数据库实验报告

|  |  |
| --- | --- |
| 学生姓名 | 储可运 |
| 学号 | 161840220 |
| 学院 | 计算机科学与技术学院/人工智能学院 |
| 专业 | 物联网工程 |
| 班级 | 1618403 |

**计算机科学与技术学院**

目录

[**1** 需求分析 2](#_Toc76071550)

[**1.1** 问题背景 2](#_Toc76071551)

[**1.2** 完成目标： 2](#_Toc76071552)

[**2** 概念结构设计 2](#_Toc76071553)

[**3** 逻辑结构设计 3](#_Toc76071554)

[**4** 物理结构设计 4](#_Toc76071555)

[**4.1** 运行事务分析 4](#_Toc76071556)

[**4.2** 关系模式存取方法的选择与数据库存储结构的确定 4](#_Toc76071557)

[**5** 数据库的实施 4](#_Toc76071558)

[**5.1** 搭建数据库 4](#_Toc76071559)

[5.1.1 购买弹性云服务器ESC 4](#_Toc76071560)

[5.1.2 安装opengauss数据库 5](#_Toc76071561)

[**5.2** 创建表格和生成数据 6](#_Toc76071562)

[5.2.1 使用Data Studio连接数据库 6](#_Toc76071563)

[5.2.2 创建表格 7](#_Toc76071564)

[5.2.3 生成数据 7](#_Toc76071565)

[**5.3** 成绩管理系统的设计 8](#_Toc76071566)

[5.3.1 连接数据库 8](#_Toc76071567)

[5.3.2 学生登录界面以及成绩界面的设计 9](#_Toc76071568)

[5.3.3 老师登录界面以及操作界面的设计 11](#_Toc76071569)

[**6** 数据库的维护 17](#_Toc76071570)

[**7** 实验的总结与感想 17](#_Toc76071571)

# 需求分析

## 问题背景

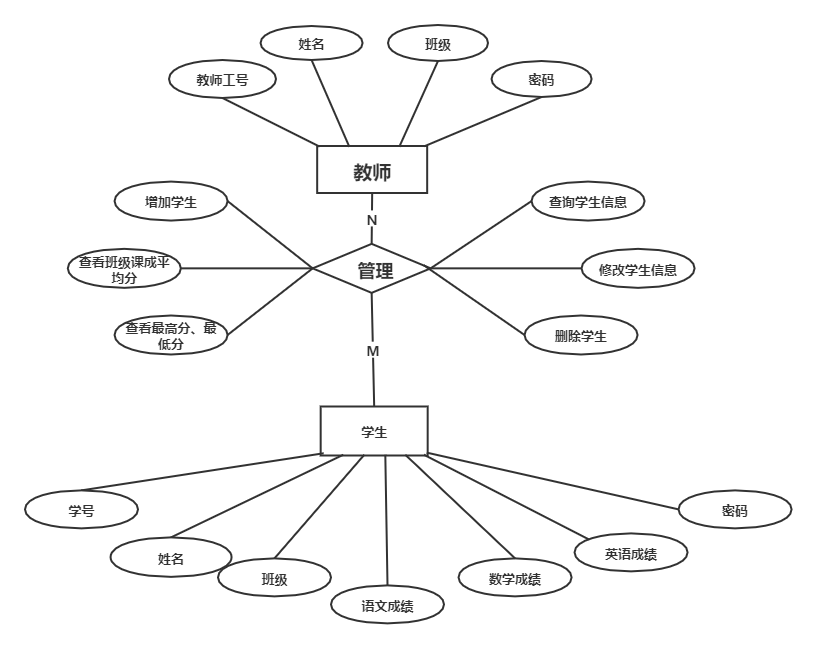
数据库解决的是小学年级的成绩管理问题，是由书中的大学生选课管理系统联想来的。回想起小学的成绩单都是在考试结束一段时间后再去学校领取，而且同时成绩单都是有老师书写，这样对于师生来说都不方便，而且统计排名与最高分、最低分也比较麻烦。由此想要设计一个学生成绩管理系统。

## 完成目标：

成绩管理系统分成学生登录与老师登录，学生登录后可以看到自己的成绩单，成绩单内有各科成绩与总分，还有各科与总分的排名。成绩单最下面，有两个按钮，一个是修改密码按钮，可以让学生修改自己的密码，一个是退出按钮，单击即可退出。老师登录后，可以查询和修改学生的信息（包括成绩），可以插入学生，删除学生，也可以看自己班级的各个科目的平均分等学生情况。关于账号，原本的设想是学生自己注册，老师自己注册（为了防止学生注册老师的账号，本来设想老师注册需要验证码，由学校发放），最后经同学提醒，最后决定是老师账号由管理人员创建，而学生的账号由老师添加。

# 概念结构设计

对于概念模型，以下通过E-R图来展现：



**教师-学生-ER图**

从E-R图中可以清晰的看出两个表之间的关系。

# 逻辑结构设计

对于逻辑结构，通过关系模型来展现，而关系模型可以由E-R图转换过来，所以关系模型如下：

学生（学号，姓名，班级，语文，数学，英语。密码）

老师（教师工号，姓名，班级，密码）

管理（教师工号，学号，姓名，班级，语文成绩，数学成绩，英语成绩）

其中，带下划线的为主码。对于管理，老师可以修改学生的各个信息（除了密码）。

# 物理结构设计

## 运行事务分析

对于这个成绩数据库，运行的事务最多的肯定是查询操作，其次还存在增删改操作。对于查询，涉及到学生的各个属性，对于更新，也是在学生表上进行更新，删除也是如此。所以对于学生的表的操作居多，并且速度肯定是越快越好

## 关系模式存取方法的选择与数据库存储结构的确定

由于自己的数据库的数据规模并不大，对于存取方法并没有考虑太多。但是在数据规模变得很大后，需要在学生表上建立索引，应该在班级这个字段建立索引方便数据库操作。

对于数据库本身，存放在华为云服务器上，对于数据的存放并没有过多的设计，对于数据库本身的系统配置，全部是默认，没有过多的更改。

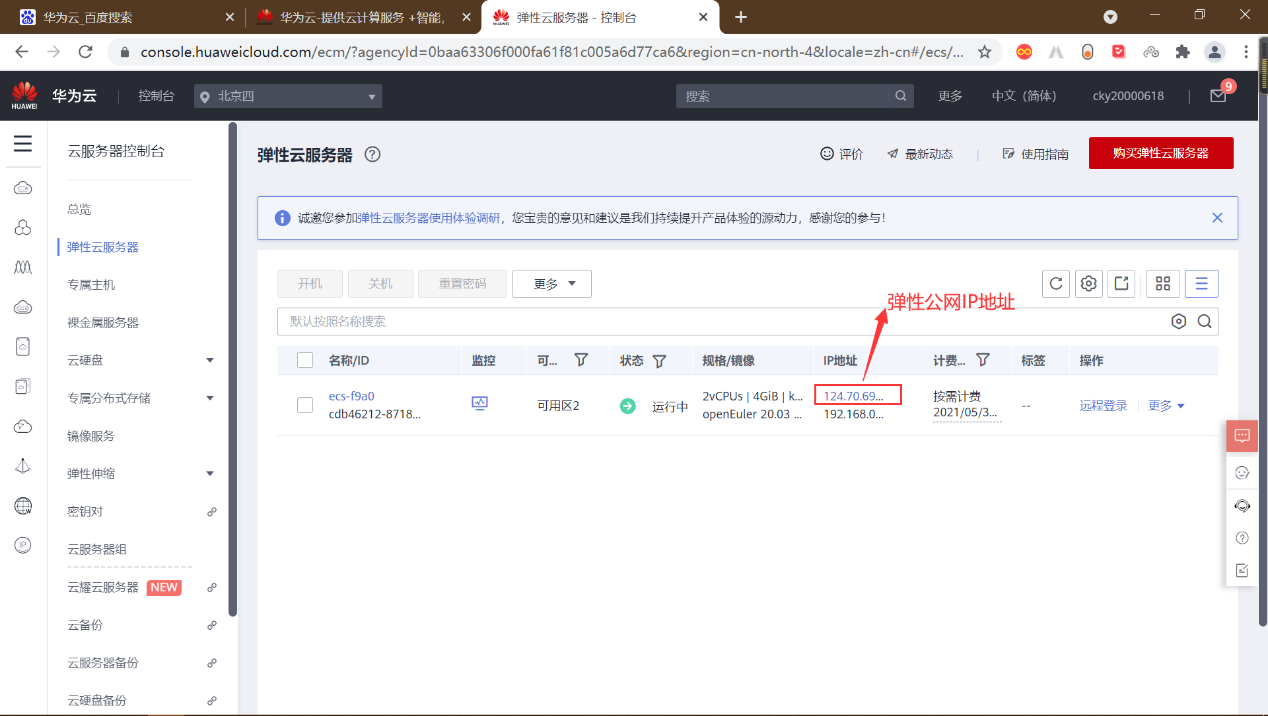
# 数据库的实施

## 搭建数据库

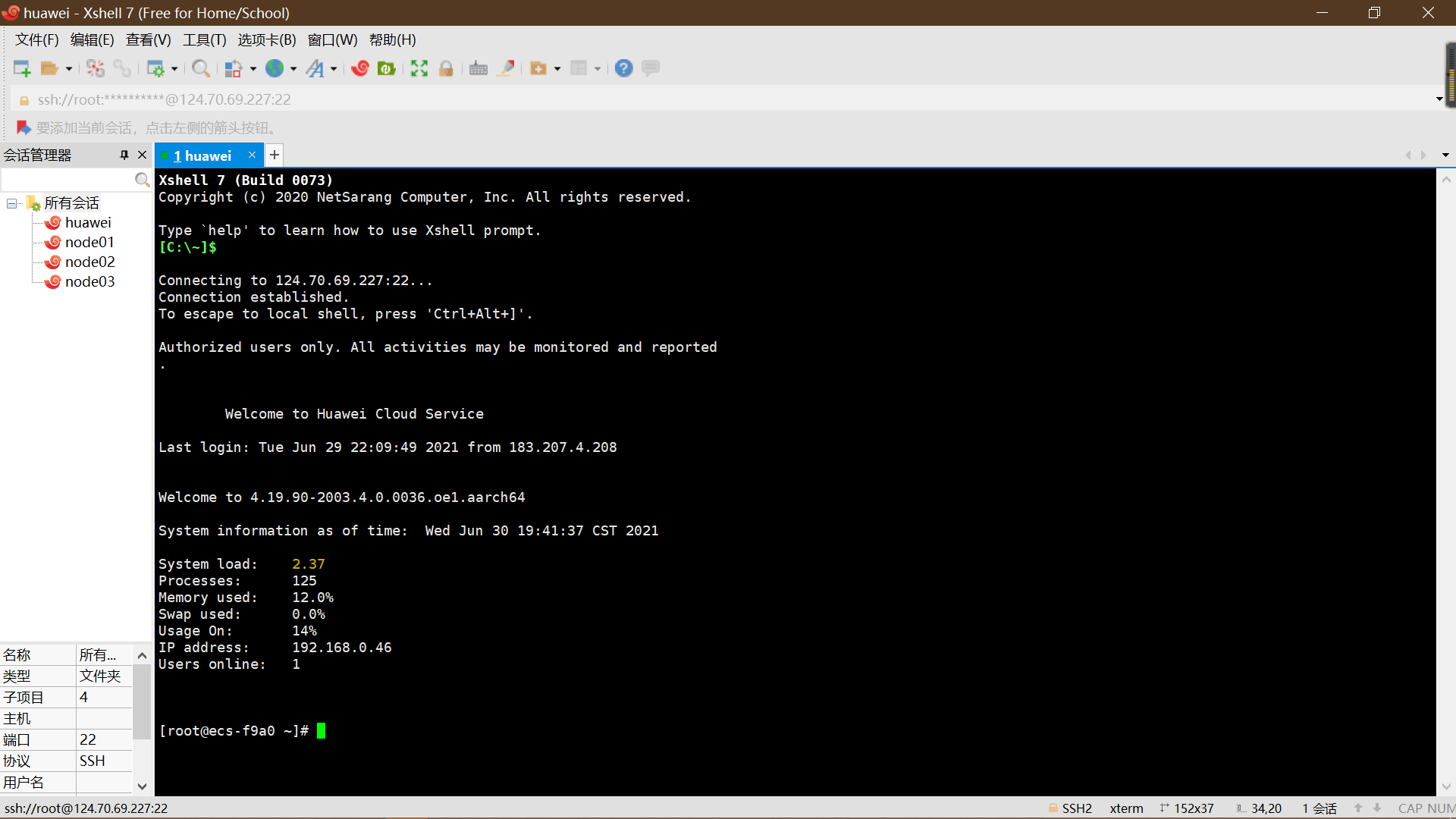
数据库环境的搭建是依照着华为给的官方数据库指导手册来做的，这里只介绍简单的流程，具体操作细节不在赘述。

### 购买弹性云服务器ESC

首先登录华为云，购买弹性云服务器ECS（购买时对其进行了相关配置），购买成功后即可在列表中看到，如下图所示：



在此可以执行开机、挂机等操作。然后根据云服务器显示的弹性公网IP地址，利用Xshell进行连接。然后即可在其中看到连接华为云服务器的提示。

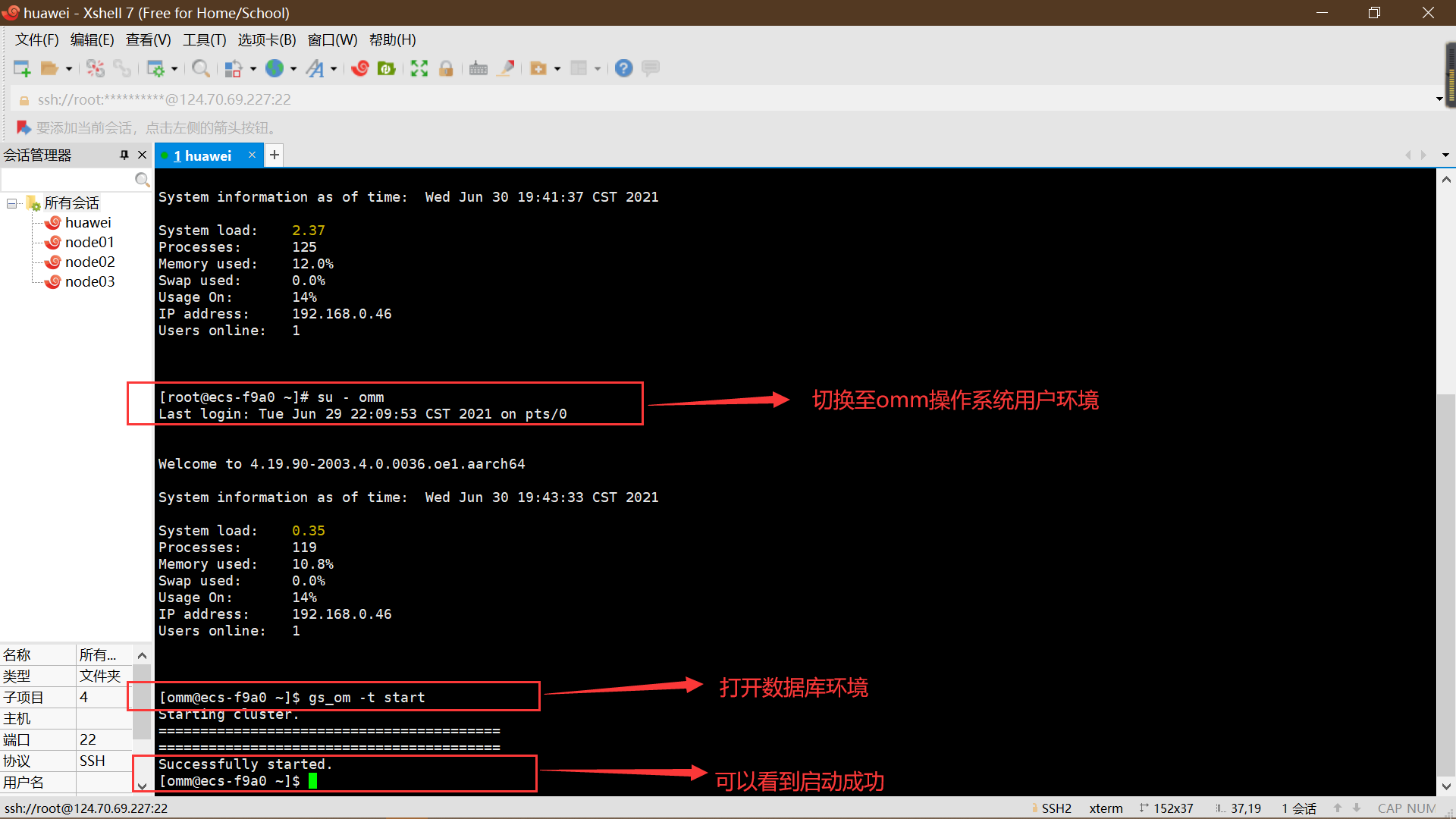


由上图所示即可看出通过Xshell连接华为云服务器是成功的。

### 安装opengauss数据库

根据数据库指导手册，首先修改python版本并安装libaio包，然后便可以下载数据库的安装包。使用wget下载数据库安装包到安装包目录，下载成功后创建XML配置文件，然后初始化安装环境（例如修改performance.sh文件等一系列步骤），最后执行安装即可。

最终是否安装成功呢？只需要进入数据库主节点服务器上，切换至omm操作系统用户环境，然后启动数据库服务即可。下图是启动成功的数据库服务的运行截图。



对于以后每次打开服务器后，都要用Xshell连接进行此操作启动数据库服务，否则无法去连接数据库。

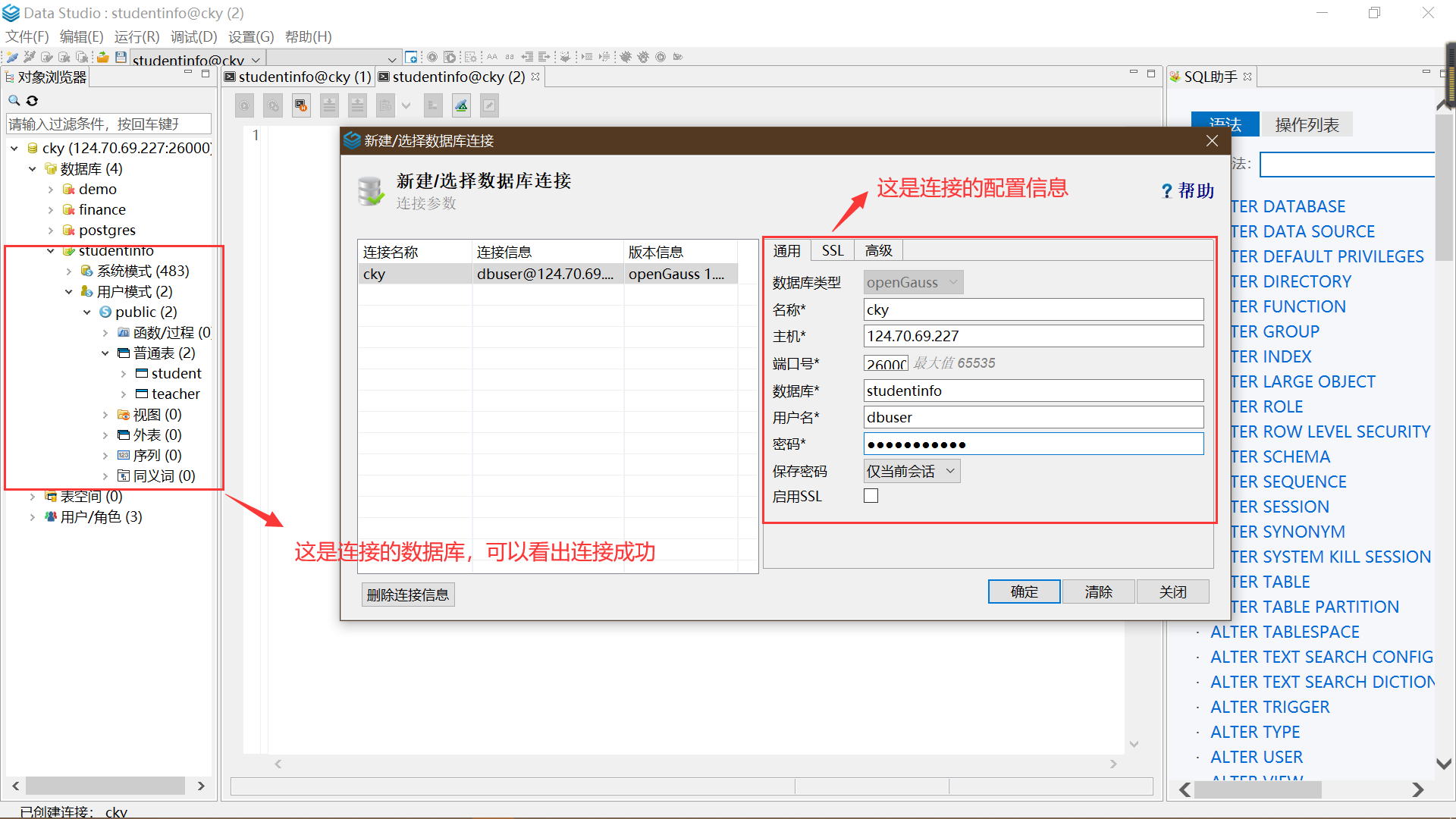
## 创建表格和生成数据

上面创建了数据库，之后便就要创建我们所需要的表了。

### 使用Data Studio连接数据库

Data Studio是一个集成开发环境（IDE），帮助数据库开发人员便捷地构建应用程序，以图形化界面形式提供数据库关键特性。这是华为提供的一款连接opengauss数据库并可以使用图形化的操作的软件 。

根据指导手册，首先要准备连接环境，将数据库的各种文件进行修改，其中，最主要的是确定服务器的26000端口是否开放，如未开放还需添加进去。各种配置完成后就可以下载Data Studio,然后运行 连接即可，连接成功后可以看到数据库内的各种表格，并且可以直接图形化的查看、修改数据非常方便，连接成功后的截图如下：



### 创建表格

从上图其实已经可以看出表格已创建完毕，对于成绩管理系统，分析后需要创建两个表格。一个是student（学生）表，有七个字段，分别是id（学号）、name（姓名）、class（班级）、chinese（语文成绩）、math（数学成绩）、english（英语成绩）、password（密码）。其中id字段是主码，对于name、class与password都应该是非空，三科成绩在0~100之间。另一个是teacher（老师）表，有四个字段，分别是id（教师工号）、name（姓名）、class（班级）、password（密码）。其中id是主码，其他三个字段都应该非空。

实验结束后对于字段的反思：对于学生和老师的描述数据过于的少，并没有性别、年龄等基本信息，算是一点小瑕疵，不过问题不大，只是对于数据的展现有些少，也算是一个小教训。

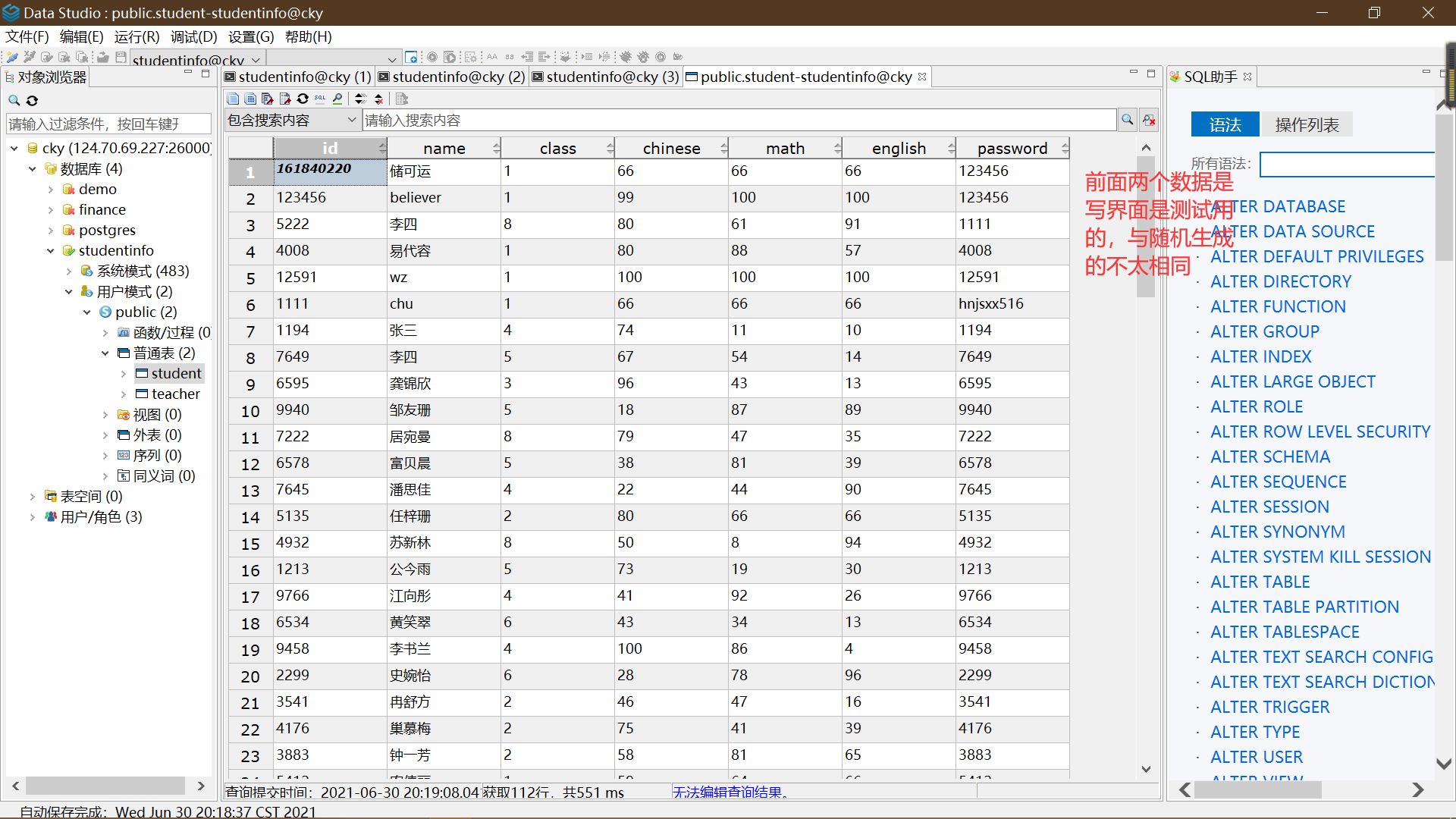
表格的具体介绍如上，对于班级和成绩用的是int型，其他都是character varying。创建表格可以用上面提到的Data Studio软件，创建过程为可视化，较为方便快捷，具体过程不再赘述。

### 生成数据

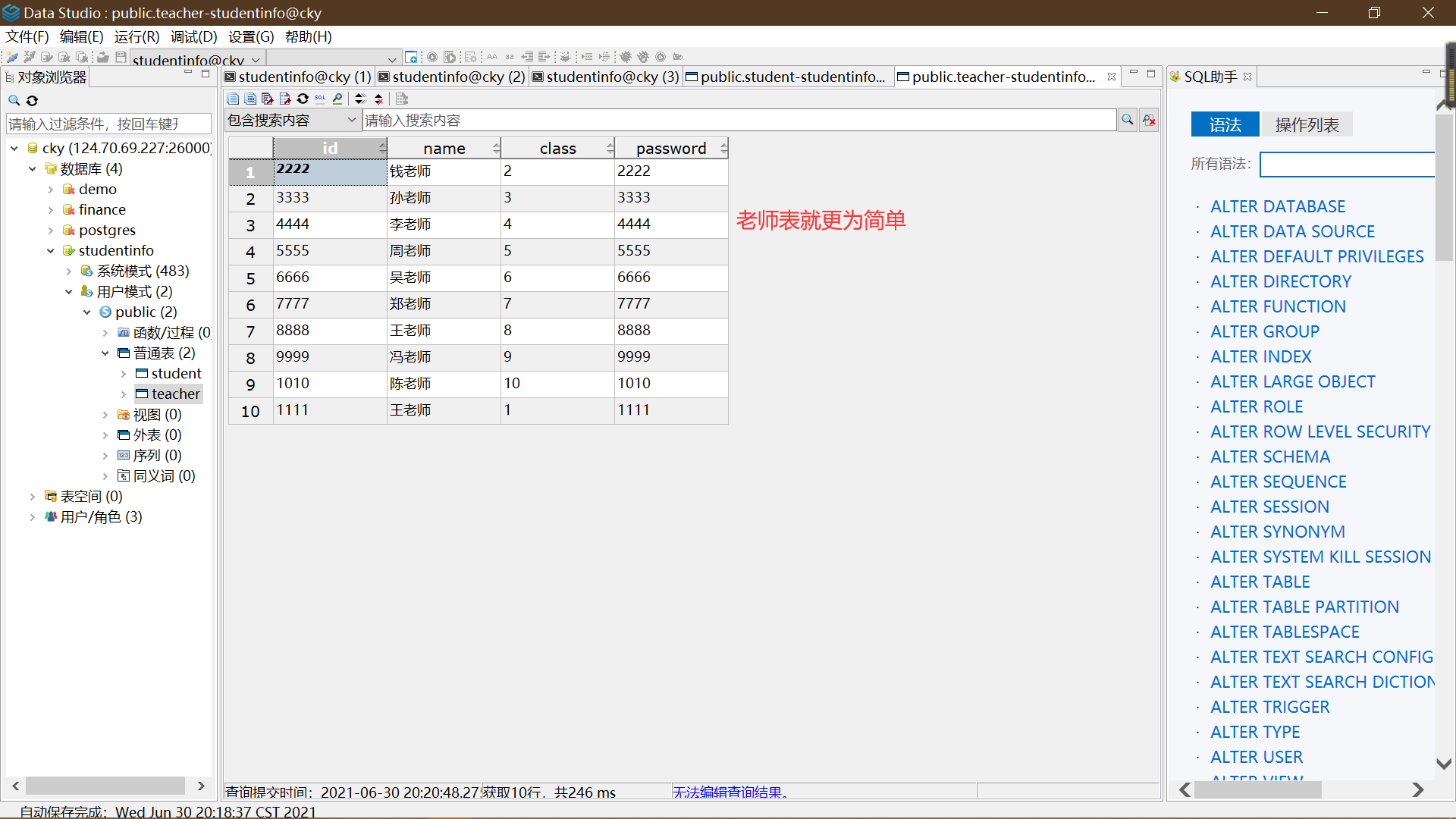
表格建立好之后，还需要向其中添加数据。对于student（学生）表，id字段采用1000~9999随机数生成，然后去重即可；name字段从网上名字生成器中随机生成；class字段采用1~10随机数生成；成绩均采用0~100随机数生成（其实采取正态分布生成更好，但是时间紧未在Excel中实现）；password与id相同。对于teacher（老师）表，生成方法与student（学生）表相同。

数据生成后，在Data Studio软件中，还可以通过txt等文件直接将数据导入到创建的数据库表格中，很是方便。下图是导入成功后的学生表与老师表：

学生表：



老师表：



## 成绩管理系统的设计

### 连接数据库

首先最重要的是连接数据库，下面是连接数据库最为重要的语句：

        db=new OperationMySql();

        // 连接数据库

        db.setDburl("jdbc:postgresql://124.70.69.227:26000/studentinfo?ApplicationName=app1");

        // 加载驱动

        db.setDbdriver("org.postgresql.Driver");

        // 这里的用户名和密码是要和你的数据库对应的，也是唯一需要更改的地方

        db.setUsername("dbuser");db.setPassword("Gauss#3demo");

        // 如果单击【退出】按钮则程序退出

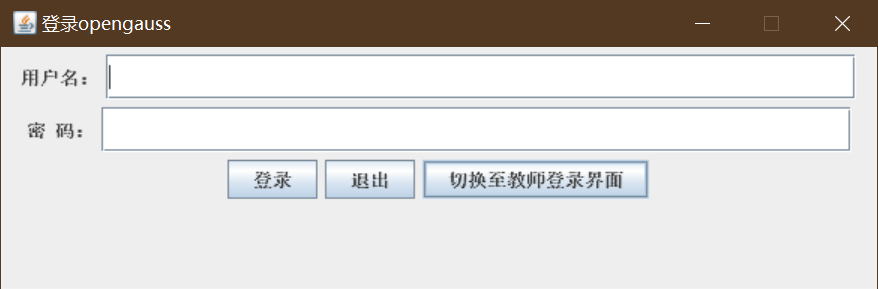
这里设置了数据库的URL、账号、密码，还有加载的驱动。Opengauss的驱动是从官网下载的在vscode中需要将驱动加入到Referenced Libraries中才能正常运行。对于数据库的操作函数，都写在OperationSql.java中。对于每一项的操作下面介绍到了再详细解说。

### 学生登录界面以及成绩界面的设计

对于界面的设计，都采用了java的Swing包，Swing是一个为Java设计的GUI工具包，可以不算很方便的进行界面设计。

对于学生，有Login.java、ChangePassword.java、studentGUI.java这三个java文件。下面对其做一一说明。

Login.java，完成的是学生登录界面的设计，由于界面设计的不方便，登录界面也就没有花里胡哨，就两个提示输入学号密码的标签，后面对应两个文本框，还有三个按钮，登录界面如下：



其中，切换至教师登录界面，就可以转换成教师的登录界面。

账号密码的匹配，是从用户名读取数据，再到数据库中去查询相关学生，如果没查到就是输入用户不存在；若查到了数据，还要去比较用户输入的密码与数据库中存储的密码是否相同，相同则跳转至studentGUI，不同则提示密码错误。

StudentGUI.java,完成的是学生成绩的展现，由于此界面需要添加的组件过多，还是挺麻烦的。不在赘述各个组件，直接看成绩单界面图：

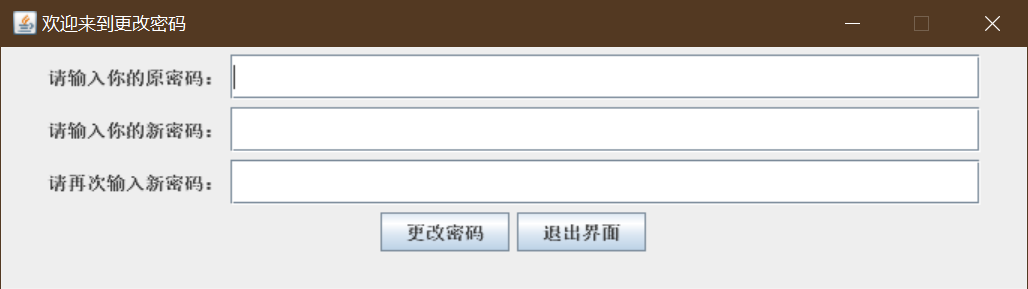


图中的信息都是根据用户登录时的id进行查询得到的，关于排名，是根据用户的班级，再到数据库中查询进行计算得到的。

点击退出界面就可以退出此界面

点击更改密码，就会跳到changepassword界面。

ChangePassword.java，完成的是更改密码的功能，界面如下：



对于原密码，要与数据库中的密码一致，否则无法修改

对于两次输入的新密码，要保持一致，否则会弹窗提示两次输入的密码不一致。

更改密码的关键语句如下：

            String sql = "update student set " + "password" + " = ? where ID = ?";

当上面条件都完成是，就可以对于数据库中的这个同学的密码进行修改，修改才做在OperationSql.java中，就是把信息拼成sql语句，再传给数据库执行即可。

整体就是学生操作的各个界面，我并没有设计学生的注册界面，自己的想法就是学生的添加删除都应该由来时来操作，学生应该只需要查询即可，学生的账号都是由老师发放。

### 老师登录界面以及操作界面的设计

对于老师，有teacherLogin.java、teacherGUI.java两个java文件，下面对其进行一一解释。

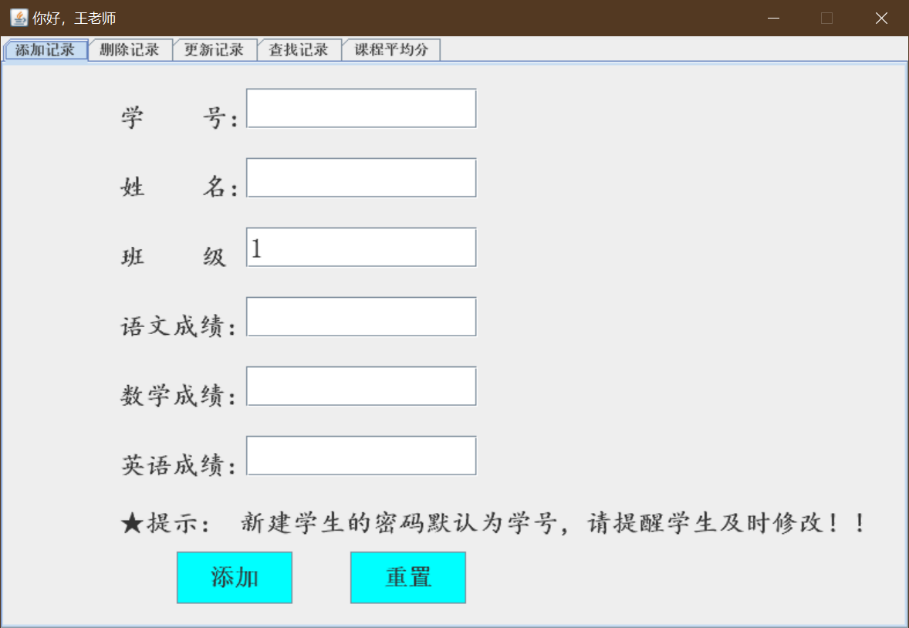
TeacherLogin.java,实现的是教师的登录功能，与学生的类似，不在过多解释：



当然从教师的登录界面也可以切换到学生的界面。

teacherGUI.java，实现的是教师的许多功能，界面的设计与代码书写及其的复杂（头秃），下面对每一个功能进行展示：

首先是添加学生

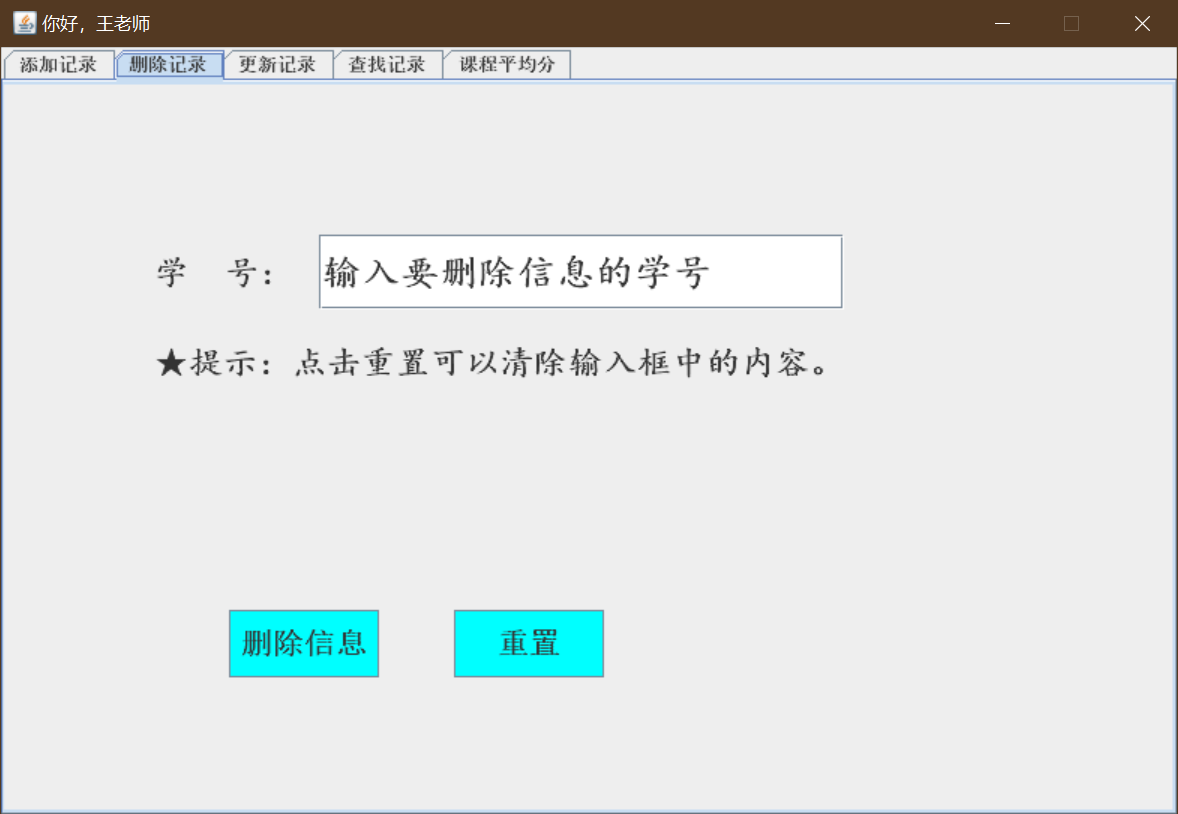


对于添加，学号和姓名不能为空，班级默认为老师的班级，成绩可以为空（可以在之后的更新中进行修改）。新插入的学生的密码默认为学号，已在界面中提示。关于向数据库中添加数据，核心语句为：

           stmt = conn.prepareStatement("insert into student values(?,?,?,?,?,?,?)");

只要将？替换成刚才文本框中输入的数据即可（成绩为空则替换为null）。

然后是删除学生：

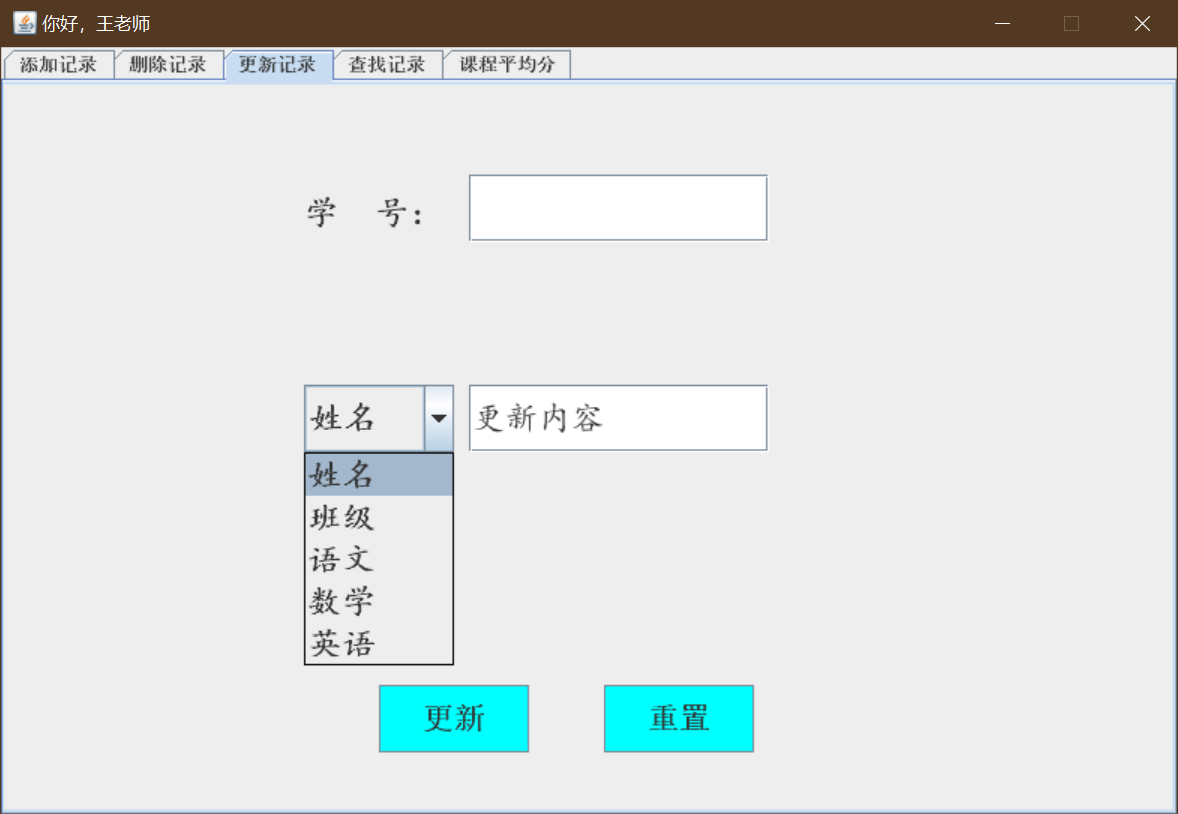


删除学生是需要知道学生的学号（学号为主码），点击重置后再输入想要删除的学生学号即可。如果没有该学号就会报错提示没有该学生。删除操作最主要的代码如下：

           stmt = conn.prepareStatement("delete from student where ID = ?");

只要将？替换成输入的学号即可。再执行操作就行了

更新信息：

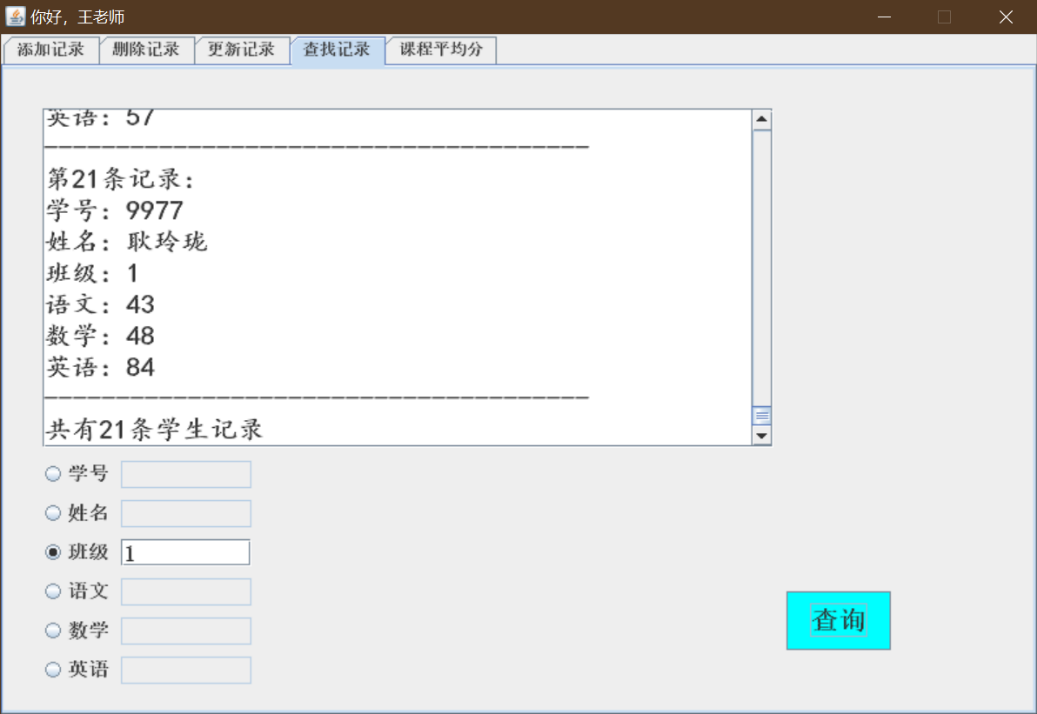


对于更新首先是要输入想要更新的学生的学号，然后可以选择想要更新的字段，点击重置就可以清除嗲文本框的中内容，再输入想要更新的内容即可。更新最主要的代码如下：

            String sql = "update student set " + UpdateItem + " = ? where ID = ?";

其中，UpdateItem是上面选择的想要更新的字段，替换掉第一个的 ？，第二个？是用户登录的id，然后执行该操作即可。

查找记录



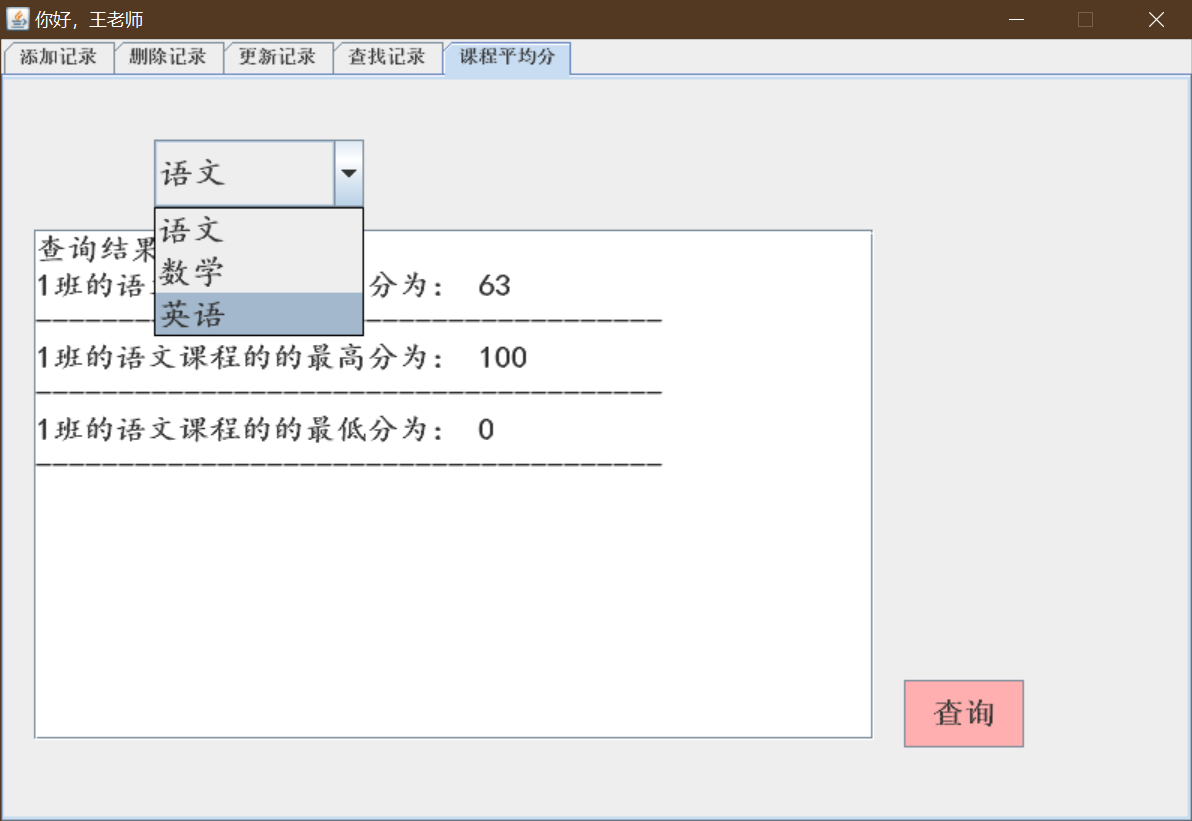
对于查找，可以通过学号、姓名、班级、各科成绩（默认是查询比输入成绩还高的人）来查询数据，也可以选择多个条件进行查询。下面是查询的关键代码：

            String sql = "select \* from student where ID like ? and Name like ? and class like ? and Chinese >= ? " +

                    "and Math >= ? and English >= ? order by ID asc";

对于没有选中的条件，？替换为%，表示都选中，对于选中的条件，只要将？替换成输入的数据即可。把sql语句拼凑成功后再运行就可以了。

课程平均分



可以选择三门科目，然后点击查询按键，就可以知道自己所在班级的这个科目的成绩的最高分，最低分，平均分。对于这些成绩的计算，是根据老师的班级，查询所有学生的成绩，然后根据查询的所有学生的成绩进行计算平均分、最高分、最低分。

以上就是老师所有界面的介绍。

最终，再写一个开始函数（主函数），命名为StartSql.java。代码如下：

package StudentInfo;

public class StartSql {

    // 启动登录界面

    public static void main(String[] args) {

        new teacherLogin();

    }

}

这可以作为老师的应用程序，将主函数中的函数换成学生的登录界面Login（），即可作为学生的应用程序。

实验最后，我又上网查询了一下如何将java项目打包成exe文件。具体过程如下：首先在vscode中将项目导出成.jar文件，然后利用exe4j Wizard软件将.jar文件再打包成.exe文件即可。这里有一个小插曲，本来打包完成后我将exe文件发送给同学测试，发现有的同学无法运行，是因为其电脑上没有jre环境，再次搜索教程后，并没有顺利解决（还花了一个多小时，累死），最后也不知道其中的原因是什么，明明已经打包好了环境，却无法运行，最后打包过程遗憾结束。

# 数据库的维护

对于才开始写成绩管理系统时，学生表格我只输入了自己的信息，老师也只添加了一个，然后进行界面的开发与试运行，这样方便对于界面的书写与管理。然后对于之后的数据库中的数据可能存在的数据增长，首先是性能问题，数据过多的话查询效率就会降低，需要添加索引，对于学生来说在班级上添加索引最合适；其次是数据库的备份问题，学生的数据，尤其是成绩还是十分重要的，需要做好备份与恢复工作。

# 实验的总结与感想

这个为期四周多的数据库实验，最终告一段落。刚刚接触这个实验，感觉毫无头绪，难如登天，而且自己还头铁选择了opengauss数据库。第一周的学习中，连数据库的搭建都未完成，之后自己课余时间跟着指导手册完成了数据库的搭建，之后创建表格的过程就比较快了。但是，对于界面的设计与书写，这是之前很少接触到的，而且连接opengauss数据库使用的是JDBC，然后需要用java来开发界面。本来是想做一个网页端的界面，然后放到服务器上，但是由于IDEA并不能连接到opengauss数据库上，最后只能作罢。这个过程又耽误了好几天的时间，最终还是选择了最为原始的java的Swing库进行界面的开发。这个GUI库对于界面的开发需要自己一个个的调试，去摆放组件的位置，还是比较麻烦的。虽说这个过程比较复杂与繁琐，但是还是从中学到了许多东西，还有了一次不同的体验。首先是对于课上所学的知识有了一次实践，巩固了自己所学是数据库知识，然后对于界面的开发，是一次从无到有的经历，也有了开发java项目的经验。最后这里要感谢老师的教导与同学的帮助，没有你们的支持与帮助我是很难完成实验的。