

# Java Full Stack Lab Manual - JDBC Experiment

## Aim

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

- Create a Student table.
- Insert, display, update, and delete records using the Statement object.

## 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. Create a Statement object.
4. Perform operations using Statement:
  - Create Table
  - Insert Initial Records
  - Display Records
  - Insert 2 New Records
  - Update 1 Record
  - Delete 1 Record
  - Display Final Records

# Java Full Stack Lab Manual - JDBC Experiment

## Java Program

```
import java.sql.*;

public class StudentJDBCApp {
    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))");

            stmt.executeUpdate("INSERT INTO Student VALUES (1, 'John', 'Hyderabad')");
            stmt.executeUpdate("INSERT INTO Student VALUES (2, 'Alice', 'Mumbai')");
            stmt.executeUpdate("INSERT INTO Student VALUES (3, 'Bob', 'Chennai')");

            ResultSet rs1 = stmt.executeQuery("SELECT * FROM Student");
            while (rs1.next()) {
                System.out.println(rs1.getInt("RollNo") + ", " +
                                   rs1.getString("Name") + ", " +
                                   rs1.getString("Address"));
            }

            stmt.executeUpdate("INSERT INTO Student VALUES (4, 'David', 'Delhi')");
            stmt.executeUpdate("INSERT INTO Student VALUES (5, 'Eva', 'Bangalore')");
            stmt.executeUpdate("UPDATE Student SET Address='Pune' WHERE RollNo=2");
            stmt.executeUpdate("DELETE FROM Student WHERE RollNo=1");

            ResultSet rs2 = stmt.executeQuery("SELECT * FROM Student");
            while (rs2.next()) {
                System.out.println(rs2.getInt("RollNo") + ", " +
                                   rs2.getString("Name") + ", " +
                                   rs2.getString("Address"));
            }

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

## Java Full Stack Lab Manual - JDBC Experiment

### Expected Output

```
Student table created successfully.  
Initial records inserted.
```

```
--- 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 Statement object to perform create, read, update, and delete operations on the Student table.