Madan Bhandari Memorial College
Department of Computer Science and Information Technology (B.Sc.CSIT)
Ninayak Nagar, New Baneshwor, Kathmandu

Practical Sheet

Submitted By:- Utsav Acharya Sharma
Submitted To Debesh Adhikari
Submission Date:- 2080/08/15

Program No:- 06
Lab Date:- 2080/08/12
T.U.Roll.No. :- 24179

Title: Java Database Connectivity.

Introduction

Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC) is a Java-based API that provides a standard interface for interacting with relational database. JDBC enables Java applications to connect to and interact with databases, execute SQ queries and retrieve or update data.

JDBC Methods:

i) DriverManager.getConnection(url, dusername, dpassword)
→ Establishes a connection to the database.

ii) Connection.prepareStatement(insertQuery)
→ Executing parameterized SQL queries.

iii) stmt.executeQuery("select id, name, email, country, password from users")
→ Executes a query that retrieves data and return a ResultSet.

iv) rs.get[~~variable~~ DataType]("variable_name");

Steps:

Step 1: Establishing a Connection → Connection with the database.
Step 2: Create a statement using connection object.
Step 3: Execute the query or update query
Step 4: Process the ResultSet object.

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

puleli class SelectStatementExample{
    pulelic static void main (String[] args){
        System.out.println("Inside the main function");
        String url = "jdbc:mysql://localhost:3306/javaprac";
        String username = "root";
        String dpassword = "";
        Connection connection = null;
        Statement stmt = null;
        ResultSet rs = null;
        try{
            connection = DriverManager.getConnection(url,username,
            stmt = connection.createStatement();  dpassword);
            insertData(connection);
            deleteData(connection,4);
            rs = stmt.executeQuery("selet id,name,email,country
                                    password from users");
            while(rs.next()){
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String email = rs.getString("email");
                String country = rs.getString("country");
                String password = rs.getString("password");
                System.out.println(id+","+name+","+email+
                                    ","+country+","+password);
            }
        }
```

```
catch (SQLException e){
    System.out.println("An error occurred while connecting
                        MySQL database");
    e.printStackTrace();

}
finally {
  try {
      if(rs!=null){
          rs.close();
      }
      if(stmt!=null){
          stmt.close();
      }
      if (connection!=null){
          connection.close();
      }
  } catch (SQLException e){
      e.printStackTrace();
  }
  System.out.println("~~Exit~~ Finally Done, Pheo!");

}
}
public static void insertData(Connection connection){
    String insertQuery = "INSERT INTO users(name,email,country,
                                password)VALUES
                                (?,?,?,?)";
    PreparedStatement pstmt=null;

    try{
        pstmt= connection.prepareStatement(insertQuery);
        pstmt.setString(1,"Ravi");
        pstmt.setString(2,"ravi@gmail.com");
        pstmt.setString(3,"Nepal");
        pstmt.setString(4,"ravi123");
```

```
        int rowsInserted = pstmt.executeUpdate();
        if (rowsInserted > 0) {
            System.out.println("Data inserted successfully!");
        }
    } catch (SQLException e) {
        System.out.println("An error occured while inserting data
                            into the database");
        e.printStackTrace();
    } finally {
        try {
            if (pstmt != null) {
                pstmt.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
public static void deleteData(Connection connection, int userId) {
    String deleteQuery = "DELETE FROM users WHERE id=?";
    PreparedStatement pstmt = null;
    try {
        pstmt = connection.prepareStatement(deleteQuery);
        pstmt.setInt(1, userId);
        int rowsDeleted = pstmt.executeUpdate();
        if (rowsDeleted > 0) {
            System.out.println("Data deleted successfully");
        } else {
            System.out.println("No data found with
                                the specified ID.");
        }
    } catch (SQLException e) {
        System.out.println("An error occured while
                            deleting data from
                            database");
        e.printStackTrace();
    }
```

```
        finally {
            try {
                if (pstmt != null) {
                    pstmt.close();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
<terminated> SelectStatementExample [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 30, 2023, 7:59:10 AM – 7:59:13 AM) [pid: 1232]
Inside the Main function
Data inserted successfully!
Data deleted successfully!
1,Utsav,utsav@gmail.com,Nepal,utsav123
2,Ram Bahadur,ram@gmail.com,Nepal,2468010
3,Shyam,shyam@gmail.com,Nepal,123456789
5,Hari,hari@gmail.com,Nepal,12345678
6,Sita,sita@gmail.com,Nepal,sita123
7,John,john@gmail.com,USA,john123
8,Ravi,ravi@gmail.com,Nepal,ravi123
Finally Done, Phew!
```