai");
            psInsert.executeUpdate();

            System.out.println("--- Initial Records ---");
            PreparedStatement psSelect = con.prepareStatement("SELECT * FROM Student");
            ResultSet rs1 = psSelect.executeQuery();
            while (rs1.next()) {
                System.out.println(rs1.getInt("RollNo") + ", " +
                                   rs1.getString("Name") + ", " +
                                   rs1.getString("Address"));
            }

            psInsert.setInt(1, 4);
            psInsert.setString(2, "David");
            psInsert.setString(3, "Delhi");
```

```
            psInsert.executeUpdate();

            psInsert.setInt(1, 5);
            psInsert.setString(2, "Eva");
            psInsert.setString(3, "Bangalore");
            psInsert.executeUpdate();

            String updateSQL = "UPDATE Student SET Address=? WHERE RollNo=?";
            PreparedStatement psUpdate = con.prepareStatement(updateSQL);
            psUpdate.setString(1, "Pune");
            psUpdate.setInt(2, 2);
            psUpdate.executeUpdate();

            String deleteSQL = "DELETE FROM Student WHERE RollNo=?";
            PreparedStatement psDelete = con.prepareStatement(deleteSQL);
            psDelete.setInt(1, 1);
            psDelete.executeUpdate();

            System.out.println("--- Final Records ---");
            ResultSet rs2 = psSelect.executeQuery();
            while (rs2.next()) {
                System.out.println(rs2.getInt("RollNo") + ", " +
                                   rs2.getString("Name") + ", " +
                                   rs2.getString("Address"));
            }

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

Java Full Stack Lab Manual - JDBC Experiment

Expected Output

```
Student table created successfully.

--- Initial Records ---
1, John, Hyderabad
2, Alice, Mumbai
3, Bob, Chennai

Two more records inserted.

Record updated where RollNo=2.

Record deleted where RollNo=1.

--- Final Records ---
2, Alice, Pune
3, Bob, Chennai
4, David, Delhi
5, Eva, Bangalore
```

Result

Successfully implemented a JDBC application using the PreparedStatement object to perform create, read, update, and delete operations on the Student table.