

## LAB 6: JDBC NÂNG CAO



### MỤC TIÊU:

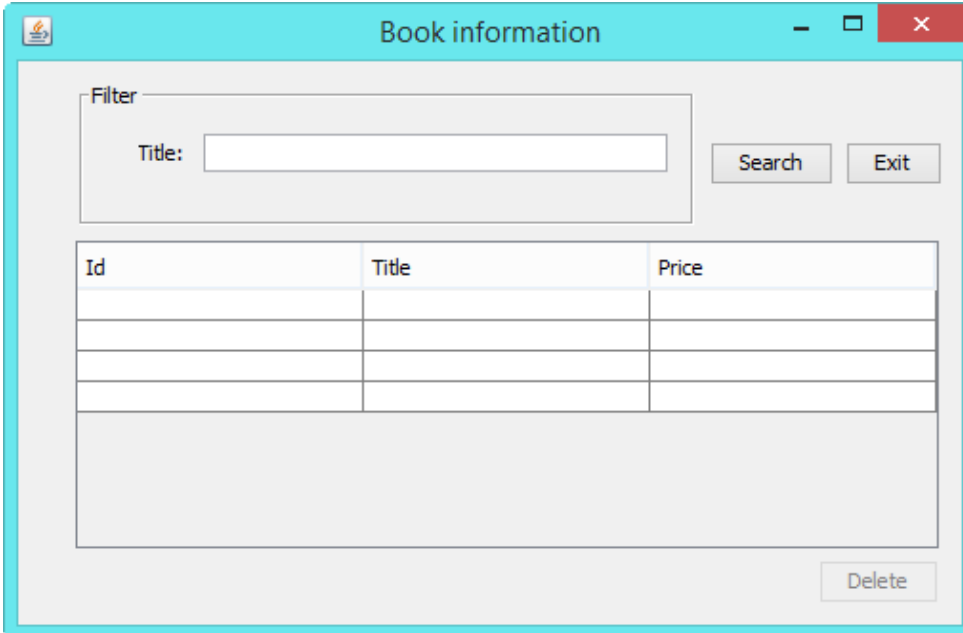
Kết thúc bài thực hành này bạn có khả năng

- ✓ Load dữ liệu từ Database lên JTextField, JComboBox và Jtable
- ✓ Thực hiện các chức năng thêm, xóa, sửa, update, tìm kiếm

### PHẦN I

#### Bài 1: JDBC – Thiết kế ứng dụng như hình và xử lý chức năng tìm kiếm và xóa dữ liệu (5 điểm)

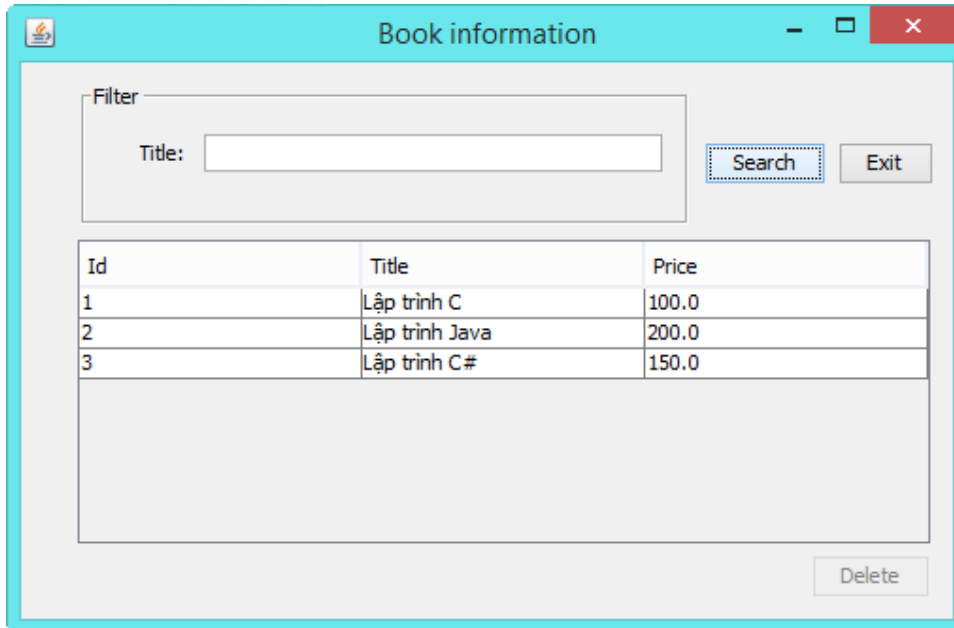
Cho phép người dùng xem và xóa dữ liệu của bảng Books. Ngoài ra chương trình cũng cho phép người dùng tìm kiếm một sách theo tiêu đề (title).



Id	Title	Price

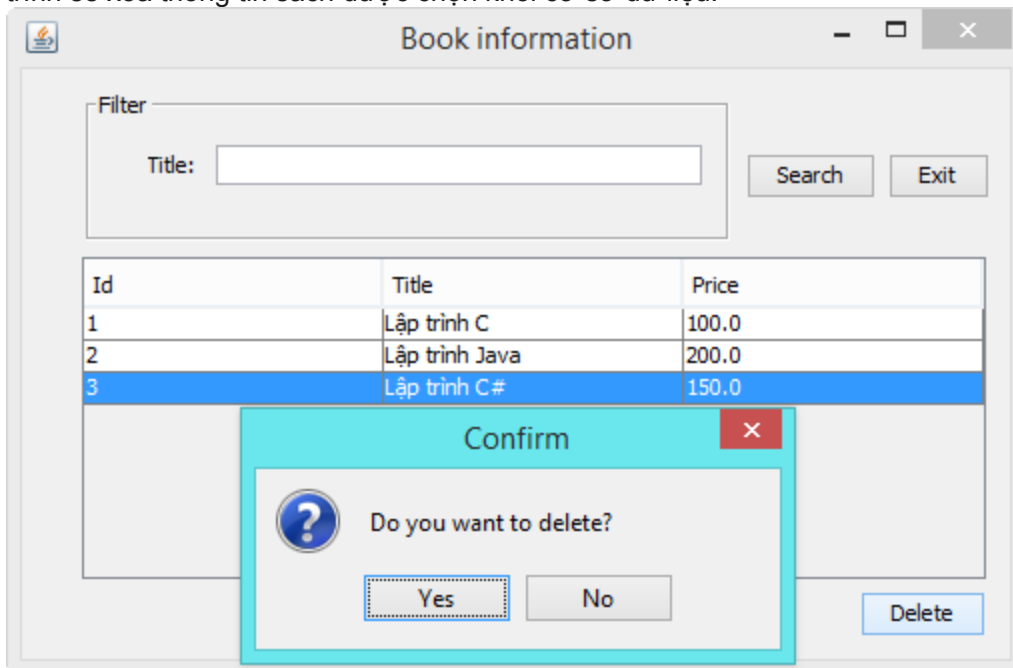
#### Mô tả yêu cầu

Đối với nút “Search” nếu vùng filter không nhập, chương trình sẽ hiển thị tất cả dữ liệu của bảng Books. Ngược lại chương trình chỉ hiển thị đúng với thông tin mà người dùng đã nhập hoặc ứng dụng sẽ thông báo “The book is not available!” trong trường hợp không tìm thấy.



Id	Title	Price
1	Lập trình C	100.0
2	Lập trình Java	200.0
3	Lập trình C#	150.0

Khi người dùng chọn một dòng trên table, nút “Delete” sẽ ở trạng thái sử dụng và chương trình sẽ hiển thị thông báo xác nhận “Do you want to delete?” khi người dùng chọn nút “Delete”. Nếu chọn “Yes”, chương trình sẽ xoá thông tin sách được chọn khỏi cơ sở dữ liệu.



Id	Title	Price
1	Lập trình C	100.0
2	Lập trình Java	200.0
3	Lập trình C#	150.0

Nút “Exit” sẽ đóng chương trình

### Viết xử lý cho nút “Search”

**Bước 1:** Khai báo tiêu đề và model cho Table

```
private String header[] = { "Id", "Title", "Price" };  
private DefaultTableModel tblModel = new DefaultTableModel(header, 0);
```

**Bước 2:** Truy xuất dữ liệu của bảng Books

```
Connection conn = null;
```

```
Statement st = null;
ResultSet rs = null;

try {
    conn = DriverManager.getConnection("jdbc:sqlserver://localhost;DatabaseName=Library", "sa", "sa");

    // Câu lệnh xem dữ liệu
    String sql = "select * from Books ";

    // Nếu tìm kiếm theo title
    if (txtTitle.getText().length() > 0) {
        sql = sql + " where title like '%" + txtTitle.getText() + "%'";
    }

    // Tạo đối tượng thực thi câu lệnh Select
    st = conn.createStatement();

    // Thực thi
    rs = st.executeQuery(sql);
    Vector data = null;

    tblModel.setRowCount(0);

    // Nếu sách không tồn tại
    if (rs.isBeforeFirst() == false) {
        JOptionPane.showMessageDialog(this, "The book is not available!");
        return;
    }

    // Trong khi chưa hết dữ liệu
    while (rs.next()) {
        data = new Vector();
        data.add(rs.getInt("id"));
        data.add(rs.getString("title"));
        data.add(rs.getString("price"));

        // Thêm một dòng vào table model
        tblModel.addRow(data);
    }

    jTable1.setModel(tblModel); // Thêm dữ liệu vào table
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (conn != null) {
```

```

        conn.close();
    }
    if (st != null) {
        st.close();
    }
    if (rs != null) {
        rs.close();
    }
} catch (Exception ex) {
    ex.printStackTrace();
}
}

```

### Viết xử lý cho nút “Delete”

Trước khi xoá dữ liệu, người dùng phải chọn một dòng trên table và chương trình sẽ lấy giá trị của cột đầu tiên của dòng được chọn để làm điều kiện xoá (cột id)

```

int ret = JOptionPane.showConfirmDialog(this, "Do you want to delete?", "Confirm",
JOptionPane.YES_NO_OPTION);
if(ret != JOptionPane.YES_OPTION) {
    return;
}

Connection c = null;
PreparedStatement ps = null;

try {
    c = DriverManager.getConnection("jdbc:sqlserver://localhost;DatabaseName=Library", "sa", "sa");
    ps = c.prepareStatement("Delete From Books where id = ?");
    ps.setInt(1, id); //giá trị của cột đầu tiên tại dòng được chọn trong table
    ret = ps.executeUpdate();
    if (ret != -1) {
        JOptionPane.showMessageDialog(this, "This book has been deleted");
    }
} catch (Exception ex) {
    ex.printStackTrace();
} finally {
    try {
        if (c != null) {
            c.close();
        }
        if (ps != null) {
            ps.close();
        }
    } catch (Exception ex2) {
        ex2.printStackTrace();
    }
}

```

}

### Chuẩn bị cơ sở dữ liệu (Cơ sở dữ liệu này được tạo với MS SQL Server)

```
CREATE DATABASE Library
```

```
USE
```

```
Library
```

```
GO
```

```
CREATE TABLE Books (
  ID INT IDENTITY PRIMARY KEY
  ,title NVARCHAR(50)
  ,price FLOAT
)
```

```
GO
```

```
INSERT INTO Books (title, price) VALUES (N'Lập trình C',100)
```

```
INSERT INTO Books (title, price) VALUES (N'Lập trình Java',200)
```

```
INSERT INTO Books (title, price) VALUES (N'Lập trình C#',150)
```

```
GO
```

## PHẦN II

### Bài 2: JDBC – Thiết kế ứng dụng như hình và xử lý các chức năng sau(5 điểm)

Name	Standard

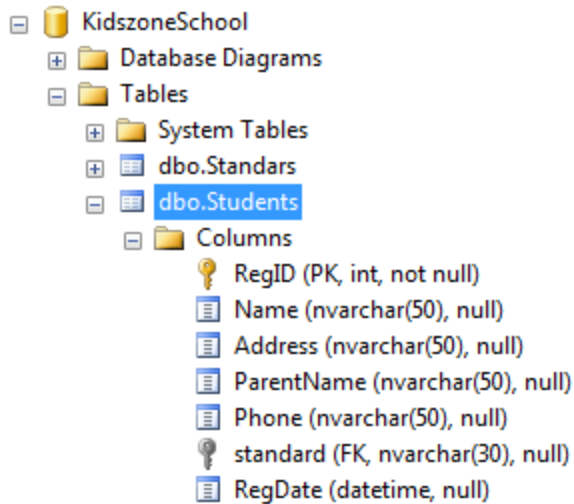
  

Name:   
Address:   
ParentName:   
ContactNo:   
Standard:   
Fees:

New Insert Edit Update  
Next Pre... Delete Exit

## Database



### Xử lý sự kiện:

- Khi người dùng click vào Jtable, hiển thị các thông tin tương ứng lên các controls bên phải.
- Viết xử lý cho nút “New”: Clear tất cả các giá trị controls bên phải, cho người dùng nhập mới
- Viết xử lý cho nút “Insert”: Lấy các giá trị người dùng đã nhập insert vào table Student.
- Viết xử lý cho nút “Edit”: Cho người dùng chỉnh thông tin sinh viên
- Viết xử lý cho nút “Update”: Cập nhật các giá trị đã thay đổi vào table.
- Viết xử lý cho nút “Next”: Đọc record tiếp theo trong table hiển thị lên các controls bên phải
- Viết xử lý cho nút “Previous”: Đọc record đứng trước dòng hiện tại trong table hiển thị lên các controls bên phải
- Viết xử lý cho nút “Delete”: Xóa dòng hiện tại.

### Code tham khảo:

```
import java.util.*;
import java.sql.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
public class Assignment extends javax.swing.JFrame {
    boolean addNew=false;
    boolean fill=false;
    Vector data=new Vector();
    Vector header=new Vector();
    Vector col =new Vector();
    Connection con=null;
    PreparedStatement pstDetails=null;
    PreparedStatement pstInsert=null;
    PreparedStatement pstDelete=null;
    PreparedStatement pstUpdate=null;
    String sqlInsert="Insert into Students ([Name],Address,ParentName,Phone,standard) values(?,?,?,?,?)";
    String sqlDelete="Delete from Students where Name=?";
    String sqlUpdate="Update Students set Address=?, ParentName=?,Phone=?,standard=? where Name=?";
```

```
ResultSet rts;
/** Creates new form Assignment */
public Assignment() {
    initComponents();
    // this.cbMID.setVisible(false);
    // this.pnlDetails.setVisible(false);
    //connect database
    try
    {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

con=DriverManager.getConnection("jdbc:sqlserver://localhost:1433;database=KidszoneSchool;userName=sa;password=");

        pstInsert=con.prepareStatement(this.sqlInsert);
        pstUpdate=con.prepareStatement(this.sqlUpdate);
        pstDelete=con.prepareStatement(this.sqlDelete);
        pstDetails=con.prepareStatement("select * from
Students",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
        rts=pstDetails.executeQuery();
        JOptionPane.showMessageDialog(this,"Connection Database Successful!");
        this.loadCombobox();
        this.loadData();
        fill=true;
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e);
    }
    btnUpdate.setEnabled(false);
}

private void loadCombobox(){
    String sql1="select * from dbo.Standars";
    try{
        Statement stm=con.createStatement();
        ResultSet rs=stm.executeQuery(sql1);
        Vector<String> standards=new Vector<String>();
        Vector<Integer> fees=new Vector<Integer>();
        while(rs.next()){
            cbb_Stan.addItem(rs.getString(1));
            cbb_Fees.addItem(rs.getInt(2));
        }
    }catch(Exception e){
        JOptionPane.showMessageDialog(this,e.getMessage());
        System.exit(0);
    }
}

private void loadData() throws SQLException
{
    String sql="Select [Name], standard from Students";
    try
    {
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery(sql);
        data.removeAllElements();
    }
}
```

```

        if(!fill)
        {
            //get header
            ResultSetMetaData rsmd=rs.getMetaData();
            int n=rsmd.getColumnCount();

            col.add(rsmd.getColumnName(1));
            col.add(rsmd.getColumnName(2));
        }
        //get data
        while(rs.next())
        {
            Vector v=new Vector();
            v.add(rs.getString(1));
            v.add(rs.getString(2));
            data.add(v);
        }
        TableModel tbl=new DefaultTableModel(data,col);
        this.tblStudent.setModel(tbl);
        rs.close();
        st.close();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e);
    }
}

boolean validates(){

    if(txt_Name.getText().matches("^\\w+$")==false){
        JOptionPane.showMessageDialog(this, "Name ko dc trong ", "Chu y", 1);
        txt_Name.requestFocus();
        return false;
    }
    String pName=this.txt_Name.getText().trim();
    Iterator it=data.iterator();
    while(it.hasNext())
    {
        Vector v=(Vector)it.next();
        String name=((String)v.get(0)).trim();
        if(pName.equalsIgnoreCase(name))
        {
            JOptionPane.showMessageDialog(this,"Ten Sinh Vien nay da ton tai!");
            this.txt_Name.grabFocus();
            return false;
        }
    }
    if(txt_Address.getText().matches("^\\w+$")==false){
        JOptionPane.showMessageDialog(this, "Address khong duoc de trong ", "Chu y",1);
        txt_Address.requestFocus();
        return false;
    }
    if(txt_PName.getText().matches("^\\w+$")==false){
        JOptionPane.showMessageDialog(this, "ParentsName khong duoc de trong ", "Chu y",1);
        txt_PName.requestFocus();
    }
}

```



```

        return false;
    }

    if(txt_Con.getText().matches("^\\d{7,11}$")==false){
        JOptionPane.showMessageDialog(this, "Contact(Phone) khong duoc de trong va phai la 7-12 so ", "Chu y", 1);
        txt_Con.requestFocus();
        return false;
    }
    if((cbb_Stan.getSelectedIndex()==0))
    {
        JOptionPane.showMessageDialog(this, cbb_Stan.getSelectedItem());
    }

    return true;
}

boolean duplicate(String s){
    if(addNew==false) return false;
    for (int i=0; i<data.size(); i++){
        Vector v=(Vector)data.get(i);
        if(s.equalsIgnoreCase((String)v.get(0))) return true;
    }
    return false;
}

private void clearForm() {

    txt_Name.setText("");
    txt_Address.setText("");
    txt_PName.setText("");
    txt_Con.setText("");
    cbb_Stan.setSelectedIndex(0);
    cbb_Fees.setSelectedIndex(0);
    txt_Name.requestFocus();
}

private void cbb_StanActionPerformed(java.awt.event.ActionEvent evt) {
    cbb_Fees.setSelectedIndex(cbb_Stan.getSelectedIndex());
}

private void cbb_FeesActionPerformed(java.awt.event.ActionEvent evt) {
    cbb_Stan.setSelectedIndex(cbb_Fees.getSelectedIndex());
}

private void tblStudentMouseReleased(java.awt.event.MouseEvent evt) {
    if(this.tblStudent.getCellEditor()!=null)
        this.tblStudent.getCellEditor().cancelCellEditing();
}

private void tblStudentMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try{
        rts.beforeFirst();
        this.clearForm();
        int row=tblStudent.getSelectedRow();
        String name=(String) tblStudent.getValueAt(row, 0);
        while(rts.next()){
            String str=rts.getString(2); //JOptionPane.showMessageDialog(null, ""+jrs.getString(2)+" "+name);

```

```

        if(str.equalsIgnoreCase(name))
        {
            txt_Name.setText(rts.getString(2));
            txt_Address.setText(rts.getString(3));
            txt_Con.setText(rts.getInt(5)+"");
            //String parent=dencry(rts.getString(5));
            txt_PName.setText(rts.getString(4));
            cbb_Stan.setSelectedItem(rts.getString(6));
            //com_fees.setSelectedItem(rts.getInt(7));
            break;
        }
    }
} catch(Exception e){}
txt_Name.setEnabled(false);
txt_Address.setEnabled(false);
txt_PName.setEnabled(false);
txt_Con.setEnabled(false);
cbb_Stan.setEnabled(false);
cbb_Fees.setEnabled(false);
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        int n;
        if(addNew)
        {
            //them moi
            //set cac tham so
            pstInsert.setString(1,this.txt_Name.getText().trim());
            pstInsert.setString(2,txt_Address.getText().trim());
            pstInsert.setString(3,txt_PName.getText().trim());
            pstInsert.setString(4,txt_Con.getText().trim());
            pstInsert.setString(5,(String) cbb_Stan.getSelectedItem());
            n=pstInsert.executeUpdate();//thuc thi
            this.loadData();
        }
        else
        {
            //update
            //set cac tham so
            pstUpdate.setString(5,this.txt_Name.getText().trim());
            pstUpdate.setString(2,this.txt_Address.getText().trim());
            pstUpdate.setString(1,this.txt_PName.getText().trim());
            pstUpdate.setString(3,this.txt_Con.getText().trim());
            pstUpdate.setString(4,(String) cbb_Stan.getSelectedItem());
            n=pstUpdate.executeUpdate();//thuc thi
            this.loadData();
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e);
    }
}

```

```
private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {
    clearForm();
    txt_Name.setEnabled(true);
    txt_Address.setEnabled(true);
    txt_PName.setEnabled(true);
    txt_Con.setEnabled(true);
    cbb_Fees.setEnabled(true);
    cbb_Stan.setEnabled(true);
    txt_Name.requestFocus();
}

private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    btnUpdate.setEnabled(true);
    btnCancel.setEnabled(true);
    btnEdit.setEnabled(true);
    txt_Name.setEnabled(true);
    txt_Address.setEnabled(true);
    txt_PName.setEnabled(true);
    txt_Con.setEnabled(true);
    cbb_Stan.setEnabled(true);
}

private void btnPreActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        //JOptionPane.showMessageDialog(this, "Co loi xay ra");
        rts.previous();
        // JOptionPane.showMessageDialog(this, "Co loi xay ra");
        btnNext.setEnabled(true);
        if(rts.isBeforeFirst()){
            // System.out.println("co loi xay ra");
            btnPre.setEnabled(false);
            btnNext.setEnabled(true);
            JOptionPane.showMessageDialog(null, "You have reached the first record " +
                "of the ResultSet!!!!");
        }
        else {
            txt_Name.setText(rts.getString(2));
            txt_Address.setText(rts.getString(3));
            txt_PName.setText(rts.getString(4));
            txt_Con.setText(rts.getString(5));
            cbb_Stan.setSelectedItem(rts.getString(6));
        }
    } catch (Exception e) {
        //JOptionPane.showConfirmDialog(this,"co loi xay ra");
    }
}

private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {

    //loadCombobox();
    try {
        rts.next();
        btnPre.setEnabled(true);
        if(rts.isAfterLast() || rts.isBeforeFirst()){
            btnNext.setEnabled(false);
        }
    }
}
```

```

        btnPre.setEnabled(true);
        JOptionPane.showMessageDialog(null, "You have reached the last record" +
            " of the ResultSet!!!!");
    } else {
        txt_Name.setText(rts.getString(2));
        txt_Address.setText(rts.getString(3));
        txt_PName.setText(rts.getString(4));
        txt_Con.setText(rts.getString(5));
        cbb_Stan.setSelectedItem(rts.getString(6));
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {
    fill=false;
    System.exit(0);
}

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        int n=this.tblStudent.getSelectedRow();
        if(n>=0)//nguoi dung co chon
        {
            //this.pnlDetails.setVisible(false);
            Vector v=(Vector)data.get(n);
            int ans=JOptionPane.showConfirmDialog(this,"Ban co thuc su muon xoa Sinh Vien " +
                ((String)v.get(0)).trim() + " khong?");
            if(ans==JOptionPane.YES_OPTION)
            {
                pstDelete.setString(1,(String)v.get(0));
                pstDelete.executeUpdate();
                this.loadData();
            }
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e);
    }
}

private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
    if(!validates()) return;
    String name=txt_Name.getText();
    String addr=txt_Address.getText();
    String parentName=txt_PName.getText();
    String phone=txt_Con.getText();
    String standard=(String) cbb_Stan.getSelectedItem();
    try{
        pstInsert.setString(1,name);
        pstInsert.setString(2,addr);
        pstInsert.setString(3,parentName);
        pstInsert.setString(4,phone);
        pstInsert.setString(5,standard);
    }
}

```

```
        int addRows= pstInsert.executeUpdate();
        this.loadData();
        clearForm();
        if(addRows>0)
            JOptionPane.showMessageDialog(this, "Students Details Have Been
Save", "Successfull",JOptionPane.INFORMATION_MESSAGE);
        }catch(Exception ex){
            JOptionPane.showMessageDialog(this, "Failed to save data in database", "Error",
JOptionPane.ERROR_MESSAGE);
            ex.printStackTrace();
        }
    }

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Assignment().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnCancel;
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnEdit;
private javax.swing.JButton btnExit;
private javax.swing.JButton btnInsert;
private javax.swing.JButton btnNext;
private javax.swing.JButton btnPre;
private javax.swing.JButton btnUpdate;
private javax.swing.JComboBox cbb_Fees;
private javax.swing.JComboBox cbb_Stan;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable tblStudent;
private javax.swing.JTextArea txt_Address;
private javax.swing.JTextField txt_Con;
private javax.swing.JTextField txt_Name;
private javax.swing.JTextField txt_PName;
// End of variables declaration
}
```

