Aashish Shrestha.

Unit 4

Q) Write a program to illustrate the process of executing SQL statements in JDBC? [2073, 2074]
Q) Implement CRUD (Create/Insert, Read/Select, Update, Delete) operations for student table. Ask for user input where applicable.
Q) Implement CRUD operations for student table using prepared statements. Ask for user input where applicable.
Q) Implement CRUD operations for student table in Swing. Ask for user input where applicable.

Program:

All the above questions requirements are fulfilled in the following program with the swing implementation.

```
package crudswing;

/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
import com.sun.xml.internal.txw2.output.ResultFactory;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.Action;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.xml.ws.BindingProvider;

/**
 *
```

Aashish Shrestha.

```
 * @author user
 */
public class CRUDSwing extends JFrame {

    /**
     * @param args the command line arguments
     */
    String comboBoxOperation;

    public static void main(String[] args) throws Exception {
        CRUDSwing app = new CRUDSwing();
        CRUDSwing result = new CRUDSwing();
        app.getContentPane().setBackground(Color.LIGHT_GRAY);
        result.getContentPane().setBackground(Color.LIGHT_GRAY);
        app.setTitle("CRUD Swing");
        app.setVisible(true);
    }

    public CRUDSwing() throws SQLException {
        setSize(700, 700);
        setLayout(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();

        String inputNameGlobal, inputDistrictGlobal, resultTextGlobal;
        final int inputAgeGlobal, inputidGlobal;

        String url = "jdbc:mariadb://localhost:3306/trinity";
        String Username = "root";
        String password = "";
        Connection   connection   =   DriverManager.getConnection(url,
Username, password);
//        Statement statement = connection.createStatement();

        c.insets = new Insets(5, 5, 5, 5);

        JLabel  optionText  =  new  JLabel("Choose  the  desired  service
type");
        c.gridx = 0;
        c.gridy = 0;
        c.weightx = 0.75;
        c.gridwidth = 2;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(optionText, c);

        String[]  serviceOptions  =  {"--Choose--",  "Create",  "Search",
"Edit", "Delete"};
        JComboBox service = new JComboBox(serviceOptions);
        c.gridx = 2;
```

```
c.gridy = 0;
c.weightx = 0.25;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(service, c);

JLabel IDLabel = new JLabel("Enter the ID");
c.gridx = 0;
c.gridy = 1;
c.weightx = 0.75;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(IDLabel, c);

JTextField ID = new JTextField("", 50);
ID.setEditable(false);
c.gridx = 2;
c.gridy = 1;
c.weightx = 0.25;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(ID, c);

JLabel usernameLabel = new JLabel("Enter the Username");
c.gridx = 0;
c.gridy = 2;
c.weightx = 0.75;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(usernameLabel, c);

JTextField username = new JTextField("", 50);
username.setEditable(false);
c.gridx = 2;
c.gridy = 2;
c.weightx = 0.75;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(username, c);

JLabel districtLabel = new JLabel("Enter the district");
c.gridx = 0;
c.gridy = 3;
c.weightx = 0.75;
c.gridwidth = 2;
c.fill = GridBagConstraints.HORIZONTAL;
add(districtLabel, c);
```

```
        JTextField district = new JTextField("", 50);
        district.setEditable(false);
        c.gridx = 2;
        c.gridy = 3;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(district, c);

        JLabel ageLabel = new JLabel("Enter the Age");
        c.gridx = 0;
        c.gridy = 4;
        c.weightx = 0.75;
        c.gridwidth = 2;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(ageLabel, c);

        JTextField age = new JTextField("", 50);
        age.setEditable(false);
        c.gridx = 2;
        c.gridy = 4;
        c.weightx = 0.25;
        c.gridwidth = 2;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(age, c);

        JButton Submit = new JButton("Submit");
        Submit.setEnabled(false);
        c.gridx = 0;
        c.gridy = 5;
        c.gridwidth = 4;
        c.weightx = 1;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(Submit, c);

        JLabel resultlabel = new JLabel("Result ::");
        c.gridx = 0;
        c.gridy = 6;
        c.gridwidth = 2;
        c.weightx = 0.75;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultlabel, c);
        resultlabel.setVisible(false);

        JTextField resultField = new JTextField("", 50);
        c.gridx = 2;
        c.gridy = 6;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultField, c);
        resultField.setVisible(false);
        resultField.setEditable(false);

        JLabel resultIDLabel = new JLabel("ID :");
        c.gridx = 0;
        c.gridy = 7;
        c.gridwidth = 2;
        c.weightx = 0.75;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultIDLabel, c);
        resultIDLabel.setVisible(false);

        JTextField resultIDField = new JTextField("", 50);
        c.gridx = 2;
        c.gridy = 7;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultIDField, c);
        resultIDField.setVisible(false);
        resultIDField.setEditable(false);

        JLabel resultNameLabel = new JLabel("Name :");
        c.gridx = 0;
        c.gridy = 8;
        c.gridwidth = 2;
        c.weightx = 0.75;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultNameLabel, c);
        resultNameLabel.setVisible(false);

        JTextField resultNameField = new JTextField("", 50);
        c.gridx = 2;
        c.gridy = 8;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultNameField, c);
        resultNameField.setEditable(false);
        resultNameField.setVisible(false);

        JLabel resultDistrictLabel = new JLabel("District :");
        c.gridx = 0;
        c.gridy = 9;
        c.gridwidth = 2;
        c.weightx = 0.75;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultDistrictLabel, c);
        resultDistrictLabel.setVisible(false);

        JTextField resultDistrictField = new JTextField("", 50);
        c.gridx = 2;
        c.gridy = 9;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultDistrictField, c);
        resultDistrictField.setVisible(false);
        resultDistrictField.setEditable(false);

        JLabel resultAgeLabel = new JLabel("Age :");
        c.gridx = 0;
        c.gridy = 10;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultAgeLabel, c);
        resultAgeLabel.setVisible(false);

        JTextField resultAgeField = new JTextField("", 50);
        c.gridx = 2;
        c.gridy = 10;
        c.gridwidth = 2;
        c.weightx = 0.25;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(resultAgeField, c);
        resultAgeField.setVisible(false);
        resultAgeField.setEditable(false);

        JButton finish = new JButton("Finish");
        c.gridx = 2;
        c.gridy = 11;
        c.gridwidth = 2;
        c.weightx = 1;
        c.fill = GridBagConstraints.HORIZONTAL;
        add(finish, c);
        finish.setVisible(false);

        service.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent ae) {
                if (ae.getSource() == service) {
```

Aashish Shrestha.

```
                    JComboBox cb = (JComboBox) ae.getSource();
                    String       choosenService       =       (String)
cb.getSelectedItem();
                    comboBoxOperation = choosenService;
                    System.out.println(choosenService);
                    switch (choosenService) {
                        case "Create": {
                            ID.setText("This    is    a    autogenerated
number");
                            ID.setEditable(false);
                            username.setText("");
                            username.setEditable(true);
                            district.setText("");
                            district.setEditable(true);
                            Submit.setEnabled(true);
                            age.setText("");
                            age.setEditable(true);

                            resultlabel.setVisible(false);
                            resultField.setVisible(false);

                            resultIDLabel.setVisible(false);
                            resultIDField.setVisible(false);

                            resultNameLabel.setVisible(false);
                            resultNameField.setVisible(false);

                            resultDistrictLabel.setVisible(false);
                            resultDistrictField.setVisible(false);

                            resultAgeLabel.setVisible(false);
                            resultAgeField.setVisible(false);

                            finish.setVisible(false);
                            pack();

                        }
                        break;
                        case "Search": {
                            ID.setText("");
                            ID.setEditable(true);
                            username.setText("");
                            username.setEditable(false);
                            district.setText("");
                            district.setEditable(false);
                            Submit.setEnabled(true);
                            age.setText("");
                            age.setEditable(false);
```

```
            resultlabel.setVisible(false);
            resultField.setVisible(false);

            resultIDLabel.setVisible(false);
            resultIDField.setVisible(false);

            resultNameLabel.setVisible(false);
            resultNameField.setVisible(false);

            resultDistrictLabel.setVisible(false);
            resultDistrictField.setVisible(false);

            resultAgeLabel.setVisible(false);
            resultAgeField.setVisible(false);

            finish.setVisible(false);
            pack();
        }
        break;
        case "Edit": {
            ID.setText("");
            ID.setEditable(true);
            username.setText("");
            username.setEditable(true);
            district.setText("");
            district.setEditable(true);
            Submit.setEnabled(true);
            age.setText("");
            age.setEditable(true);
            resultlabel.setVisible(false);
            resultField.setVisible(false);

            resultIDLabel.setVisible(false);
            resultIDField.setVisible(false);

            resultNameLabel.setVisible(false);
            resultNameField.setVisible(false);

            resultDistrictLabel.setVisible(false);
            resultDistrictField.setVisible(false);

            resultAgeLabel.setVisible(false);
            resultAgeField.setVisible(false);

            finish.setVisible(false);
            pack();
        }
        break;
```

```
case "Delete": {
    ID.setText("");
    ID.setEditable(true);
    username.setText("");
    username.setEditable(false);
    district.setText("");
    district.setEditable(false);
    Submit.setEnabled(true);
    age.setText("");
    age.setEditable(false);
    resultlabel.setVisible(false);
    resultField.setVisible(false);

    resultIDLabel.setVisible(false);
    resultIDField.setVisible(false);

    resultNameLabel.setVisible(false);
    resultNameField.setVisible(false);

    resultDistrictLabel.setVisible(false);
    resultDistrictField.setVisible(false);

    resultAgeLabel.setVisible(false);
    resultAgeField.setVisible(false);

    finish.setVisible(false);
    pack();
}
break;
case "--Choose--": {
    ID.setText("");
    username.setText("");
    district.setText("");
    ID.setEditable(false);
    username.setEditable(false);
    district.setEditable(false);
    Submit.setEnabled(false);
    age.setEditable(false);
    resultlabel.setVisible(false);
    resultField.setVisible(false);

    resultIDLabel.setVisible(false);
    resultIDField.setVisible(false);

    resultNameLabel.setVisible(false);
    resultNameField.setVisible(false);

    resultDistrictLabel.setVisible(false);
```

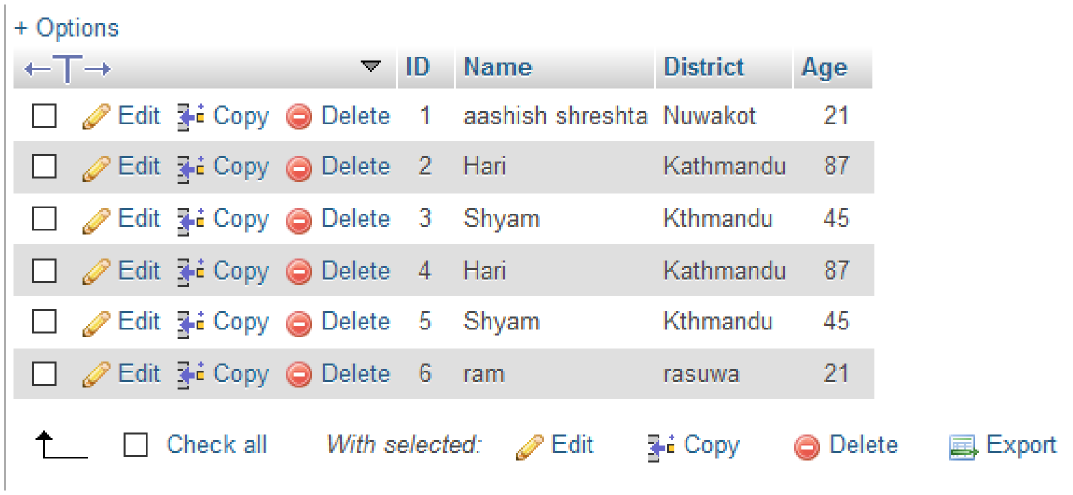
```
                            resultDistrictField.setVisible(false);

                            resultAgeLabel.setVisible(false);
                            resultAgeField.setVisible(false);

                            finish.setVisible(false);
                            pack();
                        }
                        break;

                    }
                }
            }
        });

        Submit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                ///Database operation here

                String  inputName  =  null,  inputDistrict  =  null,
resultText = null, ag = null;
                int inputAge, inputid;

                if (comboBoxOperation.equalsIgnoreCase("create")) {

                    String sql = "INSERT INTO students (Name, District,
Age)VALUES (?, ?, ?)";

                    PreparedStatement preparedStmt;
                    try {
                        inputName = username.getText();
                    inputDistrict = district.getText();
                    inputAge = Integer.parseInt(age.getText());
                        preparedStmt                                =
connection.prepareStatement(sql);
                        preparedStmt.setString(1, inputName);
                        preparedStmt.setString(2, inputDistrict);
                        preparedStmt.setInt(3, inputAge);


                        int rowsUpdate = preparedStmt.executeUpdate();
                        if (rowsUpdate > 0) {
                            resultText = "Inserted sucessfully";

                        } else {
                            resultText = "Failed";
                        }
                    } catch (SQLException ex) {
                        resultText = "Failed (SQL Error)";
                    } finally {
                        resultlabel.setVisible(true);
                        resultField.setText(resultText);
                        resultField.setVisible(true);

                        finish.setVisible(true);
                        pack();
                    }

                }

                if (comboBoxOperation.equalsIgnoreCase("Edit")) {



                    String sql = "update students set Name = ?, District
= ?, Age = ? where ID = ?";

                    PreparedStatement preparedStmt;
                    try {
                         inputName = username.getText();
                    inputDistrict = district.getText();
                    inputAge = Integer.parseInt(age.getText());
                    inputid = Integer.parseInt(ID.getText());
                        preparedStmt                                 =
connection.prepareStatement(sql);
                        preparedStmt.setString(1, inputName);
                        preparedStmt.setString(2, inputDistrict);
                        preparedStmt.setInt(3, inputAge);
                        preparedStmt.setInt(4, inputid);


                        int rowsUpdate = preparedStmt.executeUpdate();
                        if (rowsUpdate > 0) {
                            resultText = "Updated sucessfully";

                        } else {
                            resultText = "Failed";
                        }

                    } catch (SQLException ex) {
                        resultText = "Failed (SQL Error)";
                    } finally {
                        resultlabel.setVisible(true);
                        resultField.setText(resultText);
                        resultField.setVisible(true);

                        resultIDLabel.setVisible(true);
                        resultIDField.setText(ID.getText());
                        resultIDField.setVisible(true);

                        resultNameLabel.setVisible(true);
                        resultNameField.setText(username.getText());
                        resultNameField.setVisible(true);

                        resultDistrictLabel.setVisible(true);

resultDistrictField.setText(district.getText());
                        resultDistrictField.setVisible(true);

                        resultAgeLabel.setVisible(true);
                        resultAgeField.setText(age.getText());
                        resultAgeField.setVisible(true);

                        finish.setVisible(true);
                        pack();
                    }

                }

                if (comboBoxOperation.equalsIgnoreCase("Search")) {



                    String sql = "select * from students where id =?";

                    PreparedStatement preparedStmt;
                    try {
                        inputid = Integer.parseInt(ID.getText());
                        preparedStmt                                    =
connection.prepareStatement(sql);
                        preparedStmt.setInt(1, inputid);

                        ResultSet rs = preparedStmt.executeQuery();
                        while (rs.next()) {
                            inputName = rs.getString("Name");
                            ag = Integer.toString(rs.getInt("Age"));
                            resultText = "Search Complete";
                            inputDistrict = rs.getString("District");

                        }
                    } catch (SQLException ex) {
                        resultText = "Failed (SQL Error)";
                    } finally {

                        resultlabel.setVisible(true);
                        resultField.setText(resultText);
                        resultField.setVisible(true);

                        resultIDLabel.setVisible(true);
                        resultIDField.setText(ID.getText());
                        resultIDField.setVisible(true);

                        resultNameLabel.setVisible(true);

                        resultNameField.setVisible(true);
                        resultNameField.setText(inputName);

                        resultDistrictLabel.setVisible(true);
                        resultDistrictField.setText(inputDistrict);
                        resultDistrictField.setVisible(true);

                        resultAgeLabel.setVisible(true);
                        resultAgeField.setText(ag);
                        resultAgeField.setVisible(true);

                        finish.setVisible(true);
                        pack();

                    }

                }

                if (comboBoxOperation.equalsIgnoreCase("Delete")) {



                    String sql = "delete from students where id = ?";

                    PreparedStatement preparedStmt;
                    try {
                        inputid = Integer.parseInt(ID.getText());
                        preparedStmt                                    =
connection.prepareStatement(sql);
                        preparedStmt.setInt(1, inputid);
```

```
                        int rowsUpdate = preparedStmt.executeUpdate();
                        if(rowsUpdate >  0){

                        resultText = "Deleted sucessfully";
                        }
                        else{
                        resultText = "Failed"; }


                    } catch (SQLException ex) {
                        resultText = "Failed (SQL Error)";
                    } finally {
                        resultlabel.setVisible(true);
                        resultField.setText(resultText);
                        resultField.setVisible(true);

                        finish.setVisible(true);
                        pack();
                    }

                }

            }

        });

        finish.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent ae) {

                System.exit(0);
            }
        });
        pack();
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

}
```

<u>Output:</u>

The data in the defined database before any operation is:

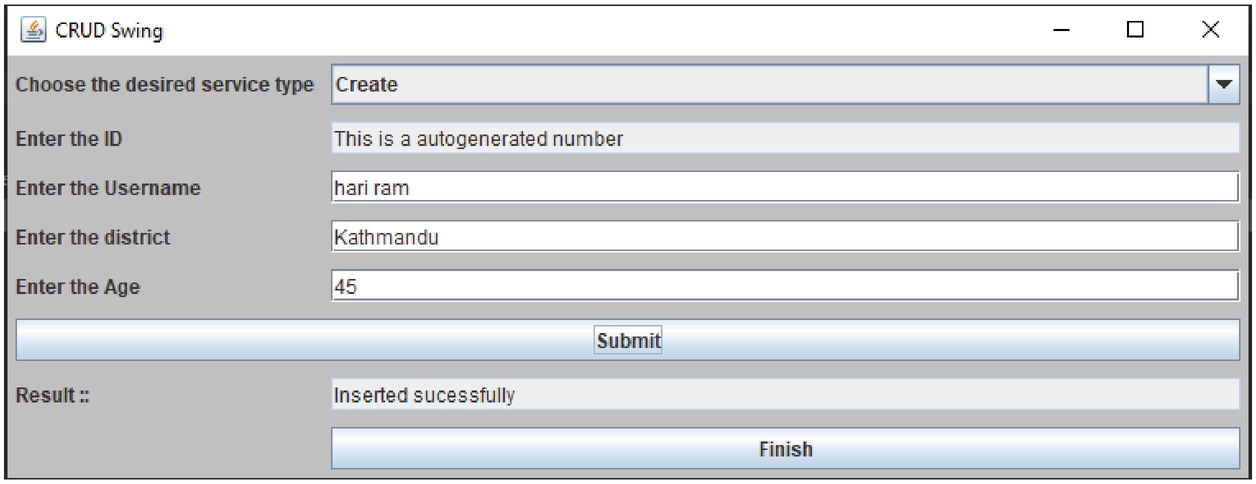
Aashish Shrestha.


+ Options

	ID	Name	District	Age
☐ Edit Copy Delete	1	aashish shreshta	Nuwakot	21
☐ Edit Copy Delete	2	Hari	Kathmandu	87
☐ Edit Copy Delete	3	Shyam	Kthmandu	45
☐ Edit Copy Delete	4	Hari	Kathmandu	87
☐ Edit Copy Delete	5	Shyam	Kthmandu	45
☐ Edit Copy Delete	6	ram	rasuwa	21

☐ Check all *With selected:* Edit Copy Delete Export

The output of the program while creating a user is:


CRUD Swing

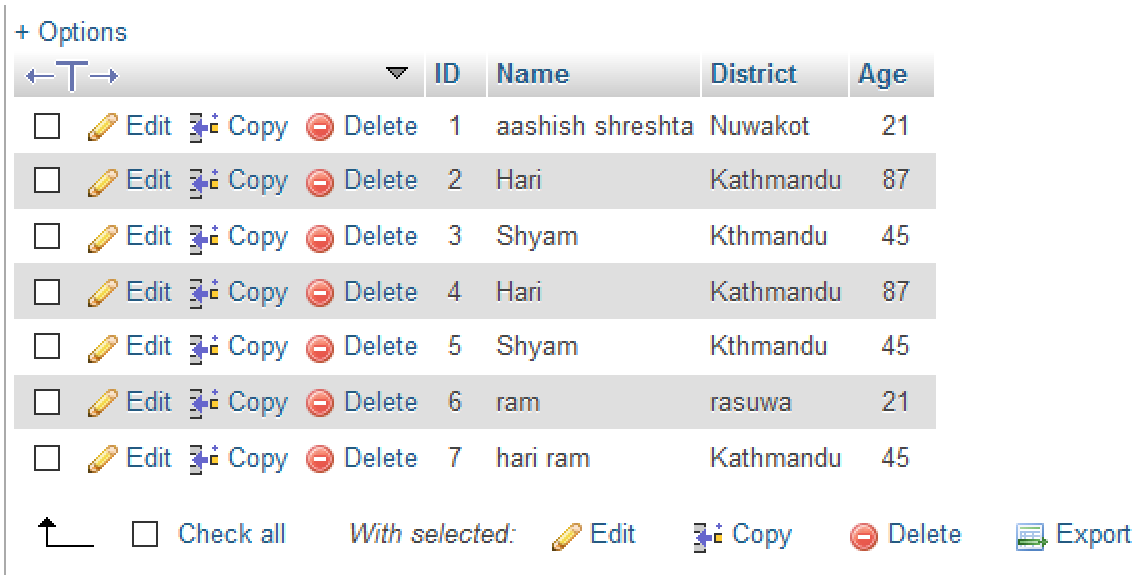
Choose the desired service type	Create
Enter the ID	This is a autogenerated number
Enter the Username	hari ram
Enter the district	Kathmandu
Enter the Age	45

Submit

Result ::	Inserted sucessfully

Finish

And the output after running the create operation program is:

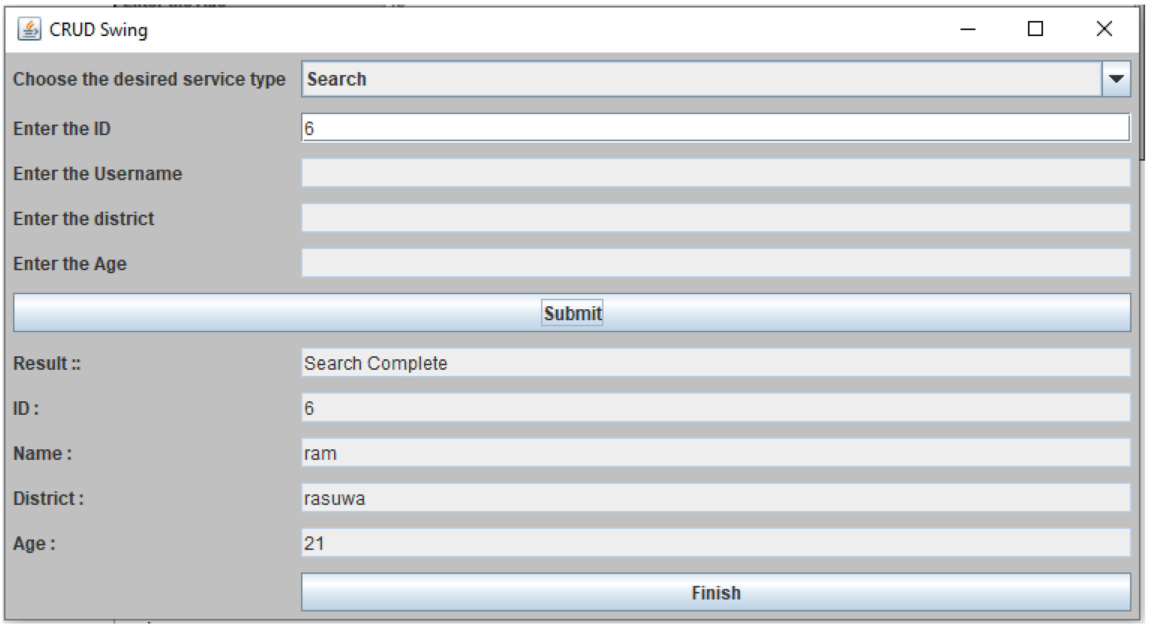

+ Options

	ID	Name	District	Age
☐ Edit Copy Delete	1	aashish shreshta	Nuwakot	21
☐ Edit Copy Delete	2	Hari	Kathmandu	87
☐ Edit Copy Delete	3	Shyam	Kthmandu	45
☐ Edit Copy Delete	4	Hari	Kathmandu	87
☐ Edit Copy Delete	5	Shyam	Kthmandu	45
☐ Edit Copy Delete	6	ram	rasuwa	21
☐ Edit Copy Delete	7	hari ram	Kathmandu	45

☐ Check all *With selected:* Edit Copy Delete Export

The output program while search is:

Aashish Shrestha.

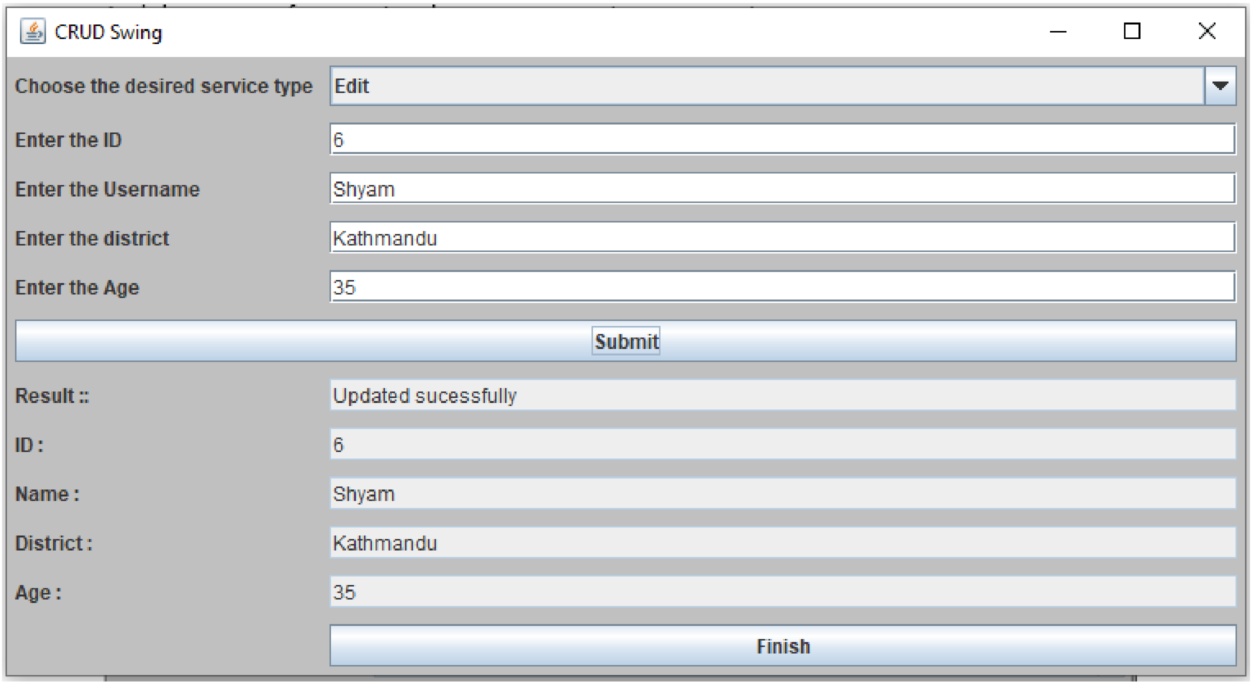


CRUD Swing

Choose the desired service type	Search
Enter the ID	6
Enter the Username	
Enter the district	
Enter the Age	

Result ::	Search Complete
ID :	6
Name :	ram
District :	rasuwa
Age :	21

The output of the Edit program is:



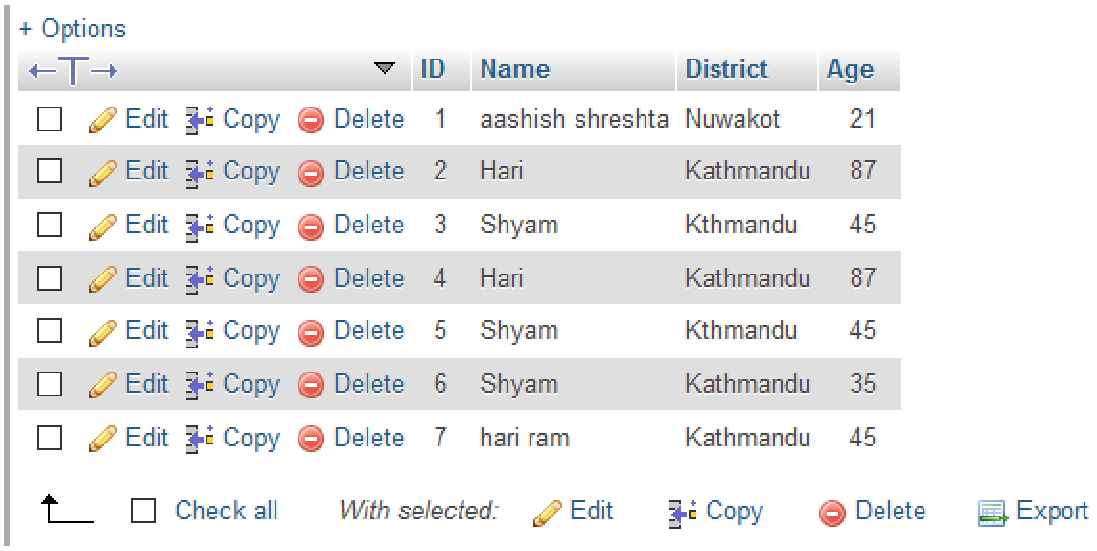
CRUD Swing

Choose the desired service type	Edit
Enter the ID	6
Enter the Username	Shyam
Enter the district	Kathmandu
Enter the Age	35

Result ::	Updated sucessfully
ID :	6
Name :	Shyam
District :	Kathmandu
Age :	35

Now that output in that database is:

Aashish Shrestha.

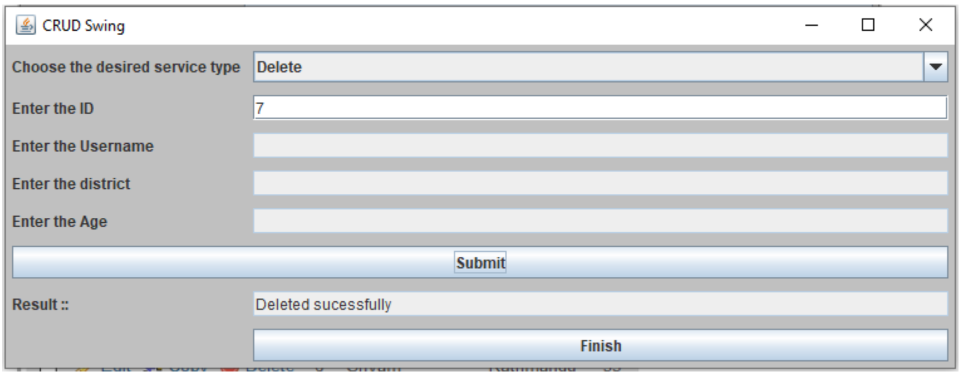


+ Options

	ID	Name	District	Age
☐ Edit Copy Delete	1	aashish shreshta	Nuwakot	21
☐ Edit Copy Delete	2	Hari	Kathmandu	87
☐ Edit Copy Delete	3	Shyam	Kthmandu	45
☐ Edit Copy Delete	4	Hari	Kathmandu	87
☐ Edit Copy Delete	5	Shyam	Kthmandu	45
☐ Edit Copy Delete	6	Shyam	Kathmandu	35
☐ Edit Copy Delete	7	hari ram	Kathmandu	45

☐ Check all *With selected:* Edit Copy Delete Export

Now the delete operation is:



CRUD Swing

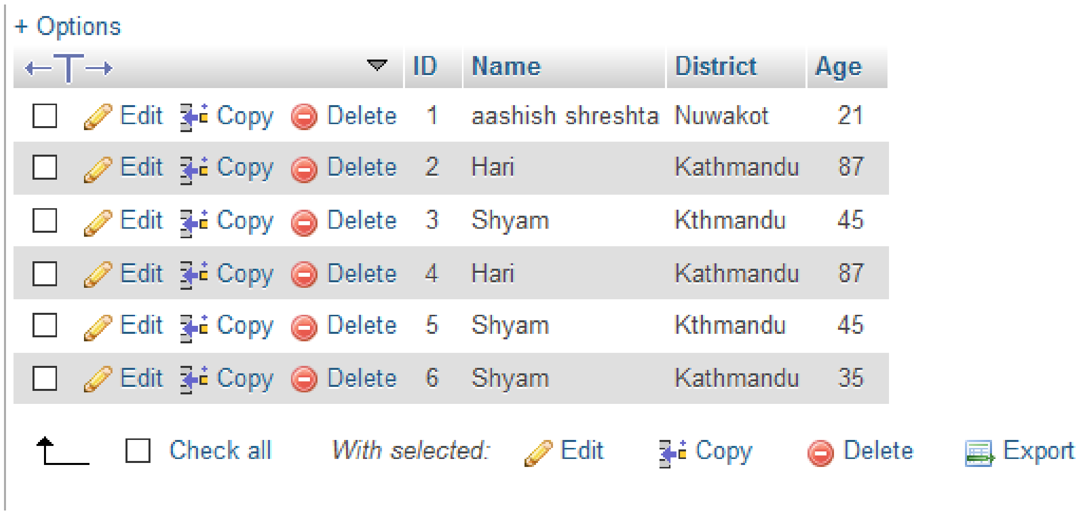
Choose the desired service type	Delete
Enter the ID	7
Enter the Username	
Enter the district	
Enter the Age	

Submit

Result ::	Deleted sucessfully

Finish

The change in database is:



+ Options

	ID	Name	District	Age
☐ Edit Copy Delete	1	aashish shreshta	Nuwakot	21
☐ Edit Copy Delete	2	Hari	Kathmandu	87
☐ Edit Copy Delete	3	Shyam	Kthmandu	45
☐ Edit Copy Delete	4	Hari	Kathmandu	87
☐ Edit Copy Delete	5	Shyam	Kthmandu	45
☐ Edit Copy Delete	6	Shyam	Kathmandu	35

☐ Check all *With selected:* Edit Copy Delete Export