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 ‘Information Technology’
      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 3 Managers

5. Presentation of Results

```
insert into manager values('AKASH','10101','IT','Rajasthan',78000.0);
insert into manager values('HARSHIT','10102','IT','Rajasthan',78000.0);
insert into manager values('ACHINTYA','10103','ECE','Rajasthan',68000.0);
insert into manager values('KARTIK','10104','EEE','PUNJAB',58000.0);
insert into manager values('AISHWARYA','10105','IT','Rajasthan',68000.0);
select * from manager;
```

select * from manager

Max. rows: 100 Fetched Rows: 5 Matching Rows:

#	Name	ID	Depatment	Address	Salary
1	AKASH	10101	IT	Rajasthan	78000.0
2	HARSHIT	10102	IT	Rajasthan	78000.0
3	ACHINTYA	10103	ECE	Rajasthan	68000.0
4	KARTIK	10104	EEE	PUNJAB	58000.0
5	AISHWARYA	10105	IT	Rajasthan	68000.0

Figure 1 Initial table

```
create view manager_view as select name,depatment from manager;
select name from manager_view where depatment='IT';
```

select name from manager_...

Max. rows: 100 Fetched Rows: 3 Matchi

#	name
1	AKASH
2	HARSHIT
3	AISHWARYA

Figure 2 View Creation and Selection of Name attribute from View

```
drop view manager_view;
```

Figure 3 Dropping of View

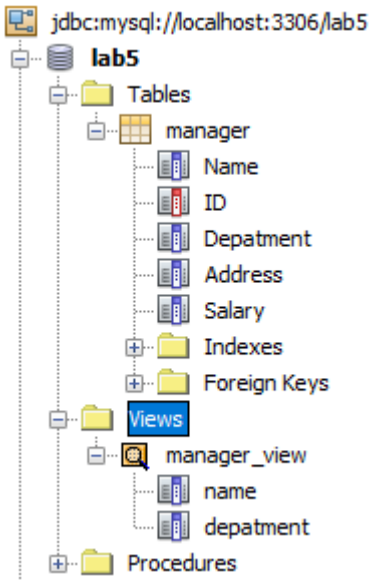

Figure 4 Database with Created View

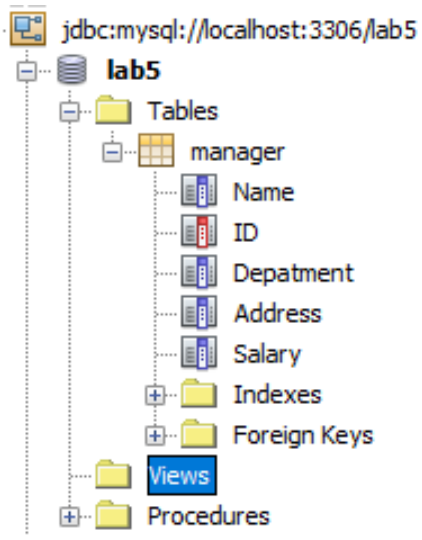

Figure 5 Database after View Dropped

```java
package lab5java;
import java.sql.*;
public class Lab5java {
    public static void main(String[] args) {
        try{
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lab5", "root", "ruas");
            Statement st = con.createStatement();
            // b.1 INSERTION OF DETAILS OF A MANAGER
            st.executeUpdate("insert into manager values('RASHMI','10106','IT','Rajasthan',78000.0)");
            System.out.println("Inserted details of Manager 'RASHMI'\n");

            // b.2 DETAILS OF MANAGERS IN ASCENDING ORDER
            ResultSet rs = st.executeQuery("select * from manager order by name ASC");
            System.out.println("Details of Managers in Ascending order");
            while (rs.next()){
                String name=rs.getString(1);
                if (name.length()>7)
                    System.out.println(name+"\t"+rs.getString(2)+"\t"
                            +rs.getString(3)+"\t"+rs.getString(4)+"\t"+rs.getFloat(5));
                else
                    System.out.println(name+"\t\t"+rs.getString(2)+"\t"
                            +rs.getString(3)+"\t"+rs.getString(4)+"\t"+rs.getFloat(5));
            }
            // b.3 COUNT OF MANAGERS STAYING IN SAME ADDRESS
            ResultSet rs1=st.executeQuery("select address,count(*) from manager group by address");
            System.out.println("\nCount of Managers staying in same Address");
            while(rs1.next())
                System.out.println(rs1.getString(1)+"\t"+rs1.getString(2));

            // b.4 NUMBER OF MANAGERS IN SAME ADDRESS WHERE COUNT > 3
            ResultSet rs2=st.executeQuery("Select address,count(*) from manager group by address having count(*)>3");
            System.out.println("\nManagers staying in same Address and Count>3");
            while(rs2.next())
                System.out.println(rs2.getString(1)+"\t"+rs2.getString(2));
            con.close();
        }
        catch(Exception ex){
            System.out.println(ex);
        }}}
```

```
Inserted details of Manager 'RASHMI'

Details of Managers in Ascending order
ACHINTYA        10103   ECE     Rajasthan       68000.0
AISHWARYA       10105   IT      Rajasthan       68000.0
AKASH           10101   IT      Rajasthan       78000.0
HARSHIT         10102   IT      Rajasthan       78000.0
KARTIK          10104   EEE     PUNJAB          58000.0
RASHMI          10106   IT      Rajasthan       78000.0

Count of Managers staying in same Address
PUNJAB  1
Rajasthan       5

Managers staying in same Address and Count>3
Rajasthan       5
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. Analysis and Discussions

In the lab views are designed and executed. A manager table is created with attributes such as ID, name, city, department and salary. The manager view is to be created to display name and department of each manager and display the names of manager from the IT department and the views are dropped.

Next, using the java programming the details of manager are to be entered into the table and the details are displayed in ascending order of their names. The names of the managers are to be counted along their location and display address with total number and the number of managers in each location.

JDBC Java Database Connectivity is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity.

7. Conclusions

In the lab, implementation and design of views are done on the data using the SQL commands and using the java programming in the NetBeans IDE relational database I connected to java. 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.