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

   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

   a. Create a table MANGER with attributes such as Name, Id, Department, Address, and Salary. Write SQL statements for the following expressions.

      i. Create a view 'MANAGER_VIEW' to display the details such as name and department of each manager
      ii. Display the name of the manager from MANAGER_VIEW whose department is 'CSE'
      iii. Drop the views generated

b. Write a Java program to do the following operations

   i. Insert the details of the Managers into the table

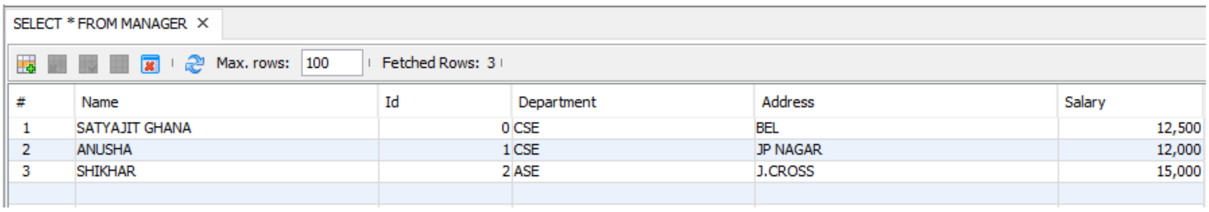
   ii. Display all the details of the Managers in the ascending order of their names

   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

```sql
USE students;

-- Drop the table if it exists
DROP TABLE IF EXISTS MANAGER;

-- Create the table
CREATE TABLE MANAGER (Name VARCHAR(50), Id INT PRIMARY KEY, Department CHAR(20), Address VARCHAR(255), Salary FLOAT);

-- Insert Values
INSERT INTO MANAGER VALUES
('SATYAJIT GHANA', 0, 'CSE', 'BEL', 12500),
('ANUSHA', 1, 'CSE', 'JP NAGAR', 12000),
('SHIKHAR', 2, 'ASE', 'J.CROSS', 15000);

-- Show the table
SELECT * FROM MANAGER;
```



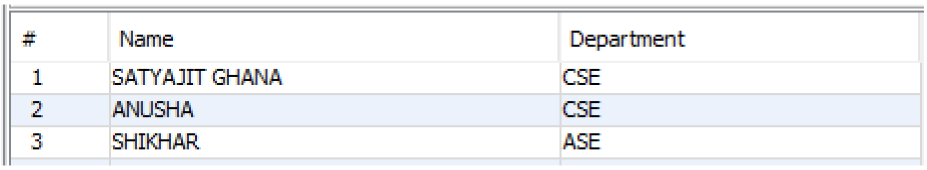
Max. rows: 100 Fetched Rows: 3

#	Name	Id	Department	Address	Salary
1	SATYAJIT GHANA	0	CSE	BEL	12,500
2	ANUSHA	1	CSE	JP NAGAR	12,000
3	SHIKHAR	2	ASE	J.CROSS	15,000

Figure 0-1 Created Table MANAGER

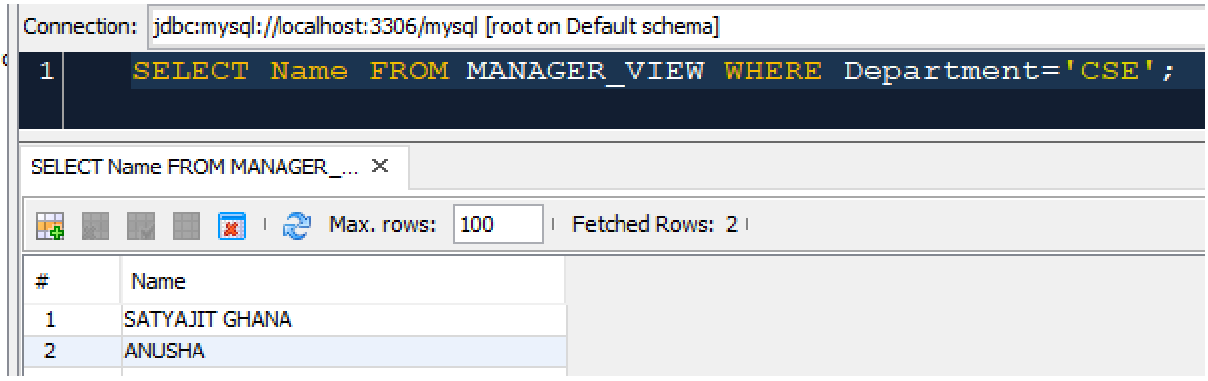
```sql
USE students;

DROP VIEW IF EXISTS MANAGER_VIEW;

-- Create the View
CREATE VIEW MANAGER_VIEW AS SELECT Name, Department FROM MANAGER;
```

```
-- Show the View
SELECT * FROM MANAGER_VIEW;
```



#	Name	Department
1	SATYAJIT GHANA	CSE
2	ANUSHA	CSE
3	SHIKHAR	ASE

Figure 0-2 Created View MANAGER_VIEW



Connection: jdbc:mysql://localhost:3306/mysql [root on Default schema]

```
SELECT Name FROM MANAGER_VIEW WHERE Department='CSE';
```

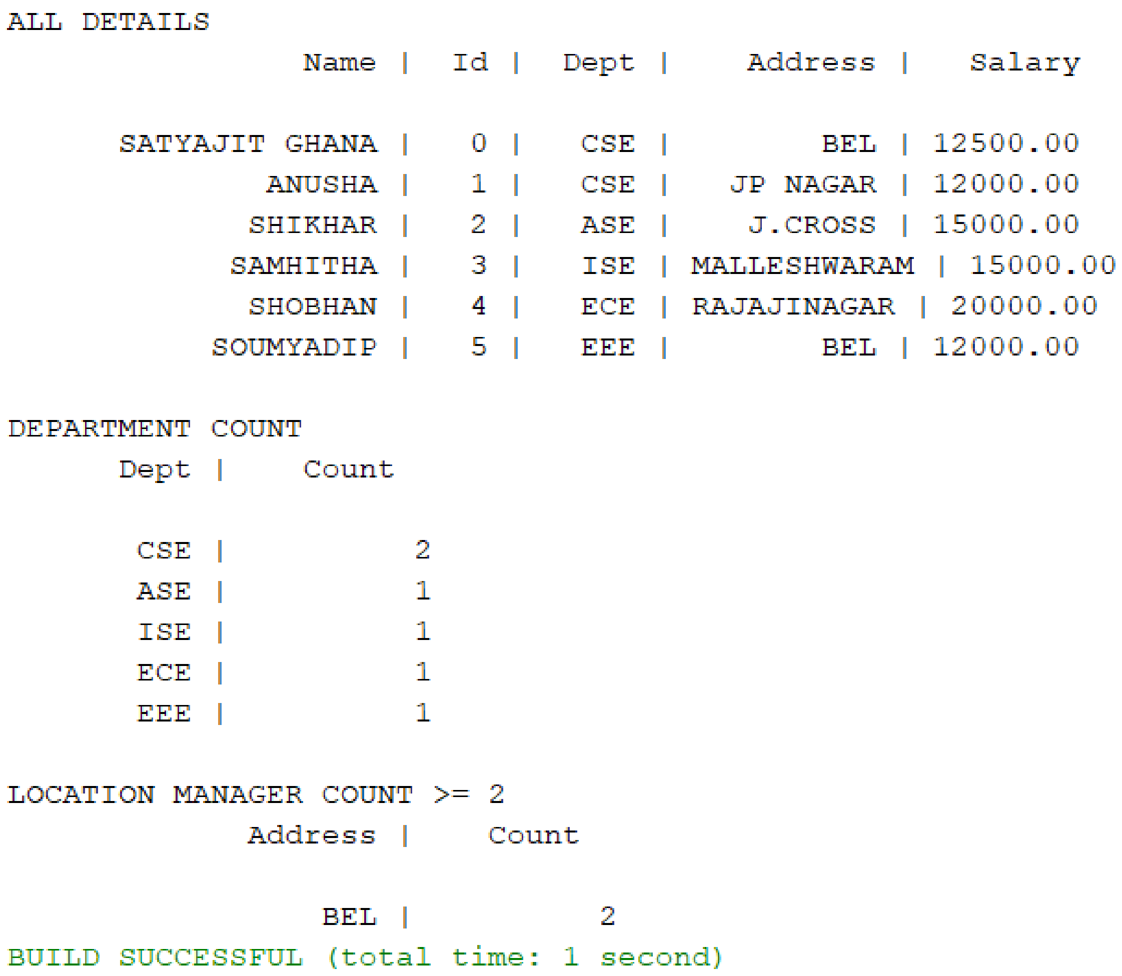
#	Name
1	SATYAJIT GHANA
2	ANUSHA

Figure 0-3 SELECT from MANAGER_VIEW

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab05;

import java.sql.*;

/**
 *
 * @author shadowleaf
 *
 * Table Details : MANAGER (Name VARCHAR(50), Id INT PRIMARY KEY, Department CHAR(20)
, Address VARCHAR(255), Salary FLOAT);
 */
public class Lab05 {

    public static void addData(Statement stmt) throws SQLException {
            stmt.executeUpdate("INSERT INTO MANAGER VALUE ('SAMHITHA', 3, 'ISE', 'MAL
LESHWARAM', 15000)");
            stmt.executeUpdate("INSERT INTO MANAGER VALUE ('SHOBHAN', 4, 'ECE', 'RAJA
JINAGAR', 20000)");
```

```
            stmt.executeUpdate("INSERT INTO MANAGER VALUE ('SOUMYADIP', 5, 'EEE', 'BE
L', 12000)");
    }

    public static void viewAllData(Statement stmt) throws SQLException {
        ResultSet rs = stmt.executeQuery("SELECT * FROM MANAGER");

        System.out.printf("%20s | %3s | %5s | %10s | %9s\n", "Name", "Id", "Dept", "A
ddress", "Salary\n");
        // iterate through the result
        while (rs.next()) {
            String name = rs.getString("Name");
            Integer id = rs.getInt("Id");
            String dept = rs.getString("Department");
            String address = rs.getString("Address");
            Double salary = rs.getDouble("Salary");

            System.out.printf("%20s | %3d | %5s | %10s | %8.2f\n", name, id, dept, ad
dress, salary);
        }
    }

    public static void viewDeptCount(Statement stmt) throws SQLException {
        ResultSet rs = stmt.executeQuery("SELECT Department, COUNT(Department) FROM M
ANAGER GROUP BY Department");

        System.out.printf("%10s | %10s", "Dept", "Count\n\n");
        while (rs.next()) {
            String dept = rs.getString("Department");
            Integer count = rs.getObject(2, Integer.class);

            System.out.printf("%10s | %10d\n", dept, count);
        }
    }

    public static void findManagers(Statement stmt) throws SQLException {
        ResultSet rs = stmt.executeQuery("SELECT Address, COUNT(Address) FROM MANAGER
 GROUP BY Address HAVING COUNT(Address) >= 2");

        System.out.printf("%20s | %10s", "Address", "Count\n\n");
        while (rs.next()) {
            String address = rs.getString("Address");
            Integer count = rs.getObject(2, Integer.class);

            System.out.printf("%20s | %10d\n", address, count);
        }
    }

    /**
```

```
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        try {
            // load the registered driver
            Class.forName("com.mysql.jdbc.Driver");

            // create a connection object
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/stu
dents", "shadowleaf", "redosmiumtetroxide");

            // create a statement object
            Statement st = conn.createStatement();

            // add data to the table
            addData(st);

            System.out.println("ALL DETAILS");
            // display all the details of the managers
            viewAllData(st);

            System.out.println();

            System.out.println("DEPARTMENT COUNT");
            // view the department count
            viewDeptCount(st);

            System.out.println();

            System.out.println("LOCATION MANAGER COUNT >= 2");
            // view the locations with managers count >=2
            findManagers(st);


        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Exception Occured : " + e.getMessage());
        }
    }

}
```



```
ALL DETAILS
                Name |  Id |  Dept |     Address |    Salary

      SATYAJIT GHANA |   0 |   CSE |         BEL | 12500.00
              ANUSHA |   1 |   CSE |    JP NAGAR | 12000.00
             SHIKHAR |   2 |   ASE |     J.CROSS | 15000.00
            SAMHITHA |   3 |   ISE | MALLESHWARAM | 15000.00
             SHOBHAN |   4 |   ECE | RAJAJINAGAR | 20000.00
           SOUMYADIP |   5 |   EEE |         BEL | 12000.00

DEPARTMENT COUNT
      Dept |     Count

       CSE |          2
       ASE |          1
       ISE |          1
       ECE |          1
       EEE |          1

LOCATION MANAGER COUNT >= 2
             Address |     Count

                 BEL |          2
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 0-4 Perform SQL Operations using JAVA

6. Analysis and Discussions

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition.

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 create and drop a view.

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