

Laboratory 5

Title of the Laboratory Exercise: Java database programming

1. Introduction and Purpose of Experiment

The SQL includes commands to define view on the data. A view contains rows and columns, just like a real table. Java uses JDBC (Java Database Connectivity) to connect to databases. JDBC allows to connect to a wide-range of databases such as Oracle, MySQL, etc. By doing this lab, students will be able to implement views in SQL and connect the developed database with the application.

2. Aim and Objectives

Aim

- To design and implement views on the data using SQL commands
- To connect to the relational database in Java

Objectives

At the end of this lab, the student will be able to

- Design and execute views using SQL commands
- Perform database programming in Java

3. Experimental Procedure

- Analyse the problem statement
- Execute the built-in functions in SQL
- Design and execute the view statements in SQL
- Test the executed commands
- Document the Results
- Analyse and discuss the outcomes of your experiment

4. Questions

- Write a Java program to do the following operations
 - Insert the details of the Managers into the table
 - Display all the details of the Managers in the ascending order of their names
 - Count the number of Managers staying in each location and display the address and the total number
 - Display the number of Managers in each location. Only include locations with more than 3 Manager ssle

5. Presentation of Results

Java Program

Main

```

You, 17 hours ago | 1 author (You)
1  import java.math.BigInteger;
2  import java.sql.*;
3  import java.util.Scanner;
4  import java.sql.Connection;
5  import java.sql.DriverManager;
You, 17 hours ago | 1 author (You)
6  public class Lab5b {
    Run | Debug
7      public static void main(String[] args) throws Exception {
8          Scanner input = new Scanner(System.in);    Resource leak: 'input' is never closed
9          Class.forName("com.mysql.cj.jdbc.Driver");
10         Connection con=DriverManager.getConnection(
11             "jdbc:mysql://localhost:3306/lab5b","root","Sri123");
12         Statement stmt=con.createStatement();
13         while(true) {
14             System.out.println("");
15             System.out.println("***** K Srikanth *****");
16             System.out.println("");
17             System.out.println("Press 1 to Enter the Details to Database");
18             System.out.println("Press 2 to Display Detials Of Manager");
19             System.out.println("Press 3 to Count the No of Managers in each Location");
20             System.out.println("Press 4 to Display Each Manager Location");
21             int choice = input.nextInt();

```

Figure 1 Java Program with JDBC Connection and MENU

Choice 1

```

24         case 1:
25             System.out.println("");
26             System.out.println("Enter the Manager Name : ");
27             String Name = input.next();
28             System.out.println("Enter the Location : ");
29             String Location = input.next();
30             System.out.println("Enter the Phone Number : ");
31             BigInteger PhoneNumber= input.nextBigInteger();
32             stmt.executeUpdate("insert into manager(Manager_Name,Location,Phone_Number) values('"+Name+"','"+Location+"','"+PhoneNumber+"')");
33             break;

```

Figure 2 Java Program to take user input for database using JDBC

Choice 2

```

35         case 2:
36             System.out.println("");
37             try{
38                 ResultSet rs=stmt.executeQuery("select * from manager order by Manager_Name ASC");
39                 while(rs.next())
40                     System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getString(4));
41             }catch(Exception e){
42                 System.out.println(e);}
43             break;

```

Figure 3 Java Program to display all the Manager Name in Ascending order using JDBC

Choice 3

```

45     case 3:
46         System.out.println("");
47         try{
48             ResultSet rs=stmt.executeQuery("select location,count(*) from manager group by location;");
49             while(rs.next())
50                 System.out.println(rs.getString(1)+" "+rs.getString(2));
51         }catch(Exception e){
52             System.out.println(e);}
53         break;

```

Figure 4 Java Program to count the number of managers in each location using JDBC

Choice 4

```

55     case 4:
56         System.out.println("");
57         try{
58             ResultSet rs=stmt.executeQuery("select location,count(*) from manager group by location having count(*)> 3");
59             while(rs.next())
60                 System.out.println(rs.getString(1)+" "+rs.getString(2));
61         }catch(Exception e){
62             System.out.println(e);}
63         break;
64
65     default: System.out.println("Invalid Input Please Try Again");
66             break;
67     }
68 }
69

```

Figure 5 Java Program to count the number of managers in each location greater than 3 using JDBC

Result / Output**Menu**

```

***** K Srikanth *****
Press 1 to Enter the Details to Database
Press 2 to Display Detials Of Manager
Press 3 to Count the No of Managers in each Location
Press 4 to Display Each Manager Location

```

Figure 6 Java Program output switch case menu choice

Choice 1

```

***** K Srikanth *****
Press 1 to Enter the Details to Database
Press 2 to Display Detials Of Manager
Press 3 to Count the No of Managers in each Location
Press 4 to Display Each Manager Location
1

Enter the Manager Name :
Deekshitha
Enter the Location :
Bangalore
Enter the Phone Number :
2345678910

```

Figure 7 Java Program output for choice 1

Choice 2

```

***** K Srikanth *****
Press 1 to Enter the Details to Database
Press 2 to Display Detials Of Manager
Press 3 to Count the No of Managers in each Location
Press 4 to Display Each Manager Location
2
5 Akansha Sydney 2468013579
6 Deekshitha Bangalore 2345678910
2 Naveen Bangalore 7019462108
1 Srikanth Bangalore 9493364308
3 Supraja Bangalore 1234567890
4 Sushanth Sydney 987654321

```

Figure 8 Java Program output for choice 2

Choice 3

```

***** K Srikanth *****
Press 1 to Enter the Details to Database
Press 2 to Display Detials Of Manager
Press 3 to Count the No of Managers in each Location
Press 4 to Display Each Manager Location
3
Bangalore 4
Sydney 2

```

Figure 9 Java Program output for choice 3

Choice 4

```

***** K Srikanth *****
Press 1 to Enter the Details to Database
Press 2 to Display Detials Of Manager
Press 3 to Count the No of Managers in each Location
Press 4 to Display Each Manager Location
4
Bangalore 4

```

Figure 10 Java Program output for choice 4

6. Analysis and Discussions

JDBC offers a programming-level interface that handles the mechanics of Java applications communicating with a database or RDBMS. The JDBC API supports communication between the Java application and the JDBC manager. Import MySQL Library using `import java.sql.*;` Under Main Function for connecting to MySQL database

```

Class.forName("com.mysql.cj.jdbc.Driver");
Connection con=DriverManager.getConnection(
    "jdbc:mysql://localhost:3306/databaseName","UserName","Password");

```

Now that our database is connected with our application to perform database queries we create a statement object with the help of connection object to write queries using statement object

```

Statement stmt=con.createStatement();

```

Using stmt we can now start writing queries using executeupdate

```
stmt.executeUpdate("SQL Qurery HERE ")
```

Using a while loop we can print the result for the queries.

7. Conclusions

JDBC driver is used to make SQL queries. JDBC provides an interface to make SQL queries in java program.

8. Comments

1. Learning happened

Learned to connect MySQL database with java project using JDBC