

## LAB 5: JDBC



### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

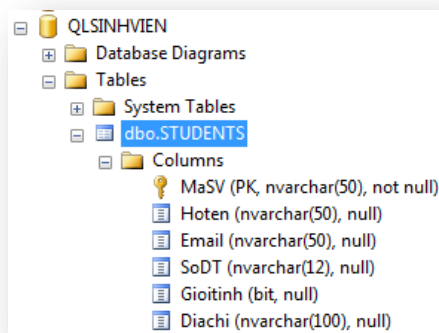
- ✓ Kết nối database, thực hiện các chức năng thêm, xóa, sửa, update, tìm kiếm
- ✓ Hiểu được và ứng dụng các lớp tác vụ cơ bản của JDBC:
  - Statement
  - PreparedStatement
  - CallableStatement
  - ResultSet

### PHẦN I

#### Bài 1: Kết nối CSDL SQL Server 2008, Type 4 (2 điểm)

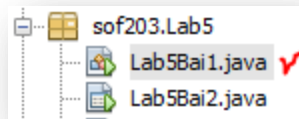
Yêu cầu: Viết chương trình đọc dữ liệu từ table hiển thị lên màn hình.

**Database:** tạo database và table có cấu trúc như hình

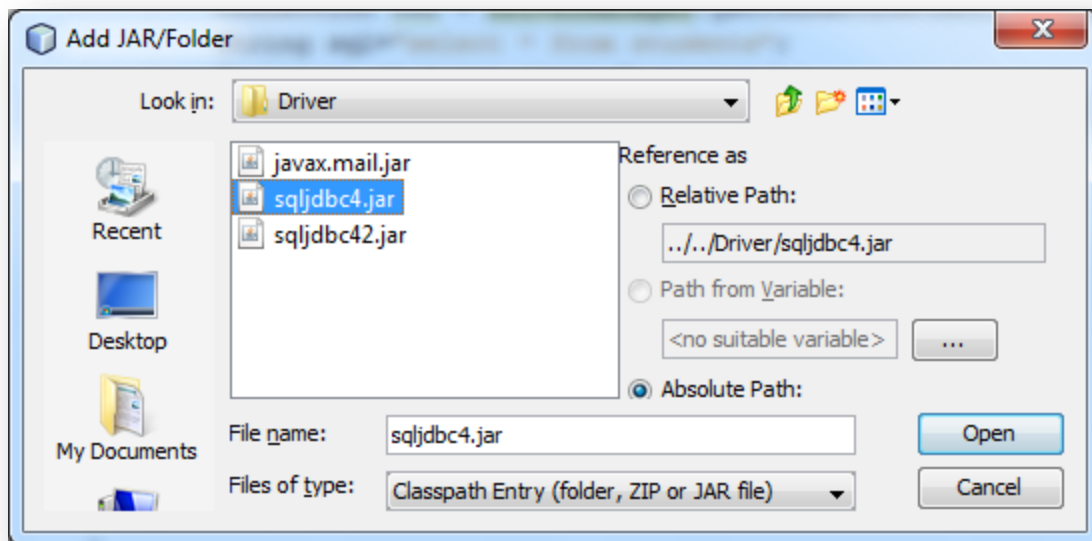
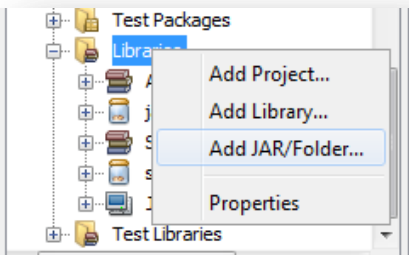


SCD050718.QLSL... - dbo.STUDENTS						
	MaSV	Hoten	Email	SoDT	Gioitinh	Diachi
	SV001	Le Van Phung	phunglv@fpt.edu.vn	0903414749	True	Ninh Thuan
	SV002	Le Quang Trung	trunglv@gmail.com	0901234567	True	222 Le Van Si
	SV003	Le Thi Bao Hieu	hieultb@gmail.com	0683872432	False	Phan Rang
	SV004	Le Thi H Hanh	hanhlth@gmail.com	0909999999	False	Quan 12
▶*	NULL	NULL	NULL	NULL	NULL	NULL

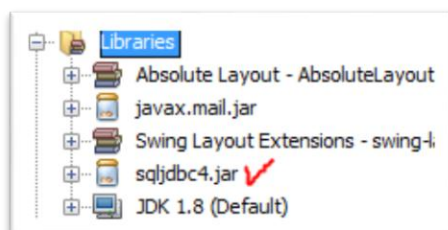
Tạo project Java:



**Add Library:** click phải chọn Library\Add Jar/Folder..., chọn file sqljdbc4.jar

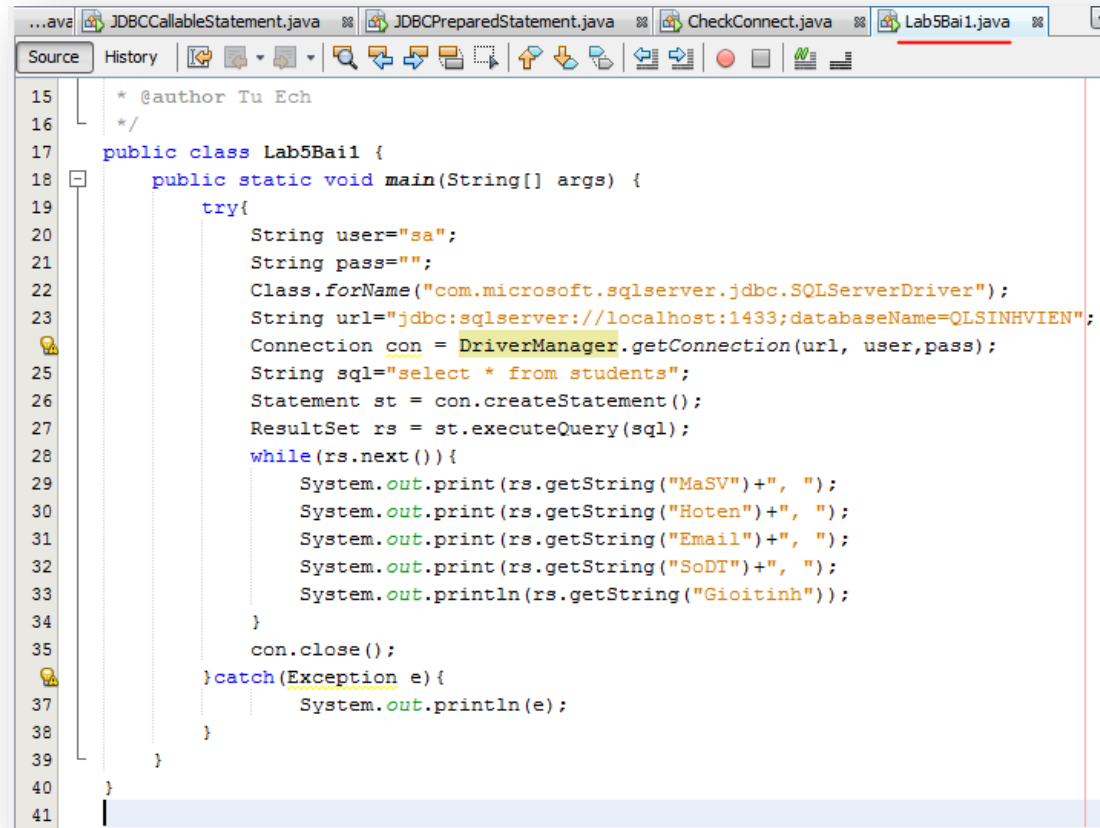


Kết quả như hình:



Link file sqljdbc4.jar: <https://www.fshare.vn/file/8PKE3VWM4FI3>

Code :

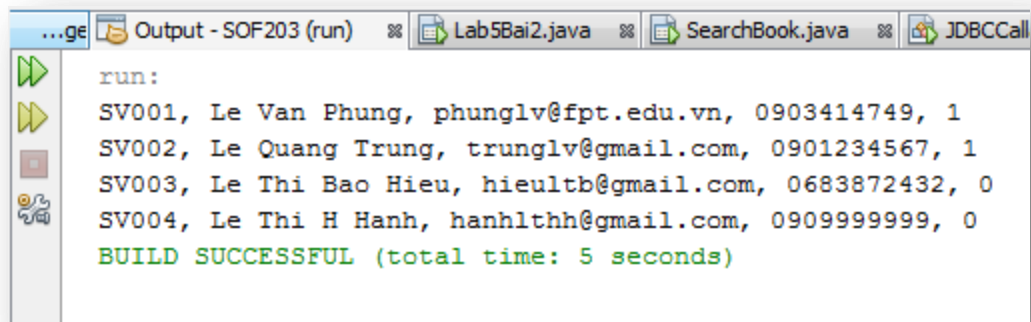


```

15  * @author Tu Ech
16  */
17  public class Lab5Bai1 {
18      public static void main(String[] args) {
19          try{
20              String user="sa";
21              String pass="";
22              Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
23              String url="jdbc:sqlserver://localhost:1433;databaseName=QLSINHVIEN";
24              Connection con = DriverManager.getConnection(url, user,pass);
25              String sql="select * from students";
26              Statement st = con.createStatement();
27              ResultSet rs = st.executeQuery(sql);
28              while(rs.next()){
29                  System.out.print(rs.getString("MaSV")+", ");
30                  System.out.print(rs.getString("Hoten")+", ");
31                  System.out.print(rs.getString("Email")+", ");
32                  System.out.print(rs.getString("SoDT")+", ");
33                  System.out.println(rs.getString("Gioitinh"));
34              }
35              con.close();
36          }catch(Exception e){
37              System.out.println(e);
38          }
39      }
40  }
41

```

Kết quả:




```

run:
SV001, Le Van Phung, phunglv@fpt.edu.vn, 0903414749, 1
SV002, Le Quang Trung, trunglv@gmail.com, 0901234567, 1
SV003, Le Thi Bao Hieu, hieultb@gmail.com, 0683872432, 0
SV004, Le Thi H Hanh, hanhlthh@gmail.com, 0909999999, 0
BUILD SUCCESSFUL (total time: 5 seconds)

```

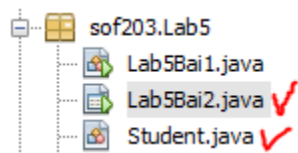
**PHẦN II****Bài 2: Thiết kế ứng dụng như hình và xử lý chức năng theo yêu cầu (7 điểm)**

Cho phép người dùng thêm, xoá, sửa, update dữ liệu của bảng Students. Ngoài ra chương trình còn cho phép duyệt danh sách sinh viên.



**Database:** như bài 1

**Cấu trúc chương trình gồm:**



**Tạo file Student.java có nội dung như sau:**

```

* @author Tu Ech
*/
public class Student {
    String MaSV;
    String Hoten;
    String Email;
    String SoDT;
    String Diachi;
    boolean Gioitinh;
    public Student(String MaSV, String Hoten, String Email, String SoDT,
        String Diachi, boolean Gioitinh) {
        this.MaSV = MaSV;
        this.Hoten = Hoten;
        this.Email = Email;
        this.SoDT = SoDT;
        this.Diachi = Diachi;
        this.Gioitinh = Gioitinh;
    }
}

```

**Khai báo các biến toàn cục:**

```

public class Lab5Bai2 extends javax.swing.JFrame {
    ArrayList<Student> list = new ArrayList<Student>();
    int current=0;
    String userName="sa";
    String password="";
    String url= "jdbc:sqlserver://localhost:1433;databaseName=QLSINHVIEN";
    public Lab5Bai2() {
        initComponents();
        LoadDataToArray(); //load du lieu tu table vao arraylist
        Display(current); // hien thi du lieu dau tien len frame
    }
}

```

**Hàm Display(int i):** dùng để hiển thị sv thứ i trong arraylist lên Frame

```

public void Display(int i){
    Student sv = list.get(i);
    txtDiachi.setText(sv.Diachi);
    txtEmail.setText(sv.Email);
    txtHoten.setText(sv.Hoten);
    txtMaSV.setText(sv.MaSV);
    txtSoDT.setText(sv.SoDT);
    boolean gt = sv.Gioitinh;
    if(gt==true){
        rdoNam.setSelected(true);
    }else{
        rdoNu.setSelected(true);
    }
}
}

```

**Hàm LoadDataToArray():** đọc tất cả dữ liệu trong table students vào arraylist.

```
public void LoadDataToArray() {  
    try{  
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
        Connection con = DriverManager.getConnection(url, userName, password);  
        Statement st = con.createStatement();  
        String sql="select * from STUDENTS";  
        ResultSet rs = st.executeQuery(sql);  
        list.clear();  
        while(rs.next()){  
            String masv = rs.getString(1);  
            String hoten = rs.getString(2);  
            String email = rs.getString(3);  
            String sodt = rs.getString(4);  
            boolean gt = rs.getBoolean(5);  
            String diachi = rs.getString(6);  
            Student sv = new Student(masv, hoten, email, sodt, diachi, gt);  
            list.add(sv);  
        }  
        con.close();  
    }catch(Exception ex){  
        System.out.println(ex);  
    }  
}
```

**Viết xử lý cho các nút:** First, Next, Previous, Last

## FPT POLYTECHNIC

```
private void btnLastActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    current = list.size()-1;
    Display(current);
}
```

```
private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    current++;
    if(current >= list.size()){
        JOptionPane.showMessageDialog(null, "Đang ở cuối!");
        return;
    }
    Display(current);
}
```

```
private void btnPreActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    current--;
    if(current < 0){
        JOptionPane.showMessageDialog(null, "Đang ở đầu danh sách!");
        return;
    }
    Display(current);
}
```

### Viết xử lý cho nút “Delete”:

```
private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    if(txtMaSV.getText().equals("")){
        JOptionPane.showMessageDialog(this, "Nhập Mã SV");
        txtMaSV.requestFocus();
        return;
    }
    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection con = DriverManager.getConnection(url, userName, password);
        String sql="delete from STUDENTS where MaSV = ?";
        PreparedStatement st = con.prepareStatement(sql);
        st.setString(1, txtMaSV.getText());
        st.execute();
        JOptionPane.showMessageDialog(this, "Delete thành công!");
        con.close();
        Display(current--);
    }catch(Exception e){
        System.out.println(e);
        JOptionPane.showMessageDialog(this, "Error");
    }
}
```

Viết xử lý cho nút “Add”:

```
private void btnThemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtDiachi.setText("");
    txtEmail.setText("");
    txtHoten.setText("");
    txtMaSV.setText("");
    txtSoDT.setText("");
    txtMaSV.requestFocus();
}
```

Viết xử lý cho nút “Update”:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    if(txtMaSV.getText().equals("")){
        JOptionPane.showMessageDialog(this, "Nhập Mã SV");
        txtMaSV.requestFocus(); return;
    }
    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection con = DriverManager.getConnection(url, userName, password);
        String sql="update STUDENTS set Hoten=?, Email=?, SoDT=?, Diachi=?, "
            + "Gioitinh=? where MaSV = ?";
        PreparedStatement st = con.prepareStatement(sql);
        st.setString(1, txtHoten.getText());
        st.setString(2, txtEmail.getText());
        st.setString(3, txtSoDT.getText());
        st.setString(4, txtDiachi.getText());
        boolean gt;
        if(rdoNam.isSelected())
            gt=true;
        else
            gt=false;
        st.setBoolean(5, gt);
        st.setString(6, txtMaSV.getText());
        st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Update thành công!");
        con.close(); LoadDataToArray();
    }catch(Exception e){
        System.out.println(e);
        JOptionPane.showMessageDialog(this, "Error");
    }
}
```

Viết xử lý cho nút “Save”:



```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    //check du lieu nhap, kiem tra trung MaSV sinh vien tu lam
    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection con = DriverManager.getConnection(url, userName,password);
        String sql="insert into STUDENTS values(?,?,?,?,?,?)";
        PreparedStatement st = con.prepareStatement(sql);
        st.setString(1, txtMaSV.getText());
        st.setString(2, txtHoten.getText());
        st.setString(3, txtEmail.getText());
        st.setString(4, txtSoDT.getText());
        boolean gt;
        if(rdoNam.isSelected())
            gt=true;
        else
            gt=false;
        st.setBoolean(5, gt);
        st.setString(6, txtDiachi.getText());
        st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Save thành công!");
        con.close(); LoadDataToArray();
    }catch(Exception e){
        System.out.println(e);
        JOptionPane.showMessageDialog(this, "Error");
    }
}
```

### Bài 3: Phát triển thêm chức năng tìm kiếm cho bài 2 (1 điểm)

Cho phép người dùng nhập MaSV cần tìm, hiển thị thông tin sinh viên nếu tìm thấy, else thông báo không tìm thấy.