

Laboratory 6

Title of the Laboratory Exercise: interface to the system

1. Introduction and Purpose of Experiment

A database connection is the means by which a database server and its client software communicate with each other. The client and the server can be on the same machine or on different machines. The client uses a database connection to send commands to and receive replies from the server. A database is stored as a file or a set of files on magnetic disk or tape, optical disk, or some other secondary storage device. By doing this lab, students will be able to connect the developed application with the database.

2. Aim and Objectives

Aim

- To design an interface and connect to the database

Objectives

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

- Design and implement an interface for the application
- Connect the developed application with the database

3. Experimental Procedure

- i. Analyse the problem statement
- ii. Design an interface for the given problem statement
- iii. Connect the application with the database
- iv. Test the implemented program
- v. Document the Results
- vi. Analyse and discuss the outcomes of your experiment

4. Questions

- a. Consider the problem statement that you selected in Laboratory 2. Design a GUI with provision for insertion, deletion and display of a particular record in the database. Use appropriate components to display the page.

5. Calculations/Computations/Algorithms

Step1: First, create a database according to the requirements.

Step2: In java, java swing program is created for GUI.

Step3: For Insertion:

1. First Establish connection with database
2. Retrieve data from the text field
3. Using the insert command, insert data in database
4. Close the connection.

Step4: For Deletion

1. First establish connection with database.
2. Retrieve data from the text field and since the primary key is the ID so the user will be able to delete using the primary key.
3. Using the delete command, delete the row from the table in database.
4. Close the connection.

Step5: For Display

1. First establish connection with database
2. Using (select * from employee), retrieve data from database and store it in resultSet.
3. From the resultSet the data is displayed in the text area
4. Close the connection.

6. Presentation of Results

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con= DriverManager.getConnection ("jdbc:mysql://localhost:3306/lab6","root","ruas");
        Statement st=con.createStatement();
        String name= jTextField1.getText();
        String companyId= jTextField2.getText();
        String password= jTextField3.getText();
        String departmentId= jTextField4.getText();
        String query="insert into employee values('"+name+"','"+companyId+"','"+password+"','"+departmentId+"')";
        st.executeUpdate(query);
        con.close();
    }
    catch(ClassNotFoundException ex){
        System.out.println(ex);
    }
    catch(SQLException ce){
        System.out.println(ce);
    }
}
```

Figure 1 Insert into table

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con= DriverManager.getConnection ("jdbc:mysql://localhost:3306/lab6","root","ruas");
        Statement st=con.createStatement();
        String companyId= jTextField5.getText();
        String query="delete from employee where companyId='"+companyId+"'";
        st.executeUpdate(query);
        con.close();
    }
    catch(ClassNotFoundException ex){
        System.out.println(ex);
    }
    catch(SQLException ce){
        System.out.println(ce);
    }
}
```

Figure 2 Delete from table

```

private void DisplayActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con= DriverManager.getConnection ("jdbc:mysql://localhost:3306/lab6","root","ruas");
        Statement st=con.createStatement();
        String query="select * from employee";
        ResultSet rs =st.executeQuery(query);
        String display="Name\tCompanyId\tPassword\tDepartmentId\n\n";
        while(rs.next()){
            display=display+rs.getString(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.getString(4)+"\n";
        }
        JTextArea1.setText(display);
        con.close();
    }
    catch(ClassNotFoundException ex){
        System.out.println(ex);
    }
    catch(SQLException ce){
        System.out.println(ce);
    }
}

```

Figure 3 Display the table

Insertion

Name: Company Id:

Password: Department Id:

Deletion

Company Id:

Displaying

Display

Name	CompanyId	Password	DepartmentId
trishita	1011	5001	1011
AKASH	1111	7891	1000
Harshit	1112	7412	1001
Achintya	1113	7485	1002
Aishwarya	1114	8475	1004
Alok	1115	7415	1008
sdasd	asas	asas	asasa

Figure 4 Inserting and displaying Trishita's details

Insertion

Name: Company Id:

Password: Department Id:

Deletion

Company Id:

Displaying

Display

Name	CompanyId	Password	DepartmentId
trishita	1011	5001	1011
AKASH	1111	7891	1000
Harshit	1112	7412	1001
Achintya	1113	7485	1002
Aishwarya	1114	8475	1004
Alok	1115	7415	1008

Figure 5 Deleting details of person with company id as "asas" (Company id is string)

Insertion

Name Company Id

Password Department Id

Deletion

Company Id

Displaying

Display			
Name	CompanyId	Password	DepartmentId
trishita	1011	5001	1011
AKASH	1111	7891	1000
Harshit	1112	7412	1001
Achintya	1113	7485	1002
Aishwarya	1114	8475	1004
Alok	1115	7415	1008

Figure 6 Displaying the whole table

7. Conclusions

In the above lab, a user defined interface is designed to insert, delete and display from the records which is connected to the database. Here, java swing is used to create the user defined interface.

For the above problem, the scenario given was for an employee management system for any of the one attribute, insertion deletion and display are to be carried out. Here, the database is connected to the user-defined GUI.