**Jdbc:**

CREATE TABLE student (

id NUMBER,

name VARCHAR2(81),

dept VARCHAR2(50)

);

INSERT INTO student (id, name, dept) VALUES (1, 'sai', 'cse');

INSERT INTO student (id, name, dept) VALUES (2, 'prasad', 'it');

**PARAMETRISED:**

import java.sql.\*;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver loaded successfully");

            // Establish connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "saiprasad");

            System.out.println("Connection established");

            // SQL Query with placeholders

            String SQLQuery = "INSERT INTO Student VALUES (?, ?, ?)";

            // Create PreparedStatement

            PreparedStatement pst = conn.prepareStatement(SQLQuery);

            // Initialize Scanner for user input

            Scanner sc = new Scanner(System.in);

            // Take user inputs

            System.out.print("Enter ID: ");

            int id = sc.nextInt();

            pst.setInt(1, id);

            System.out.print("Enter Name: ");

            String name = sc.next();

            pst.setString(2, name);

            System.out.print("Enter Branch: ");

            String branch = sc.next();

            pst.setString(3, branch);

            // Execute Update

            pst.executeUpdate();

            System.out.println("Record inserted successfully");

            // Close resources

            pst.close();

            conn.close();

            System.out.println("Connection closed");

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**SELECT OPERATION:**

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        // Use try-with-resources to ensure resources are closed automatically

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver class loaded successfully");

            // Establish connection

            try (Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system",

                    "saiprasad");

                    Statement stmt = conn.createStatement()) {

                System.out.println("Connection established");

                // Execute query

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

                // Uncomment the following line if you intend to insert a new record

                // int count = stmt.executeUpdate("INSERT INTO student VALUES(5, 'Srikar',

                // 'cse')");

                // System.out.println(count + " row(s) inserted");

                // Process result set

                while (rs.next()) {

                    System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getString(3));

                }

                // Close result set

                rs.close();

                System.out.println("Connection closed");

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**Insert update delete:**

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver loaded successfully");

            // Establish connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "saiprasad");

            System.out.println("Connection established");

            // Create a statement

            Statement stmt = conn.createStatement();

            // Insert operation (Ensure column names and values match your table schema)

            int count = stmt.executeUpdate("INSERT INTO Student (id, name, dept) VALUES (10, 'Sai', 'CSE')");

            if (count == 1) {

                System.out.println("Record successfully inserted");

            } else {

                System.out.println("Some error occurred");

            }

            // Update operation

            int count2 = stmt.executeUpdate("UPDATE Student SET dept = 'ece' WHERE id = 1");

            if (count2 == 1) {

                System.out.println("Record updated successfully");

            } else {

                System.out.println("Record updated successfully");

            }

            // Delete operation

            int count3 = stmt.executeUpdate("DELETE FROM Student WHERE id = 105");

            if (count3 == 1) {

                System.out.println("Record successfully deleted");

            } else {

                System.out.println("Some error occurred");

            }

            // Close resources

            stmt.close();

            conn.close();

            System.out.println("Connection closed");

        } catch (SQLException e) {

            e.printStackTrace();

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        }

    }

}

**STORED PROCEDURES:**

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver loaded successfully");

            // Establish connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "saiprasad");

            System.out.println("Connection established");

            // Prepare the CallableStatement to call the stored procedure

            CallableStatement cst = conn.prepareCall("{call sum(?, ?, ?)}");

            // Set input parameters

            cst.setInt(1, 10);

            cst.setInt(2, 10);

            // Register the output parameter

            cst.registerOutParameter(3, Types.INTEGER);

            // Execute the stored procedure

            cst.execute();

            // Retrieve the output parameter

            int result = cst.getInt(3);

            System.out.println("Result: " + result);

            // Close resources

            cst.close();

            conn.close();

            System.out.println("Connection closed");

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

CREATE OR REPLACE PROCEDURE sum (

num1 IN NUMBER,

num2 IN NUMBER,

result OUT NUMBER

) AS

BEGIN

result := num1 + num2;

END;

/

**DATABASEMETADATA:**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver loaded successfully");

            // Establish connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "saiprasad");

            System.out.println("Connection established");

            // Get Database Metadata

            java.sql.DatabaseMetaData dbmd = conn.getMetaData();

            // Display database information

            System.out.println("Product Name: " + dbmd.getDatabaseProductName());

            System.out.println("Product Version: " + dbmd.getDatabaseProductVersion());

            System.out.println("Driver Version: " + dbmd.getDriverVersion());

            System.out.println("Driver Name: " + dbmd.getDriverName());

            System.out.println("Database Major Version: " + dbmd.getDatabaseMajorVersion());

            System.out.println("Database Minor Version: " + dbmd.getDatabaseMinorVersion());

            // Close connection

            conn.close();

            System.out.println("Connection closed");

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**RESULTSET METADATA:**

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Establish a connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "system", "saiprasad");

            // Create a statement

            Statement stmt = conn.createStatement();

            // Execute a query

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

            // Get metadata

            ResultSetMetaData rsmd = rs.getMetaData();

            // Get the column count

            int columnCount = rsmd.getColumnCount();

            // Loop through each column and display its metadata

            for (int i = 1; i <= columnCount; i++) {

                System.out.println("Column Number: " + i);

                System.out.println("Column Name: " + rsmd.getColumnName(i));

                System.out.println("Column Type: " + rsmd.getColumnTypeName(i));

                System.out.println("Column Display Size: " + rsmd.getColumnDisplaySize(i));

                System.out.println(

                        "Is Nullable: " + (rsmd.isNullable(i) == ResultSetMetaData.columnNullable ? "Yes" : "No"));

                System.out.println("Is Auto Increment: " + rsmd.isAutoIncrement(i));

                System.out.println();

            }

            // Close resources

            rs.close();

            stmt.close();

            conn.close();

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**SCROLLABLE RESULT SET:**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Main {

    public static void main(String[] args) {

        try {

            // Load Oracle JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Establish a connection to the database

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "System", "saiprasad");

            // Create a scrollable and read-only statement

            Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

            // Execute query

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

            // Display records in forward direction

            System.out.println("Records in forward direction:");

            System.out.println("Id\tName\tBranch");

            while (rs.next()) {

                System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));

            }

            // Display records in backward direction

            System.out.println("\nRecords in backward direction:");

            System.out.println("Id\tName\tBranch");

            while (rs.previous()) {

                System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));

            }

            // Display first record

            System.out.println("\nFirst record in the table:");

            rs.first();

            System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));

            // Display second record

            System.out.println("\nSecond record in the table:");

            rs.absolute(2);

            System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));

            // Display last record

            System.out.println("\nLast record in the table:");

            rs.last();

            System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));

            // Additional information

            System.out.println("\nIs first: " + rs.isFirst());

            System.out.println("Is last: " + rs.isLast());

            rs.next();

            System.out.println("Is after last: " + rs.isAfterLast());

            // Close resources

            rs.close();

            stmt.close();

            conn.close();

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**UPDATABLE RESULT SET:**

import java.sql.\*;

public class Main {

    public static void main(String[] args) {

        try {

            // Load MySQL JDBC Driver

            Class.forName("oracle.jdbc.driver.OracleDriver");

            System.out.println("Driver loaded successfully.");

            // Establish the connection

            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "saiprasad");

            System.out.println("Connection established.");

            // Create a scrollable and updatable ResultSet

            Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_UPDATABLE);

            ResultSet rs = stmt.executeQuery("SELECT id, name FROM student");

            // Display original ResultSet

            System.out.println("Original ResultSet:");

            while (rs.next()) {

                int id = rs.getInt("id");

                String name = rs.getString("name");

                System.out.println("ID: " + id + ", Name: " + name);

            }

            // Update the first record

            rs.beforeFirst(); // Reset cursor before the first row

            if (rs.next()) {

                rs.updateString("name", "UpdatedName");

                rs.updateRow();

            }

            // Insert a new record

            rs.moveToInsertRow();

            rs.updateInt("id", 101); // Ensure '101' is unique

            rs.updateString("name", "NewEmployee");

            rs.insertRow();

            // Display the updated ResultSet

            System.out.println("\nUpdated ResultSet:");

            rs.beforeFirst();

            while (rs.next()) {

                int id = rs.getInt("id");

                String name = rs.getString("name");

                System.out.println("ID: " + id + ", Name: " + name);

            }

            // Close resources

            rs.close();

            stmt.close();

            conn.close();

            System.out.println("Connection closed.");

        } catch (SQLException e) {

            e.printStackTrace();

        } catch (ClassNotFoundException e) {

            System.err.println("MySQL JDBC Driver not found.");

            e.printStackTrace();

        }

    }

}