Lab 9: Writing Java Database Applications using the JDBC API (Part 3)

Objectives

- Develop an Employee management system for an organization
- Using JTable

Tool(s)/Software

- NetBeans IDE

Description

This lab requires students to develop a database driven application. The data contains data about employee for an organization. This data can be displayed in the Java application through JTable. Code snippets have been provided and the students are required to view these snippets and develop a fully working application.

Tasks/Assignments

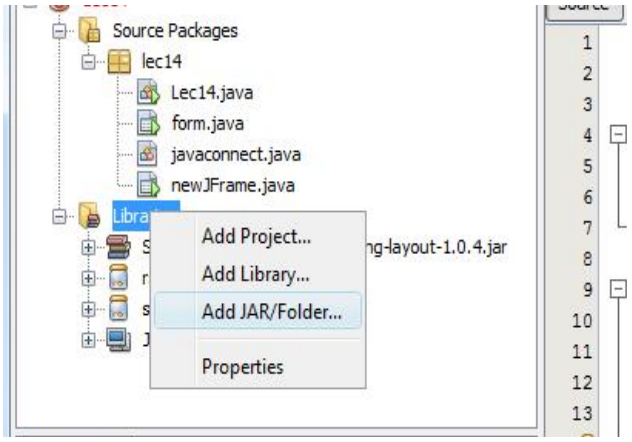
Follow the following steps:

1. Create Employees Database that has one table named Employee. Note that Emp ID field is the primary key. After creating the database, insert some sample data.

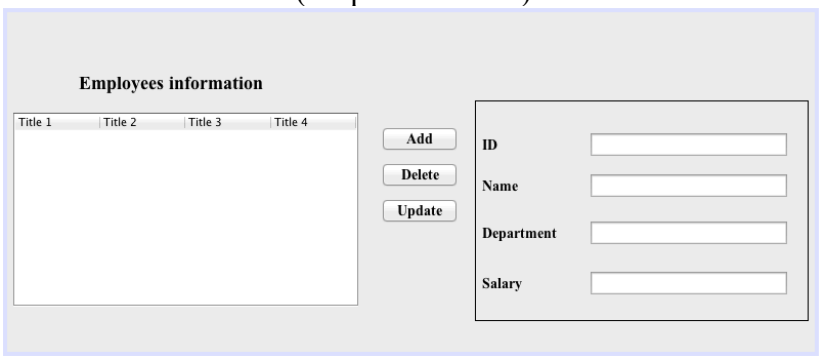
Emp ID	Name	Department	Salary
1255	Mohammed	IT	8500
1256	Omar	IT	9000
1300	Lama	HR	6000
1310	Sara	HR	5900

2. Add the required jar/library files

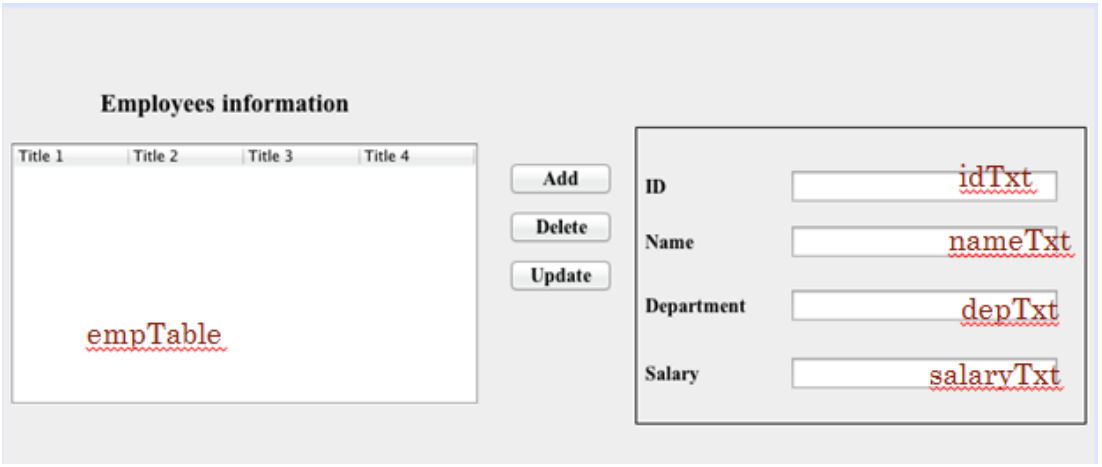
- Download the Jar File "rs2xml.jar" from BlackBoard. This is also available at:
  - https://sourceforge.net/projects/finalangelsanddemons/files/latest/download
- Add the Jar file to your project.
  - Right Click on Libraries -> add JAR/Folder then select the "rs2xml.jar" file
- Add the JDBC Driver.
  - Right Click on Libraries -> add Library then choose the appropriate driver



3. Design the interface using Netbeans design tools.
   - Create a new JFrame Form. (EmpTableFrame)



   - Variables Names:



4. Start Coding

- Class EmpTableFrame

```
public class EmpTableFrame extends JFrame{

    Connection con;
    Statement st;
    ResultSet rs;

    public EmpTableFrame() {
        super("Emplyees Info Application");
        initComponents();
        fillTable();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        addWindowListener(new WindowAdapter(){
              @Override
              public void windowClosed(WindowEvent event)
              {//close everything and terminate program
                    closeDB();
                    System.exit(0);
              }//end of windowClosed
           });

        setVisible(true);
    }
```

- FillTable Method in class EmpTableFrame
- FillTable Method in class EmpTableFrame. To use this statement you need to:
  - add rs2xml.jar
  - import net.proteanit.sql.DbUtils;

```
private void fillTable()
{
     try {
          con=DriverManager.getConnection("jdbc:derby://localhost:1527/Employees","esra","esra");
          st=con.createStatement();
          rs=st.executeQuery("select * from Employee");

          empTable.setModel(DbUtils.resultSetToTableModel(rs));

     }
     catch (SQLException ex)
     {
          JOptionPane.showMessageDialog(null, ex);
     }
}//end of fillTable method
```

- closeDB Method in class EmpTableFrame

```
public void closeDB()
{
    // close Statement and Connection
    try
    {
        rs.close();
        st.close();
        con.close();
    }
    catch (SQLException sqlException)
    {
        sqlException.printStackTrace();
    }
}
```

- Table Mouse Clicked

```
private void empTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_empTableMouseClicked
        try
        {
            int row=empTable.getSelectedRow();

            int tableClickID=Integer.parseInt(empTable.getModel().getValueAt(row, 0).toString());
            String tableClickname=empTable.getModel().getValueAt(row, 1).toString();
            String tableClickDep=empTable.getModel().getValueAt(row, 2).toString();
            int tableClickSal=Integer.parseInt(empTable.getModel().getValueAt(row, 3).toString());

            txtID.setText(""+tableClickID);
            txtName.setText(tableClickname);
            txtDep.setText(tableClickDep);
            txtSalary.setText(""+tableClickSal);

        }//end of try
        catch (Exception ex)
        {   JOptionPane.showMessageDialog(null,ex.getMessage());}
}//GEN-LAST:event_empTableMouseClicked
```

Alternative way:

```
private void empTableMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_empTableMouseClicked
        try
        {
            int row=empTable.getSelectedRow();
            int tableClickID=Integer.parseInt(empTable.getModel().getValueAt(row, 0).toString());

            //Another way to get the info using SQL query

            String sql="select * from employee where empID="+tableClickID;
            rs=st.executeQuery(sql);
            //To move the curser to the first row.
            rs.next();
            txtID.setText(String.format("%d",rs.getInt("empID")));
            txtName.setText(rs.getString("name"));
            txtDep.setText(rs.getString("department"));
            txtSalary.setText(String.format("%.2f",rs.getDouble("salary")));

        }//end of try
        catch (Exception ex)
        {   JOptionPane.showMessageDialog(null,ex.getMessage());}
}//GEN-LAST:event_empTableMouseClicked
```

- Add Button

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnAddActionPerformed
        try
        {
            PreparedStatement insertSt=con.prepareStatement("INSERT INTO Employee (empID,name,department,salary) VALUES (?,?,?,?)");

            insertSt.setInt(1,Integer.valueOf(txtID.getText()));
            insertSt.setString(2,txtName.getText());
            insertSt.setString(3,txtDep.getText());
            insertSt.setDouble(4,Double.valueOf(txtSalary.getText()));
            //We can check the retured value if it is 1 that means the row is updated
            insertSt.executeUpdate();

            JOptionPane.showMessageDialog(null, "Added..");

            closeDB();
            fillTable();
        }//end of try
        catch (SQLException ex)
        {   JOptionPane.showMessageDialog(null,ex.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);}
        //GEN-LAST:event_btnAddActionPerformed
        catch (Exception ex)
        {   JOptionPane.showMessageDialog(null,"Exception:"+ex.getMessage(),"Error",JOptionPane.ERROR_MESSAGE);}

    }//GEN-LAST:event_btnAddActionPerformed
```

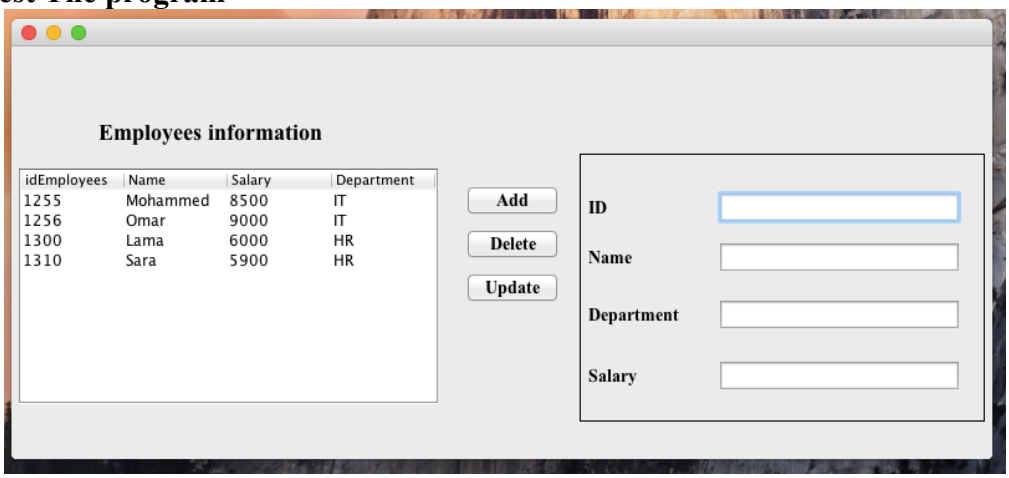
- Delete Button

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnDeleteActionPerformed
        try
        {
            int answer=JOptionPane.showConfirmDialog(null, "Are you sure you want to delete the selected record?",
                    "Delete Record",JOptionPane.YES_NO_OPTION);
            if (answer==JOptionPane.YES_OPTION)
            {   String sql="delete from employee where empID="+txtID.getText();
                int rows=st.executeUpdate(sql);
                if (rows == 1)
                    JOptionPane.showMessageDialog(null, "Deleted..");

            txtID.setText("");
            txtName.setText("");
            txtDep.setText("");
            txtSalary.setText("");

            closeDB();
            fillTable();
            }
        }//end of try
        catch (SQLException ex)
        {   JOptionPane.showMessageDialog(null,ex.getMessage());}
}//GEN-LAST:event_btnDeleteActionPerformed
```

- Update Button

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_btnUpdateActionPerformed
        try
        {
            PreparedStatement updateSt=con.prepareStatement("Update Employee set name=?,department=?,salary=? where empID= ?");

            updateSt.setString(1,txtName.getText());
            updateSt.setString(2,txtDep.getText());
            updateSt.setDouble(3,Double.valueOf(txtSalary.getText()));
            updateSt.setInt(4,Integer.valueOf(txtID.getText()));

            int updatedRows=updateSt.executeUpdate();

            if (updatedRows > 0)
                JOptionPane.showMessageDialog(null, "Updated..");
            else
                JOptionPane.showMessageDialog(null, "Cannot apply update process !");

            closeDB();
            fillTable();
        }//end of try
        catch (SQLException ex)
        {   JOptionPane.showMessageDialog(null,ex.getMessage());}
}//GEN-LAST:event_btnUpdateActionPerformed
```

- **Test The program**



Deliverables:

The deliverable for this lab is a working prototype of the employee management system. You should demonstrate your application to your instructor at the end of the lab.