南昌大学实验报告

学生姓名：丁俊　　学　　号：8003119100　　专业班级：信息安全 193 班

实验类型：☐ 验证 ☐ 综合 ☐ 设计 ☐ 创新　　实验日期：**4.20**　　实验成绩：

一、实验项目名称

数据库操作

说明：（该实验为期中考试，20 分，提交到教学平台 10 分，5.12 前演示 10 分）

二、实验目的

1. 连接数据库实现身份验证

2. 掌握滚动结果集和可更新结果集的使用

三、实验要求

1）利用 JDBC 连接数据库并显示数据内容

2）建立用户数据库表（至少包含 10 条记录）

3）实现用户登录，用户验证以及页面跳转（验证成功，跳转到 success.jsp，并在该页面使用 session 获取中文用户名）

4）在 success.jsp 中显示所有用户信息，通过 更新链接或按钮 利用可更新结果集添加一条用户信息、删除第 6 条用户信息以及更改第 9 条用户信息中的密码，修改完后以表格形式倒序显示新的所有用户信息。

5）使用任一新密码重新登录验证。

四、主要仪器设备及耗材

Windows10 系统　　Intellij IDEA2020.2.3x64 软件

五、实验步骤

create datebase student;
use student;
create table user(
`id` int primary key auto_increment,
`username` varchar(255),
`userid` varchar(255),
`password` decimal(255)
);
创建用户数据库表，id 为 int 自增类型便于用户进行排序。
JDBCdemo.java 文件代码：
有关数据库的连接、断开、插入、查询和删除操作都被封装在 JDBCdemo 类中，这样避免了代码的冗余和重复，直接在 jsp 文件代码中调用 java 类和代码方法即可。

```
1.  public class JDBCdemo extends HttpServlet {
2.      // 账号密码
3.      String uname = "root";
4.      String uword = "1234";
5.      // 数据库链接 url
6.      String url = "jdbc:mysql://localhost:3306/student?useSSL=true&serverTimezone=UT
    C&characterEncoding=utf-8";
7.
8.      Connection con = null;
9.
10.     PreparedStatement sta1 = null;
11.     ResultSet res = null;
12.
13.     // 连接数据库
14.     public void startlink() {
15.         try {
16.             Class.forName("com.mysql.cj.jdbc.Driver");
17.         } catch (Exception e) {
18.             e.printStackTrace();
19.         }
20.         try {
21.             con = DriverManager.getConnection(url, uname, uword);
22.         } catch (Exception e) {
23.             e.printStackTrace();
24.         }
25.     }
26.
27.
28.     public String isExist(String u1,String p1) // 判断是否有该用户且密码正确
29.     {
```

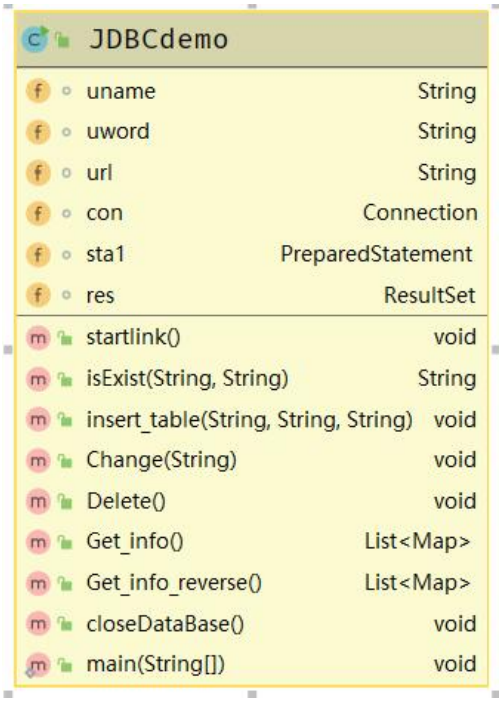
```
30.         String user;
31.         try{
32.             String sql = "select * from user where userid= ? AND password=?";
33.             sta1 = con.prepareStatement(sql,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABLE);
    // 执行查询
34.             sta1.setString(1,u1);
35.             sta1.setString(2,p1);
36.             res = sta1.executeQuery(); // 查询返回结果
37.             if(res.next()){
38.                 user = res.getString(1);
39.                 return user; // 返回中文用户名
40.             }else{
41.                 return null;
42.             }
43.         }catch (Exception e){
44.             e.printStackTrace();
45.         }finally {
46.
47.         }
48.         return null;
49.     }
50.
51.     // 执行添加语句
52.     public void insert_table(String a1,String u1,String p1){
53.         if(a1.length()==0 || u1.length()==0 || p1.length()==0 || a1 == null ||
54.         u1 == null || p1 == null)  return;
55.         try{
56.             String sql1 = "select* from user "; // 选择所有用户信息
57.             sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABLE)
    ; // 可更新结果集
58.             res = sta1.executeQuery();
59.             res.afterLast();
60.             res.moveToInsertRow();
61.             res.updateString(1,a1);
62.             res.updateString(2,u1);
63.             res.updateString(3,p1);
64.             res.insertRow();
65.             res.beforeFirst();
66.         }catch (Exception e){
67.             e.printStackTrace();
68.         }finally {
69.
70.         }
71.     }
```

```
72.
73.     public void Change(String p1) // 更改第 9 条的密码
74.     {
75.         if(p1.length()==0 || p1 == null)  return;
76.         try{
77.             String sql1 = "select* from user order by CONVERT(username USING gbk)";
    // 选择所有用户信息
78.             sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABLE)
   ; // 可更新结果集
79.             res = sta1.executeQuery();
80.             res.absolute(9);
81. //            res.updateString(1,u1);
82.             res.updateString(3,p1);
83.             res.updateRow();
84.             res.beforeFirst();
85.         }catch (Exception e){
86.             e.printStackTrace();
87.         }finally {
88.
89.         }
90.     }
91.     public void Delete()
92.     {
93.         try{
94.             String sql1 = "select* from user order by CONVERT(username USING gbk)";
    // 选择所有用户信息
95.             sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABLE)
   ; // 可更新结果集
96.             res = sta1.executeQuery();
97.             res.absolute(6);
98.             res.deleteRow(); // 删除第六行
99.             res.beforeFirst();
100.     }catch (Exception e){
101.             e.printStackTrace();
102.         }finally {
103.
104.         }
105.     }
106.     public List<Map> Get_info() // 正序输出
107.     {
108.         List<Map> list = new ArrayList<Map>();
109.         try{
110.             String sql1 = "select* from user order by CONVERT(username USING gbk)";
    // 选择所有用户信息
```

```
111.             sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABL
    E); // 可更新结果集
112.             res = sta1.executeQuery();
113.             while (res.next()){
114.                 Map map = new HashMap();
115.                 String user = res.getString(1);
116.                 String name = res.getString(2);
117.                 String pword = res.getString(3);
118.                 map.put("user",user);
119.                 map.put("name",name);
120.                 map.put("word",pword);
121.                 list.add(map);
122.             }
123.         }catch (Exception e){
124.             e.printStackTrace();
125.         }finally {
126.
127.         }
128.         return list;
129.     }
130.
131.     public List<Map> Get_info_reverse() // 逆序
132.     {
133.         List<Map> list = new ArrayList<Map>();
134.         try{
135.             String sql1 = "select* from user order by CONVERT(username USING gbk)";
    // 选择所有用户信息
136.             sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABL
    E); // 可更新结果集
137.             res = sta1.executeQuery();
138.             res.afterLast(); // 最后一行的后面
139.             while(res.previous()){
140.                 Map map = new HashMap();
141.                 String user = res.getString(1);
142.                 String name = res.getString(2);
143.                 String pword = res.getString(3);
144.                 map.put("user",user);
145.                 map.put("name",name);
146.                 map.put("word",pword);
147.                 list.add(map);
148.             }
149.         }catch (Exception e){
150.             e.printStackTrace();
151.         }finally {
```

```
152.
153.         }
154.         return list;
155.     }
156.
157.     // 关闭数据库
158.     public void closeDataBase()
159.     {
160.         try{
161.             res.close();
162.             sta1.close();
163.             con.close();
164.         }catch (Exception e){
165.             e.printStackTrace();
166.         }
167.     }
168. }
```

UML 图:



JDBCdemo

- startlink(): 连接数据库函数
- isExist(String,String): 判断用户名密码是否存在和正确，如果存在返回中文用户名
- insert_table(String,String,String): 插入一个用户信息
- Change(String): 改变第九个用户的密码
- Delete(): 从数据库删除第六个用户信息
- Get_info(): 按照字母顺序正序返回用户信息
- Get_info_reverse(): 按照字母顺序逆序返回用户信息
- closeDataBase(): 关闭数据库连接

login.jsp 文件-主登录界面

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="com.mysql.jdbc.Driver" %>
<%@ page import="java.sql.*" %>
<html>
  <head>
    <title>数据库连接</title>
  </head>
  <body>
  <form action="login_check.jsp" method="post">
    <table border="10" align="center">
      <tr>
        <td colspan="2">用户登录</td>
      </tr>
      <tr>
        <td>登录 ID：</td>
        <td><input type="text" name="id"></td>
```

```
      </tr>
      <tr>
        <td>登录密码：</td>
        <td><input type="password" name="password"></td>
      </tr>
      <tr>
        <td colspan="2">
          <input type="submit" value="登录">
          <input type="reset" value="重置">
        </td>
      </tr>
    </table>
  </form>
  </body>
</html>
```

Login_check.jsp 文件-用户登录检查界面

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="ding.*" %>
<%@ page import="java.util.Map" %>
<%@ page import="java.util.List" %>
<%@ page import="java.util.ArrayList" %>
<%@ page import="jakarta.servlet.http.HttpServlet"%>
<html>
<head>
    <title>登录检查页面</title>
</head>
<body>
    <%
        String getid = null;
        String getpword = null;
        String st = null;
        List<Map> list = new ArrayList<Map>();
    %>
    <%
        JDBCdemo jc = new JDBCdemo();
        jc.startlink();
        getid = request.getParameter("id");
        getpword = request.getParameter("password");
        st = jc.isExist(getid,getpword);
    %>
        <%
        if (st != null) {
```

```
            session.setAttribute("username",st);
            jc.closeDataBase();
            response.sendRedirect("login_success.jsp");

    } else {
            jc.closeDataBase();
    %>
        <jsp:forward page="login_failure.jsp"/>
        <%
        }
    %>
</body>

</html>
```

Login_success.jsp 文件-登录成功界面

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>
<head>
    <title>登录成功页面</title>
</head>
<body>
    <%
        String str1 = (String) session.getAttribute("username");
        List<Map> list;
        JDBCdemo jc = new JDBCdemo();
        jc.startlink();
        list = jc.Get_info();
    %>
    <h2 align="center">登录成功</h2><hr/>
    <h2 align="center">欢迎<font color="red"><%=str1%></font>光临！ </h2>
    <h2 align="center">所有注册用户信息:</h2>
    <table border="1"  align = "left" cellspacing="0">
        <caption>用户信息</caption>
        <tr>
            <th>用户名称</th>
            <th>账号</th>
            <th>密码</th>
        <tr/>
        <%
            for(Map t:list)
            {%>
```

```
                <tr>
                    <td><%=t.get("user")%></td>
                    <td><%=t.get("name") %></td>
                   <td><%=t.get("word") %></td>
                </tr>
            <%}
        %>
    </table>
    <hr/>
    <h2 align="center">---请添加一条用户信息和修改第 9 条用户的密码:---</h2>

    <form action="change.jsp" method="post">
        <table borde = "2" align="center" cellspacing="0">
            <tr>
                <td>新的用户中文名:</td>
                <td><input type = "text" name = "user"></td>
            </tr>
            <tr>
                <td>新的 ID:</td>
                <td><input type = "text" name = "newid"></td>
            </tr>
            <tr>
                <td>新的密码:</td>
                <td><input type = "text" name = "neword"></td>
            </tr>
            <tr>
                <td>修改第九个用户的密码:</td>
                <td><input type = "text" name = "nineword"></td>
            </tr>
            <tr>
                <td colspan="2">
                    <input type = "submit" value="提交">
                    <input type = "reset" value="重置">
                </td>
            </tr>
        </table>
    </form>


    <%
        jc.closeDataBase();
    %>
</body>
</html>
```

Change.jsp 文件-改变信息操作文件，增加一个新用户，删除第 6 个用户并修改第 9 个用户的密码。

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>修改后的倒序表格</title>
</head>
<body>
<%
    String user = null;
    String newname = null;
    String newword = null;
    String nine = null;
%>
    <%
        String str1 = (String) session.getAttribute("username");
        user = request.getParameter("user");
        newname = request.getParameter("newid");
        newword = request.getParameter("neword");
        nine = request.getParameter("nineword");
        List<Map> list;
        JDBCdemo jc = new JDBCdemo();
        jc.startlink();
        if(nine.length()!=0){
                jc.Change(nine);
        }
        jc.Delete();
        if(user != null && newname !=null && newword!=null){
            jc.insert_table(user,newname,newword);
        }
        list = jc.Get_info_reverse();

        jc.closeDataBase();

    %>
    <h2>修改完毕,你可以重新登录哦<a href="login.jsp">登录</a>! </h2>
    <h2>---下面是倒序的用户信息---</h2>

```

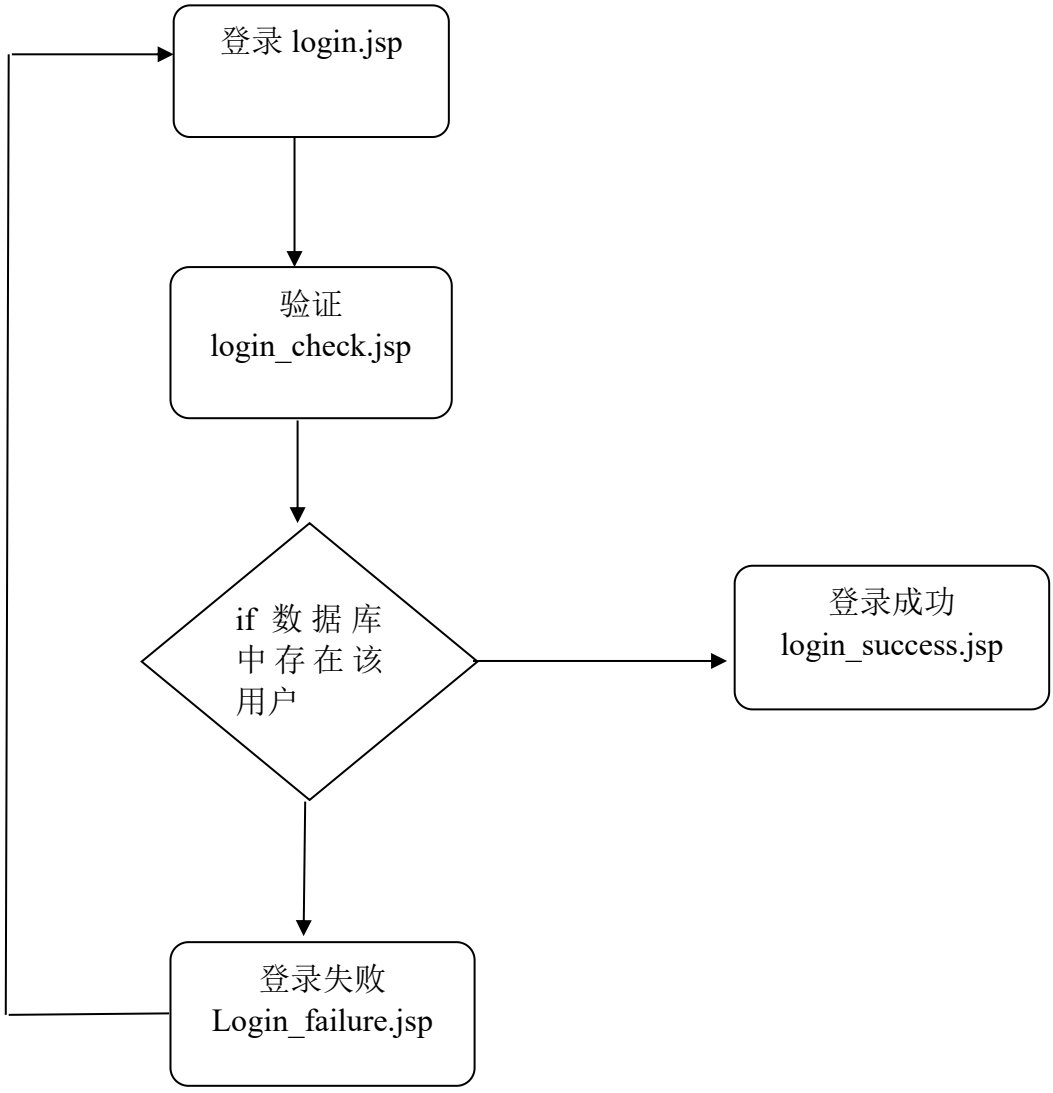
```
    <table border="1"  align = "left" cellspacing="0">
        <caption>倒序的用户信息</caption>
        <tr>
            <th>用户名称</th>
            <th>账号</th>
            <th>密码</th>
        <tr/>
        <%
            for(Map t:list)
            {%>
        <tr>
            <td><%=t.get("user")%></td>
            <td><%=t.get("name") %></td>
            <td><%=t.get("word") %></td>
        </tr>
        <%}
        %>
    </table>


</body>
</html>
```

Login_failure.jsp-登录失败界面

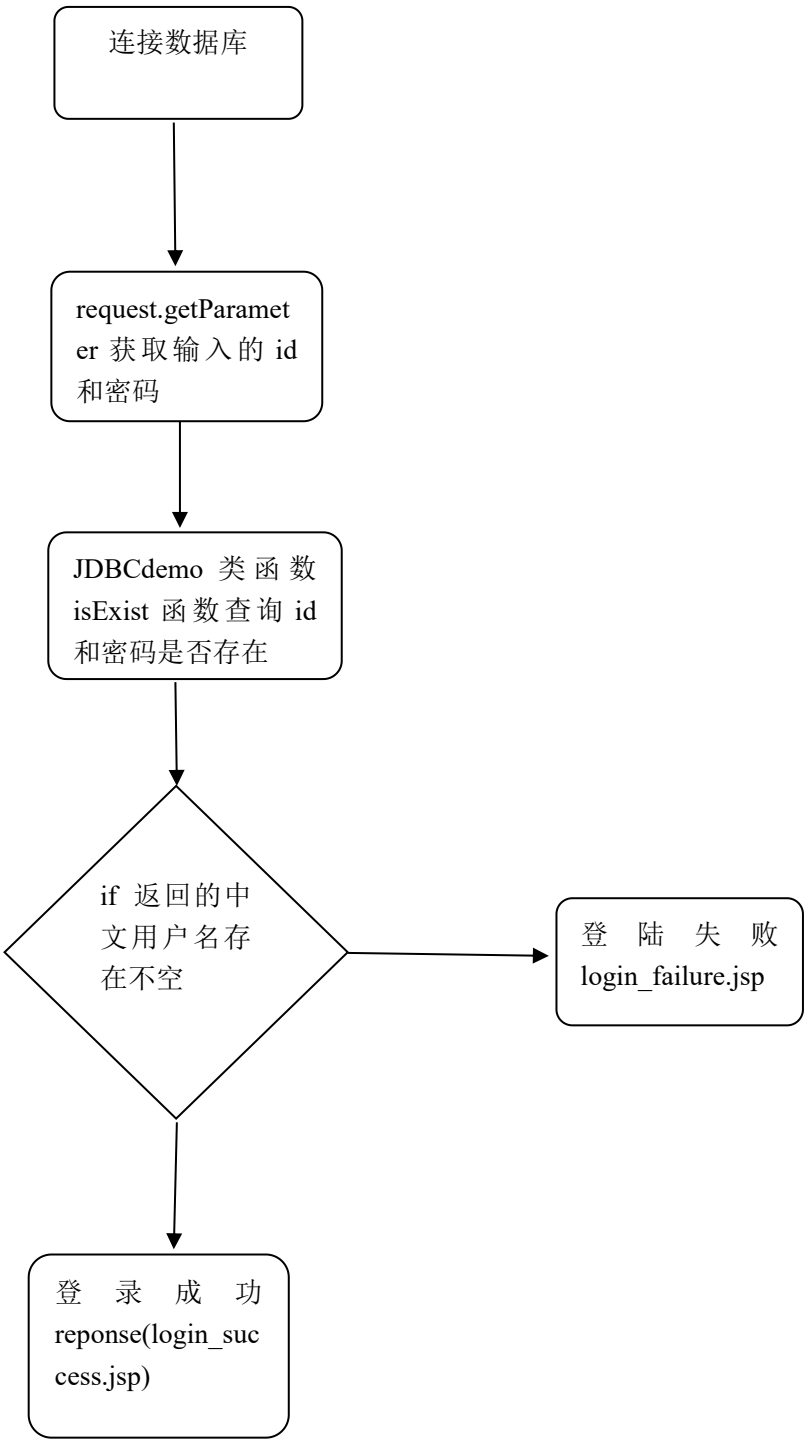
```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>登录失败页面</title>
</head>
<body>
    <h1>登录失败</h1><hr>
    <h2>登录失败，请重新<a href="login.jsp">登录</a>! </h2>
  </body>
</html>
```

流程图

主要功能界面流程



登陆检查流程


连接数据库
request.getParamet er 获取输入的 id 和密码
JDBCdemo 类函数 isExist 函数查询 id 和密码是否存在
if 返回的中文用户名存在不空
登陆失败 login_failure.jsp
登录成功 reponse(login_suc cess.jsp)

六、主要实现代码及流程

1、数据库连接类和方法

在 src.ding 中定义了一个 java 类，用于数据库连接和数据操作，分别为数据库连接、判断输入的用户名和密码是否正确、添加记录函数、更新语句、删除语句、按用户名信息正序和逆序排列输入数据。

```
List<Map> list = new ArrayList<Map>();
try{
    String sql1 = "select* from user order by CONVERT(username USING gbk)"; // 选择所有用户信息
    sta1 = con.prepareStatement(sql1,TYPE_SCROLL_SENSITIVE,CONCUR_UPDATABLE); // 可更新结果集
    res = sta1.executeQuery();
    while (res.next()){
        Map map = new HashMap();
        String user = res.getString( columnIndex: 1);
        String name = res.getString( columnIndex: 2);
        String pword = res.getString( columnIndex: 3);
        map.put("user",user);
        map.put("name",name);
        map.put("word",pword);
        list.add(map);
    }
```

2、数据获取和显示

在获取数据库所有记录的函数方法中使用了 List 和 HashMap 类，用于接收用户信息的键值对，并返回 list 给客户端。

```jsp
<body>
    <%
        String getid = null;
        String getpword = null;
        String st = null;
    %>
    <%
        request.setCharacterEncoding("UTF-8");
        JDBCdemo jc = new JDBCdemo();
        jc.startlink();
        getid = request.getParameter("id");
        getpword = request.getParameter("password");
        st = jc.isExist(getid,getpword);
    %>
        <%
        if (st != null) {
            session.setAttribute("username",st);
            jc.closeDataBase();
            response.sendRedirect("login_success.jsp");

    } else {
            jc.closeDataBase();
    %>
        <jsp:forward page="login_failure.jsp"/>
        <%
        }
    %>
```

在 login_check.jsp 检查页面中 isExits 函数判断输入的用户名和密码是否存在，如果存在会返回中文用户名，并用 session 在服务端保存这个数据，方便显示这个用户名，此时跳转到 login_success.jsp 登录成功页面；反之跳转到 login_failure.jsp 页面提示登录失败。

```jsp
%
    String str1 = (String) session.getAttribute("username");
    List<Map> list;
    JDBCdemo jc = new JDBCdemo();
    jc.startlink();
    list = jc.Get_info();
```

```jsp
    <h2 align="center">登录成功</h2><hr/>
    <h2 align="center">欢迎<font color="red"><%=str1%></font>光临！ </h2>
```

```jsp
  <%
      for(Map t:list)
      {%>
        <tr>
            <td><%=t.get("user")%></td>
            <td><%=t.get("name") %></td>
           <td><%=t.get("word") %></td>
        </tr>
      <%}
  %>
```

在登录成功页面通过 session 获取原来存储的中文用户名并显示，定义了 List 存储 Get_info 函数返回的用户信息表，并以表格形式展现。

3、数据更新

通过获取在 login_success.jsp 中填入的表格信息对用户信息进行添加一条用户信息、删除第 6 条用户信息以及更改第 9 条用户信息的中文密码，这时候只需要调用数据库连接类里的函数方法传入参数即可。

```
request.setCharacterEncoding("UTF-8");
String str1 = (String) session.getAttribute("username");
user = request.getParameter("user");
newname = request.getParameter("newid");
newword = request.getParameter("neword");
nine = request.getParameter("nineword");
List<Map> list;
JDBCdemo jc = new JDBCdemo();
jc.startlink();
if(nine.length()!=0){
        jc.Change(nine);
}
jc.Delete();
if(user != null && newname !=null && newword!=null){
    jc.insert_table(user,newname,newword);
}
list = jc.Get_info_reverse();

jc.closeDataBase();
```

七、实验数据及处理结果

Mysql 数据库中的表和记录如下：

开始事务 | 文本 | 筛选 | 排序 | 导入 | 导出

id	username	userid	password
1	Sara	7415851	asw1822
2	丁俊	147258	1314521
3	张灿	7893116	asd188
4	徐兵	489621	147852
5	成吉思汗	121asss	asfffw
8	武则天	aswdad	121344
9	王韬	147258	iloveyou
10	罗峰	123789	2314123
11	阿文	7894111	123456
12	陈冰	1448189	1544891
13	邹霞	12314134	iloeyou
14	丁峰	123444	swaw2

输入丁俊的账号和密码：

点击登录跳转到登录成功页面：

登录成功

欢迎丁俊光临！

所有注册用户信息:

用户信息

用户名称	账号	密码
Sara	7415851	asw1822
丁俊	147258	1314521
张灿	7893116	asd188
徐兵	489621	147852
成吉思汗	121asss	asfffw
武则天	aswdad	121344
王韬	147258	iloveyou
罗峰	123789	2314123
阿文	7894111	123456
陈冰	1448189	1544891
邹霞	12314134	iloeyou
丁峰	123444	swaw2

---请添加一条用户信息和修改第9条用户的密码:---

新的用户中文名: 刘天
新的ID: 12345
新的密码: wwee
修改第九个用户的密码: 132134
提交 重置

这里第 6 条记录是武则天，第 9 条记录是阿文。
显示的信息是按照 id 次序来排列的，即添加进数据库的顺序，右边输入信息点击提交可以添加或更新部分用户信息。

修改完毕,你可以重新登录哦登录!

---下面是倒序的用户信息---

倒序的用户信息

用户名称	账号	密码
刘天	12345	wwee
丁峰	123444	swaw2
邹霞	12314134	iloeyou
陈冰	1448189	1544891
阿文	7894111	132134
罗峰	123789	2314123
王韬	147258	iloveyou
成吉思汗	121asss	asfffw
徐兵	489621	147852
张灿	7893116	asd188
丁俊	147258	1314521
Sara	7415851	asw1822

输入添加和更新的信息后点击提交跳转到新的显示页面，在新的页面中有一条跳转到登录页面 login.jsp 的链接。即使输入的信息为空或者没有输入也有错误校验，在数据库连接类中每一个更新、插入函数都有判断参数是否为 null 的功能。
可以看到第 6 条记录武则天被删除了，第 9 条记录阿文的密码变成了“132134”,并添加了“刘天”这一条记录。

随后再次点击登录链接，尝试用新的用户 id 和密码登录，发现登录成功。

用户登录

登录ID: 12345

登录密码: wwee

登录 重置

八、思考讨论题或体会或对改进实验的建议