

实验 8 JDBC 编程

实验目的

1. 掌握使用 Navicat 建立数据库和表；
2. 对 Mysql 数据库实现增删改查操作。

实验步骤

- 步骤 1：**创建工程，工程名以学号姓名方式命名，“2020010505-薛飞宇”。
- 步骤 2：**为每道实习题目创建对应的包，如“work1”、“work2”……；
- 步骤 3：**按照要求创建源代码，进行编写，注意编码格式，如缩进、命名规范等；
- 步骤 4：**规范书写实习报告；
- 步骤 5：**实现与测试。附上代码或以附件的形式提交，同时贴上必要的代码运行截图；
- 步骤 6：**及时总结心得体会与备忘。

实验内容

实验题 1 JDBC 编程。

应用 test 数据库，在该数据库中，建一个表 employee，通过 JDBC 完成如下操作：

1. 执行 sql 语句在表中添加如下记录；
2. 查询“Peter”的信息并输出
3. 将表中数据读出并将每人工资加 1500 后存入原表。
4. 以“name salary”的格式输出男性员工中工资最高的记录。

关于 JDBC 的编程都有固定的套路。

0. 在项目文件夹中加入驱动程序
 1. 首先，加载驱动。
 2. 建立连接。
 3. 执行 SQL 语句。
 4. 释放资源，关闭连接。

所有的 JDBC 编程的前 3 步都是一样的，多次练习之后达到熟练即可。真正有区别的是实行 SQL 语句。刚开始，我就遇到了问题。在插入记录的语句中，字段的类型如果是字符串的话，在插入内容的两边要加上单引号，如下图，否则会报错。

```
String executeString1 = "insert into employee "
    + "(id, name, sex, salary) "
    + "values "
    + "(1002, 'Tom', 'male', 2600)";
```

图 1.1 实验 1 插入数据

修改过后，成功执行。

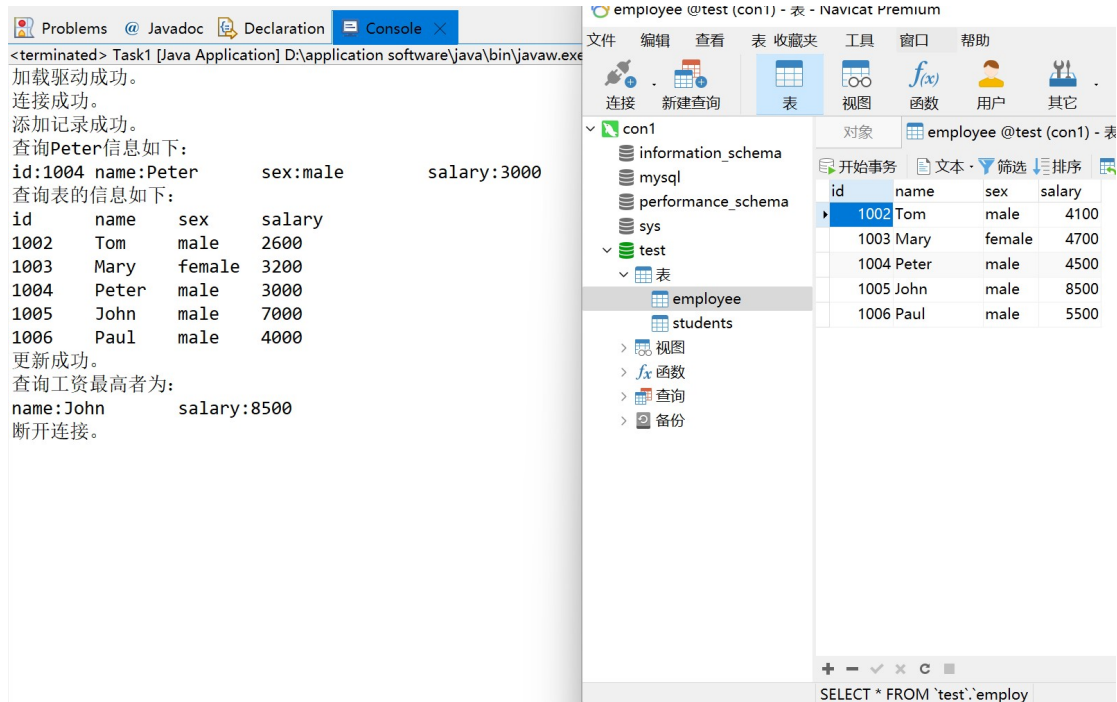


图 1.2 实验 1 测试结果

第二步，查询记录的时候，需要将 ResultSet 的对象先调用 next()方法，否则依然会报错，如下图。

```
if(rs.next()) {
    System.out.println("查询Peter信息如下: ");
    int id = rs.getInt(1);
    String name = rs.getString(2);
    String sex = rs.getString(3);
    int salary = rs.getInt(4);
    System.out.println("id:"+id
        + " name:"+name
        + " sex:"+sex
        + " salary:"+salary);
}
```

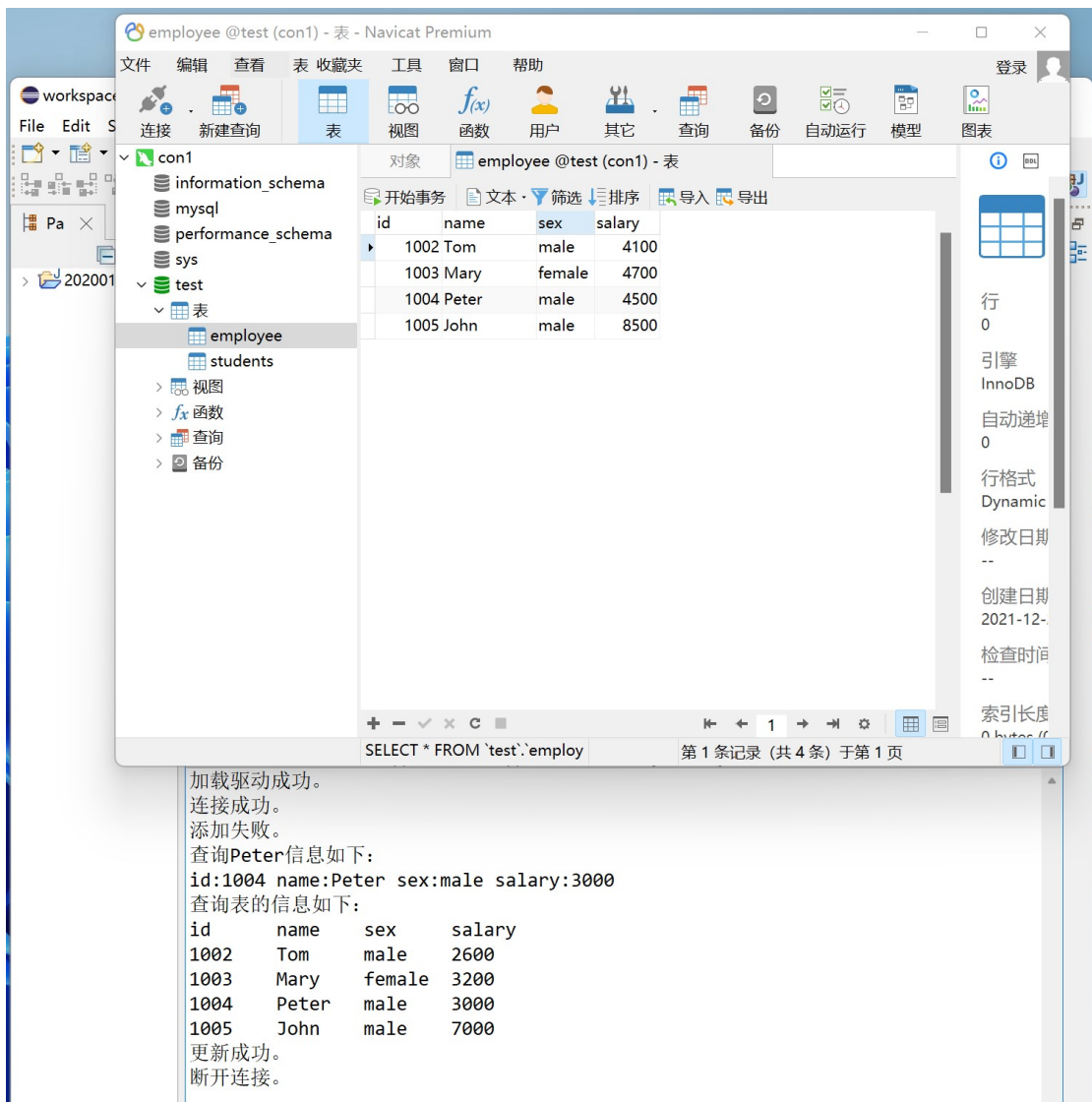
图 1.3 next()方法

查询信息如下图。

```
<terminated> Task1 [Java Application] D:\application software\java\bin\javaw.exe (2021年12月24日上午7:45:14 -
加载驱动成功。
连接成功。
查询Peter信息如下:
id:1004 name:Peter sex:male salary:3000
断开连接。
```

图 1.4 查询结果

第三步，将表中数据读出并将每人工资加 1500 后存入原表，读出数据直接用 `select * from employee` 即可。更新数据用 `update employee set salary = salary + 1500` 即可，下面是测试结果。



The screenshot shows the Navicat Premium interface. The left sidebar displays the database structure, including the 'test' database and the 'employee' table. The main window shows the 'employee' table with the following data:

id	name	sex	salary
1002	Tom	male	4100
1003	Mary	female	4700
1004	Peter	male	4500
1005	John	male	8500

Below the table, the SQL query `SELECT * FROM `test`.`employ` is shown. The results of the query are displayed, showing the same 4 records with salaries 2600, 3200, 3000, and 7000.

```
加载驱动成功。
连接成功。
添加失败。
查询Peter信息如下:
id:1004 name:Peter sex:male salary:3000
查询表的信息如下:
id      name  sex  salary
1002    Tom   male  2600
1003    Mary  female 3200
1004    Peter male  3000
1005    John  male  7000
更新成功。
断开连接。
```

图 1.5 更新结果

第四步，查询工资最高的记录，这里就要用一点小技巧了。 `select * from`

employee order by salary desc limit 1。先将数据按 salary 降序排序，然后选中第一条记录，这就是查询到工资最高的记录。

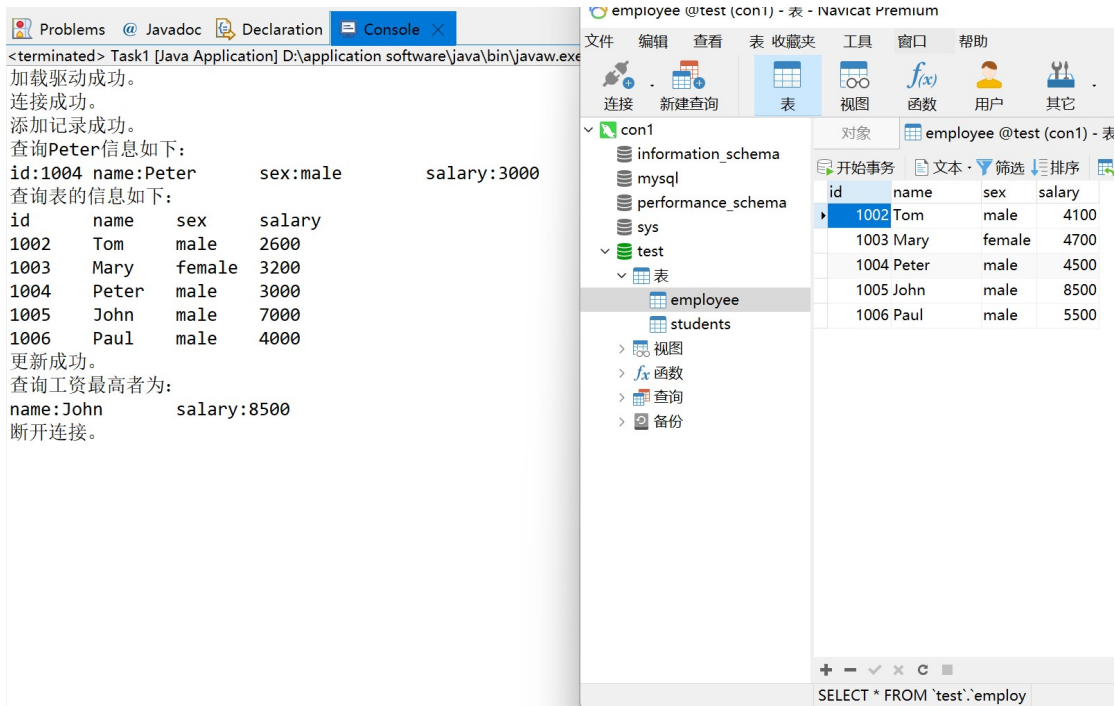


图 1.6 查询工资最高的记录

下面是本程序的 UML 类图。

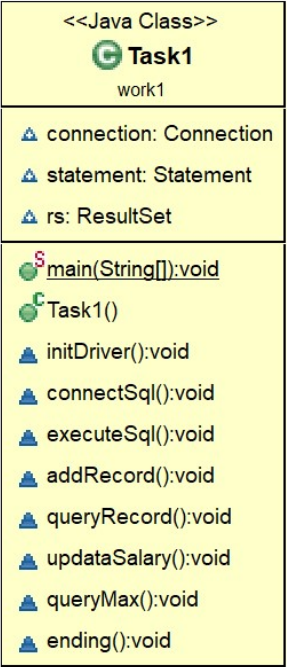


图 1.7 实验 1UML 类图

实验题 2 JDBC 编程

在 test 数据库中，建一个表 admin，根据分析在表中添加若干记录，完成如下登录界面案例。

本题还是很简单，先根据要求建立相应的表添加相应的数据记录。

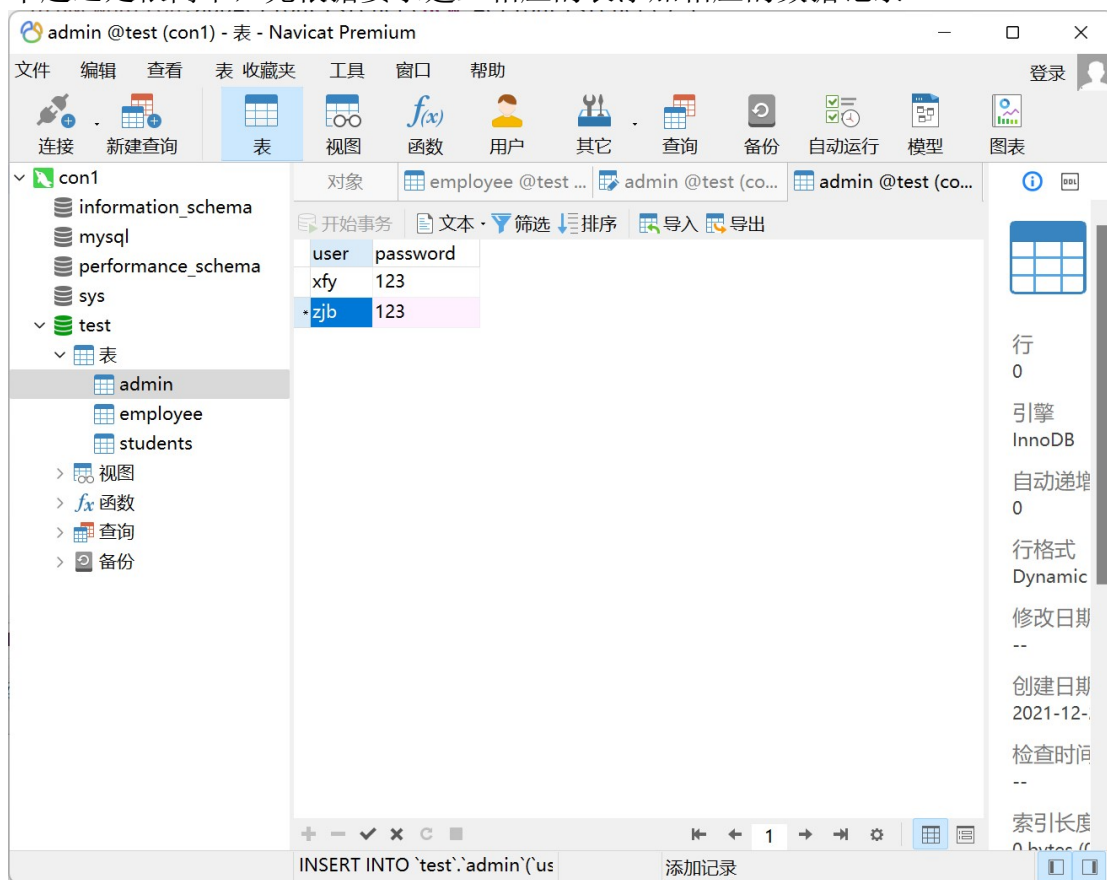


图 2.1 添加记录

在 LoginConnection 类中设计一个 check() 方法，每当用户点击确定按钮后，LoginConnection 的一个对象就调用 check() 方法，根据查询结果跳出不同的窗口。

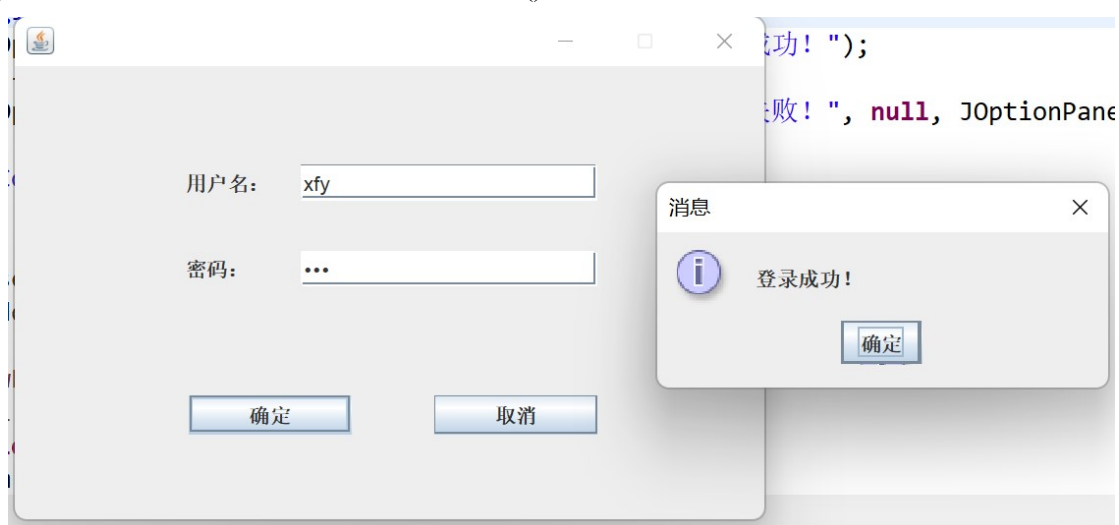


图 2.2 实验 2 测试

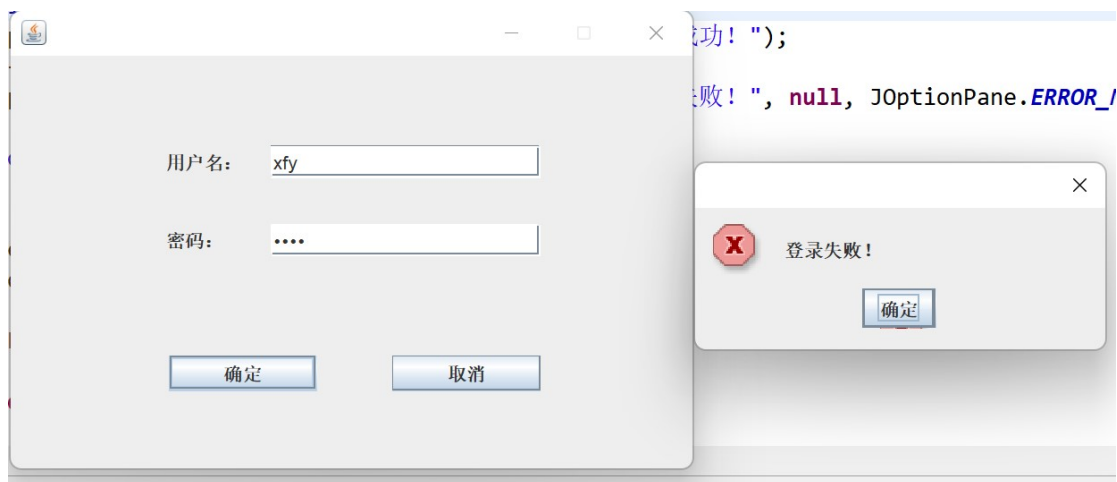


图 2.3 实验 2 测试

本实验还有一个值得注意的点，在查询的时候，查询语句应为
`select user,password from admin where user="" + uu + ""`
 查询的对象为变量时，应该以这样的形式出现："`"" + uu + ""`".
 下面是本实验的 UML 类图。

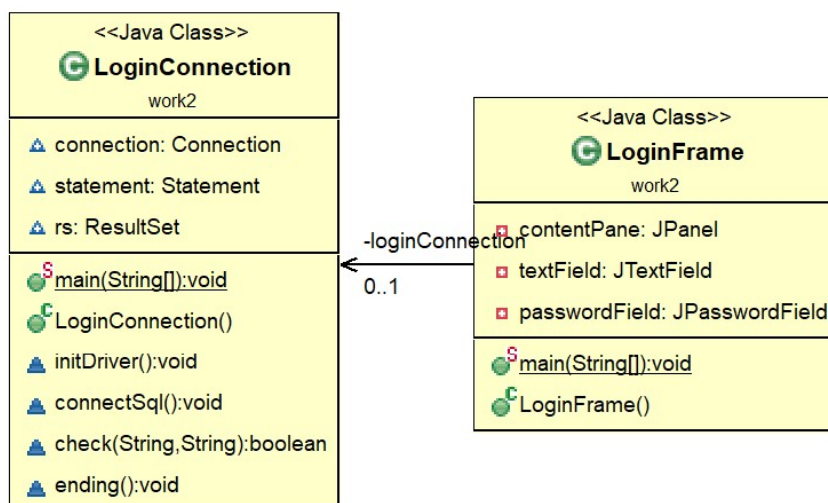


图 2.4 实验 2UML 类图

本次实验尚未解决的疑难：

本实验所有问题都已解决。

心得体会：

本次实验，我掌握 JDBC 编程的概念和实现，掌握、使用 mysql 进行数据库的连接以及表的增删改查，在此基础上实现了简单的登录界面。Java 和数据库的结合给我的编程世界带来了无限的可能。