

# Java Full Stack Lab Manual - JDBC Experiment

## Aim

To design a JDBC application that demonstrates the use of Updatable ResultSet functionality.

## Technologies Used

Java, JDBC, MySQL Database, MySQL JDBC Driver

## Database Table Structure

Table Name: Employee

Columns:

- EmpID (INT)
- Name (VARCHAR)
- Salary (DOUBLE)

## Procedure

1. Load JDBC driver.
2. Establish a connection to the MySQL database.
3. Create an updatable ResultSet using Statement with TYPE\_SCROLL\_SENSITIVE and CONCUR\_UPDATABLE.
4. Update a record in the ResultSet using update methods and commit using updateRow().
5. Display the updated records.

# Java Full Stack Lab Manual - JDBC Experiment

## Java Program

```
import java.sql.*;

public class UpdatableResultSetDemo {
    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(
                ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE
            );

            ResultSet rs = stmt.executeQuery("SELECT * FROM Employee");

            while (rs.next()) {
                if (rs.getInt("EmpID") == 101) {
                    rs.updateDouble("Salary", 60000.0);
                    rs.updateRow();
                    System.out.println("Updated salary for EmpID 101.");
                }
            }

            rs.beforeFirst();
            System.out.println("--- Updated Employee Records ---");
            while (rs.next()) {
                System.out.println(rs.getInt("EmpID") + ", " +
                    rs.getString("Name") + ", " +
                    rs.getDouble("Salary"));
            }

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

## Java Full Stack Lab Manual - JDBC Experiment

### Expected Output

```
Updated salary for EmpID 101.  
  
--- Updated Employee Records ---  
101, Rahul, 60000.0  
102, Priya, 55000.0  
103, Ajay, 60000.0
```

### Result

Successfully demonstrated the use of Updatable ResultSet in JDBC to modify data directly within the ResultSet and commit changes using `updateRow()`.