

# Projekt code

## SQL-code

### CREATE database

```
create database rock_festival_g26;
```

### CREATE employee

```
create table employee(  
  person_nr int primary key,  
  name varchar (30) not null,  
  address varchar (40) not null);
```

### CREATE bands

```
create table bands(  
  band_name varchar(30) primary key,  
  county varchar (20) not null,  
  person_nr int references employee(person_nr));
```

### CREATE scene

```
create table scene(  
  scene_name varchar(30) primary key,  
  size varchar(6) not null);
```

### CREATE schedule

```
create table schedule(  
  band_name varchar(30) references bands(band_name),  
  scene_name varchar(30) references scene(scene_name),  
  start_time char(5) not null,  
  finish_time char(5) not null,  
  date varchar(20) not null);
```

### CREATE members

```
create table members  
  (member_id serial primary key,  
   member_name varchar(30) not null,  
   info varchar(200));
```

### CREATE band\_member

```
create table band_member  
  (band_name varchar(30) references bands(band_name),  
   member_id int references member(member_id));
```

### INSERT member

```
insert into member(member_name, info) values  
( 'Agnetha Fältskog', '' ),  
( 'Anni-frid Lyngstad', 'Performs solo and in the band Abba' ),  
( 'Benny Andersson', '' ),  
( 'Björn Ulvaeus', '' ),
```

```

('Bono', ''),
('The Edge', ''),
('Adam Clayton', ''),
('Larry Mullen jr', 'Performs during first act'),
('James Hetfield', ''),
('Lars Ulrich', ''),
('Kirk Hammett', ''),
('Robert Trujillo', 'Only in for the second half'),
('Mick Jagger', 'Information about performance coming shortly'),
('Keith Richards', ''),
('Row Wood', ''),
('Charlie Watts', 'Also doing an appearance with the Eagles'),
('Don Henley', ''),
('Joe Walsh', ''),
('Timothy B Schmitt', '');

```

### INSERT bands

```

insert into bands(band_name, country, person_nr) values
('Abba', 'Sweden', '910318-2334'),
('U2', 'Ireland', '901018-4556'),
('Metallica', 'USA', '880226-7283'),
('The rolling stones', 'England', '880226-7283'),
('The eagles', 'USA', '940127-9289');

```

### INSERT band\_member

```

insert into band_members(band_name, member_id) values
('Abba', '1'),
('Abba', '2'),
('Abba', '3'),
('Abba', '4'),
(U2, '5'),
(U2, '6'),
(U2, '7'),
(U2, '8'),
(Metallica, '9'),
(Metallica, '10'),
(Metallica, '11'),
(Metallica, '12'),
(The rolling stones, '13'),
(The rolling stones, '14'),
(The rolling stones, '15'),
(The rolling stones, '16'),
(The eagles, '17'),
(The eagles, '18'),
(The eagles, '19');

```

### INSERT employee

```

insert into employee(person_nr,name,address) values
('910318-2334', 'Amanda Eriksson', 'Amiralsgatan 1'),
('901018-4556', 'Ramy Behnam', 'Amiralsgatan 2'),
('880226-7283', 'Jonas Drewling', 'Amiralsgatan 3'),
('940127-9289', 'Elena Tayar', 'Amiralsgatan 4');

```

### INSERT scene

```

insert into scene(scene_name, size) values
('Mallorca Scene', 'Large'),

```

```
('Swedbank Stadion', 'Large'),
('Konserthuset', 'Large'),
('Opera', 'Large'),
('Malmö live', 'Large'),
('Folkets park', 'Medium'),
('Stortorget', 'Large'),
('Lilla torg', 'Small'),
('Gustav Adolfs Torg', 'Medium');
```

### Join

```
select bands.band_name, bands.country, members.member_name, members.info
from (( bands
inner join band_member on bands.band_name = band_member.band_name)
inner join members on band_member.member_name = member.member_name);
```

# Java code

## OfficeUser

```
package rockFestival.rockFestival;

import java.awt.EventQueue;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.border.LineBorder;
import java.sql.*;

public class officeUser extends JFrame {
    private JPanel contentPane;
    private JPanel pnlBook;
```

```

private JTextField txtBandName;
private JTextField txtCountry;
private JTextField txtPersonNr;
private JTextField txtBandName2;
private JTextField txtScene;
private JTextField txtStart;
private JTextField txtFinish;
private JTextField txtDate;

private JButton book = new JButton("Book bands");
private JButton assignInSchedule = new JButton("Book to schedule");

/**
 * Launch the application.
 */
public static void main(String[] args) throws SQLException {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                officeUser frame = new officeUser();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */

```

```

public officeUser() throws SQLException {
    assignInSchedule.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if(e.getSource()==assignInSchedule){
                try {
                    Class.forName("org.postgresql.Driver");
                    Connection c = (Connection)
DriverManager.getConnection("jdbc:postgresql://pgserver.mah.se/rock_festival_g
26", "m10p2724", "ogn810b8");

                    System.out.println("Database Opened successfully");

                    Statement stmt = c.createStatement();

                    String sql = "INSERT INTO schedule (band_name, scene_name,
start_time, finish_time, date) "
                        + "VALUES ('" + txtBandName.getText() + "','" +
txtScene.getText() + "','" + txtStart.getText() + "','" + txtFinish.getText() + "','"
+ txtDate.getText() + "');"
                    stmt.executeUpdate(sql);

                    stmt.close();
                    c.commit();
                    c.close();

                } catch (Exception ee) {
                    ee.printStackTrace();
                    System.exit(0);
                }
            }
        }
    });
    book.addActionListener(new ActionListener() {

        @Override

```

```

public void actionPerformed(ActionEvent e) {

    if (e.getSource() == book) {
        try {
            Class.forName("org.postgresql.Driver");
            Connection c = (Connection)
DriverManager.getConnection("jdbc:postgresql://pgserver.mah.se/rock_festival_g
26", "m10p2724", "ogn810b8");
            c.setAutoCommit(false);
            System.out.println("Database Opened successfully");

            Statement stmt = c.createStatement();
            String sql = ("INSERT INTO bands (band_name, country,
person_nr) VALUES ('" + txtBandName2.getText() + "', '" + txtCountry.getText() +
"', '" + txtPersonNr.getText() + "');");
            stmt.executeUpdate(sql);
            stmt.close();
            c.close();
        } catch (Exception e1) {
            System.out.println(e1);
        }
        System.out.println("Operation successfully");
    }
}

setTitle("Mörtfors Rock Festival - Office");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(80, 20, 1200, 2000);
contentPane = new JPanel();
contentPane.setBackground(Color.WHITE);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);

```

```
contentPane.setLayout(null);
```

```
//Titel label
```

```
JLabel lblTitel = new JLabel();
```

```
lblTitel.setForeground(new Color(178, 34, 34));
```

```
lblTitel.setFont(new Font("Helvetica Neue", Font.BOLD, 42));
```

```
lblTitel.setText("Mörtfors Rock Festival - Booking");
```

```
lblTitel.setBounds(290, 20, 860, 72);
```

```
ImageIcon logo = new ImageIcon("images/rockestival.png");
```

```
lblTitel.setIcon(logo);
```

```
contentPane.add(lblTitel);
```

```
//Booking panel/box
```

```
pnlBook = new JPanel();
```

```
pnlBook.setBackground(Color.WHITE);
```

```
pnlBook.setBorder(new LineBorder(Color.LIGHT_GRAY, 1, true));
```

```
pnlBook.setBounds(80, 110, 1200 - 85 - 85, 600);
```

```
contentPane.add(pnlBook);
```

```
pnlBook.setLayout(null);
```

```
//Booking bands
```

```
JLabel lblBand = new JLabel("Band:");
```

```
lblBand.setFont(new Font("Helvetica Neue", Font.BOLD, 18));
```

```
lblBand.setForeground(new Color(178, 34, 34));
```

```
lblBand.setBounds(20, 10, 200, 72);
```

```
pnlBook.add(lblBand);
```

```
JLabel lblCountry = new JLabel("Country:");
```

```
lblCountry.setFont(new Font("Helvetica Neue", Font.BOLD, 18));
```

```
lblCountry.setForeground(new Color(178, 34, 34));
```

```
lblCountry.setBounds(20, 150, 200, 72);
```

```
pnlBook.add(lblCountry);
```

```
JLabel lblPersonnel = new JLabel("Person Number:");  
lblPersonnel.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
lblPersonnel.setForeground(new Color(178, 34, 34));  
lblPersonnel.setBounds(20, 300, 200, 72);  
pnlBook.add(lblPersonnel);
```

```
//booking bands into schedule
```

```
JLabel lblBand2 = new JLabel("Band:");  
lblBand2.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
lblBand2.setForeground(new Color(178, 34, 34));  
lblBand2.setBounds(70 + 360, 10, 200, 72);  
pnlBook.add(lblBand2);
```

```
JLabel lblScene = new JLabel("Scene:");  
lblScene.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
lblScene.setForeground(new Color(178, 34, 34));  
lblScene.setBounds(70 + 360, 100, 200, 72);  
pnlBook.add(lblScene);
```

```
JLabel start = new JLabel("Start:");  
start.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
start.setForeground(new Color(178, 34, 34));  
start.setBounds(70 + 360, 180, 200, 72);  
pnlBook.add(start);
```

```
JLabel finish = new JLabel("Finish:");  
finish.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
finish.setForeground(new Color(178, 34, 34));  
finish.setBounds(70 + 360, 270, 200, 72);  
pnlBook.add(finish);
```



```
JLabel date = new JLabel("Date:");  
date.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
date.setForeground(new Color(178, 34, 34));  
date.setBounds(70 + 360, 350, 200, 72);  
pnlBook.add(date);
```

```
//textfields for booking bands
```

```
txtBandName2 = new JTextField();  
txtBandName2.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtBandName2.setForeground(new Color(178, 34, 34));  
txtBandName2.setBounds(20, 10 + 72, 160, 20);  
pnlBook.add(txtBandName2);
```

```
txtCountry = new JTextField();  
txtCountry.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtCountry.setForeground(new Color(178, 34, 34));  
txtCountry.setBounds(20, 150+72, 160, 20);  
pnlBook.add(txtCountry);
```

```
txtPersonNr = new JTextField();  
txtPersonNr.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtPersonNr.setForeground(new Color(178, 34, 34));  
txtPersonNr.setBounds(20, 300 +72, 160, 20);  
pnlBook.add(txtPersonNr);
```

```
//textfields for adding bands in schedule
```

```
txtBandName = new JTextField();  
txtBandName.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtBandName.setForeground(new Color(178, 34, 34));  
txtBandName.setBounds(70 + 360, 10 + 50, 160, 20);  
pnlBook.add(txtBandName);
```

```
txtScene = new JTextField();  
txtScene.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtScene.setForeground(new Color(178, 34, 34));  
txtScene.setBounds(70 + 360, 100+50, 160, 20);  
pnlBook.add(txtScene);
```

```
txtStart = new JTextField();  
txtStart.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtStart.setForeground(new Color(178, 34, 34));  
txtStart.setBounds(70 + 360, 150 +80, 160, 20);  
pnlBook.add(txtStart);
```

```
txtFinish = new JTextField();  
txtFinish.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtFinish.setForeground(new Color(178, 34, 34));  
txtFinish.setBounds(70 + 360, 260+60, 160, 20);  
pnlBook.add(txtFinish);
```

```
txtDate = new JTextField();  
txtDate.setFont(new Font("Helvetica Neue", Font.BOLD, 18));  
txtDate.setForeground(new Color(178, 34, 34));  
txtDate.setBounds(70 + 360, 340 +60, 160, 20);  
pnlBook.add(txtDate);
```

```
//for two buttons  
book.setBounds(20, 500, 200, 20);  
pnlBook.add(book);  
assignInSchedule.setBounds(70 + 360 , 500, 200, 20);  
pnlBook.add(assignInSchedule);
```

```
}
```

```
}
```

## **visitorUser**

```
package rockFestival;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.JTable;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JTextArea;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.awt.event.ActionEvent;

public class viewerUser extends JFrame {

    //    JPanel
    private JPanel contentPane;
    private JPanel pnlBand = new JPanel();
    private JPanel pnlMember = new JPanel();
```

```
//JTextArea

    JTextArea taMemberInfo = new JTextArea();
    JTextArea taBandMembers = new JTextArea();

//    JLabes
    private JLabel lblSchedule = new JLabel("Schedule");
    private JLabel lblBands = new JLabel("Bands");
    private JLabel lblMember = new JLabel("Member");
    private JLabel lblTitel = new JLabel();
    private JLabel lblBName = new JLabel("Band name:");
    private JLabel lblBCountry = new JLabel("Country:");
    private JLabel lblBMembers = new JLabel("Members:");
    private JLabel lblEnterBandName = new JLabel("Enter band name");
    private JLabel lblBandName = new JLabel("");
    private JLabel lblBandCountry = new JLabel("");
    private JLabel lblEnterMemberName = new JLabel("Enter member name");
    private JLabel lblMName = new JLabel("Member name:");
    private JLabel lblMInfo = new JLabel("Info:");
    private JLabel lblMemberName = new JLabel("");

//    JTable
    private JTable scheduleTable;

//    JButtons
    JButton btnBandSearch = new JButton("Search");
    JButton btnMemberSearch = new JButton("Search");

// JTextField
    private JTextField txtBandName;
    private JTextField txtMemberName;
```

```

public viewerUser() {
    initComponents();
    showSchedule();
}

```

```

public ArrayList<Schedule> scheduleList(){
    ArrayList<Schedule> scheduleList = new ArrayList<>();
    try {
        Class.forName("org.postgresql.Driver");
        Connection c = (Connection)
DriverManager.getConnection("jdbc:postgresql://pgserver.mah.se/rock_festival_g
26", "m10p2724", "ogn810b8");
        System.out.println("Database Opened successfylt");
        Statement stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery("select b.band_name,
b.country, s.scene_name, s.start_time, s.finish_time, s.date\n" +
            "from bands as b\n" +
            "inner join schedule as s on b.band_name =
s.band_name;");
        Schedule schedule;
        while (rs.next()) {
            schedule = new Schedule(rs.getString("band_name"),
rs.getString("country"), rs.getString("scene_name"), rs.getString("start_time"),
rs.getString("finish_time"), rs.getString("date"));
            scheduleList.add(schedule);
        }
        rs.close();
        stmt.close();
        c.close();
    }catch (Exception e1) {
        System.out.println(e1);
    }
}

```

```

        System.out.println("Operation successfully");
        return scheduleList;
    }

    public void showSchedule() {
        ArrayList<Schedule> sList = scheduleList();
        DefaultTableModel model =
(DefaultTableModel)scheduleTable.getModel();
        model.setColumnCount(6);
        Object[] row = new Object[6];
        for(int i=0; i<sList.size(); i++) {
            row[0] = sList.get(i).getBand();
            row[1] = sList.get(i).getCountry();
            row[2] = sList.get(i).getScene();
            row[3] = sList.get(i).getStart();
            row[4] = sList.get(i).getFinish();
            row[5] = sList.get(i).getDate();
            model.addRow(row);
        }
    }

    /**
     * Create the frame.
     */
    public void initComponents() {
        setTitle("Mörtfors Rock Festival");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 1200, 1000);
        contentPane = new JPanel();
        contentPane.setBackground(Color.WHITE);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
    }

```

```
contentPane.setLayout(null);

scheduleTable = new JTable();
scheduleTable.setBorder(new LineBorder(Color.LIGHT_GRAY, 1,
true));

scheduleTable.setToolTipText("gsdgsegsegse");
scheduleTable.setBackground(Color.WHITE);
scheduleTable.setBounds(400, 180, 768, 766);
contentPane.add(scheduleTable);

lblSchedule.setForeground(new Color(178, 34, 34));
lblSchedule.setFont(new Font("Helvetica Neue", Font.BOLD, 22));
lblSchedule.setBackground(Color.WHITE);
lblSchedule.setBounds(750, 145, 100, 16);
contentPane.add(lblSchedule);

lblBands.setForeground(new Color(178, 34, 34));
lblBands.setFont(new Font("Helvetica Neue", Font.BOLD, 22));
lblBands.setBackground(Color.WHITE);
lblBands.setBounds(165, 145, 70, 16);
contentPane.add(lblBands);

lblMember.setForeground(new Color(178, 34, 34));
lblMember.setFont(new Font("Helvetica Neue", Font.BOLD, 22));
lblMember.setBackground(Color.WHITE);
lblMember.setBounds(155, 570, 90, 16);
contentPane.add(lblMember);

lblTitel.setForeground(new Color(178, 34, 34));
lblTitel.setFont(new Font("Helvetica Neue", Font.BOLD, 42));
lblTitel.setText("Mörtfors Rock Festival");
lblTitel.setBounds(370, 40, 460, 72);
```

```
contentPane.add(lblTitel);
```

```
pnlBand.setBackground(Color.WHITE);  
pnlBand.setBorder(new LineBorder(Color.LIGHT_GRAY, 1, true));  
pnlBand.setBounds(32, 180, 336, 358);  
contentPane.add(pnlBand);  
pnlBand.setLayout(null);
```

```
txtBandName = new JTextField();  
txtBandName.setToolTipText("");  
txtBandName.setForeground(Color.BLACK);  
txtBandName.setBounds(32, 50, 188, 45);  
pnlBand.add(txtBandName);  
txtBandName.setFont(new Font("Helvetica Neue", Font.PLAIN, 13));  
txtBandName.setColumns(10);
```

```
lblBName.setFont(new Font("Helvetica Neue", Font.BOLD, 13));  
lblBName.setForeground(new Color(178, 34, 34));  
lblBName.setBounds(35, 115, 80, 16);  
pnlBand.add(lblBName);
```

```
lblBCountry.setFont(new Font("Helvetica Neue", Font.BOLD, 13));  
lblBCountry.setForeground(new Color(178, 34, 34));  
lblBCountry.setBounds(35, 155, 61, 16);  
pnlBand.add(lblBCountry);
```

```
lblBMembers.setFont(new Font("Helvetica Neue", Font.BOLD, 13));  
lblBMembers.setForeground(new Color(178, 34, 34));  
lblBMembers.setBounds(35, 195, 65, 16);  
pnlBand.add(lblBMembers);
```



```

12));

lblEnterBandName.setFont(new Font("Helvetica Neue", Font.ITALIC,

lblEnterBandName.setForeground(Color.LIGHT_GRAY);
lblEnterBandName.setBounds(42, 32, 110, 16);
pnlBand.add(lblEnterBandName);


lblBandName.setBounds(127, 115, 175, 18);
pnlBand.add(lblBandName);


lblBandCountry.setBounds(108, 155, 195, 16);
pnlBand.add(lblBandCountry);


btnBandSearch.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        taBandMembers.setText("");

        String members;
        try {
            Class.forName("org.postgresql.Driver");
            Connection c = (Connection)
DriverManager.getConnection("jdbc:postgresql://pgserver.mah.se/rock_festival_g
26", "m10p2724", "ogn810b8");
            System.out.println("Database Opened
successfylt");

            Statement stmt = c.createStatement();
            ResultSet rs = stmt.executeQuery("select
b.band_name, b.country, m.member_name "
                                + "from bands as b "
                                + "inner join band_member as bm
on b.band_name = bm.band_name "
                                + "inner join member as m on
bm.member_id = m.member_id "
                                + "where b.band_name =
""+txtBandName.getText()+"");

```

```

        while (rs.next()) {
            String bandName =
rs.getString("band_name");

            String country= rs.getString("country");
            members =
rs.getString("member_name");

            lblBandName.setText(bandName);
            lblBandCountry.setText(country);
            taBandMembers.append(members
+"\\n" );

        }

        rs.close();
        stmt.close();
        c.close();
    }catch (Exception e1) {
        System.out.println(e1);
    }
    System.out.println("Operation successfully");
}

});
btnBandSearch.setFont(new Font("Helvetica Neue", Font.BOLD,
13));

btnBandSearch.setBounds(232, 50, 72, 45);
pnlBand.add(btnBandSearch);

taBandMembers = new JTextArea(5,20);
taBandMembers.setTabSize(100);
taBandMembers.setLineWrap(true);
taBandMembers.setFont(new Font("Helvetica Neue", Font.PLAIN,
13));

taBandMembers.setEditable(false);

```

```
taBandMembers.setBounds(35, 226, 266, 103);  
pnlBand.add(taBandMembers);
```

```
pnlMember.setBackground(Color.WHITE);  
pnlMember.setBorder(new LineBorder(Color.LIGHT_GRAY, 1, true));  
pnlMember.setLayout(null);  
pnlMember.setBounds(32, 606, 336, 340);  
contentPane.add(pnlMember);
```

```
txtMemberName = new JTextField();  
txtMemberName.setForeground(Color.BLACK);  
txtMemberName.setFont(new Font("Helvetica Neue", Font.PLAIN,  
13));  
txtMemberName.setColumns(10);  
txtMemberName.setBounds(34, 50, 188, 45);  
pnlMember.add(txtMemberName);
```

```
lblEnterMemberName.setForeground(Color.LIGHT_GRAY);  
lblEnterMemberName.setFont(new Font("Helvetica Neue",  
Font.ITALIC, 12));  
lblEnterMemberName.setBounds(42, 32, 120, 16);  
pnlMember.add(lblEnterMemberName);
```

```
lblMName.setForeground(new Color(178, 34, 34));  
lblMName.setFont(new Font("Helvetica Neue", Font.BOLD, 13));  
lblMName.setBounds(35, 115, 95, 16);  
pnlMember.add(lblMName);
```

```
lblMInfo.setForeground(new Color(178, 34, 34));  
lblMInfo.setFont(new Font("Helvetica Neue", Font.BOLD, 13));  
lblMInfo.setBounds(35, 155, 65, 16);  
pnlMember.add(lblMInfo);
```

```

lblMemberName.setBounds(142, 114, 162, 18);
pnlMember.add(lblMemberName);
btnMemberSearch.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            Class.forName("org.postgresql.Driver");
            Connection c = (Connection)
DriverManager.getConnection("jdbc:postgresql://pgserver.mah.se/rock_festival_g
26", "m10p2724", "ogn810b8");
            System.out.println("Database Opened
successfylt");

            Statement stmt = c.createStatement();
            ResultSet rs = stmt.executeQuery("select * from
member where member_name = '"+txtMemberName.getText()+"'");

            while (rs.next()) {
                String memberName =
rs.getString("member_name");

                String info= rs.getString("info");

                lblMemberName.setText(memberName);
                taMemberInfo.setText(info);

            }
            rs.close();
            stmt.close();
            c.close();
        } catch (Exception e1) {
            System.out.println(e1);
        }
        System.out.println("Operation successfully");
    }
}

```

```

        });

        btnMemberSearch.setFont(new Font("Helvetica Neue", Font.BOLD,
13));

        btnMemberSearch.setBounds(232, 50, 72, 45);
        pnlMember.add(btnMemberSearch);
        taMemberInfo.setLineWrap(true);
        taMemberInfo.setEditable(false);
        taMemberInfo.setBounds(35, 187, 266, 121);
        pnlMember.add(taMemberInfo);
    }
}

```

## **Schedule**

```

package rockFestival;

public class Schedule {

    private String band, country, scene, start, finish, date;

    public Schedule(String band, String country, String scene, String start,
String finish, String date) {
        this.band = band;
        this.country = country;
        this.scene = scene;
        this.start = start;
        this.finish = finish;
        this.date = date;
    }

    public String getBand () {

```

```
        return band;
    }

    public String getCountry () {
        return country;
    }

    public String getScene () {
        return scene;
    }

    public String getStart () {
        return start;
    }

    public String getFinish() {
        return finish;
    }

    public String getDate() {
        return date;
    }

}
```