

Laboratory 5.b

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

- i. Analyse the problem statement
- ii. Execute the built-in functions in SQL
- iii. Design and execute the view statements in SQL
- iv. Test the executed commands
- v. Document the Results
- vi. Analyse and discuss the outcomes of your experiment

4. Questions

- b. Write a Java program to do the following operations
 - i. Insert the details of the Managers into the table
 - ii. Display all the details of the Managers in the ascending order of their names

- iii. Count the number of Managers staying in each location and display the address and the total number
- iv. Display the number of Managers in each location. Only include locations with more than or equal to 2 Managers

5. Presentation of Results

```

1 DROP TABLE IF EXISTS MANAGER;
2
3 -- Create the table
4 CREATE TABLE MANAGER (Name VARCHAR(50), Id INT PRIMARY KEY, Department CHAR(20), Address VARCHAR(255), Salary FLOAT);
5
6 -- Insert Values
7 INSERT INTO MANAGER VALUES
8 ('Deepak R', 12, 'CSE', 'BEL', 125000),
9 ('Virat', 1, 'CSE', 'JP NAGAR', 12000),
10 ('Dhoni', 3, 'ASE', 'RR NAGAR', 15000),
11 ('Raina', 4, 'ISE', 'MN NAGAR', 16000),
12 ('Warner', 2, 'ASE', 'J.CROSS', 15000);

```

Figure 0-1 Created Table MANAGER

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package javaapplication20;
7
8   import java.sql.*;
9
10  /**
11   *
12   * @author Deepak R
13   *
14   * Table Details : MANAGER (Name VARCHAR(50), Id INT PRIMARY KEY, Department CHAR(20), Address VARCHAR(255), Salary FLOAT);
15   */
16  public class Lab05 {
17
18      public static void addData(Statement stmt) throws SQLException {
19          stmt.executeUpdate("INSERT INTO MANAGER VALUE ('Ravan', 9, 'ISE', 'MALLESHWARAM', 15000)");
20          stmt.executeUpdate("INSERT INTO MANAGER VALUE ('Maxwell', 10, 'ECE', 'RAJAJINAGAR', 20000)");
21          stmt.executeUpdate("INSERT INTO MANAGER VALUE ('Smith', 11, 'EEE', 'BEL', 12000)");
22      }
23
24      public static void viewAllData(Statement stmt) throws SQLException {
25          ResultSet rs = stmt.executeQuery("SELECT * FROM MANAGER");
26
27          System.out.printf("%20s | %3s | %5s | %10s | %9s\n", "Name", "Id", "Dept", "Address", "Salary\n");
28          // iterate through the result
29          while (rs.next()) {
30              String name = rs.getString("Name");
31              Integer id = rs.getInt("Id");
32              String dept = rs.getString("Department");
33              String address = rs.getString("Address");
34              Double salary = rs.getDouble("Salary");
35
36              System.out.printf("%20s | %3d | %5s | %10s | %8.2f\n", name, id, dept, address, salary);
37          }
38      }
39
40      public static void viewDeptCount(Statement stmt) throws SQLException {
41          ResultSet rs = stmt.executeQuery("SELECT Department, COUNT(Department) FROM MANAGER GROUP BY Department");
42
43          System.out.printf("%10s | %10s", "Dept", "Count\n\n");
44          while (rs.next()) {

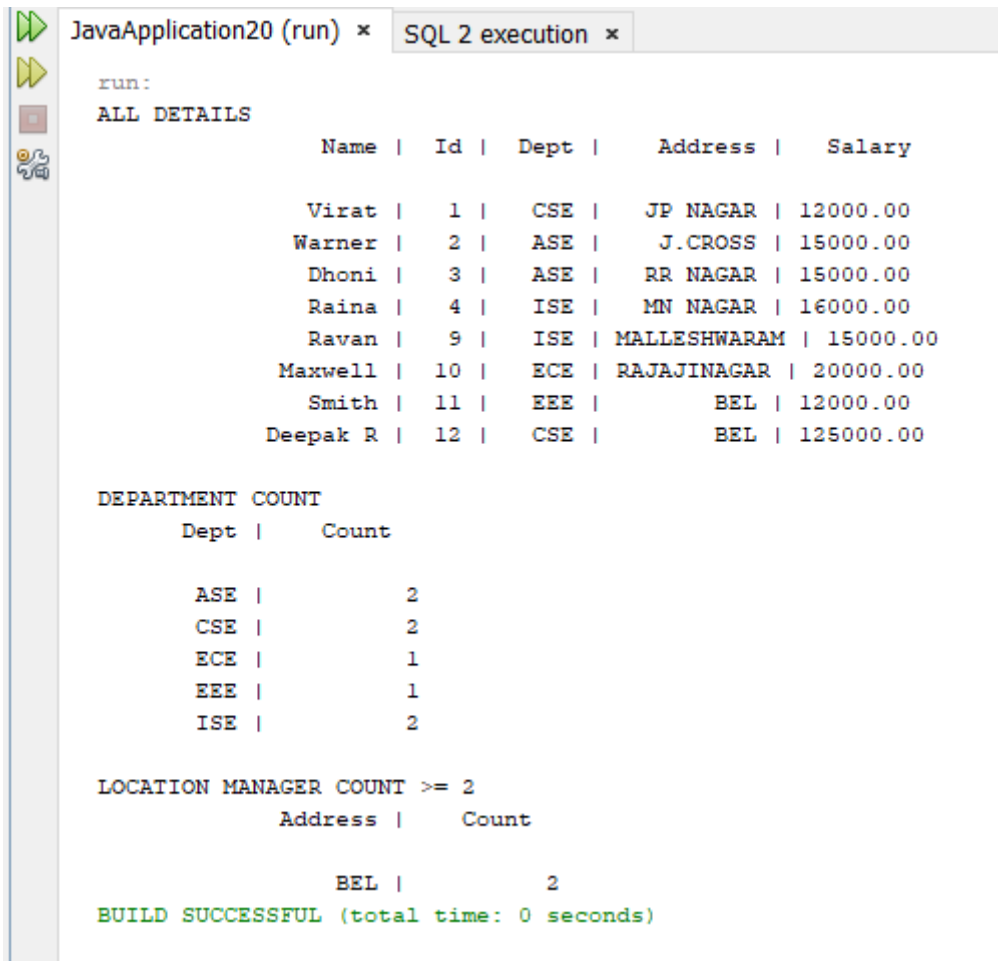
```

```

45     String dept = rs.getString("Department");
46     Integer count = rs.getObject(2, Integer.class);
47
48     System.out.printf("%10s | %10d\n", dept, count);
49 }
50
51 }
52
53 public static void findManagers(Statement stmt) throws SQLException {
54     ResultSet rs = stmt.executeQuery("SELECT Address, COUNT(Address) FROM MANAGER GROUP BY Address HAVING COUNT(Address) >= 2");
55
56     System.out.printf("%20s | %10s", "Address", "Count\n\n");
57     while (rs.next()) {
58         String address = rs.getString("Address");
59         Integer count = rs.getObject(2, Integer.class);
60
61         System.out.printf("%20s | %10d\n", address, count);
62     }
63 }
64
65 /**
66  * @param args the command line arguments
67  */
68 public static void main(String[] args) {
69     try {
70         // load the registered driver
71         Class.forName("com.mysql.jdbc.Driver");
72
73         // create a connection object
74         Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/manager_view", "root", "root");
75
76         // create a statement object
77         Statement st = conn.createStatement();
78
79         // add data to the table
80         addData(st);
81
82         System.out.println("ALL DETAILS");
83         // display all the details of the managers
84         viewAllData(st);
85
86         System.out.println();
87
88         System.out.println("DEPARTMENT COUNT");
89         // view the department count
90         viewDeptCount(st);
91
92         System.out.println();
93
94         System.out.println("LOCATION MANAGER COUNT >= 2");
95         // view the locations with managers count >=2
96         findManagers(st);
97
98     } catch (ClassNotFoundException | SQLException e) {
99         System.out.println("Exception Occured : " + e.getMessage());
100     }
101 }
102
103 }

```

Figure 0-2 Java Program that 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 or equal to 2 Managers



```

run:
ALL DETAILS

      Name | Id | Dept | Address | Salary
-----
      Virat | 1 | CSE | JP NAGAR | 12000.00
      Warner | 2 | ASE | J.CROSS | 15000.00
      Dhoni | 3 | ASE | RR NAGAR | 15000.00
      Raina | 4 | ISE | MN NAGAR | 16000.00
      Ravan | 9 | ISE | MALLESHWARAM | 15000.00
      Maxwell | 10 | ECE | RAJAJINAGAR | 20000.00
      Smith | 11 | EEE | BEL | 12000.00
      Deepak R | 12 | CSE | BEL | 125000.00

DEPARTMENT COUNT
      Dept | Count
-----
      ASE | 2
      CSE | 2
      ECE | 1
      EEE | 1
      ISE | 2

LOCATION MANAGER COUNT >= 2
      Address | Count
-----
      BEL | 2

BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 0-3 Perform SQL Operations using JAVA

6. Analysis and Discussions

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access any kind of tabular data, especially relational database.

7. Conclusions

Java code was executed successfully and appropriate results were obtained.

Database views are created using the CREATE VIEW statement. Views can be created from a single table, multiple tables or another view.

8. Comments

1. Limitations of Experiments

Views cannot be created on Temporary Tables

You cannot associate rules and defaults with views

You cannot pass parameters to SQL Server views

2. Limitations of Results

Correct drivers need to be deployed for each type of database.

Cannot update or insert multiple tables with sequence. (Sequence is always random)

3. Learning happened

Learnt how to write SQL commands in java application and execute them.