

2019 年秋季学期 计算机学院大三 计算机系统安全课程

Lab 3 实验报告

姓名	冯帅	
学号	1170301027	
班号	1703201	
电子邮件	1170301027@stu.hit.edu.cn	
手机号码	手机号码 15765513201	

一、实验环境:

Win10

二、实验工具:

Intellij IDEA, MySQL, Navicat

三、编程语言:

JAVA

四、实验要求:

1. 实验目的

熟练掌握数据库(比如 MySQL)基本权限管理命令、SQL 语言以及学习数据库系统的设计、数据库用户权限管理的实现,包括特定场景下地数据表创建和管理、数据库用户的创建和**合理的**权限分配,<u>权限分配细化到数据库、表、列和</u>行,或者视图。

学生自行设计应用场景(应用场景不要和实验指导书示例相同),为具体的应用需求建立数据库,比如产品销售、人口管理、医院、银行、股票、手机通信信息等的数据库。并为你所面对的应用需求进行用户分类和权限划分,要求权限设计<mark>合理</mark>,能够实现依据用户权限的分发、权限的收回,并能够成组的批量分发和收回权限。

系统要求:

- (1)系统自建操作界面,能够实现单条、批量权限的查询、分发和收回等操作。(利用数据库系统自带的界面进行演示,视为不合格)
 - (2) 测试权限设计的合理性。
 - (3)管理操作日志。

五、实验步骤:

1 一个简单的应用场景

- 1. 商品信息,如 id、name、price、number等。
- 2. 雇员信息,如 id、name、birthday、age 等。

- 3. 经理信息,如id、name、birthday、age等。
- 4. 老板信息, 如 id, name 等;
- 5. Boss: 等同于 root 的操作权限
- 6. Manager: 可以修改 employee 表内容, 具有 good 表的所有权限
- 7. Employee: 查看整个 good 表但仅能修改其中的 number

2 建立数据库

建立上述数据库表单,把老板、经理、雇员、作为可以登录数据库的不同的用户。

其中,老板相当于管理员,具有所有权限,可以添加删除用户,并给普通员工升级为经理或将经理降级为普通员工,经理具有查看与修改雇员及商品信息的权限,雇员只能修改查看商品信息和修改其中的价格。

添加员工时可以需要在数据库中创建用户并在相应的表中插入数据,修改商品信息时只需操作相应 good 表即可,雇员负责销售商品所以只可以修改 good 的 number 无法修改其他信息

登陆时根据登陆 id 鉴别身份,进入不同管理界面

日志采用 logger 和 filehandle 进行记录

3 实验结果

- 1. 操作界面:
 - 登陆界面:

用户名
密码
登陆

● Boss 界面:可以管理员工,经理,商品



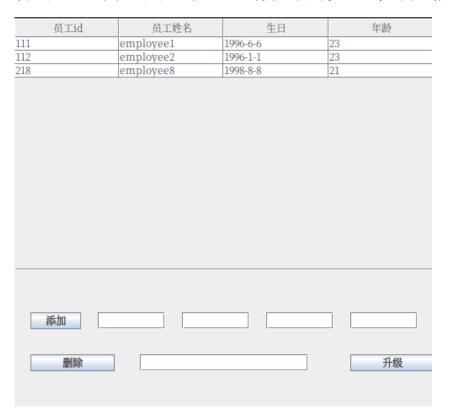
● 经理界面:可管理员工,商品



● 员工界面: 仅能修改商品的数量

商品id	商品名称	价格	数量
1111	apple	1	10
1112	banana	2	15
1113	book	10	20
1114	water	1	100
1115	cake	8	10
修改			

● 管理员工: 可以添加员工, 批量删除, 升级为经理(id 用空格个库)



● 管理经理: 可以添加经理, 批量删除, 降级为普通员工(id 用空格个库)

经理id	经理姓名	生日	年龄
11	managerl	1995-10-2	24
12	manager2	1994-5-5	25
添加			
mirA			BA Acre
删除			降权

● 经理管理员工:只能修改员工信息



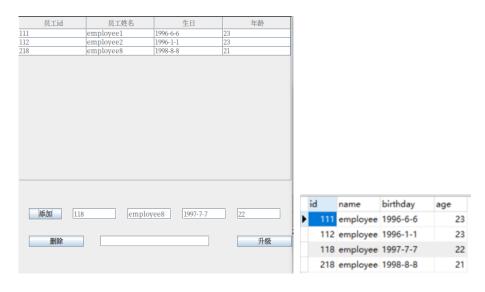
● 经理,老板,管理商品:可以添加删除商品



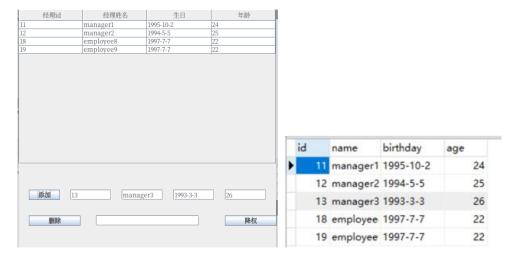
● 员工管理商品:由于员工只能管理商品,即为员工界面

2. 运行结果:

● 添加员工:



● 添加经理:



age

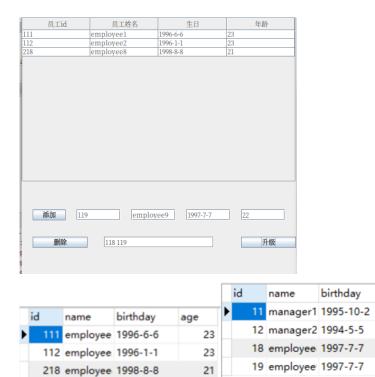
24

25

22

22

● 员工升级:



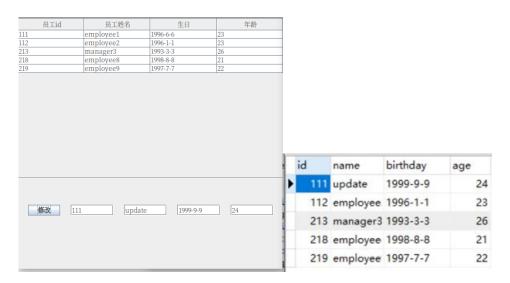
● 经理降级:





	id	name	birthday	age
•	111	employee	1996-6-6	23
	112	employee	1996-1-1	23
	213	manager3	1993-3-3	26
	218	employee	1998-8-8	21
	219	employee	1997-7-7	22

● 修改员工信息:



● 添加商品:

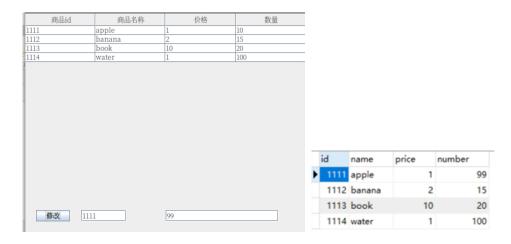


● 删除商品:

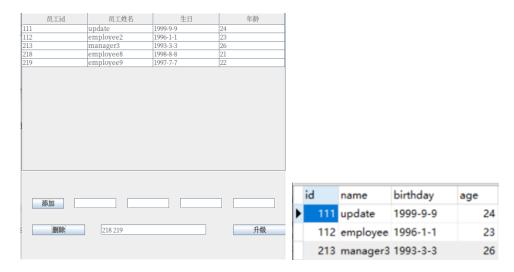


	id	name	price	number
١	1111	apple	1	10
	1112	banana	2	15
	1113	book	10	20
	1114	water	1	100

● 更改商品数量:



● 删除员工



3. 具体操作介绍:

操作使用 java 通过 jdbc 连接数据库

- 添加员工与经理: boos 具有 root 权限,添加人员时首先在数据库中根据 id 创建用户,并在 manager 或 employee 中添加人员信息,并根据身份设置权限
- 删除员工:将输入的字符串用空格分开获取每个人的 id, 删除用户及表中用户信息。
- 升级或降权:由于用户级别和 id 有关,所以需要修改用户名称,并在将原来表中用户信息删除添加到新的表中
- 删除商品: 直接删除 good 表中信息即可,只有经理和老板具有权限
- 添加商品:在 good 表中插入信息,只有经理和老板具有权限
- 修改商品数量:普通员工只有能修改 good 中的 number,所以很安全
- 修改普通员工信息: 经理具有 employee 表的 update 权限进行修改

六、实验源码:

```
public class Login extends JFrame {
    String JDBC DRIVER = "com.mysql.cj.jdbc.Driver";
    String
                                             DB URL
"jdbc:mysql://localhost:3306/lab3?useSSL=false&serverTimezone=UTC";
    Connection conn = null;
    Statement stmt = null;
    Statement stmt1 = null;
    Statement stmt2 = null;
    JPanel user = new JPanel();
    JPanel passWord = new JPanel();
    JButton getButton = new JButton("登陆");
    JTextField userField = new JTextField(10);
    JTextField passField = new JTextField(10);
    String name;
    String passwd;
    Logger logger = Logger.getLogger("lab3");
    FileHandler fileHandler = new FileHandler("lab3.log");
    public Login() throws IOException {
         fileHandler.setLevel(Level.INFO);
         fileHandler.setFormatter(new Formatter() {
              @Override
              public String format(LogRecord record) {
                  // TODO Auto-generated method stub
                  return record.getMessage() + "\n";
              }
         });
         logger.setUseParentHandlers(false);
         logger.addHandler(fileHandler);
         setLayout(null);
         setJPanel(user, "用户名");
         user.add(userField);
         setJPanel(passWord, "密码
                                      ");
         passWord.add(passField);
         user.setBounds(150,50,200,50);
```

```
passWord.setBounds(150, 100, 200, 50);
         getButton.setBounds(150, 150, 200, 20);
         this.add(user);
         this.add(passWord);
         this.add(getButton);
         this.setTitle("Login");
         this.setSize(500, 300);
         this.setLocation(200, 120);
         this.setDefaultCloseOperation(JFrame.EXIT ON CLOSE);
         this.setVisible(true);
         getButton.addActionListener(event -> {
              name = userField.getText();
              passwd = passField.getText();
              try {
                   login(name, passwd);
              } catch (ClassNotFoundException e) {
                   e.printStackTrace();
              } catch (SQLException e) {
                   e.printStackTrace();
              if (name.length() \le 1) {
                   bossGui();
              \} else if (name.length() <= 2) {
                   manageGui();
              } else {
                   try {
                       employeeGui();
                   } catch (SQLException e) {
                       e.printStackTrace();
         });
    private void setJPanel(JPanel panel, String name) {
         // TODO Auto-generated method stub
         Font font = new Font("宋体", Font.PLAIN, 22);
         JLabel label = new JLabel(name);
         label.setFont(font);
         panel.add(label);
    public void login(String name, String passwd) throws ClassNotFoundException,
SQLException {
```

```
Class.forName(JDBC DRIVER);
    conn = DriverManager.getConnection(DB URL,name,passwd);
    stmt = conn.createStatement();
    stmt1 = conn.createStatement();
    stmt2 = conn.createStatement();
    logger.info("login " + name);
}
public void bossGui() {
    JFrame | Frame = new JFrame();
    ¡Frame.setLayout(null);
    JButton employeeButton = new JButton();
    employeeButton.setBounds(110,20,100,20);
    employeeButton.setText("管理雇员");
    JButton managerButton = new JButton();
    managerButton.setBounds(110,70,100,20);
    managerButton.setText("管理经理");
    JButton goodButton = new JButton();
    goodButton.setBounds(110,120,100,20);
    goodButton.setText("管理商品");
    ¡Frame.add(employeeButton);
    jFrame.add(managerButton);
    jFrame.add(goodButton);
    jFrame.setBounds(500,500,300, 200);
    ¡Frame.setVisible(true);
    employeeButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
             try {
                  manageEmployee();
              } catch (SQLException e) {
                  e.printStackTrace();
              }
    });
    managerButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
             try {
                  manageManager();
              } catch (SQLException e) {
                  e.printStackTrace();
```

```
}
    });
    goodButton.addActionListener(new ActionListener() {
          @Override
         public void actionPerformed(ActionEvent actionEvent) {
              try {
                   manageGood();
              } catch (SQLException e) {
                   e.printStackTrace();
          }
    });
}
public void employeeGui() throws SQLException {
    JFrame frame = new JFrame();
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"商品 id","商品名称","价格","数量"});
    String sql = "SELECT * FROM good ";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String birthday = rs.getString("price");
         int age = rs.getInt("number");
         model.addRow(new Object[]{id,name, birthday, age});
    JTable fk = new JTable(model);
    JScrollPane jsp = new JScrollPane(fk);
    jsp.setBounds(50, 10, 500, 250);
    JTextField idFiled = new JTextField(10);
    JTextField numberFiled = new JTextField(10);
    JButton modifyButton = new JButton();
    modifyButton.setBounds(20, 350,60,20);
    modifyButton.setText("修改");
    idFiled.setBounds(100, 350, 80, 20);
    numberFiled.setBounds(250, 350, 200, 20);
    frame.add(modifyButton);
    frame.add(idFiled);
    frame.add(numberFiled);
    frame.setBounds(100, 100, 500, 500);
    frame.add(jsp);
    frame.setVisible(true);
    modifyButton.addActionListener(new ActionListener() {
```

```
@Override
         public void actionPerformed(ActionEvent actionEvent) {
              String id = idFiled.getText();
              String number = numberFiled.getText();
              try {
                   updateGood(id, number);
                   logger.info(name + "update good " + id + " " + number);
              } catch (SQLException e) {
                   e.printStackTrace();
    });
}
public void manageGui() {
    JFrame jFrame = new JFrame();
    jFrame.setLayout(null);
    JButton employeeButton = new JButton();
    employeeButton.setBounds(110,20,100,20);
    employeeButton.setText("管理雇员");
    JButton goodButton = new JButton();
    goodButton.setBounds(110,120,100,20);
    goodButton.setText("管理商品");
    jFrame.add(employeeButton);
    ¡Frame.add(goodButton);
    ¡Frame.setBounds(500,500,300, 200);
    jFrame.setVisible(true);
    employeeButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
              try {
                   managerManageEmployee();
              } catch (SQLException e) {
                  e.printStackTrace();
    });
    goodButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
              try {
                  manageGood();
```

```
} catch (SQLException e) {
                   e.printStackTrace();
    });
}
public void manageEmployee() throws SQLException {
    JFrame frame = new JFrame();
    frame.setLayout(null);
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"员工 id","员工姓名","生日","年龄"});
    String sql = "SELECT * FROM employee ";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String birthday = rs.getString("birthday");
         int age = rs.getInt("age");
         model.addRow(new Object[]{id,name, birthday, age});
    JTable fk = new JTable(model);
    JScrollPane jsp = new JScrollPane(fk);
    jsp.setBounds(0, 0, 500, 300);
    JTextField idFiled = new JTextField(10);
    JTextField nameFiled = new JTextField(10);
    JTextField birthdayFiled = new JTextField(10);
    JTextField ageFiled = new JTextField(5);
    JTextField removeFiled = new JTextField(40);
    JButton addButton = new JButton();
    addButton.setBounds(20, 350,60,20);
    addButton.setText("添加");
    JButton removeButton = new JButton();
    removeButton.setText("删除");
    removeButton.setBounds(20,400,100, 20);
    idFiled.setBounds(100, 350, 80, 20);
    nameFiled.setBounds(200, 350, 80, 20);
    birthdayFiled.setBounds(300, 350, 80, 20);
    ageFiled.setBounds(400, 350, 80, 20);
    removeFiled.setBounds(150, 400, 200, 20);
    frame.add(addButton);
    frame.add(idFiled);
    frame.add(nameFiled);
    frame.add(birthdayFiled);
```

```
frame.add(ageFiled);
frame.add(removeButton);
frame.add(removeFiled);
frame.setBounds(100, 100, 500, 500);
frame.add(jsp);
frame.setVisible(true);
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         String id = idFiled.getText();
         String addname = nameFiled.getText();
         String birthday = birthdayFiled.getText();
         String age = ageFiled.getText();
         try {
              addEmployee(id, addname, birthday, age);
              logger.info(name + "add employee " + id);
         } catch (SQLException e) {
              e.printStackTrace();
         }
     }
});
removeButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         try {
              deletePeople(removeFiled.getText(), "employee");
              logger.info(name + "delete employee " + removeFiled.getText());
         } catch (SQLException e) {
              e.printStackTrace();
         }
     }
});
JButton upButton = new JButton();
upButton.setBounds(400, 400, 100, 20);
upButton.setText("升级");
frame.add(upButton);
upButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         try {
              upQuanxian(removeFiled.getText());
              logger.info(name + " up level " + removeFiled.getText());
         } catch (SQLException e) {
              e.printStackTrace();
```

```
}
    });
public void manageManager() throws SQLException {
    JFrame frame = new JFrame();
    frame.setLayout(null);
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"经理 id","经理姓名","生日","年龄"});
    String sql = "SELECT * FROM manager ";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String birthday = rs.getString("birthday");
         int age = rs.getInt("age");
         model.addRow(new Object[]{id,name, birthday, age});
    }
    JTable fk = new JTable(model);
    JScrollPane jsp = new JScrollPane(fk);
    jsp.setBounds(00, 0, 500, 300);
    JTextField idFiled = new JTextField(10);
    JTextField nameFiled = new JTextField(10);
    JTextField birthdayFiled = new JTextField(10);
    JTextField ageFiled = new JTextField(5);
    JTextField removeFiled = new JTextField(40);
    JButton addButton = new JButton();
    addButton.setBounds(20, 350,60,20);
    addButton.setText("添加");
    JButton removeButton = new JButton();
    removeButton.setText("删除");
    removeButton.setBounds(20,400,100, 20);
    idFiled.setBounds(100, 350, 80, 20);
    nameFiled.setBounds(200, 350, 80, 20);
    birthdayFiled.setBounds(300, 350, 80, 20);
    ageFiled.setBounds(400, 350, 80, 20);
    removeFiled.setBounds(150, 400, 200, 20);
    frame.add(addButton);
    frame.add(idFiled);
    frame.add(nameFiled);
    frame.add(birthdayFiled);
    frame.add(ageFiled);
    frame.add(removeButton);
```

```
frame.add(removeFiled);
frame.setBounds(100, 100, 500, 500);
frame.add(jsp);
frame.setVisible(true);
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         String id = idFiled.getText();
         String addname = nameFiled.getText();
         String birthday = birthdayFiled.getText();
         String age = ageFiled.getText();
         try {
              addManager(id, addname, birthday, age);
              logger.info(name + " add manager " + id);
         } catch (SQLException e) {
              e.printStackTrace();
    }
});
removeButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         try {
              deletePeople(removeFiled.getText(), "manager");
              logger.info(name + " delete manager " + removeFiled.getText());
         } catch (SQLException e) {
              e.printStackTrace();
     }
});
JButton downButton = new JButton();
downButton.setBounds(400, 400, 100, 20);
downButton.setText("降权");
frame.add(downButton);
downButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
         try {
              deleteQx(removeFiled.getText());
              logger.info(name + " down level " + removeFiled.getText());
         } catch (SQLException e) {
              e.printStackTrace();
         }
```

```
});
}
public void manageGood() throws SQLException {
    JFrame frame = new JFrame();
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"商品 id","商品名称","价格","数量"});
    String sql = "SELECT * FROM good ";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         int price = rs.getInt("price");
         int number = rs.getInt("number");
         model.addRow(new Object[]{id,name, price, number});
    JTable fk = new JTable(model);
    JScrollPane jsp = new JScrollPane(fk);
    jsp.setBounds(50, 10, 500, 250);
    JTextField idFiled = new JTextField(10);
    JTextField nameFiled = new JTextField(10);
    JTextField priceFiled = new JTextField(10);
    JTextField numberFiled = new JTextField(5);
    JTextField removeFiled = new JTextField(40);
    JButton addButton = new JButton();
    addButton.setBounds(20, 350,60,20);
    addButton.setText("添加");
    JButton removeButton = new JButton();
    removeButton.setText("删除");
    removeButton.setBounds(20,400,100, 20);
    idFiled.setBounds(100, 350, 80, 20);
    nameFiled.setBounds(200, 350, 80, 20);
    priceFiled.setBounds(300, 350, 80, 20);
    numberFiled.setBounds(400, 350, 80, 20);
    removeFiled.setBounds(150, 400, 200, 20);
    frame.add(addButton);
    frame.add(idFiled);
    frame.add(nameFiled);
    frame.add(priceFiled);
    frame.add(numberFiled);
    frame.add(removeButton);
    frame.add(removeFiled);
    frame.setBounds(100, 100, 500, 500);
    frame.add(jsp);
```

```
frame.setVisible(true);
    addButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
              String id = idFiled.getText();
              String addname = nameFiled.getText();
              String price = priceFiled.getText();
              String number = numberFiled.getText();
              try {
                   addGood(id, addname, price, number);
                   logger.info(name + " add good " + id);
              } catch (SQLException e) {
                   e.printStackTrace();
              }
         }
    });
    removeButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent actionEvent) {
              String[] id = removeFiled.getText().split(" ");
              for (int i = 0; i < id.length; i++) {
                   String sql = "DELETE FROM good WHERE id=" + id[i];
                   try {
                       stmt.executeUpdate(sql);
                       logger.info(name + " remove good " + id[i]);
                   } catch (SQLException e) {
                        e.printStackTrace();
                   }
              }
    });
}
public void managerManageEmployee() throws SQLException {
    JFrame frame = new JFrame();
    frame.setLayout(null);
    DefaultTableModel model = new DefaultTableModel();
    model.setColumnIdentifiers(new Object[]{"员工 id","员工姓名","生日","年龄"});
    String sql = "SELECT * FROM employee ";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String birthday = rs.getString("birthday");
```

```
int age = rs.getInt("age");
              model.addRow(new Object[]{id,name, birthday, age});
         JTable fk = new JTable(model);
         JScrollPane jsp = new JScrollPane(fk);
         jsp.setBounds(0, 0, 500, 300);
         JTextField idFiled = new JTextField(10);
         JTextField nameFiled = new JTextField(10);
         JTextField birthdayFiled = new JTextField(10);
         JTextField ageFiled = new JTextField(5);
         JButton addButton = new JButton();
         addButton.setBounds(20, 350,60,20);
         addButton.setText("修改");
         idFiled.setBounds(100, 350, 80, 20);
         nameFiled.setBounds(200, 350, 80, 20);
         birthdayFiled.setBounds(300, 350, 80, 20);
         ageFiled.setBounds(400, 350, 80, 20);
         frame.add(addButton);
         frame.add(idFiled);
         frame.add(nameFiled);
         frame.add(birthdayFiled);
         frame.add(ageFiled);
         frame.setBounds(100, 100, 500, 500);
         frame.add(jsp);
         frame.setVisible(true);
         addButton.addActionListener(new ActionListener() {
              @Override
              public void actionPerformed(ActionEvent actionEvent) {
                   String id = idFiled.getText();
                   String addname = nameFiled.getText();
                   String birthday = birthdayFiled.getText();
                   String age = ageFiled.getText();
                   try {
                        updateEmployee(id, addname, birthday, age);
                        logger.info(name + " update employee " + id);
                   } catch (SQLException e) {
                        e.printStackTrace();
                   }
              }
         });
    public void addManager(String id, String name, String birthday, String age) throws
SQLException {
```

```
String sql = "create user "" + id + ""@'%' identified by 'hello";
         stmt.executeUpdate(sql);
         sql = "INSERT INTO 'employee' VALUES (" + id + ", ""+ name + "", "" + birthday +
"'," + age + ")";
         stmt.executeUpdate(sql);
         sql = "grant ALL on lab3.employee to "" + id +""";
         stmt.executeUpdate(sql);
         sql = "grant ALL on lab3.good to "" + id + """;
         stmt.executeUpdate(sql);
    public void deletePeople(String ids, String table) throws SQLException {
         String[] id = ids.split(" ");
         for (int i = 0; i < id.length; i++) {
              String sql = "drop user "" + id[i] + """;
              stmt.executeUpdate(sql);
              sql = "DELETE FROM " + table + " WHERE id=" + id[i];
              stmt.executeUpdate(sql);
         }
    }
    public void addEmployee(String id, String name, String birthday, String age) throws
SQLException {
         String sql = "create user "" + id + ""@'%' identified by 'hello";
         stmt.executeUpdate(sql);
         sql = "INSERT INTO 'employee' VALUES (" + id + ", "'+ name + "', "' + birthday +
"'," + age + ")";
         stmt.executeUpdate(sql);
         sql = "grant SELECT on lab3.good to "" + id +""";
         stmt.executeUpdate(sql);
         sql = "grant update(number) on lab3.good to "+ id + "";
         stmt.executeUpdate(sql);
    }
    public void addGood(String id, String name, String price, String number) throws
SQLException {
         String sql = "INSERT INTO 'good' VALUES (" + id + ", "+ name + "'," + price + ","
+ number + ")";
         stmt.executeUpdate(sql);
    }
    public void updateEmployee(String id, String name, String birthday, String age) throws
SQLException {
         String sql = "DELETE FROM employee WHERE id=" + id;
```

```
stmt.executeUpdate(sql);
         sql = "INSERT INTO 'employee' VALUES (" + id + ", "'+ name + "', "' + birthday +
""," + age + ")";
         stmt.executeUpdate(sql);
    public void updateGood(String id, String number) throws SQLException {
         String sql ="UPDATE good SET number=" + number + " WHERE id=" + id;
         stmt.executeUpdate(sql);
    public void upQuanxian(String ids) throws SQLException {
         String[] id = ids.split(" ");
         for (int i = 0; i < id.length; i++) {
              int idNumber = Integer.parseInt(id[i]) % 100;
              String sql = "drop user " + id[i] + """;
              stmt.executeUpdate(sql);
              sql = "create user "" + idNumber + ""@'%' identified by 'hello'";
              stmt.executeUpdate(sql);
              sql = "grant ALL on lab3.employee to "" + idNumber +""";
              System.out.println(sql);
              stmt.executeUpdate(sql);
              sql = "grant ALL on lab3.good to "" + idNumber + """;
              stmt.executeUpdate(sql);
              sql = "SELECT * FROM employee where id=" + id[i];
              ResultSet rs = stmt1.executeQuery(sql);
              while (rs.next()) {
                   int hereid = rs.getInt("id");
                   String name = rs.getString("name");
                   String birthday = rs.getString("birthday");
                   int age = rs.getInt("age");
                   sql = "INSERT INTO 'manager' VALUES (" + (hereid % 100) + ", ""+ name
+ "'," + birthday + "'," + age + ")";
                   stmt.executeUpdate(sql);
              sql = "DELETE FROM employee WHERE id=" + id[i];
              stmt.executeUpdate(sql);
         }
    public void deleteQx(String ids) throws SQLException {
         String[] id = ids.split(" ");
         for (int i = 0; i < id.length; i++) {
              int idNumber = Integer.parseInt(id[i]) + 200;
              String sql = "drop user " + id[i] + """;
              stmt.executeUpdate(sql);
              sql = "create user "" + idNumber + ""@'%' identified by 'hello'";
```

```
stmt.executeUpdate(sql);
              sql = "grant SELECT on lab3.good to "" + idNumber +""";
              stmt.executeUpdate(sql);
              sql = "grant update(number) on lab3.good to "" + idNumber + "";
              stmt.executeUpdate(sql);
              sql = "SELECT * FROM manager where id=" + id[i];
              ResultSet rs = stmt2.executeQuery(sql);
              while (rs.next()) {
                   int hereid = rs.getInt("id");
                   String name = rs.getString("name");
                   String birthday = rs.getString("birthday");
                   int age = rs.getInt("age");
                   sql = "INSERT INTO 'employee' VALUES (" + (hereid + 200) + ", ""+ name
+ "',"" + birthday + "'," + age + ")";
                   stmt.executeUpdate(sql);
              sql = "DELETE FROM manager WHERE id=" + id[i];
              stmt.executeUpdate(sql);
         }
//程序入口
public class Main {
    public static void main(String[] args) throws IOException {
         Login login = new Login();
```