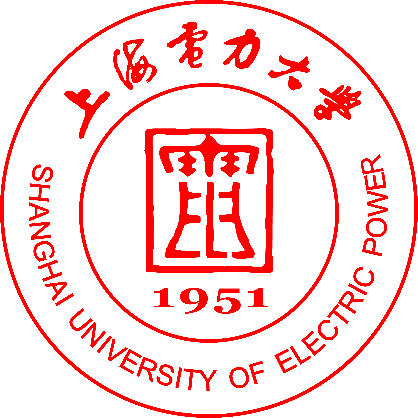
上海电力大学

数据库应用课程设计



**题　　目:**

**学生姓名： 龚释冰**

**学 号： 20211576 班 级： 2021222**

**院　　系：　　 计算机科学与技术学院**

**专业班级： 软件工程专业2021222**

**2023年 6 月 20 日**

目录

[一、需求分析 3](#_Toc138153785)

[（1）问题描述 3](#_Toc138153786)

[（2）系统功能描述 3](#_Toc138153787)

[（3）安全性与完整性方面要求 4](#_Toc138153788)

[二、概念结构设计 5](#_Toc138153789)

[三、逻辑结构设计 6](#_Toc138153790)

[**（**1）模式设计 6](#_Toc138153791)

[（2）子模式设计 9](#_Toc138153792)

[四、数据库的物理设计 15](#_Toc138153793)

[五、数据库设计实现及运行 16](#_Toc138153794)

[（1）数据库的创建 16](#_Toc138153795)

[（2）数据表的创建 16](#_Toc138153796)

[（3）视图的创建 18](#_Toc138153797)

[（4）存储过程的定义实现 22](#_Toc138153798)

[（5）触发器的定义实现 23](#_Toc138153799)

[（6）自行设计各模块中所涉及的操作语句 27](#_Toc138153800)

[（7）原子性事务显示定义：无 87](#_Toc138153801)

[六、系统详细设计及实现 87](#_Toc138153802)

[（1）系统功能模块说明 87](#_Toc138153803)

[（2）每个模块的关键语句及关键技术说明 87](#_Toc138153804)

[七、总结 87](#_Toc138153805)

一、需求分析

（1）问题描述

系统功能大致有：

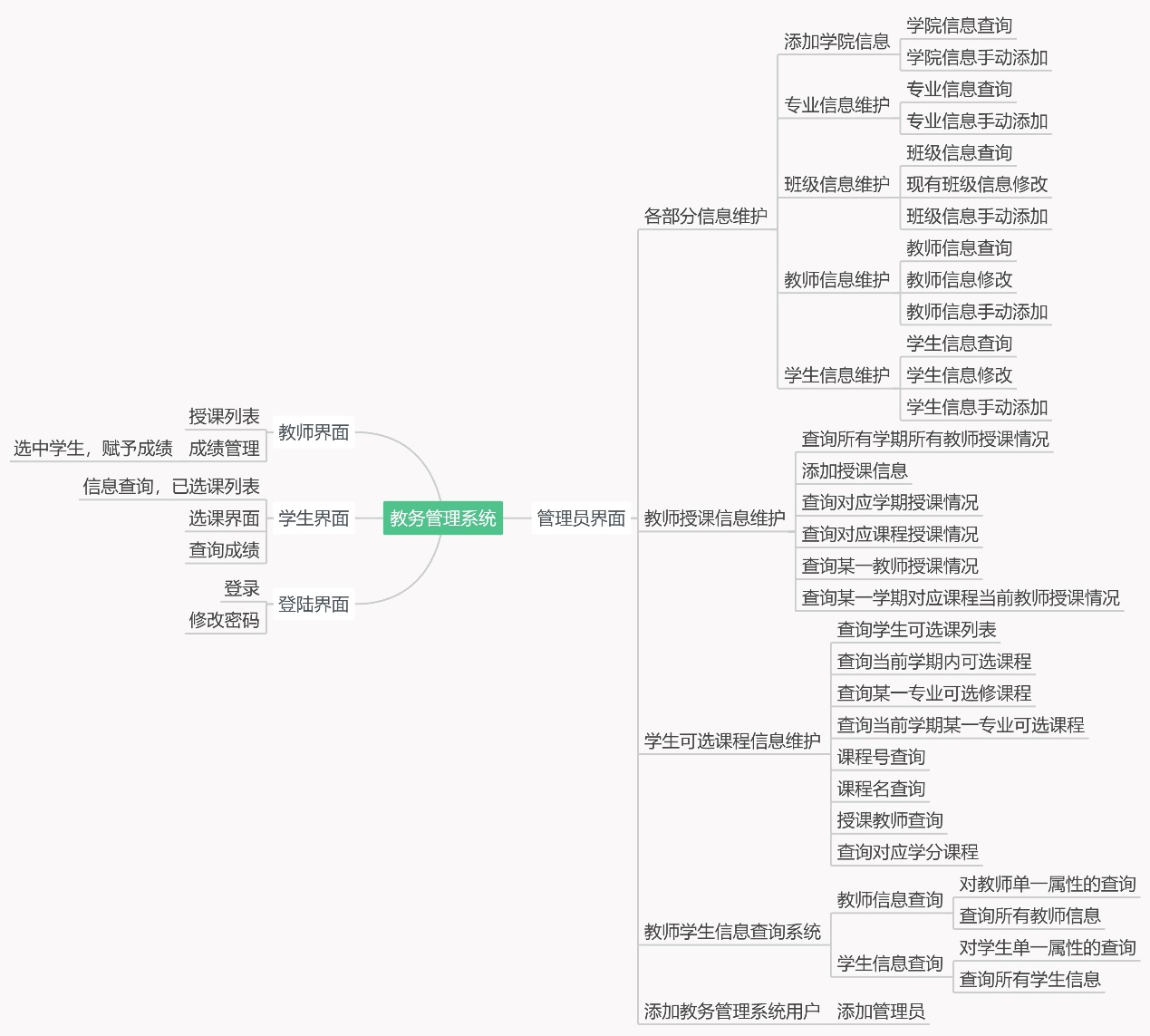
教师端：授课列表查询，对应课程赋分。

学生端：信心查询，已选课列表；选课；查询成绩。

管理员端：对学院、专业、班级、教师、学生信息的维护；对教师授课情况的维护；对学生可选课程的维护；对教师学生信息的查询；添加教务管理人员账号。

连接方法：通过登陆界面区分不同用户，分别打开各自的界面（区分选项是id），根据权限大小分别对应不同的界面。

（2）系统功能描述

对自己所负责的功能模块，可画出所设计模块的操作流程；并分析所需要存储的数据信息。

1、Academy\_20211576存储Aca\_no，Aca\_name，Aca\_introduction。

2、Sdept\_20211576存储Sde\_no，Sde\_name，Sde\_introduction，Aca\_no，Aca\_name

3、Class\_20211576存储Class\_no，Sde\_no，Entrance\_date，Num

4、Student\_20211576存储Sno，Sname，Sex，Birth，Entrance\_date，Class\_no，Home\_addr，Sde\_no，Postcode，AttendanceStatus。

5、Course\_20211576存储Cno，Aca\_no，Cname， Total\_perior，Credit。

6、Teacher\_20211576存储Emp\_no，Emp\_name，Sex，Sde\_no，Position。

7、Position\_20211576存储Position。

8、Course\_Teacher\_20211576存储Emp\_no，Cno，Cname。

9、Course\_Available\_20211576存储Semester，Emp\_no，Cno，Course\_Offered\_Sde\_no，Cname，Credit，Stu\_Total，Stu\_chosen。

10、Course\_chosen\_20211576存储Semester，Emp\_no，Sno，Cno，Course\_Offered\_Sde\_no，

Cname，Credit，Stu\_Total，Stu\_chosen。

11、SC\_20211576存储Semester，Sno，Cno，Emp\_no，Sde\_no，Grade。

12、Password\_20211576存储id， name， position， password。

（3）安全性与完整性方面要求

（一）没有定义显示定义原子事务

（二）没有定义存储过程

（三）定义了以下功能的触发器

1（trg\_UpdateEntranceDate\_Class）当Class\_20211576表中加入新数据时，会在添加数据后自动将入学时间Entrance\_date更改为同年9/1；

2（check\_records\_CourseAvail\_Course\_CourseTea）当向Course\_Available\_20211576添加数据时，先判断其对应的组合在Course\_20211576,Course\_Teacher\_202111576中是否存在，若不存在，添加失败并报错；

3（trg\_insert\_from\_course\_chosen\_to\_SC）当向Course\_chosen\_20211576中插入数据时，会在插入成功后在SC\_20211576中插入对应信息；

4（update\_stu\_chosen）当向Course\_chosen\_20211576中插入数据时，会更新在Course\_Available\_20211576中的对应行的Stu\_chosen(Stu\_chosen+=1);

5（add\_teacher\_password）当向Teacher\_202111576中插入数据时，会先判断这条信息在Password\_20211576中是否存在，若存在，则在Password\_20211576中插入对应数据，并设置默认密码为123456；

6（delete\_Teacher\_Password）当要删除Teacher\_20211576中的信息时，执行instead of delete操作，先分别删除Course\_chosen\_20211576,Password\_20211576,Course\_Available\_20211576,Course\_Teacher\_20211576这几张表中对应的信息，最后再删除Teacher\_20211576中的信息；

7（trg\_NewStudent）当向Teacher\_202111576中插入数据时，在Password\_20211576中插入对应数据，并设置默认密码为123456；

8（delete\_Student\_Password）当要删除Teacher\_20211576中的信息时，执行instead of delete操作，先分别删除Course\_chosen\_20211576,Password\_20211576,SC\_20211576这几张表中对应的信息，最后再删除Student\_20211576中的信息；

9（trg\_delete\_info）当删除Academy\_20211576中信息时，先删除Password\_20211576中属于Academy\_20211576中对应删除学院的专业的教师和学生的信息，再删除Class\_20211576中对应删除学院对应专业名下的班级，再删除SC\_20211576中对应删除学院的对应专业的考试成绩信息，再删除Course\_chosen\_20211576中对应删除学院的对应专业的已选课信息，再删除Student\_20211576对应删除学院的专业的学生信息，再删除Teacher\_20211576中对应删除学院的专业的教师信息，再删除Sdept\_20211576中对应删除学院所属的专业信息，再删除Course\_Available\_20211576中对应删除学院的对应专业的可选课程，再删除Course\_Teacher\_20211576中对应删除学院的对应课程及对应的教师授课信息，最后在删除Course\_20211576中对应删除学院开设的课程信息。

10（delete\_course\_trigger）首先在SC\_20211576表中删除与该课程相关的成绩信息，在Course\_chosen\_20211576表中删除与该课程相关的选课信息，在Course\_Available\_20211576表中删除该课程的开课信息，在Course\_Teacher\_20211576表中删除该课程的任课信息，最后在Course\_20211576表中删除该课程的基本信息

11（delete\_course\_teacher\_trigger）首先删除SC\_20211576中与该课程相关的成绩信息，然后删除Course\_chosen\_20211576中与该课程相关的选课信息，然后删除Course\_Available\_20211576中该课程的开课信息，最后删除Course\_Teacher\_20211576中该课程的任课信息。

12（delete\_course\_available\_trigger）首先删除SC\_20211576中与该课程相关的成绩信息，然后删除Course\_chosen\_20211576中与该课程相关的选课信息，最后删除Course\_Available\_20211576中该课程的开课信息。

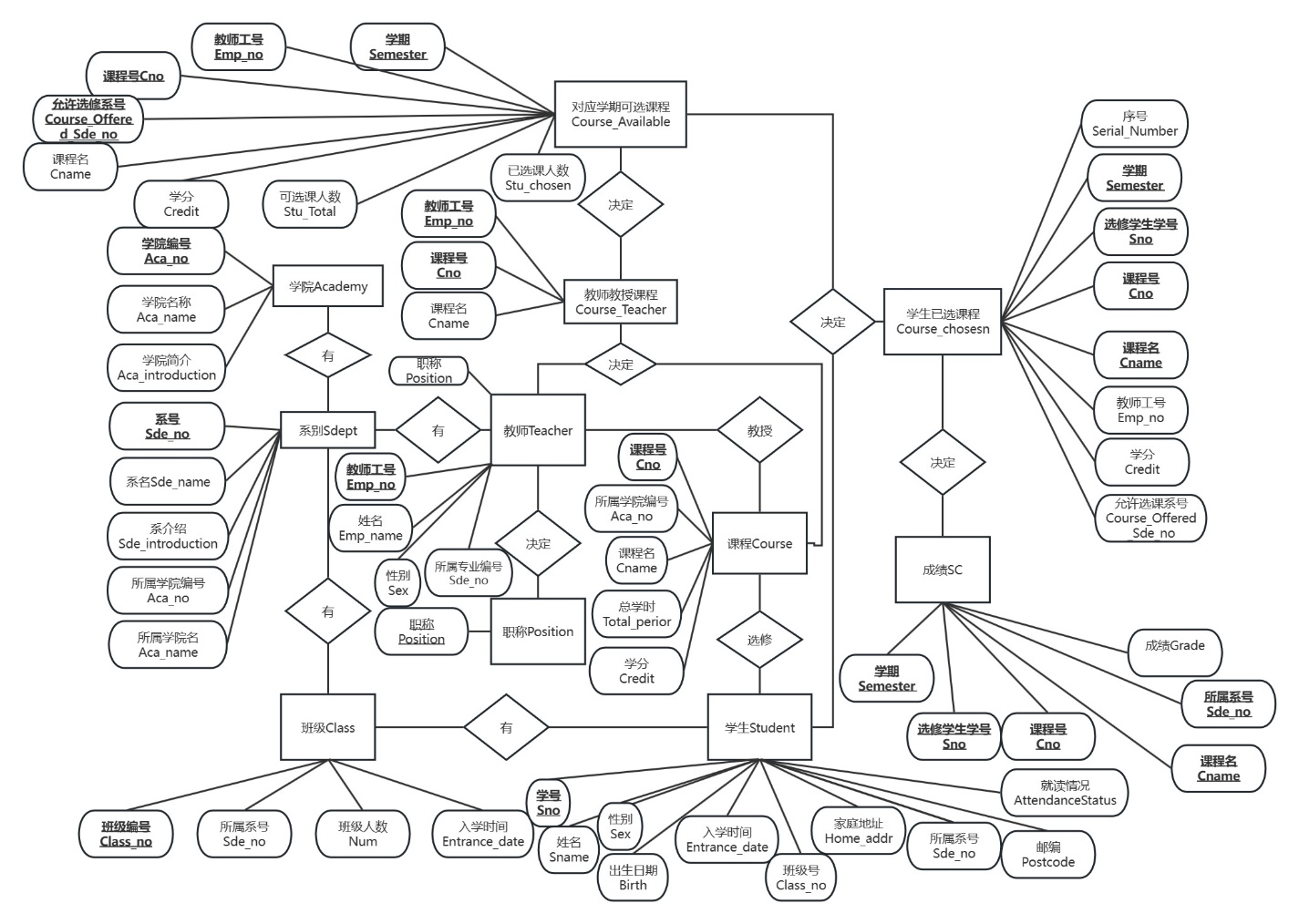
13（delete\_course\_chosen\_trigger）首先删除SC\_20211576中对应的成绩信息，再删除Course\_chosen\_20211576中的已选课信息。

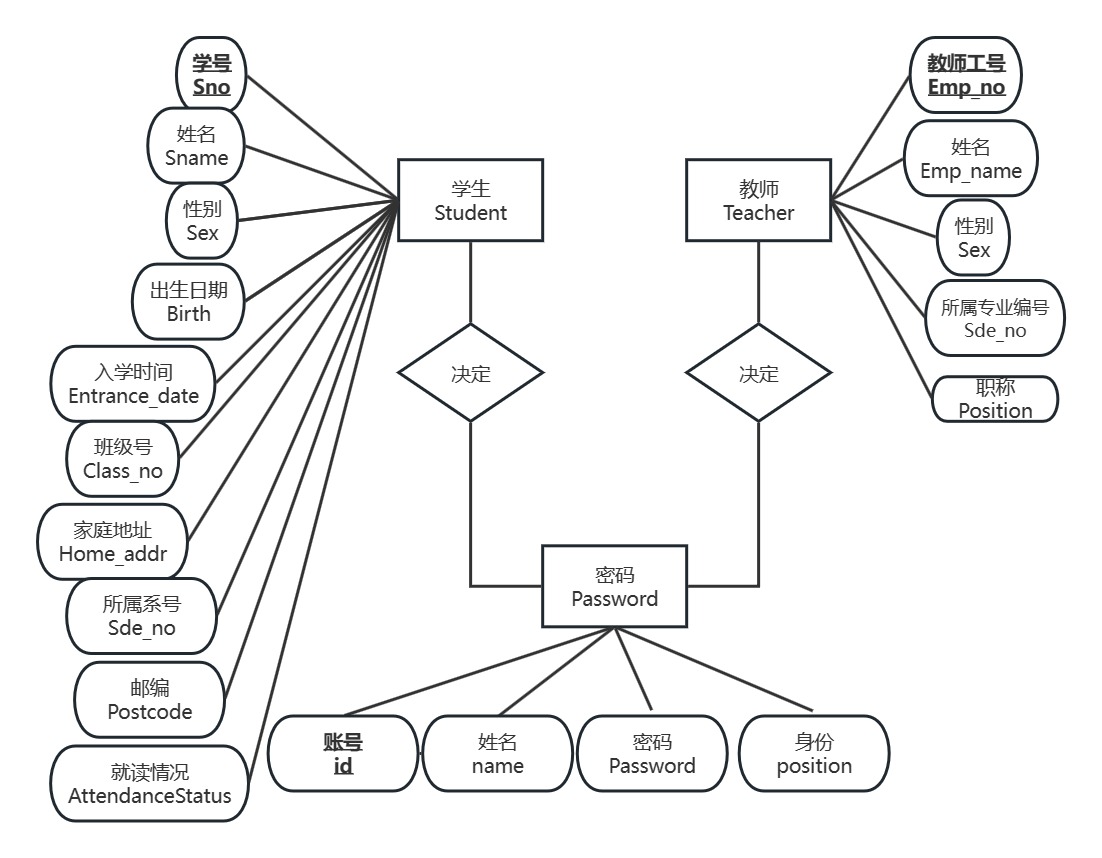
（四）对不同用户采取登陆时鉴别身份的操作

对管理员用户给予全部操作权限，对学生仅提供视图读取权限，对教师，在对学科进行赋分时给予更新权限，当输入成绩后，对已赋分部分仅有视图读取权限，对除赋分以外的其他操作仅有视图读取权限。

二、概念结构设计

画出系统整体的E-R模型；并对模型中所出现的实体及属性等信息加以说明。





实体及属性：

实体1、Academy\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Aca\_no | char | 3 | 否 | 学院编号 |
| Aca\_name | varchar | 20 | 否 | 学院名 |
| Aca\_introduction | char | 100 | 是 | 学院介绍 |

实体2、Sdept\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Sde\_no | char | 3 | 否 | 系号 |
| Sde\_name | Char | 20 | 否 | 系名 |
| Sde\_introduction | Char | 100 | 是 | 系介绍 |
| Aca\_no | Char | 3 | 否 | 所属学院编号 |
| Aca\_name | char | 20 | 是 | 所属学院名 |

实体3、Class\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Class\_no | char | 3 | 否 | 班级号 |
| Sde\_no | char | 3 | 否 | 所属系号 |
| Entrance\_date | smalldatetime | 100 | 否 | 入学时间 |
| Num | int |  | 是 | 班级人数 |

实体4、Student\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Sno | char | 9 | 否 | 学号 |
| Sname | Varchar | 8 | 否 | 姓名 |
| Sex | Char | 2 | 否 | 性别 |
| Birth | Smalldatetime |  | 是 | 出生日期 |
| Entrance\_date | Smalldatetime |  | 是 | 入学时间 |
| Class\_no | Char | 7 | 是 | 所属班级号 |
| Home\_addr | Varchar | 40 | 是 | 家庭地址 |
| Sde\_no | Char | 3 | 否 | 所属专业号 |
| Postcode | Char | 6 | 是 | 邮编 |
| AttendanceStatus | char | 20 | 是 | 就读状态 |

实体5、Course\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Cno | char | 10 | 否 | 课程号 |
| Aca\_no | Char | 3 | 否 | 所属学院编号 |
| Cname | Varchar | 40 | 否 | 课程名 |
| Total\_perior | Smallint |  | 是 | 总学时 |
| Credit | Float |  | 是 | 学分 |

实体6、Teacher\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Emp\_no | char | 9 | 否 | 工号 |
| Emp\_name | varchar | 8 | 否 | 姓名 |
| Sex | Char | 2 | 是 | 性别 |
| Sde\_no | Char | 10 | 否 | 所属系号 |
| Position | char | 10 | 否 | 职称 |

实体7、Position\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Position | char | 20 | 否 | 职称 |

实体8、Course\_Teacher\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Emp\_no | char | 9 | 否 | 工号 |
| Cno | Char | 10 | 否 | 课程号 |
| Cname | Varchar | 20 | 是 | 课程名 |

实体9、Course\_Available\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Semester | char | 20 | 否 | 学期 |
| Emp\_no | Char | 9 | 否 | 工号 |
| Cno | Char | 10 | 否 | 课程号 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 | 允许选修系的系号 |
| Cname | Varchar | 20 | 是 | 课程名 |
| Credit | Tinyint |  | 是 | 学分 |
| Stu\_Total | Tinyint |  | 是 | 可选课学生数 |
| Stu\_chosen | tinyint |  | 否 | 已选学生数 |

实体10、Course\_chosen\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Semester | char | 20 | 否 | 学期 |
| Emp\_no | Char | 9 | 否 | 工号 |
| Sno | char | 9 | 否 | 学号 |
| Cno | Char | 10 | 否 | 课程号 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 | 允许选修系的系号 |
| Cname | Varchar | 20 | 是 | 课程名 |
| Credit | Tinyint |  | 是 | 学分 |

实体11、SC\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| Semester | char | 20 | 否 | 学期 |
| Sno | Char | 9 | 否 | 学号 |
| Cno | char | 10 | 否 | 课程号 |
| Emp\_no | Char | 9 | 否 | 授课教师工号 |
| Sde\_no | Char | 10 | 否 | 允许选修系的系名 |
| Grade | tinyint |  | 是 | 分数 |

实体12、Password\_20211576

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 宽度 | 允许空值 | 说明 |
| id | char | 9 | 否 | 学号/工号 |
| name | Varchar | 8 | 否 | 姓名 |
| position | Char | 20 | 否 | 职称 |
| password | char | 20 | 是 | 密码 |

三、逻辑结构设计

**（**1）模式设计

按系统整体E-R模型，写出关系模式；并利用数据字典加以描述。

1、Academy\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 |
| Aca\_no | char | 3 | 否 |  | 主键约束 | 是 |
| Aca\_name | varchar | 20 | 否 |  |  |  |
| Aca\_introduction | char | 100 | 是 | 无 |  |  |

2、Sdept\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Sde\_no | char | 3 | 否 |  | 主键约束 | 是 |  |
| Sde\_name | Char | 20 | 否 |  |  |  |  |
| Sde\_introduction | Char | 100 | 是 | 无 |  |  |  |
| Aca\_no | Char | 3 | 否 |  | 外键约束（Academy\_20211576.Aca\_no） |  | 是 |
| Aca\_name | char | 20 | 是 |  |  |  |  |

3、Class\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Class\_no | char | 3 | 否 |  | 主键约束 | 是 |  |
| Sde\_no | char | 3 | 否 |  | 外键约束(Sdept\_20211576.Sde\_no) |  | 是 |
| Entrance\_date | smalldatetime | 100 | 否 |  |  |  |  |
| Num | int |  | 是 |  |  |  |  |

4、Student\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Sno | char | 9 | 否 |  | 主键约束 | 是 |  |
| Sname | Varchar | 8 | 否 |  |  |  |  |
| Sex | Char | 2 | 否 | 男 |  |  |  |
| Birth | Smalldatetime |  | 是 |  |  |  |  |
| Entrance\_date | Smalldatetime |  | 是 |  |  |  |  |
| Class\_no | Char | 7 | 是 |  | 外键约束（Class\_20211576.Class\_no） |  |  |
| Home\_addr | Varchar | 40 | 是 |  |  |  |  |
| Sde\_no | Char | 3 | 否 |  | 外键约束（Sdept\_20211576.Sde\_no） |  | 是 |
| Postcode | Char | 6 | 是 |  | 检查约束（Like’[0-9] [0-9] [0-9] [0-9] [0-9] [0-9]’） |  |  |
| AttendanceStatus | char | 20 | 是 |  |  |  |  |

5、Course\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Cno | char | 10 | 否 |  | 主键约束 | 是 |  |
| Aca\_no | Char | 3 | 否 |  | 外键约束（Academy\_20211576.Aca\_no） |  | 是 |
| Cname | Varchar | 40 | 否 |  |  |  |  |
| Total\_perior | Smallint |  | 是 |  |  |  |  |
| Credit | Float |  | 是 |  |  |  |  |

6、Teacher\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Emp\_no | char | 9 | 否 |  | 主键约束 | 是 |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |  |
| Sex | Char | 2 | 是 | 男 |  |  |  |
| Sde\_no | Char | 10 | 否 |  | 外键约束（Sdept\_20211576.Sde\_no） |  |  |
| Position | char | 10 | 否 |  | 外键约束（Position\_20211576.Position） |  | 是 |

7、Position\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Position | char | 20 | 否 |  | 主键约束 | 是 |  |

8、Course\_Teacher\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Emp\_no | char | 9 | 否 |  | 外键约束（Teacher\_20211576.Emp\_no） | 是 | 是 |
| Cno | Char | 10 | 否 |  | 外键约束（Course\_20211576.Cno） | 是 | 是 |
| Cname | Varchar | 20 | 是 |  |  |  |  |

主键约束：（Emp\_no,Cno）

9、Course\_Available\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Semester | char | 20 | 否 |  |  | 是 | 是 |
| Emp\_no | Char | 9 | 否 |  |  | 是 | 是 |
| Cno | Char | 10 | 否 |  | 外键约束（Course\_20211576.Cno） | 是 | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 | 是 |
| Cname | Varchar | 20 | 是 |  |  |  |  |
| Credit | Tinyint |  | 是 |  |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |  |
| Stu\_chosen | tinyint |  | 否 |  |  |  |  |

主键约束：（Semester,Emp\_no,Cno,Course\_Offered\_Sde\_no）

Emp\_no,Cno外键约束:

|  |  |
| --- | --- |
| 主键表 | 外键表 |
| Course\_Teacher\_20211576 | Course\_Available\_20211576 |
| Emp\_no | Emp\_no |
| Cno | Cno |

10、Course\_chosen\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Semester | char | 20 | 否 |  |  | 是 | 是 |
| Emp\_no | Char | 9 | 否 |  |  | 是 | 是 |
| Sno | char | 9 | 否 |  | 外键约束（Student\_20211576.Sno） | 是 | 是 |
| Cno | Char | 10 | 否 |  |  | 是 | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 | 是 |
| Cname | Varchar | 20 | 是 |  |  |  |  |
| Credit | Tinyint |  | 是 |  |  |  |  |

Semester,Cno,Emp\_no,Course\_Offered\_Sde\_no外键约束：

|  |  |
| --- | --- |
| 主键表 | 外键表 |
| Course\_Available\_20211576 | Course\_chosen\_20211576 |
| Semester | Semester |
| Cno | Cno |
| Emp\_no | Emp\_no |
| Course\_Offered\_Sde\_no | Course\_Offered\_Sde\_no |

主键约束：（Semester,Emp\_no,Sno,Cno,Course\_Offered\_Sde\_no）

11、SC\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| Semester | char | 20 | 否 |  |  | 是 | 是 |
| Sno | Char | 9 | 否 |  |  | 是 | 是 |
| Cno | char | 10 | 否 |  |  | 是 | 是 |
| Emp\_no | Char | 9 | 否 |  |  | 是 | 是 |
| Sde\_no | Char | 10 | 否 |  |  | 是 | 是 |
| Grade | tinyint |  | 是 |  |  |  |  |

主键约束：（Semester,Emp\_no,Sno,Cno,Course\_Offered\_Sde\_no）

Semester,Cno,Emp\_no,Course\_Offered\_Sde\_no外键约束：

|  |  |
| --- | --- |
| 主键表 | 外键表 |
| Course\_chosen\_20211576 | SC\_20211576 |
| Semester | Semester |
| Sno | Sno |
| Cno | Cno |
| Emp\_no | Emp\_no |
| Course\_Offered\_Sde\_no | Sde\_no |

12、Password\_20211576

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 | 外键 |
| id | char | 9 | 否 |  |  | 是 |  |
| name | Varchar | 8 | 否 |  |  |  |  |
| position | Char | 20 | 否 |  | 外键约束（Position\_20211576.Position） |  | 是 |
| password | char | 20 | 是 |  |  |  |  |

（2）子模式设计

针对自己的设计模块的应用需定义什么样的视图表结构；并利用数据字典加以描述。

（一）登录功能：Manager\_Password\_view

create view Manager\_Password\_view as

select id,name,position,password

from Password\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| id | char | 9 | 否 |  |  | 是 |
| name | Varchar | 8 | 否 |  |  |  |
| position | Char | 20 | 否 |  |  |  |
| password | char | 20 | 是 |  |  |  |

（二）登陆后教师界面

1))Teacher\_view (展示教师信息)

create view Teacher\_view as

select Emp\_no,Emp\_name,Sex,Teacher\_20211576.Sde\_no,Sde\_name,Position

from Teacher\_20211576,Sdept\_20211576

where Teacher\_20211576.Sde\_no=Sdept\_20211576.Sde\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Emp\_no | char | 9 | 否 |  |  | 是 |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 是 | 男 |  |  |
| Sde\_no | Char | 10 | 否 |  |  |  |
| Sde\_name | char | 20 | 否 |  |  |  |
| Position | char | 10 | 否 |  |  |  |

2))Manager\_Course\_Teacher\_manage\_view(查询教授课程)

CREATE VIEW Manager\_Course\_Teacher\_manage\_view AS

SELECT Semester,Course\_Offered\_Sde\_no,Sde\_name,Cno,Cname,Emp\_name,Credit,Stu\_total

FROM Course\_Available\_20211576,Teacher\_20211576,Sdept\_20211576

where Course\_Available\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and Sdept\_20211576.Sde\_no=Course\_Available\_20211576.Course\_Offered\_Sde\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 |
| Sde\_name | char | 20 | 否 |  |  | 是 |
| Cno | Char | 10 | 否 |  |  | 是 |
| Cname | Varchar | 20 | 否 |  |  |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Credit | float |  | 是 |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |

Teacher\_Manage\_SC\_view(赋分)：

create view Teacher\_Manage\_SC\_view as

select Semester,Sno,SC\_20211576.Cno,Cname,Emp\_name,SC\_20211576.Sde\_no,Grade

from Teacher\_20211576,SC\_20211576,Course\_20211576

where SC\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and SC\_20211576.Cno=Course\_20211576.Cno

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Sno | Char | 9 | 否 |  |  | 是 |
| Cno | char | 10 | 否 |  |  | 是 |
| Cname | varchar | 8 | 否 |  |  |  |
| Emp\_no | Char | 9 | 否 |  |  | 是 |
| Sde\_no | Char | 10 | 否 |  |  | 是 |
| Grade | tinyint |  | 是 |  |  |  |

（三）登陆后学生界面：

（（一））学生信息+已选课程查询：

1））Student\_Enquiry\_view(已选课程查询)

create view Student\_Enquiry\_view as

SELECT Semester,Sno,Cno,Cname,Emp\_name,Credit

FROM Course\_chosen\_20211576,Teacher\_20211576

where Course\_chosen\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Sno | Char | 9 | 否 |  |  | 是 |
| Cno | char | 10 | 否 |  |  | 是 |
| Cname | varchar | 8 | 否 |  |  |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Credit | float |  | 是 |  |  |  |

2））Student\_view(学生信息)

CREATE VIEW Manager\_Student\_view AS

SELECT Sno , Sname , Sex , Birth , Entrance\_date , Class\_no , Home\_addr , Sde\_no , Postcode ,

CASE WHEN (DATEDIFF(year,Entrance\_date,getdate())>=4) THEN '本科毕业' ELSE '本科在读' END AS AttendanceStatus

FROM Student\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Sno | char | 9 | 否 |  |  | 是 |
| Sname | Varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 否 | 男 |  |  |
| Birth | Smalldatetime |  | 是 |  |  |  |
| Entrance\_date | Smalldatetime |  | 是 |  |  |  |
| Class\_no | Char | 7 | 是 |  |  |  |
| Home\_addr | Varchar | 40 | 是 |  |  |  |
| Sde\_no | Char | 3 | 否 |  |  |  |
| Postcode | Char | 6 | 是 |  |  |  |
| AttendanceStatus | char | 20 | 是 |  |  |  |

（（二））学生选课：

Manager\_Course\_Teacher\_manage\_view (选课)

CREATE VIEW Manager\_Course\_Teacher\_manage\_view AS

SELECT Semester,Course\_Offered\_Sde\_no,Sde\_name,Cno,Cname,Emp\_name,Credit,Stu\_total

FROM Course\_Available\_20211576,Teacher\_20211576,Sdept\_20211576

where Course\_Available\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and Sdept\_20211576.Sde\_no=Course\_Available\_20211576.Course\_Offered\_Sde\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 |
| Sde\_name | char | 20 | 否 |  |  | 是 |
| Cno | Char | 10 | 否 |  |  | 是 |
| Cname | Varchar | 20 | 否 |  |  |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Credit | float |  | 是 |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |

（四）登陆后管理员界面

（（一））各部分信息维护

1))Manager\_Academy\_view (添加学院信息)

CREATE VIEW Manager\_Academy\_view AS

SELECT Aca\_no,Aca\_name,Aca\_introduction

FROM Academy\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Aca\_no | char | 3 | 否 |  |  | 是 |
| Aca\_name | varchar | 20 | 否 |  |  |  |
| Aca\_introduction | char | 100 | 是 | 无 |  |  |

2)) Manager\_Sdept\_view(专业信息维护)

CREATE VIEW Manager\_Sdept\_view AS

SELECT Sde\_no,Sde\_name,Sde\_introduction,Aca\_no,Aca\_name

FROM Sdept\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Sde\_no | char | 3 | 否 |  |  | 是 |
| Sde\_name | Char | 20 | 否 |  |  |  |
| Sde\_introduction | Char | 100 | 是 | 无 |  |  |
| Aca\_no | Char | 3 | 否 |  |  |  |
| Aca\_name | char | 20 | 是 |  |  |  |

3)) Manager\_Class\_view(班级信息维护,入学时间取出年份/月份/日/yy-mm-dd)

CREATE VIEW Manager\_Class\_view AS

SELECT Class\_no,Sde\_no,substring(CONVERT(varchar, Entrance\_date, 120),0,11) 'Entrance\_date',Num

FROM Class\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Class\_no | char | 3 | 否 |  |  | 是 |
| Sde\_no | char | 3 | 否 |  |  |  |
| Entrance\_date | smalldatetime | 100 | 否 |  |  |  |
| Num | int |  | 是 |  |  |  |

4)) Manager\_Teacher\_view(教师信息维护)

CREATE VIEW Manager\_Teacher\_view AS

SELECT Emp\_no,Emp\_name,Sex,Sde\_no,Position

FROM Teacher\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主键 |
| Emp\_no | char | 9 | 否 |  |  | 是 |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 是 | 男 |  |  |
| Sde\_no | Char | 10 | 否 |  |  |  |
| Position | char | 10 | 否 |  |  |  |

5)) Manager\_Student\_view(学生信息维护，在最后一栏增加该生是否毕业的项)

CREATE VIEW Manager\_Student\_view AS

SELECT Sno , Sname , Sex , Birth , Entrance\_date , Class\_no , Home\_addr , Sde\_no , Postcode ,

CASE WHEN (DATEDIFF(year,Entrance\_date,getdate())>=4) THEN '本科毕业' ELSE '本科在读' END AS AttendanceStatus

FROM Student\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Sno | char | 9 | 否 |  |  | 是 |
| Sname | Varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 否 | 男 |  |  |
| Birth | Smalldatetime |  | 是 |  |  |  |
| Entrance\_date | Smalldatetime |  | 是 |  |  |  |
| Class\_no | Char | 7 | 是 |  |  |  |
| Home\_addr | Varchar | 40 | 是 |  |  |  |
| Sde\_no | Char | 3 | 否 |  |  |  |
| Postcode | Char | 6 | 是 |  |  |  |
| AttendanceStatus | char | 20 | 是 |  |  |  |

（（二））教师授课管理

1))Manager\_Course\_Teacher\_manage\_view(查询教师授课信息)

CREATE VIEW Manager\_Course\_Teacher\_manage\_view AS

SELECT Semester,Course\_Offered\_Sde\_no,Sde\_name,Cno,Cname,Emp\_name,Credit,Stu\_total

FROM Course\_Available\_20211576,Teacher\_20211576,Sdept\_20211576

where Course\_Available\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and Sdept\_20211576.Sde\_no=Course\_Available\_20211576.Course\_Offered\_Sde\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 |
| Sde\_name | char | 20 | 否 |  |  | 是 |
| Cno | Char | 10 | 否 |  |  | 是 |
| Cname | Varchar | 20 | 否 |  |  |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Credit | float |  | 是 |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |

2))Course\_Available\_20211576(增加教师授课项)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Emp\_no | Char | 9 | 否 |  |  | 是 |
| Cno | Char | 10 | 否 |  |  | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 |
| Cname | Varchar | 20 | 是 |  |  |  |
| Credit | Tinyint |  | 是 |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |
| Stu\_chosen | tinyint |  | 否 |  |  |  |

（（三））学生可选课程信息维护

Manager\_Course\_Teacher\_manage\_view(查询对应系学生可选课程)

CREATE VIEW Manager\_Course\_Teacher\_manage\_view AS

SELECT Semester,Course\_Offered\_Sde\_no,Sde\_name,Cno,Cname,Emp\_name,Credit,Stu\_total

FROM Course\_Available\_20211576,Teacher\_20211576,Sdept\_20211576

where Course\_Available\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and Sdept\_20211576.Sde\_no=Course\_Available\_20211576.Course\_Offered\_Sde\_no

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Semester | char | 20 | 否 |  |  | 是 |
| Course\_Offered\_Sde\_no | Char | 10 | 否 |  |  | 是 |
| Sde\_name | char | 20 | 否 |  |  | 是 |
| Cno | Char | 10 | 否 |  |  | 是 |
| Cname | Varchar | 20 | 否 |  |  |  |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Credit | float |  | 是 |  |  |  |
| Stu\_Total | Tinyint |  | 是 |  |  |  |

（（四））教师学生信息查询系统

1））Manager\_Teacher\_view(教师信息查询)

CREATE VIEW Manager\_Teacher\_view AS

SELECT Emp\_no,Emp\_name,Sex,Sde\_no,Position

FROM Teacher\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Emp\_no | char | 9 | 否 |  |  | 是 |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 是 | 男 |  |  |
| Sde\_no | Char | 10 | 否 |  |  |  |
| Position | char | 10 | 否 |  |  |  |

2)) Manager\_Student\_view(学生信息查询，在最后一栏增加该生是否毕业的项)

CREATE VIEW Manager\_Student\_view AS

SELECT Sno , Sname , Sex , Birth , Entrance\_date , Class\_no , Home\_addr , Sde\_no , Postcode ,

CASE WHEN (DATEDIFF(year,Entrance\_date,getdate())>=4) THEN '本科毕业' ELSE '本科在读' END AS AttendanceStatus

FROM Student\_20211576

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Sno | char | 9 | 否 |  |  | 是 |
| Sname | Varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 否 | 男 |  |  |
| Birth | Smalldatetime |  | 是 |  |  |  |
| Entrance\_date | Smalldatetime |  | 是 |  |  |  |
| Class\_no | Char | 7 | 是 |  |  |  |
| Home\_addr | Varchar | 40 | 是 |  |  |  |
| Sde\_no | Char | 3 | 否 |  |  |  |
| Postcode | Char | 6 | 是 |  |  |  |
| AttendanceStatus | char | 20 | 是 |  |  |  |

（（五））添加教务管理系统管理员用户

Teacher\_20211576(插入管理员教师信息，Position为”教务管理人员”)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名称 | 类型 | 宽度 | 允许空值 | 缺省值 | 约束 | 主码 |
| Emp\_no | char | 9 | 否 |  |  | 是 |
| Emp\_name | varchar | 8 | 否 |  |  |  |
| Sex | Char | 2 | 是 | 男 |  |  |
| Sde\_no | Char | 10 | 否 |  |  |  |
| Position | char | 10 | 否 |  |  |  |

四、数据库的物理设计

为了提高在表中搜索元组的速度，在实际实现的时候应该基于某些属性建立索引。给出所建立的索引。（包括按哪些表中哪些属性按升序还是降序来创建索引及相应的T-SQL的创建语句）。

1.CREATE CLUSTERED INDEX Academy\_clustered\_index

ON Academy\_20211576(Aca\_no ASC);

2.CREATE CLUSTERED INDEX Class\_clustered\_index

ON Class\_20211576(Class\_no ASC);

3.CREATE CLUSTERED INDEX Course\_clustered\_index

ON Course\_20211576(Cno ASC);

4.CREATE CLUSTERED INDEX Course\_Available\_clustered\_index

ON Course\_Available\_20211576(Semester ASC,Emp\_no ASC,Cno ASC,Course\_Offered\_Sde\_no ASC);

5.CREATE CLUSTERED INDEX Course\_chosen\_clustered\_index

ON Course\_chosen\_20211576(Semester ASC,Sno ASC,Cno ASC,Emp\_no ASC,Course\_Offered\_Sde\_no ASC);

6.CREATE CLUSTERED INDEX Course\_Teacher\_clustered\_index

ON Course\_Teacher\_20211576(Emp\_no ASC,Cno ASC);

7.CREATE CLUSTERED INDEX Password\_clustered\_index

ON Password\_20211576(id ASC);

8.CREATE CLUSTERED INDEX Position\_clustered\_index

ON Position\_20211576(Position ASC);

9.CREATE CLUSTERED INDEX SC\_clustered\_index

ON SC\_20211576(Semester ASC,Sno ASC,Cno ASC,Emp\_no ASC,Sde\_no ASC);

10.CREATE CLUSTERED INDEX Sdept\_clustered\_index

ON Sdept\_202111576(Sde\_no ASC);

11.CREATE CLUSTERED INDEX Student\_clustered\_index

ON Student\_20211576(Sno ASC);

12.CREATE CLUSTERED INDEX Teacher\_clustered\_index

ON Teacher\_20211576(Emp\_no ASC);

五、数据库设计实现及运行

（1）数据库的创建

T-SQL语句：

Create database Academic\_Affairs\_Management\_System\_20211576

Use Academic\_Affairs\_Management\_System\_20211576

（2）数据表的创建

T-SQL语句：

以下是各个表的CREATE语句：

1.Academy\_20211576表：

CREATE TABLE Academy\_20211576 (

Aca\_no char(3) NOT NULL PRIMARY KEY,

Aca\_name varchar(20) NOT NULL,

Aca\_introduction char(100) NULL);

2.Sdept\_20211576表：

CREATE TABLE Sdept\_20211576 (

Sde\_no char(3) NOT NULL PRIMARY KEY,

Sde\_name char(20) NOT NULL,

Sde\_introduction char(100) NULL,

Aca\_no char(3) NOT NULL,

Aca\_name char(20) NULL,

FOREIGN KEY (Aca\_no) REFERENCES Academy\_20211576(Aca\_no));

3.Class\_20211576表：

CREATE TABLE Class\_20211576 (

Class\_no char(3) NOT NULL PRIMARY KEY,

Sde\_no char(3) NOT NULL,

Entrance\_date smalldatetime NOT NULL,

Num int NULL,

FOREIGN KEY (Sde\_no) REFERENCES Sdept\_20211576(Sde\_no));

4.Student\_20211576表：

CREATE TABLE Student\_20211576 (

Sno char(9) NOT NULL PRIMARY KEY,

Sname varchar(8) NOT NULL,

Sex char(2) NOT NULL DEFAULT '男',

Birth smalldatetime NULL,

Entrance\_date smalldatetime NULL,

Class\_no char(7) NULL,

Home\_addr varchar(40) NULL,

Sde\_no char(3) NOT NULL,

Postcode char(6) NULL,

AttendanceStatus char(20) NULL,

FOREIGN KEY (Sde\_no) REFERENCES Sdept\_20211576(Sde\_no),

FOREIGN KEY (Class\_no) REFERENCES Class\_20211576(Class\_no));

5.Course\_20211576表：

CREATE TABLE Course\_20211576 (

Cno char(10) NOT NULL PRIMARY KEY,

Aca\_no char(3) NOT NULL,

Cname varchar(40) NOT NULL,

Total\_perior smallint NULL,

Credit float NULL,

FOREIGN KEY (Aca\_no) REFERENCES Academy\_20211576(Aca\_no));

6.Teacher\_20211576表：

CREATE TABLE Teacher\_20211576 (

Emp\_no char(9) NOT NULL PRIMARY KEY,

Emp\_name varchar(8) NOT NULL,

Sex char(2) NULL DEFAULT '男',

Sde\_no char(10) NOT NULL,

Position char(20) NOT NULL,

FOREIGN KEY (Sde\_no) REFERENCES Sdept\_20211576(Sde\_no),

FOREIGN KEY (Position) REFERENCES Position\_20211576(Position));

7.Position\_20211576表：

CREATE TABLE Position\_20211576 (

Position char(20) NOT NULL PRIMARY KEY);

8.Course\_Teacher\_20211576表：

CREATE TABLE Course\_Teacher\_20211576 (

Emp\_no char(9) NOT NULL,

Cno char(10) NOT NULL,

Cname varchar(20) NULL,

PRIMARY KEY (Emp\_no, Cno),

FOREIGN KEY (Emp\_no) REFERENCES Teacher\_20211576(Emp\_no),

FOREIGN KEY (Cno) REFERENCES Course\_20211576(Cno));

9.Course\_Available\_20211576表：

CREATE TABLE Course\_Available\_20211576 (

Semester char(20) NOT NULL,

Emp\_no char(9) NOT NULL,

Cno char(10) NOT NULL,

Course\_Offered\_Sde\_no char(10) NOT NULL,

Cname varchar(20) NULL,

Credit tinyint NULL,

Stu\_Total tinyint NULL,

Stu\_chosen tinyint NOT NULL,

PRIMARY KEY (Semester, Emp\_no, Cno, Course\_Offered\_Sde\_no),

FOREIGN KEY (Emp\_no,Cno) REFERENCES Course\_Teacher\_20211576(Emp\_no,Cno),

FOREIGN KEY (Cno) REFERENCES Course\_20211576(Cno));

10.Course\_chosen\_20211576表：

CREATE TABLE Course\_chosen\_20211576 (

Serial\_Number int NULL,

Semester char(20) NOT NULL,

Emp\_no char(9) NOT NULL,

Sno char(9) NOT NULL,

Cno char(10) NOT NULL,

Course\_Offered\_Sde\_no char(10) NOT NULL,

Cname varchar(20) NULL,

Credit tinyint NULL,

Stu\_Total tinyint NULL,

Stu\_chosen tinyint NOT NULL,

PRIMARY KEY (Semester, Emp\_no, Sno, Cno, Course\_Offered\_Sde\_no),

FOREIGN KEY (Semester, Emp\_no, Sno, Cno, Course\_Offered\_Sde\_no) REFERENCES Course\_Available\_20211576(Semester, Emp\_no, Sno, Cno, Course\_Offered\_Sde\_no),

FOREIGN KEY (Sno) REFERENCES Student\_20211576(Sno));

11.SC\_20211576表：

CREATE TABLE SC\_20211576 (

Semester char(20) NOT NULL,

Sno char(9) NOT NULL,

Cno char(10) NOT NULL,

Emp\_no char(9) NOT NULL,

Sde\_no char(10) NOT NULL,

Grade tinyint NULL,

PRIMARY KEY (Semester, Sno, Cno, Emp\_no, Sde\_no),

FOREIGN KEY (Semester, Sno, Cno, Emp\_no, Sde\_no) REFERENCES Course\_chosen\_20211576(Semester, Sno, Cno, Emp\_no, Sde\_no));

12.Password\_20211576表：

CREATE TABLE Password\_20211576 (

id char(9) NOT NULL PRIMARY KEY,

name varchar(8) NOT NULL,

position char(20) NOT NULL,

password char(20) NULL,

FOREIGN KEY (position) REFERENCES Position\_20211576(Position));

说明：若有完整性约束的一方面可以创建时同时实现。按需求及自行设计模块中所实现的情况，在该部分对完整性约束加以实现。

（3）视图的创建

T-SQL语句：

1、Manager\_Student\_view：

CREATE VIEW Manager\_Student\_view AS

SELECT Sno , Sname , Sex , Birth , Entrance\_date , Class\_no , Home\_addr , Sde\_no , Postcode ,

CASE WHEN (DATEDIFF(year,Entrance\_date,getdate())>=4) THEN '本科毕业' ELSE '本科在读' END AS AttendanceStatus

FROM Student\_20211576

2、Manager\_Teacher\_view：

CREATE VIEW Manager\_Teacher\_view AS

SELECT Emp\_no,Emp\_name,Sex,Sde\_no,Position

FROM Teacher\_20211576

3、Manager\_Class\_view：

CREATE VIEW Manager\_Class\_view AS

SELECT Class\_no,Sde\_no,substring(CONVERT(varchar, Entrance\_date, 120),0,11) 'Entrance\_date',Num

FROM Class\_20211576

4、Manager\_Academy\_view：

CREATE VIEW Manager\_Academy\_view AS

SELECT Aca\_no,Aca\_name,Aca\_introduction

FROM Academy\_20211576

5、Manager\_Sdept\_view：

CREATE VIEW Manager\_Sdept\_view AS

SELECT Sde\_no,Sde\_name,Sde\_introduction,Aca\_no,Aca\_name

FROM Sdept\_20211576

6、Course\_teacher\_view:

create view Course\_teacher\_view as

select Emp\_no,Cno,Cname

from Course\_Teacher\_20211576

7、Manager\_Course\_Teacher\_manage\_view：

CREATE VIEW Manager\_Course\_Teacher\_manage\_view AS

SELECT Semester,Course\_Offered\_Sde\_no,Sde\_name,Cno,Cname,Emp\_name,Credit,Stu\_total

FROM Course\_Available\_20211576,Teacher\_20211576,Sdept\_20211576

where Course\_Available\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and Sdept\_20211576.Sde\_no=Course\_Available\_20211576.Course\_Offered\_Sde\_no

8、Student\_Enquiry\_view:

create view Student\_Enquiry\_view as

SELECT Semester,Sno,Cno,Cname,Emp\_name,Credit

FROM Course\_chosen\_20211576,Teacher\_20211576

where Course\_chosen\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

9、Student\_view:

create view Student\_view as

select Sno,Sname,Sex,substring(CONVERT(varchar, Birth, 120),0,11) 'Birth',substring(CONVERT(varchar, Entrance\_date, 120),0,11) 'Entrance\_date',Class\_no,Home\_addr,Student\_20211576.Sde\_no,Sde\_name,Postcode,

CASE WHEN (DATEDIFF(year,Entrance\_date,getdate())>=4) THEN '本科毕业' ELSE '本科在读' END AS AttendanceStatus

from Student\_20211576,Sdept\_20211576

where Student\_20211576.Sde\_no=Sdept\_20211576.Sde\_no

10、Teacher\_view:

create view Teacher\_view as

select Emp\_no,Emp\_name,Sex,Teacher\_20211576.Sde\_no,Sde\_name,Position

from Teacher\_20211576,Sdept\_20211576

where Teacher\_20211576.Sde\_no=Sdept\_20211576.Sde\_no

11. Teacher\_Score\_Manage\_view:查看已经被选的课程的课程信息

create view Teacher\_Score\_Manage\_view as

select Semester,Sno,Cno,Cname,Emp\_no,Credit,Course\_Offered\_Sde\_no

from Course\_chosen\_20211576

12、Teacher\_Manage\_SC\_view:给学生赋分

create view Teacher\_Manage\_SC\_view as

select Semester,Sno,SC\_20211576.Cno,Cname,Emp\_name,SC\_20211576.Sde\_no,Grade

from Teacher\_20211576,SC\_20211576,Course\_20211576

where SC\_20211576.Emp\_no=Teacher\_20211576.Emp\_no

and SC\_20211576.Cno=Course\_20211576.Cno

13、Student\_Score\_Enquiry\_view:

create view Student\_Score\_Enquiry\_view as

select Semester,SC\_20211576.Sno,Sname,SC\_20211576.Cno,Cname,SC\_20211576.Emp\_no,Emp\_name,Grade

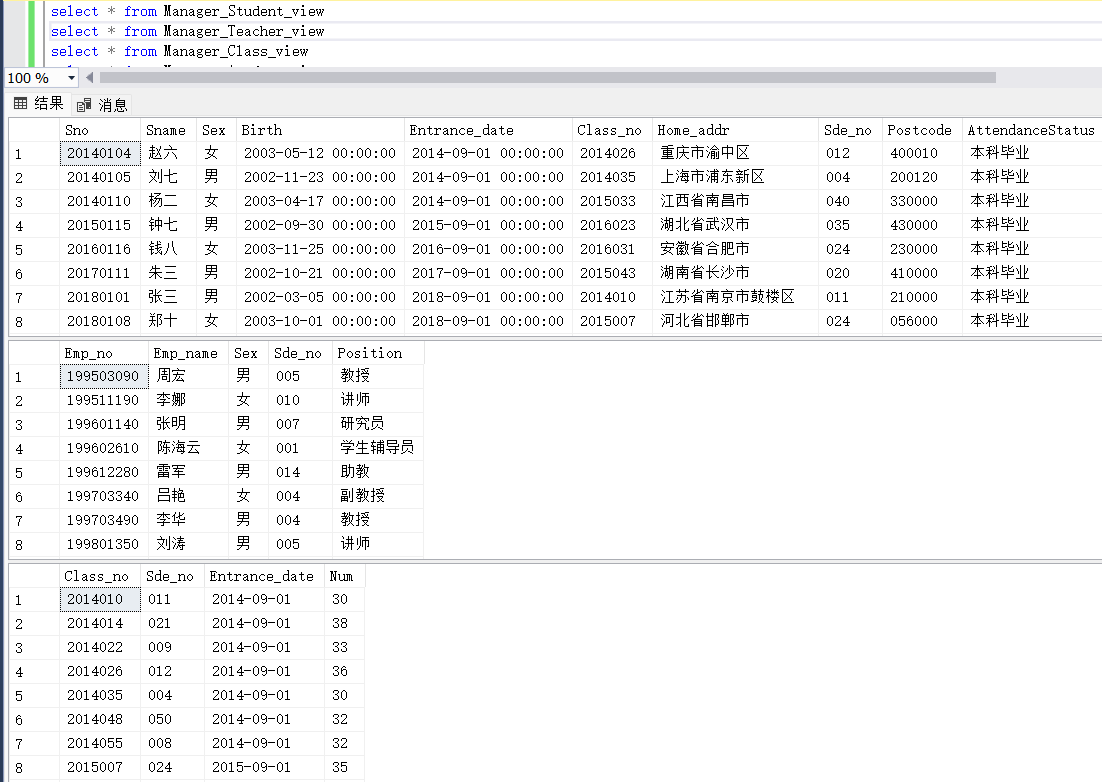
from SC\_20211576,Student\_20211576,Course\_20211576,Teacher\_20211576

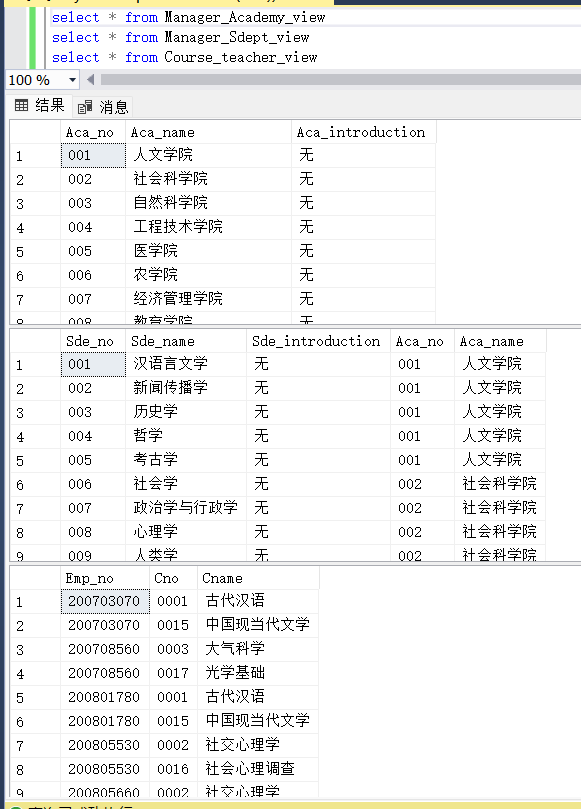
where SC\_20211576.Sno=Student\_20211576.Sno

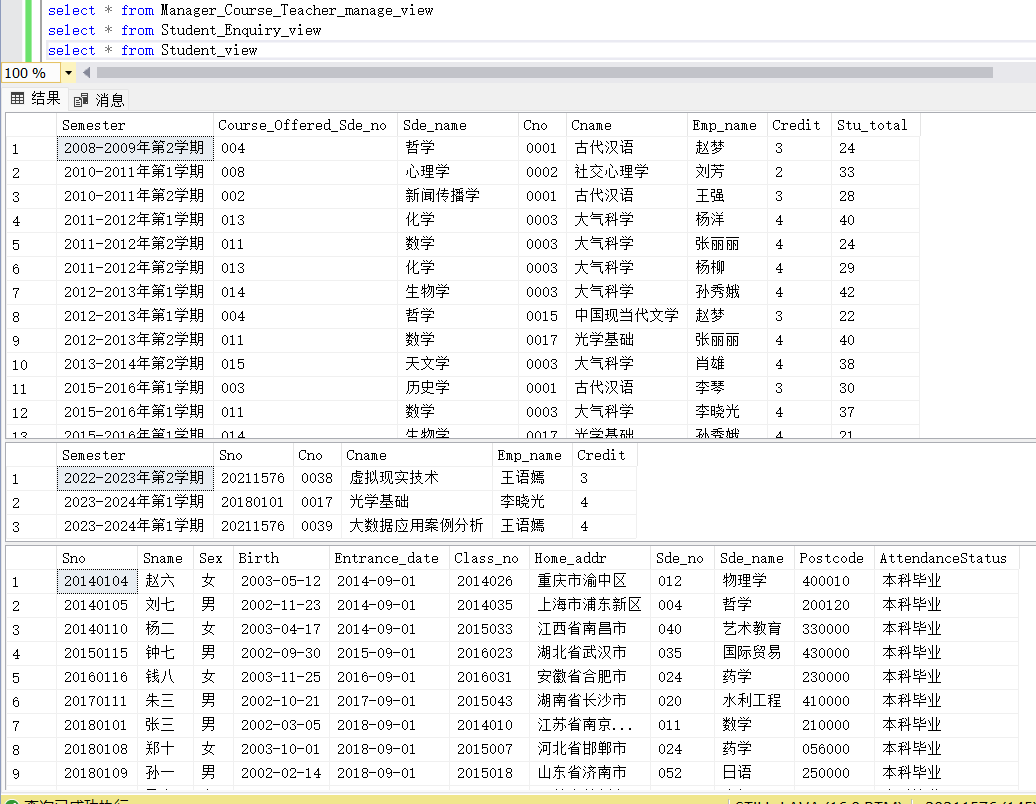
and Course\_20211576.Cno=SC\_20211576.Cno

and Teacher\_20211576.Emp\_no=SC\_20211576.Emp\_no

测试结果粘贴：









（4）存储过程的定义实现

T-SQL语句：

无

在宿主语言中调用存储过程代码段（粘贴）：

无

测试结果粘贴：

无

（5）触发器的定义实现

T-SQL语句：

1）trg\_UpdateEntranceDate\_Class：

CREATE TRIGGER trg\_UpdateEntranceDate\_Class

ON Class\_20211576

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE S

SET Entrance\_date = SUBSTRING(S.Class\_no, 1, 4) + '/09/01'

FROM Class\_20211576 AS S

INNER JOIN (

SELECT DISTINCT Class\_no FROM inserted

) AS T ON S.Class\_no = T.Class\_no

END

2)check\_records\_CourseAvail\_Course\_CourseTea

CREATE TRIGGER check\_records\_CourseAvail\_Course\_CourseTea

ON Course\_Available\_20211576

AFTER INSERT

AS

BEGIN

IF NOT EXISTS (

SELECT \*

FROM inserted AS i

WHERE NOT EXISTS (

SELECT \*

FROM Course\_20211576

WHERE Cno = i.Cno AND Credit = i.Credit

) OR NOT EXISTS (

SELECT \*

FROM Course\_Teacher\_20211576

WHERE Cno = i.Cno AND Emp\_no = i.Emp\_no

)

)

BEGIN

RAISERROR ('插入失败！Course\_20211576或Course\_Teacher\_20211576不存在相同数据！', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END

END

3)trg\_insert\_from\_course\_chosen\_to\_SC：

CREATE TRIGGER trg\_insert\_from\_course\_chosen\_to\_SC

ON Course\_chosen\_20211576

AFTER INSERT

AS

BEGIN

INSERT INTO SC\_20211576(Semester, Sno, Cno, Emp\_no, Sde\_no, Grade)

SELECT i.Semester, i.Sno, i.Cno, i.Emp\_no, i.Course\_Offered\_Sde\_no,0

FROM inserted i;

END

1. update\_stu\_chosen：

CREATE TRIGGER update\_stu\_chosen

ON Course\_Chosen\_20211576

AFTER INSERT

AS

BEGIN

UPDATE Course\_Available\_20211576

SET Stu\_chosen = Stu\_chosen + 1

WHERE Semester IN (SELECT Semester FROM inserted)

AND Emp\_no IN (SELECT Emp\_no FROM inserted)

AND Cno IN (SELECT Cno FROM inserted);

END

4)add\_teacher\_password：

CREATE TRIGGER add\_teacher\_password

ON Teacher\_20211576

AFTER INSERT

AS

BEGIN

INSERT INTO Password\_20211576 (id, name, position, password)

SELECT i.Emp\_no, i.Emp\_name, i.Position, '123456'

FROM inserted i

WHERE NOT EXISTS (

SELECT \* FROM Password\_20211576 WHERE id = i.Emp\_no AND position = i.Position

);

END

5)delete\_Teacher\_Password：

CREATE TRIGGER delete\_Teacher\_Password

ON Teacher\_20211576

INSTEAD OF DELETE

AS

BEGIN

DECLARE @DeletedEmp\_no TABLE (Emp\_no char(9));

INSERT INTO @DeletedEmp\_no (Emp\_no)

SELECT Emp\_no FROM DELETED;

DELETE FROM Course\_chosen\_20211576 WHERE Cno IN (SELECT Cno FROM Course\_Available\_20211576 WHERE Emp\_no IN (SELECT Emp\_no FROM @DeletedEmp\_no));

DELETE FROM Password\_20211576 WHERE id IN (SELECT Emp\_no FROM @DeletedEmp\_no);

DELETE FROM Course\_Available\_20211576 WHERE Emp\_no IN (SELECT Emp\_no FROM @DeletedEmp\_no);

DELETE FROM Course\_Teacher\_20211576 WHERE Emp\_no IN (SELECT Emp\_no FROM @DeletedEmp\_no);

DELETE FROM Teacher\_20211576 WHERE Emp\_no IN (SELECT Emp\_no FROM @DeletedEmp\_no);

END

6)trg\_NewStudent：

CREATE TRIGGER trg\_NewStudent

ON Student\_20211576

AFTER INSERT

AS

BEGIN

INSERT INTO Password\_20211576 (id, name, position, password)

SELECT i.Sno, i.Sname, '学生', '123456'

FROM inserted i

END

7)delete\_Student\_Password：

CREATE TRIGGER delete\_Student\_Password

ON Student\_20211576

AFTER DELETE

AS

BEGIN

DELETE FROM Course\_chosen\_20211576 WHERE Sno IN (

SELECT Sno FROM deleted

);

DELETE FROM Password\_20211576 WHERE id IN (

SELECT Sno FROM deleted

);

DELETE FROM SC\_20211576 WHERE Sno IN (

SELECT Sno FROM deleted

);

UPDATE Course\_Available\_20211576 SET Stu\_chosen = Stu\_chosen - 1 WHERE Cno IN (

SELECT DISTINCT Cno FROM SC\_20211576 WHERE Sno IN (

SELECT Sno FROM deleted

)

);

END;

8)trg\_delete\_info：

CREATE TRIGGER trg\_delete\_info

ON Academy\_20211576

INSTEAD OF DELETE

AS

BEGIN

DELETE FROM Password\_20211576 WHERE id IN (SELECT Sno FROM Student\_20211576 WHERE Sde\_no IN (SELECT Sde\_no FROM Sdept\_20211576 JOIN deleted d ON Sdept\_20211576.Aca\_no = d.Aca\_no));

DELETE FROM Password\_20211576 WHERE id IN (SELECT Emp\_no FROM Teacher\_20211576 WHERE Sde\_no IN (SELECT Sde\_no FROM Sdept\_20211576 JOIN deleted d ON Sdept\_20211576.Aca\_no = d.Aca\_no));

DELETE c FROM Class\_20211576 c JOIN deleted d ON c.Sde\_no IN (SELECT s.Sde\_no FROM Sdept\_20211576 s WHERE s.Aca\_no = d.Aca\_no);

DELETE sc FROM SC\_20211576 sc WHERE sc.Cno IN (SELECT co.Cno FROM Course\_20211576 co JOIN deleted d ON co.Aca\_no = d.Aca\_no);

DELETE ccs FROM Course\_chosen\_20211576 ccs WHERE ccs.Cno IN (SELECT co.Cno FROM Course\_20211576 co JOIN deleted d ON co.Aca\_no = d.Aca\_no);

DELETE st FROM Student\_20211576 st JOIN deleted d ON st.Sde\_no IN (SELECT s.Sde\_no FROM Sdept\_20211576 s WHERE s.Aca\_no = d.Aca\_no);

DELETE t FROM Teacher\_20211576 t JOIN deleted d ON t.Sde\_no IN (SELECT s.Sde\_no FROM Sdept\_20211576 s WHERE s.Aca\_no = d.Aca\_no);

DELETE s FROM Sdept\_20211576 s JOIN deleted d ON s.Aca\_no = d.Aca\_no;

DELETE cas FROM Course\_Available\_20211576 cas WHERE cas.Cno IN (SELECT co.Cno FROM Course\_20211576 co JOIN deleted d ON co.Aca\_no = d.Aca\_no);

DELETE cts FROM Course\_Teacher\_20211576 cts WHERE cts.Cno IN (SELECT co.Cno FROM Course\_20211576 co JOIN deleted d ON co.Aca\_no = d.Aca\_no);

DELETE co FROM Course\_20211576 co JOIN deleted d ON co.Aca\_no = d.Aca\_no;

END

9)delete\_course\_trigger：

CREATE TRIGGER delete\_course\_trigger

ON Course\_20211576

INSTEAD OF DELETE

AS

BEGIN

BEGIN TRANSACTION;

DELETE FROM SC\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_chosen\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_Available\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_Teacher\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

COMMIT TRANSACTION;

END

10)delete\_course\_teacher\_trigger：

CREATE TRIGGER delete\_course\_teacher\_trigger

ON Course\_Teacher\_20211576

INSTEAD OF DELETE AS

BEGIN

DELETE FROM SC\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_chosen\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_Available\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_Teacher\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

END;

11)delete\_course\_available\_trigger：

CREATE TRIGGER delete\_course\_available\_trigger

ON Course\_Available\_20211576

INSTEAD OF DELETE AS

BEGIN

DELETE FROM SC\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_chosen\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_Available\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

END;

12)delete\_course\_chosen\_trigger：

CREATE TRIGGER delete\_course\_chosen\_trigger

ON Course\_chosen\_20211576

INSTEAD OF DELETE AS

BEGIN

DELETE FROM SC\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

DELETE FROM Course\_chosen\_20211576 WHERE Cno IN (SELECT Cno FROM deleted);

END;

测试结果粘贴：

（6）自行设计各模块中所涉及的操作语句

1）插入数据操作

功能模块1：教务管理系统学生界面-学生选课界面选课

功能界面：



描述：点击选课按钮前需要先选中表格中一项，若未选会弹出报错窗口，点击选课后，若该课已经被该同学选过，则会弹窗提示该课程已选，若未选过，则会弹窗提示选课成功。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）；

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnChoose) {

if(sqlInsert.equals("")) {

JOptionPane.showMessageDialog(null, "请先选择课程！", "提示", JOptionPane.ERROR\_MESSAGE);

}else {

String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

Connection dbConn = null;

try {

Class.forName(Driver2);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (ClassNotFoundException ec) {

throw new RuntimeException(ec);

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

try {

Statement stmt = dbConn.createStatement();

System.out.println("sqlInsert:"+sqlInsert);

if(checkInsert(sqlCheck) == true){//checkInsert:检查该生是否已选过这门课

JOptionPane.showMessageDialog(null, "该课程已选！", "提示", JOptionPane.ERROR\_MESSAGE);

dbConn.close();

return;

}

else{

JOptionPane.showMessageDialog(null, "选课成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

}

ResultSet rs = stmt.executeQuery(sqlInsert);

} catch (SQLException ee) {

throw new RuntimeException(ee);

} finally {

try {

dbConn.close();

} catch (SQLException exc) {

throw new RuntimeException(exc);

}

}

}

}

// 添加监听器，监听表格的行选中事件

table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

// 获取当前选中行的索引

int row = table.getSelectedRow();

if (row >= 0) {

// 获取选中行的各列的数据

String insertSemester = table.getValueAt(row, 0).toString();

String insertSno = loginID.toString();

String insertCourseOfferedSdeNo = table.getValueAt(row, 1).toString();

String insertCno = table.getValueAt(row, 3).toString();

String insertCname = table.getValueAt(row, 4).toString();

//中间变量insertEmpName

String insertEmpName = table.getValueAt(row, 5).toString();

String insertEmpNo = getEmpNo(insertEmpName);

String insertCredit = table.getValueAt(row, 6).toString();

String insertStuTotal = table.getValueAt(row, 7).toString();

sqlInsert = "INSERT INTO Course\_chosen\_20211576(Semester,Sno,Cno,Cname,Emp\_no,Credit,Course\_Offered\_Sde\_no) VALUES ('" + insertSemester.trim() + "','" + insertSno.trim() + "','" + insertCno.trim() + "','" + insertCname.trim() + "','" + insertEmpNo.trim() + "'," + insertCredit + ",'" + insertCourseOfferedSdeNo.trim() + "')";

sqlCheck = "SELECT \* FROM Course\_chosen\_20211576 where Sno='" + insertSno.trim() + "' and Cno='" + insertCno.trim() + "' and Semester='" + insertSemester.trim() + "' and Course\_Offered\_Sde\_no='" + insertCourseOfferedSdeNo.trim() + "' and Emp\_no='" + insertEmpNo.trim() + "'";

}

}

});

}

private boolean checkInsert(String sqlCheck) {

String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

boolean flag = false;

Connection dbConn = null;

try {

Class.forName(Driver2);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (ClassNotFoundException ex) {

throw new RuntimeException(ex);

} catch (SQLException exc) {

throw new RuntimeException(exc);

}

int columnCount;

try {

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sqlCheck);

ResultSetMetaData rsmd = rs.getMetaData();

columnCount = 0;

while (rs.next()) {

flag = true;

break;

}

} catch (SQLException e) {

throw new RuntimeException(e);

} finally {

try {

dbConn.close();

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

return flag;

}

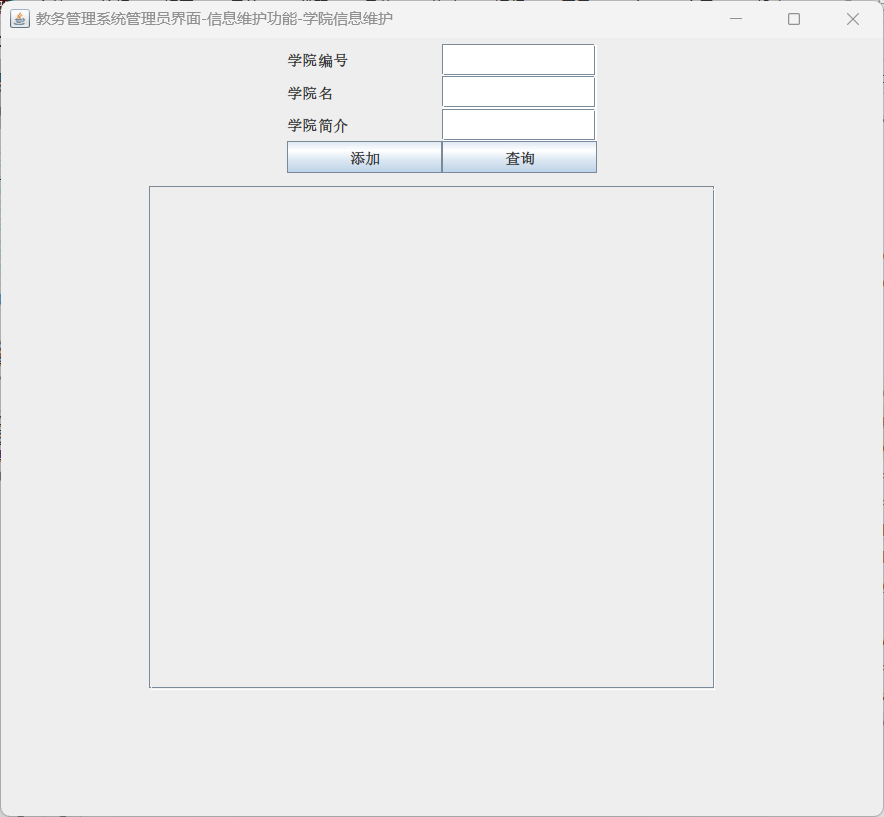
测试结果粘贴：



功能模块2：教务管理系统管理员界面-添加学院信息

功能界面：

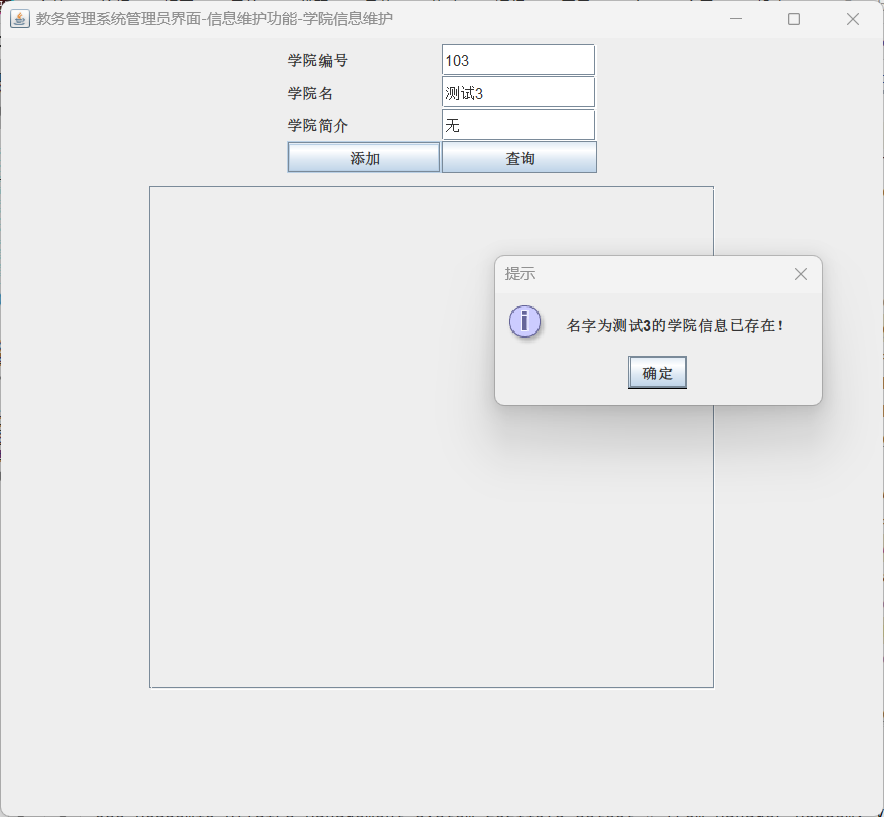


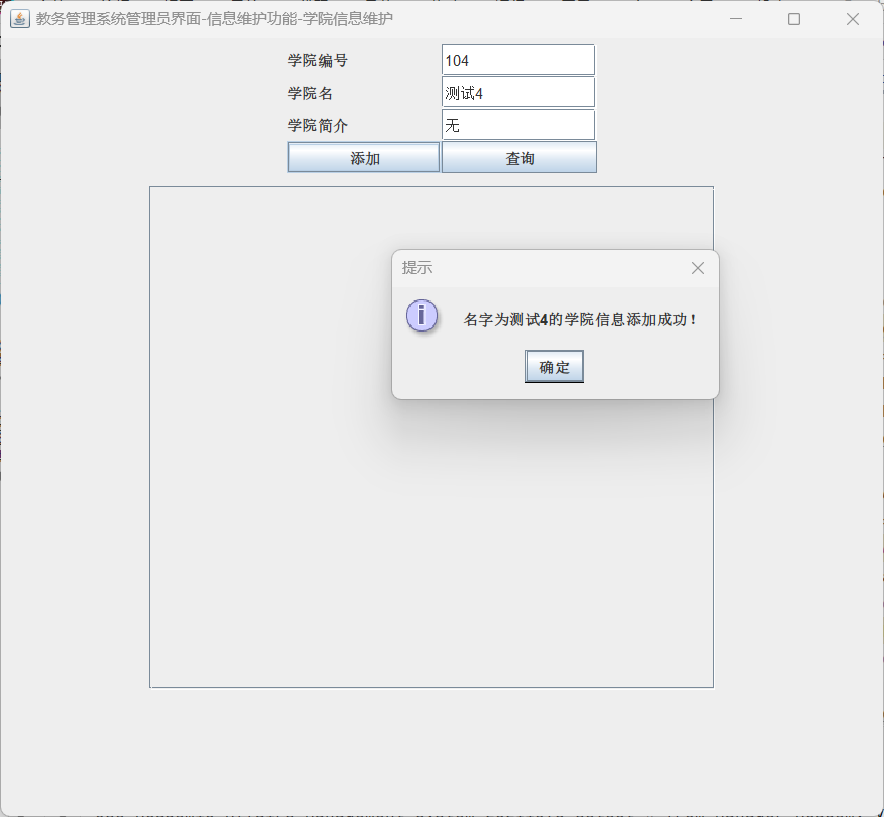
描述：点击教务管理系统管理员界面，打开如图界面，点击查询按钮，会从Manager\_Academy\_view中查询学院的相关信息，若为空，查询所有学院信息，若不为空，从上到下，按照没空的顺序依序查询（单项查询）。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

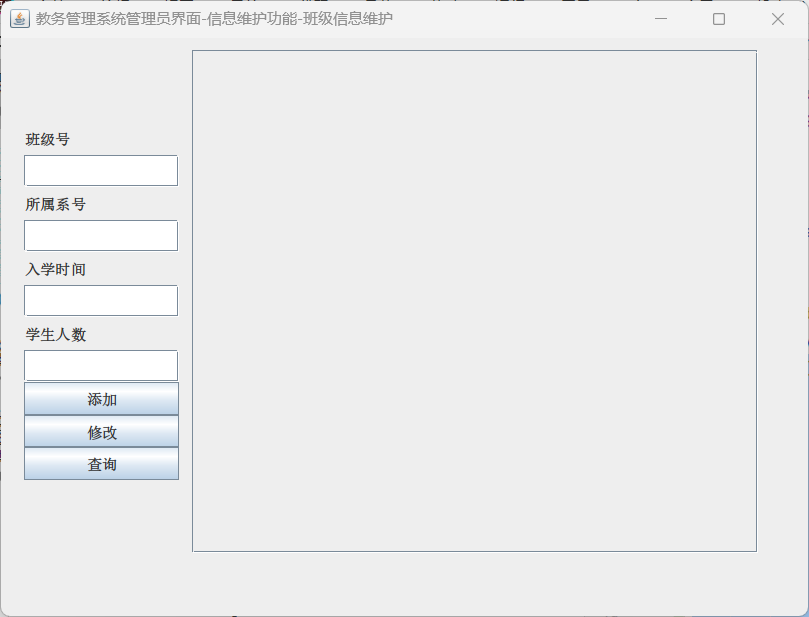
@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnSearch){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textAcaNo.getText().toString().equals("") == true  
 && textAcaName.getText().toString().equals("") == true  
 && textAcaIntroduction.getText().toString().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textAcaNo中内容进行查询，当textAcaNo中为空时方可对textAcaName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textAcaNo.getText().toString().equals("") == false) {  
 Input = textAcaNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_no = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textAcaName.getText().toString().equals("") == false) {  
 Input = textAcaName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_name = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textAcaIntroduction.getText().toString().equals("") == false) {  
 Input = textAcaIntroduction.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_introduction = " + "'" +Input + "'";  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while(rs.next()){  
 data[count][0] = rs.getString("Aca\_no");  
 data[count][1] = rs.getString("Aca\_name");  
 data[count][2] = rs.getString("Aca\_introduction");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
}

测试结果：





功能模块3：教务管理系统管理员界面-增加班级信息

功能界面：

描述：点击教务管理系统管理员界面，打开如图界面，点击查询按钮，会从Manager\_Class\_view中查询班级的相关信息，若为空，查询所有班级信息，若不为空，从上到下，按照没空的顺序依序查询（单项查询）。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnAdd) {

if (textClassNo.getText().trim().equals("") == false

&& textSdeNo.getText().trim().equals("") == false

&& textEntranceDate.getText().trim().equals("") == false

&& textNum.getText().trim().equals("") == false) {

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Class\_no ='" + textClassNo.getText() + "'";

String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576(Class\_no,Sde\_no,Entrance\_date,Num) values('" + textClassNo.getText() + "','" + textSdeNo.getText() + "','" + textEntranceDate.getText() + "'," + textNum.getText() + ")";

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

System.out.println(sql0); // 如果连接成功 控制台输出

System.out.println(sql);

Statement stmt = dbConn.createStatement();

ResultSet rs0 = stmt.executeQuery(sql0);

String Class\_no = "";

while (rs0.next()) {

Class\_no += rs0.getString("Class\_no");

}

if (Class\_no.equals("") == false) {

JOptionPane.showMessageDialog(null, "班级号为" + textClassNo.getText() + "的班级信息已存在！", "提示", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(null, "班级号为" + textClassNo.getText() + "的班级信息添加成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

ResultSet rs = stmt.executeQuery(sql);

rs.close();

}

rs0.close();

stmt.close();

dbConn.close();

} catch (Exception em) {

em.printStackTrace();

} finally {

try {

} catch (Exception ec) {

ec.printStackTrace();

}

}

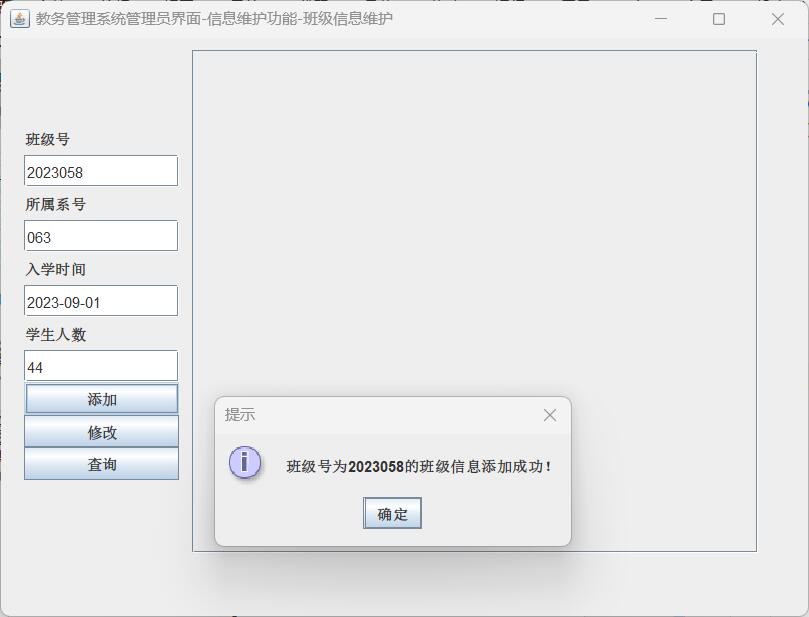
} else {

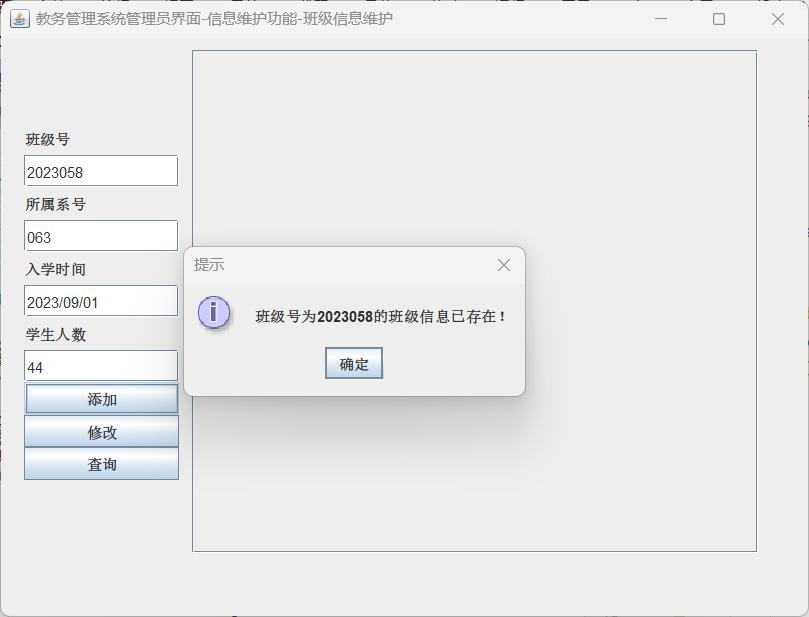
JOptionPane.showMessageDialog(null, "请填写完整信息!");

}

}

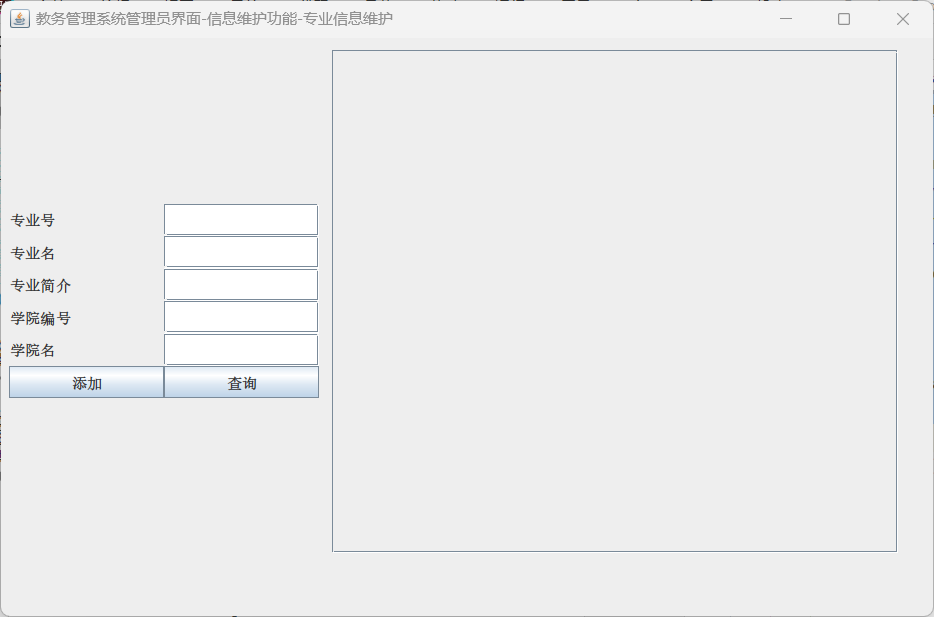
运行结果：





数据模块4：教务管理系统管理员界面-增加专业信息

功能界面：



描述：点击教务管理系统管理员界面，打开如图界面，点击查询按钮，会从Manager\_Sdept\_view中查询专业的相关信息，若为空，查询所有专业信息，若不为空，从上到下，按照没空的顺序依序查询（单项查询）。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnAdd) {

if (textSdeNo.getText().trim().equals("") == false

&& textSdeName.getText().trim().equals("") == false

&& textSdeIntroduction.getText().trim().equals("") == false

&& textAcaNo.getText().trim().equals("") == false

&& textAcaName.getText().trim().equals("") == false) {

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Sde\_no ='" + textSdeNo.getText() + "'";

String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Sdept\_20211576(Sde\_no,Sde\_name,Sde\_introduction,Aca\_no,Aca\_name) values('"+textSdeNo.getText()+"','"+textSdeName.getText()+"','"+textSdeIntroduction.getText()+"','"+textAcaNo.getText()+"','"+textAcaName.getText()+"')";

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

System.out.println(sql0); // 如果连接成功 控制台输出

System.out.println(sql);

Statement stmt = dbConn.createStatement();

ResultSet rs0 = stmt.executeQuery(sql0);

String Class\_no = "";

while (rs0.next()) {

Class\_no += rs0.getString("Sde\_no");

}

if (Class\_no.equals("") == false) {

JOptionPane.showMessageDialog(null, "专业号为" + textSdeNo.getText() + "的专业信息已存在！", "提示", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(null, "专业号为" + textSdeNo.getText() + "的专业信息添加成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

ResultSet rs = stmt.executeQuery(sql);

rs.close();

}

rs0.close();

stmt.close();

dbConn.close();

} catch (Exception em) {

em.printStackTrace();

} finally {

try {

} catch (Exception ec) {

ec.printStackTrace();

}

}

} else {

JOptionPane.showMessageDialog(null, "请填写完整信息!");

}

}

运行结果：





数据模块5：教务管理系统管理员界面-增加学生信息

功能界面：

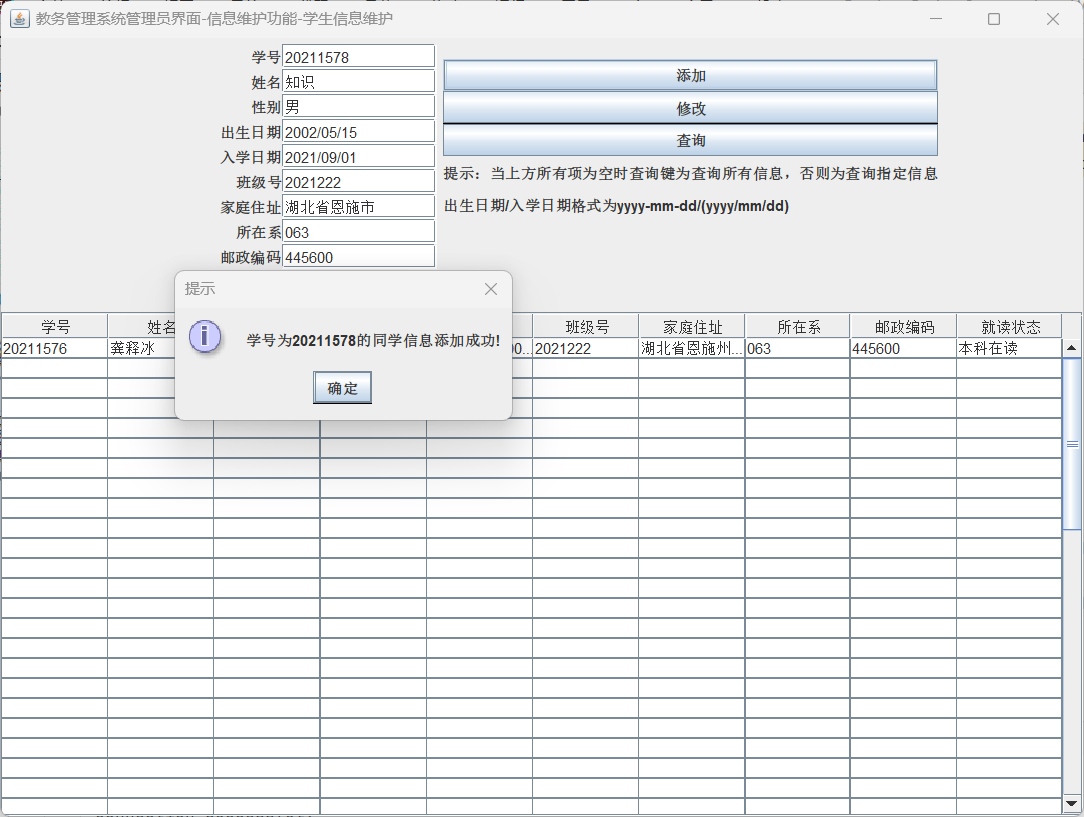


描述：点击教务管理系统管理员界面，打开如图界面，点击查询按钮，会从Manager\_Sdept\_view中查询相关学生的信息，若为空，查询所有学生信息，若不为空，从上到下，按照没空的顺序依序查询（单项查询）。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

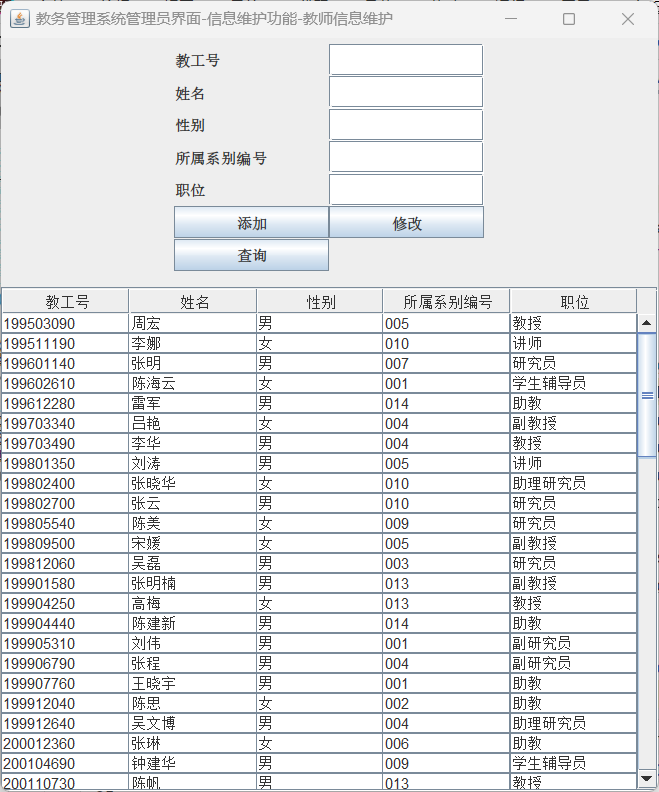
if (e.getSource() == btnAdd) {  
 if(textNo.getText().trim().equals("") == false  
 && textName.getText().trim().equals("") == false  
 && textSex.getText().trim().equals("") == false  
 && textBirth.getText().trim().equals("") == false  
 && textEntrancedate.getText().trim().equals("") == false  
 && textClassno.getText().trim().equals("") == false  
 && textHomeaddress.getText().trim().equals("") == false  
 && textSdept.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sde\_no ='" + textNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576(Sno,Sname,Sex,Birth,Entrance\_date,Class\_no,Home\_addr,Sde\_no,Postcode) values('"+textNo.getText()+"','"+textName.getText()+"','"+textSex.getText()+"',"+textBirth.getText()+","+textEntrancedate.getText()+",'"+textClassno.getText()+"','"+textHomeaddress.getText()+"','"+textSdept.getText()+"','"+textPostcode.getText()+"')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Sno");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "学号为"+textNo.getText()+"的同学信息已存在!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "学号为"+textNo.getText()+"的同学信息添加成功!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
}

运行结果：



数据模块6：教务管理系统管理员界面-增加教师信息

功能界面：

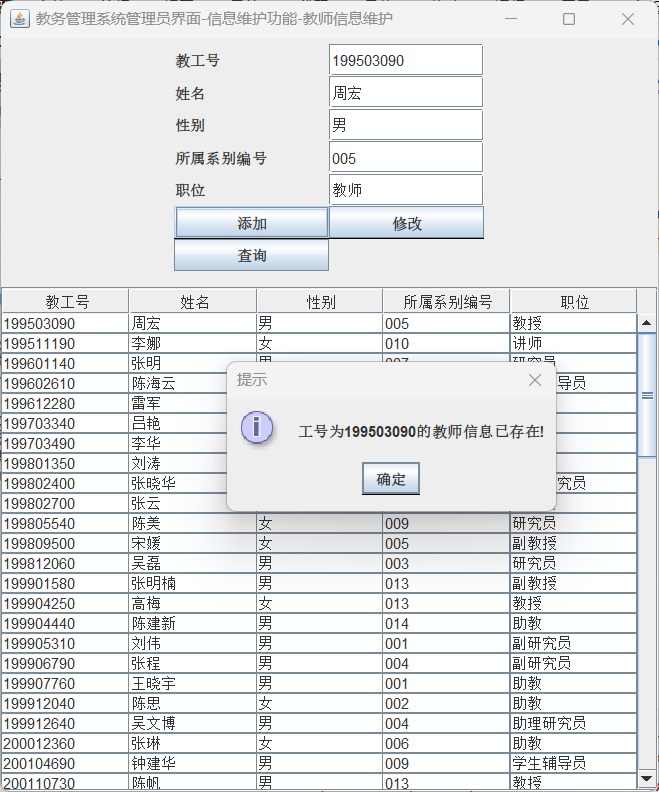


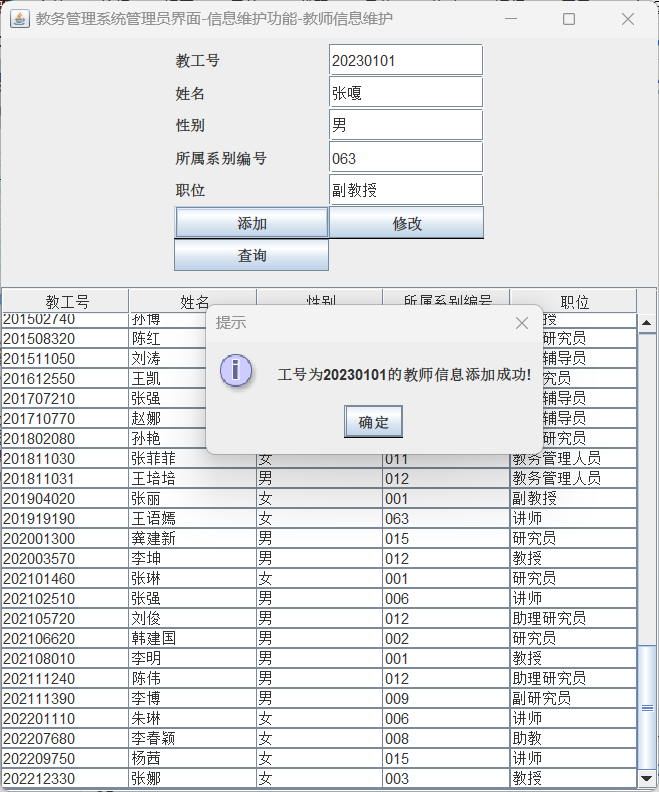
描述：点击教务管理系统管理员信息维护界面，打开如图界面，点击查询按钮，会从Manager\_Teacher\_view中查询教师的相关信息，若为空，查询所有教师信息，若不为空，从上到下，按照没空的顺序依序查询（单项查询）。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnAdd) {  
 if(textNo.getText().trim().equals("") == false  
 && textName.getText().trim().equals("") == false  
 && textSex.getText().trim().equals("") == false  
 && textSdept.getText().trim().equals("") == false  
 && textPosition.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_no ='" + textNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576(Emp\_no,Emp\_name,Sex,Sde\_no,Position) values('"+textNo.getText()+"','"+textName.getText()+"','"+textSex.getText()+"','"+textSdept.getText()+"','"+textPosition.getText()+"')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Emp\_no");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "工号为"+textNo.getText()+"的教师信息已存在!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "工号为"+textNo.getText()+"的教师信息添加成功!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
}

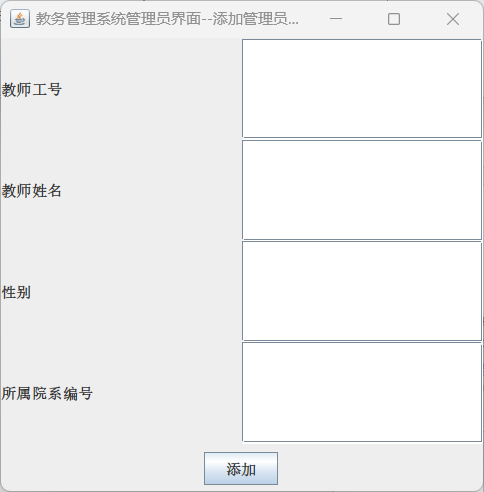
运行结果：





数据模块7：教务管理系统管理员界面-增加管理员账号

功能界面：

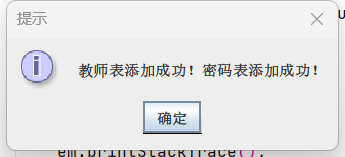


描述：点击教务管理系统管理员界面，点击添加教务系统管理员按钮，打开如图界面，会向Teacher\_20211576插入对应的信息，若其中有一项为空，弹窗提示，若无，弹窗提示添加成功。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if (txtEmp\_No.getText().equals("") || txtEmp\_name.getText().equals("") || txtSex.getText().equals("") || txtEmp\_SdeptNo.getText().equals("")) {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 return;  
 }  
 else{  
 frame1.dispose();  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql1 = "use Academic\_Affairs\_Management\_System\_20211576 insert into Teacher\_20211576 values ('" + txtEmp\_No.getText().toString() + "','" + txtEmp\_name.getText().toString() + "','" + txtSex.getText().toString() + "','" + txtEmp\_SdeptNo.getText().toString() + "','教务管理人员')"; *// SQL语句* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 JOptionPane.*showMessageDialog*(null, "教师表添加成功！密码表添加成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql1);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 }  
}

运行结果：



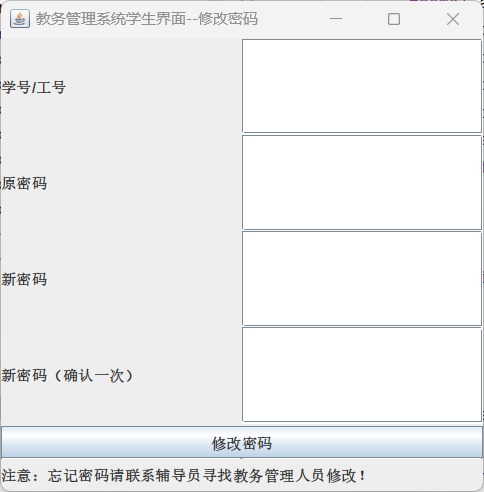
2）删除数据操作

在界面上无删除数据操作，删除只能由管理员在数据库中删除。

3）修改数据操作

功能模块1：教务系统登陆界面-修改密码

功能界面（粘贴）；



功能界面简单描述；可以对密码进行修改（修改作用于Password\_20211576表中），新密码两栏必须保持一致，否则按修改密码按钮时弹窗报错，所有项均需要填，否则按修改密码按钮时弹窗报错。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）；

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == buttonForget) {

WindowForget forget = new WindowForget();

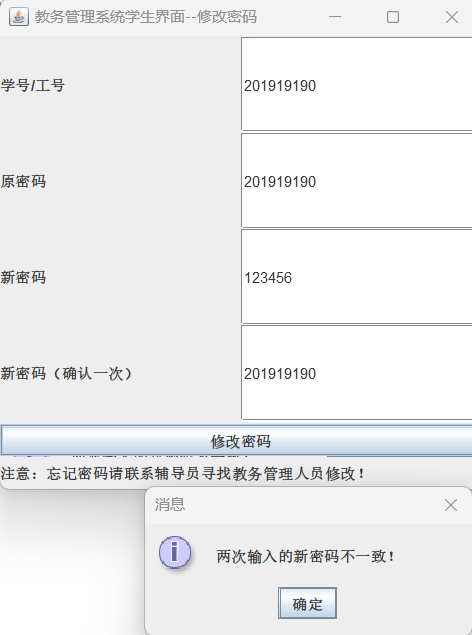
}

}

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnUpdate) {  
 if(txtNewPassword.getText().equals(txtNewPasswordTwice.getText()) == false) {  
 JOptionPane.*showMessageDialog*(null, "两次输入的新密码不一致！");  
 txtNewPassword.setText("");  
 txtNewPasswordTwice.setText("");  
 return;  
 }  
 frame1.dispose();  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select Position from Password\_20211576 where id='" + txtNo.getText() + "'";  
 String sql1 = "use Academic\_Affairs\_Management\_System\_20211576 select Password from Password\_20211576 where id='" + txtNo.getText() + "'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while(rs.next()){  
 String position = rs.getString("Position");  
 if (position.toString().trim().equals("教务管理人员")) {  
 JOptionPane.*showMessageDialog*(null, "教务管理人员无法修改密码！请至数据库内修改！");  
 }  
 else {  
 ResultSet rs1 = stmt.executeQuery(sql1);  
 while(rs1.next()){  
 String passwordcorrect = rs1.getString("password");  
 if (passwordcorrect.toString().trim().equals(txtOldPassword.getText())) {  
 String sql2 = "use Academic\_Affairs\_Management\_System\_20211576 update Password\_20211576 set Password='" + txtNewPassword.getText() + "' where id='" + txtNo.getText() + "'";  
 System.*out*.println(sql2);  
 stmt.executeUpdate(sql2);  
 JOptionPane.*showMessageDialog*(null, "修改成功！");  
 } else {  
 JOptionPane.*showMessageDialog*(null, "原密码错误！");  
 }  
 }  
 }  
  
 }  
  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
}

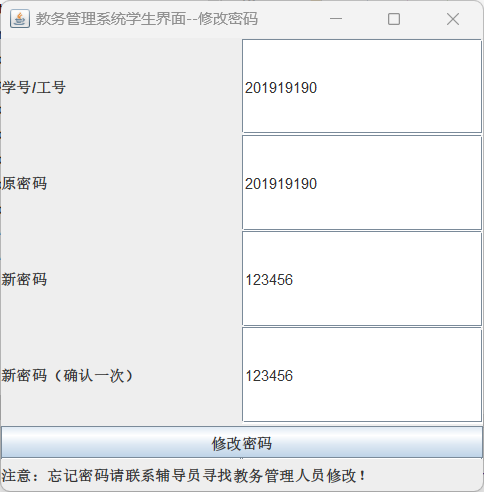
测试结果粘贴；





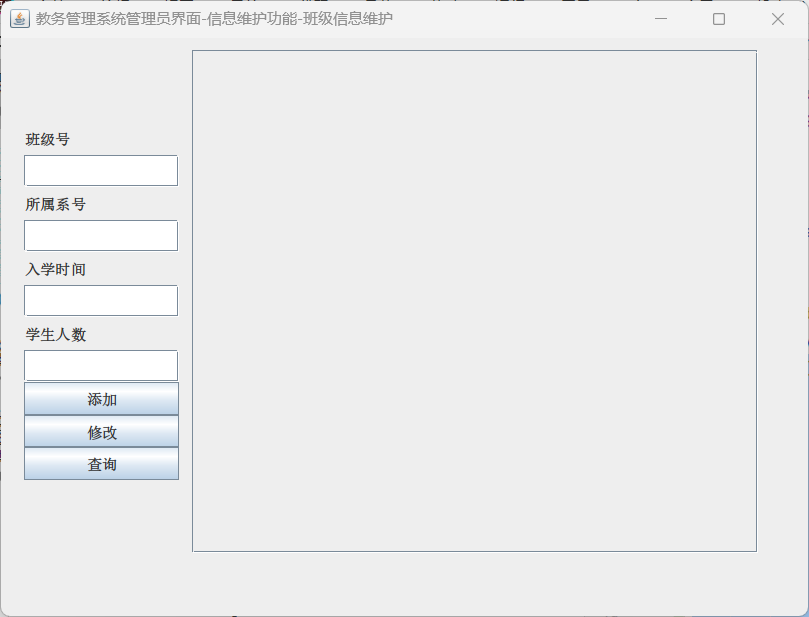








功能模块2：教务管理系统管理员界面-增加班级信息

功能界面：

描述：点击教务管理系统管理员界面，打开如图界面，点击修改按钮，会从Manager\_Class\_view中修改班级的相关信息，班级号不允许为空，否则弹窗提示。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnUpdate) {

if (textClassNo.getText().trim().equals("") == false) {

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sql = null;

if (textSdeNo.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Sde\_no = '" + textSdeNo.getText() + "' where Class\_no = '" + textClassNo.getText() + "'";

else if (textEntranceDate.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Entrance\_date = '" + textEntranceDate.getText() + "' where Class\_no = '" + textClassNo.getText() + "'";

else if (textNum.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Num = " + textNum.getText() + " where Class\_no = '" + textClassNo.getText() + "'";

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

System.out.println(sql); // 如果连接成功 控制台输出

JOptionPane.showMessageDialog(null, "班级号为" + textClassNo.getText() + "的班级信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

rs.close();

stmt.close();

dbConn.close();

} catch (Exception em) {

em.printStackTrace();

} finally {

try {

} catch (Exception ec) {

ec.printStackTrace();

}

}

}

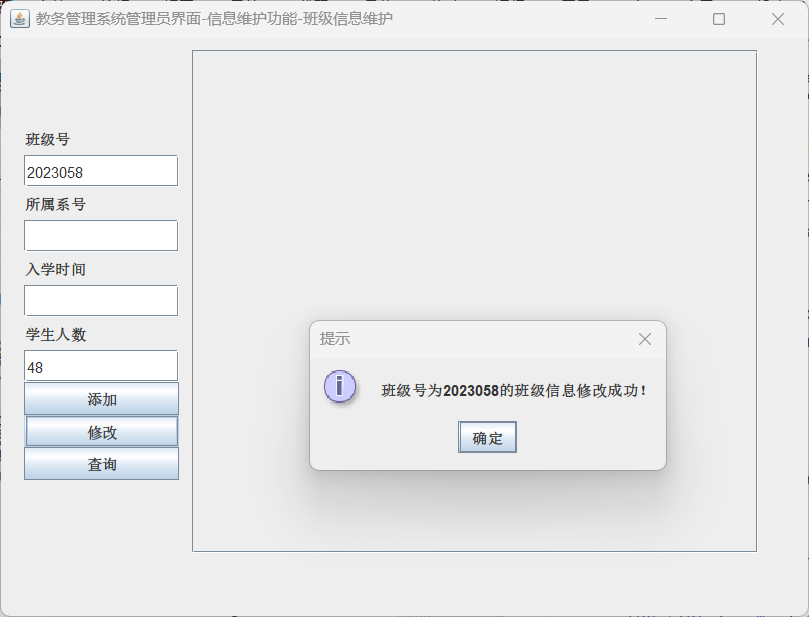
else{

JOptionPane.showMessageDialog(null, "班级号和（所属系号/入学时间/学生人数中至少有一项）不为空!", "提示", JOptionPane.INFORMATION\_MESSAGE);

}

}

运行结果：



功能模块3：教务管理系统管理员界面-修改学生信息

功能界面：

描述：点击教务管理系统管理员界面-信息维护界面，打开如图界面，点击修改按钮，会从Student\_20211576中修改学生的相关信息，学号不允许为空，否则弹窗提示。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnUpdate){

if(textNo.getText().trim().equals("") == false

&&(textName.getText().trim().equals("") == false

|| textSex.getText().trim().equals("") == false

|| textBirth.getText().trim().equals("") == false

|| textEntrancedate.getText().trim().equals("") == false

|| textClassno.getText().trim().equals("") == false

|| textHomeaddress.getText().trim().equals("") == false

|| textSdept.getText().trim().equals("") == false)){

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sql = null;

if(textName.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sname = '"+textName.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textSex.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sex = '"+textSex.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textBirth.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Birth = "+textBirth.getText()+" where Sno = '"+textNo.getText()+"'";

else if(textEntrancedate.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Entrance\_date = '"+textEntrancedate.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textClassno.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Class\_no = '"+textClassno.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textHomeaddress.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Home\_addr = '"+textHomeaddress.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textSdept.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sde\_no = '"+textSdept.getText()+"' where Sno = '"+textNo.getText()+"'";

else if(textPostcode.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Postcode = '"+textPostcode.getText()+"' where Sno = '"+textNo.getText()+"'";

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

System.out.println(sql); // 如果连接成功 控制台输出

JOptionPane.showMessageDialog(null, "学号为"+textNo.getText()+"的同学信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

rs.close();

stmt.close();

dbConn.close();

} catch (Exception em) {

em.printStackTrace();

} finally {

try {

} catch (Exception ec) {

ec.printStackTrace();

}

}

}

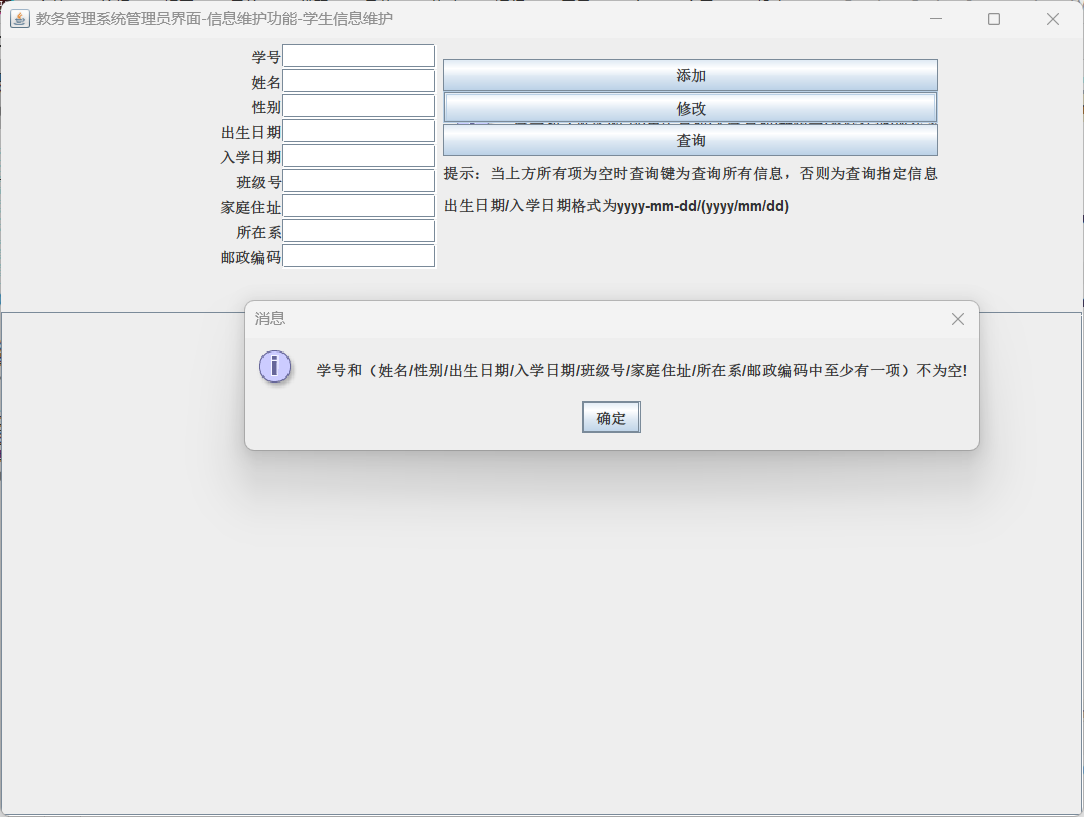
else{

JOptionPane.showMessageDialog(null, "学号和（姓名/性别/出生日期/入学日期/班级号/家庭住址/所在系/邮政编码中至少有一项）不为空!");

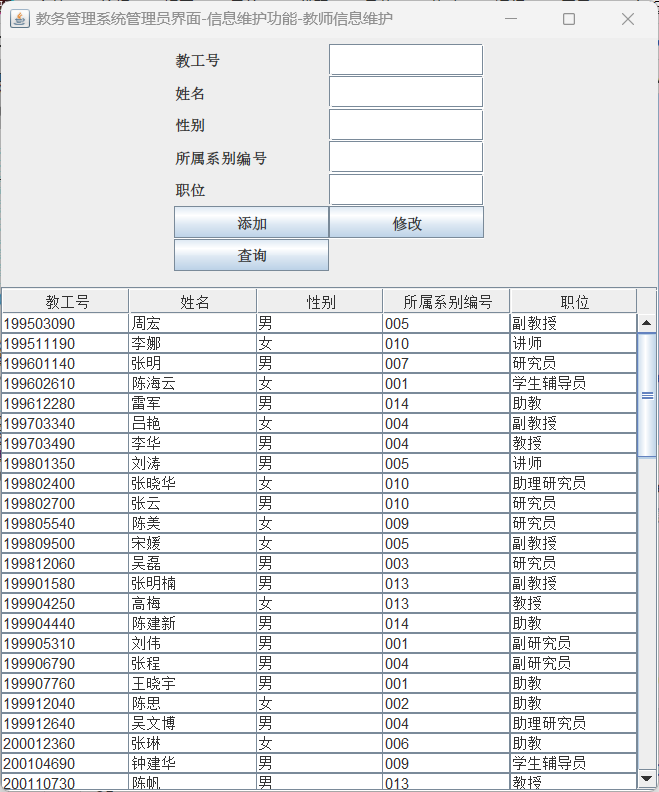
}

}

运行结果：



功能模块4：教务管理系统管理员界面-修改教师信息

功能界面：

描述：点击教务管理系统管理员界面-信息维护界面，打开如图界面，点击修改按钮，会从Teacher\_20211576中修改教师的相关信息，工号不允许为空，否则弹窗提示。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

if (e.getSource() == btnUpdate){

if(textNo.getText().trim().equals("") == false

&&(textName.getText().trim().equals("") == true

|| textSex.getText().trim().equals("") == true

|| textSdept.getText().trim().equals("") == true

|| textPosition.getText().trim().equals("") == true)){

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sql = null;

if(textName.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Emp\_name = '"+textName.getText()+"' where Emp\_no = '"+textNo.getText()+"'";

else if(textSex.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Sex = "+textSex.getText()+" where Emp\_no = '"+textNo.getText()+"'";

else if(textSdept.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Sdept = '"+textSdept.getText()+"' where Emp\_no = '"+textNo.getText()+"'";

else if(textPosition.getText().trim().equals("") == false)

sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Position = '"+textPosition.getText()+"' where Emp\_no = '"+textNo.getText()+"'";

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

System.out.println(sql); // 如果连接成功 控制台输出

JOptionPane.showMessageDialog(null, "工号为"+textNo.getText()+"的教师信息修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

rs.close();

stmt.close();

dbConn.close();

} catch (Exception em) {

em.printStackTrace();

} finally {

try {

} catch (Exception ec) {

ec.printStackTrace();

}

}

}

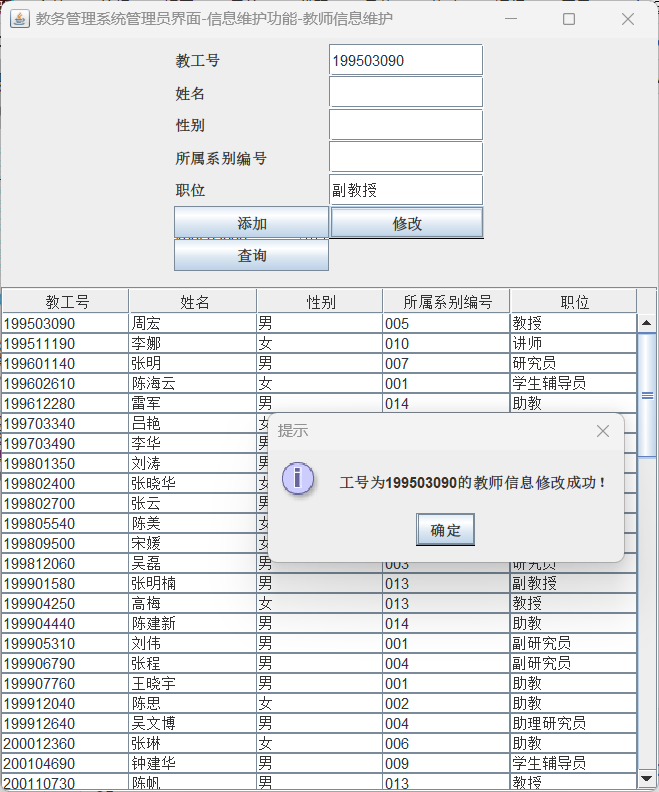
else{

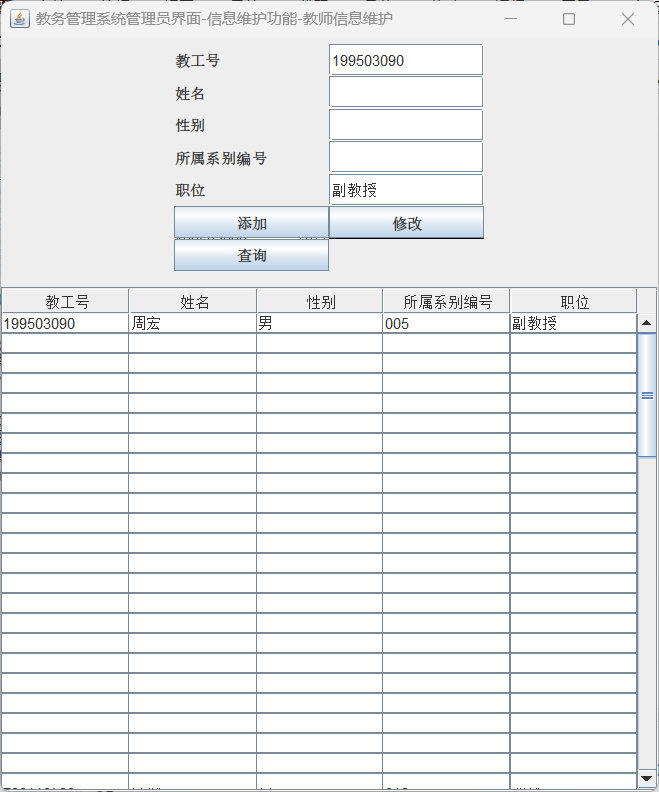
JOptionPane.showMessageDialog(null, "工号和（姓名，性别，所属系别编号职位中至少有一项）不为空!");

}

}

运行结果：





每个功能模块描述方法如下：

功能界面（粘贴）；

功能界面简单描述；

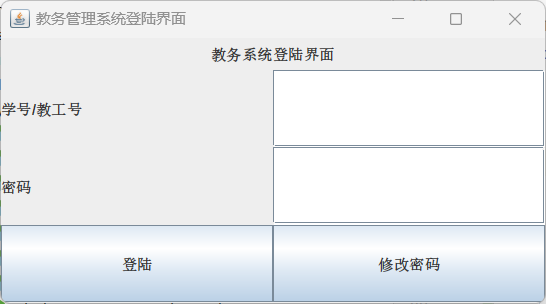
T-SQL语句与宿主语言嵌套使用代码段（粘贴）；

测试结果粘贴；

4）查询数据操作：

功能模块1：登陆界面查询操作

功能界面：



描述：包含用户名和密码输入框、登录按钮和注册按钮，用户输入用户名和密码后，若密码正确，则点击登录按钮即可进入主界面，教务管理人员对应的账号进入管理员界面，学生对应的账号进入学生界面，其余密码与用户名匹配的均进入教师界面。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == buttonRegister) {

if (txtLabelNo.getText().trim().equals("")) {

JOptionPane.showMessageDialog(null, "请输入学号/教工号", "提示", JOptionPane.ERROR\_MESSAGE);

} else if (txtLabelPassword.getText().trim().equals("")) {

JOptionPane.showMessageDialog(null, "密码不能为空", "提示", JOptionPane.ERROR\_MESSAGE);

} else {

String userInput = null;

if(txtLabelNo.getText().length()<7){

userInput = "'" + txtLabelNo.getText() + "'";

}

else{

userInput = txtLabelNo.getText();

}

String passwordInput = txtLabelPassword.getText().toString();

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;database=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String sql="use Academic\_Affairs\_Management\_System\_20211576 select Position from Password\_20211576 where id = " + userInput;

String sql1="use Academic\_Affairs\_Management\_System\_20211576 select password from Password\_20211576 where id = " + userInput;

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (Exception em) {

em.printStackTrace();

}

Statement stml;

ResultSet rs,rs1;

try {

stml = dbConn.createStatement();

rs = stml.executeQuery(sql);//SQL查询语句

while (rs.next()) {

String passwordCorrect;

String position = rs.getString("Position");

rs1 = stml.executeQuery(sql1);

while (rs1.next()) {

passwordCorrect = rs1.getString("password");

if(passwordInput.equals(passwordCorrect.toString().trim())) {

JOptionPane.showMessageDialog(null, "登陆成功");

//关闭界面，打开新界面

if (position.toString().trim().equals("学生")){

loginID = userInput;

loginName = getStudentName(loginID);

System.out.println(loginName);

System.out.println(loginID);

WindowStudent student = new WindowStudent();

}else if (position.trim().equals("教务管理人员")){

WindowManager manager = new WindowManager();

}else{

loginID = userInput;

loginName = getTeacherName(loginID);

WindowTeacher teacher = new WindowTeacher();

}

frame1.dispose();

}

else{

JOptionPane.showMessageDialog(null, "密码错误", "提示", JOptionPane.ERROR\_MESSAGE);

}

}

}

dbConn.close();

} catch (SQLException ex) {

System.out.println(ex);

}

}

}

else if (e.getSource() == buttonForget) {

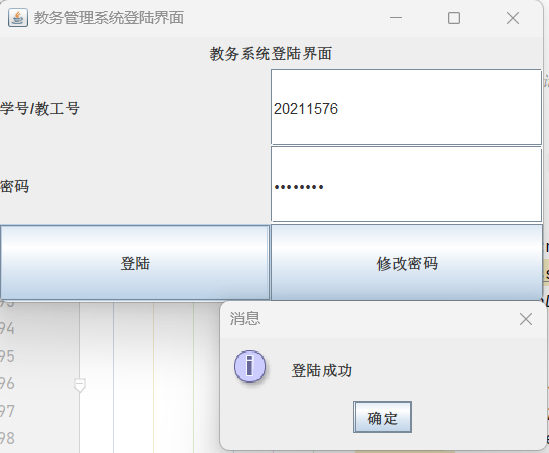
WindowForget forget = new WindowForget();

}

}

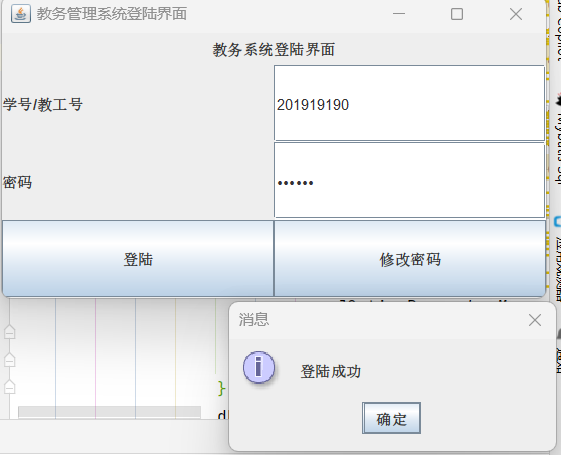
测试结果粘贴；





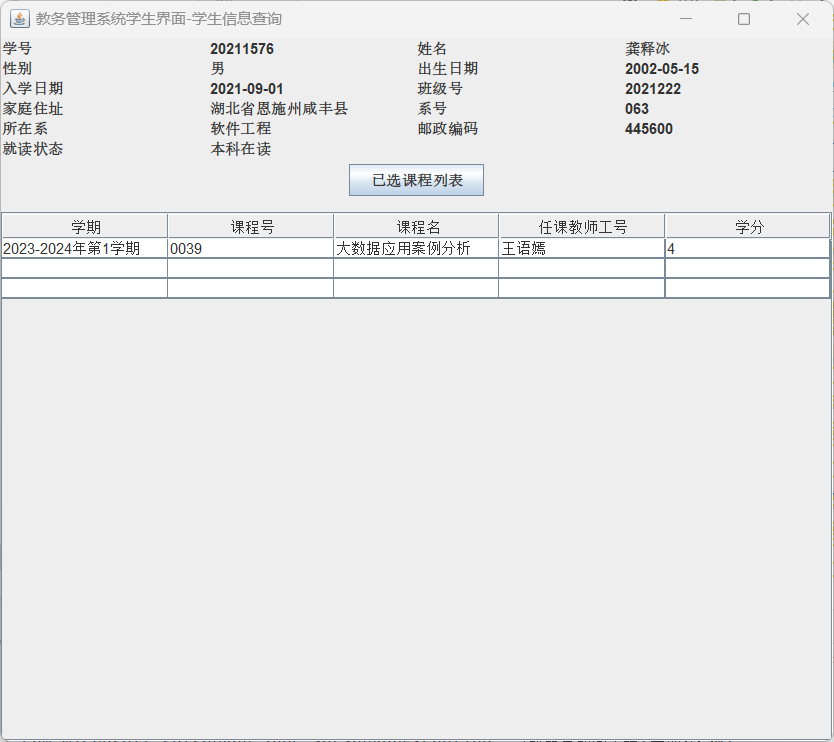






功能模块2：教务管理系统学生界面-

功能界面：



描述：点击学生界面-个人信息+已选课列表按钮，打开如图界面，点击已选课程列表按钮，通过学号查询并输出该学生已选修的所有课程到表中，学号是从登陆界面通过static string变量传过来的。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnEnquiry) {

String sqlEnquiryStudent = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_Enquiry\_view where Sno = '" + loginID.toString() + "'";

btnEnquiry(sqlEnquiryStudent);

}

}

public void btnEnquiry(String sql){

String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

Connection dbConn = null;

try {

Class.forName(Driver2);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (ClassNotFoundException ea) {

throw new RuntimeException(ea);

} catch (SQLException eb) {

throw new RuntimeException(eb);

}

int columnCount = 0;

try {

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

data[0][0] = rs.getString("Semester");

data[0][1] = rs.getString("Cno");

data[0][2] = rs.getString("Cname");

data[0][3] = rs.getString("Emp\_name");

data[0][4] = rs.getString("Credit");

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

} finally {

try {

dbConn.close();

} catch (SQLException exc) {

throw new RuntimeException(exc);

}

}

DefaultTableModel model = new DefaultTableModel(data, columnNames);

table.setModel(model);

}

功能模块3：教务管理系统学生界面-学生选课界面查询

功能界面：



描述：点击学生界面-选课按钮，打开如图界面，点击当前学期内按钮、课程号查询按钮、课程名查询按钮、授课教师查询按钮、学分对应课程按钮，对应查找符合条件的课程，暂时只支持单项查询，复合查询（多个条件一起查询）还没做，系名是通过查询Student\_view中学好对应的系名，学号是从登陆界面通过static string变量传过来的。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnSemestersEnquire) {

String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Semester='"+comSemester.getSelectedItem()+"' and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";

Connect(sql);

}else if (e.getSource() == btnCnoEnquire) {

String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cno='" + txtCno.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";

Connect(sql);

} else if (e.getSource() == btnCnameEnquire) {

String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cname='" + txtCname.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";

Connect(sql);

} else if (e.getSource() == btnTeacherEnquire) {

String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Emp\_Name='" + txtEmp\_Name.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";

Connect(sql);

} else if (e.getSource() == btnCreditEnquire) {

String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Credit='" + txtCredit.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";

Connect(sql);

}

}

public void Connect(String sql){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 8; j++) {  
 data[i][j] = "";  
 }  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}

测试结果粘贴：











功能模块4：教务管理系统学生界面-学生成绩查询界面

功能界面：



描述：点击学生界面-成绩查询按钮，打开如图界面，点击成绩查询按钮，查找该生选修的所有课程的成绩（通过Sno在Student\_Score\_Enquiry\_view中查询），学号是从登陆界面通过static string变量传过来的。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnEnquiry) {

String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

String sqlCourseEnquiry = "select \* from Student\_Score\_Enquiry\_view where Sno = '" + loginID + "' and Sname = '" + loginName + "' order by Semester asc";

System.out.println(sqlCourseEnquiry);

Connection dbConn = null;

try {

Class.forName(Driver2);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (ClassNotFoundException e1) {

throw new RuntimeException(e1);

} catch (SQLException e2) {

throw new RuntimeException(e2);

}

try {

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sqlCourseEnquiry);

ResultSetMetaData rsmd = rs.getMetaData();

int count = 0;

while (rs.next()&&count<getColumns()) {

data[count][0] = rs.getString("Semester");

data[count][1] = rs.getString("Cno");

data[count][2] = rs.getString("Cname");

data[count][3] = rs.getString("Emp\_name");

data[count][4] = rs.getString("Grade");

count++;

}

} catch (SQLException e3) {

throw new RuntimeException(e3);

} finally {

try {

dbConn.close();

} catch (SQLException e4) {

throw new RuntimeException(e4);

}

}

DefaultTableModel model = new DefaultTableModel(data, columnNames);

table.setModel(model);

}

}

测试结果粘贴：



功能模块5：教务管理系统教师界面授课列表查询

功能界面：



描述：点击教务管理系统登陆界面，打开如图界面，点击授课列表按钮，查找该教师所授的所有课程的信息（通过Emp\_name在Manager\_Course\_Teacher\_manage\_view中查询），教师姓名是从登陆界面通过static string变量loginName传过来的。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnEnquiry) {

Enquiry();

}

}

public void Enquiry(){

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";

String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Emp\_name = '" + loginName.trim() + "'";

System.out.println(sql);

String userName = "s20211576"; // 默认用户名

String userPwd = "s20211576"; // 密码

Connection dbConn = null;

try {

Class.forName(Driver);

dbConn = DriverManager.getConnection(url, userName, userPwd);

} catch (ClassNotFoundException ea) {

throw new RuntimeException(ea);

} catch (SQLException eb) {

throw new RuntimeException(eb);

}

int columnCount = 0;

try {

Statement stmt = dbConn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

data[0][0] = rs.getString("Semester");

data[0][1] = rs.getString("Course\_Offered\_Sde\_no");

data[0][2] = rs.getString("Sde\_name");

data[0][3] = rs.getString("Cno");

data[0][4] = rs.getString("Cname");

data[0][5] = rs.getString("Credit");

data[0][6] = rs.getString("Stu\_total");

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

} finally {

try {

dbConn.close();

} catch (SQLException exc) {

throw new RuntimeException(exc);

}

}

DefaultTableModel model = new DefaultTableModel(data, columnNames);

table.setModel(model);

}

测试结果粘贴：



功能模块6：教务管理系统教师界面-对应课程成绩管理界面

功能界面：



描述：点击教务管理系统登陆界面，打开如图界面，选中一列后，点击成绩管理按钮，查找该教师所授的所有课程的成绩（通过Emp\_name在Manager\_Course\_Teacher\_manage\_view中查询）并打开新界面，教师姓名是从登陆界面通过static string变量loginName传过来的，当在教务管理系统教师界面未选中课程直接点击成绩管理时，会弹窗报错，当选中课程没有学生选修，会关闭新界面，并弹窗提示。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnScoreManage){

//如果没有选中行，则弹出提示框

if (table.getSelectedRow() == -1) {

JOptionPane.showMessageDialog(null, "请先选中课程再进行此操作！", "提示", JOptionPane.ERROR\_MESSAGE);

return;

}

if(getInformationSC(TeacherRelated,sqlScore) == true){

JOptionPane.showMessageDialog(null, "该课程无人选择，无法进行赋分操作!", "提示", JOptionPane.ERROR\_MESSAGE);

return;

}

else{

WindowTeacherScoreManage windowTeacherScoreManage = new WindowTeacherScoreManage();

}

}

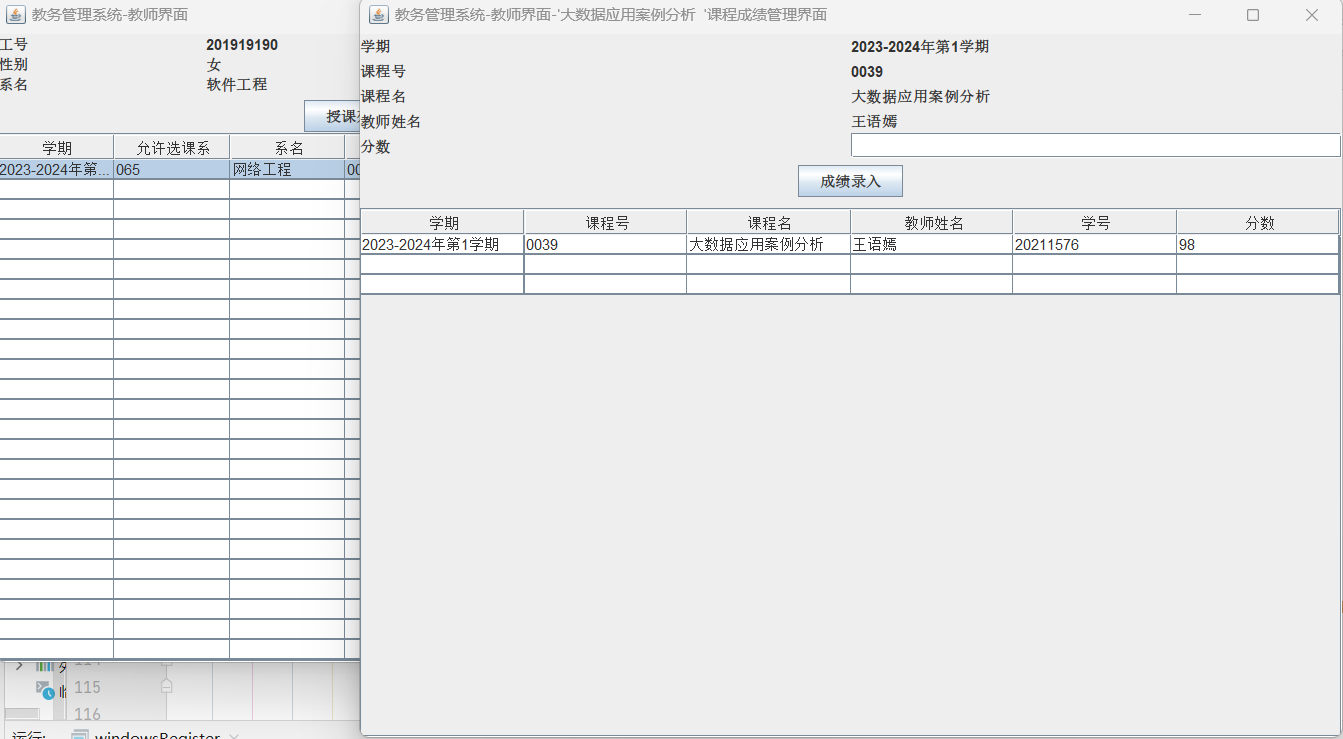
}

public boolean getInformationSC(Object[] TeacherRelated, String sql){  
 String Driver4 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* sql = *sqlScore*;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver4);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 TeacherRelated[0] = rs.getString("Semester");  
 TeacherRelated[1] = rs.getString("Cno");  
 TeacherRelated[2] = rs.getString("Cname");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
 int flag = 0;  
 for(int i=0;i<3;i++){  
 if(TeacherRelated[i].equals("")){  
 flag++;  
 }  
 }  
 if(flag==3){  
 return true;  
 }  
 else{  
 return false;  
 }  
}

测试结果截图：







功能模块7：教务管理系统管理员界面-教师授课管理界面

功能界面：



描述：点击教务管理系统管理员界面，点击教师授课信息维护按钮，打开如图界面，界面中除了添加与修改的按钮均为查询按钮，从Manager\_Course\_Teacher\_manage\_view获取相关信息。

T-SQL语句与宿主语言嵌套使用代码段（粘贴）：

测试结果截图：



（7）原子性事务显示定义：无

六、系统详细设计及实现

（1）系统功能模块说明

（一）教务管理系统登录功能模块

1、登录

2、修改密码：

（二）教务管理系统管理员功能模块

1、维护学院、专业、班级、教师、学生信息；

2、维护教师授课信息；

3、维护学生可选课程的信息；

4、查询教师、学生的各项信息；

5、添加教务管理系统管理员用户；

（三）教务管理系统学生功能模块

1、查看个人信息

2、查看已选课程

3、选课

4、查看课程成绩

（四）教务管理系统教师功能模块

1、查询所授课程

2、给有学生选修的课程进行赋分

（2）每个模块的关键语句及关键技术说明

（一）教务管理系统登录功能模块

1、登录：关键技术：数据验证、数据库查询。

（1）声明各类组件：

JPanel panelNorth = new JPanel();  
JFrame frame1 = new JFrame();  
public static String *loginID* = "";  
public static String *loginName* = "";

JLabel labelName = new JLabel("教务系统登陆界面");  
JLabel labelNo = new JLabel("学号/教工号");  
JTextField txtLabelNo = new JTextField(12);  
JLabel labelPassword = new JLabel("密码");  
JPasswordField txtLabelPassword = new JPasswordField(12);  
JButton buttonRegister = new JButton("登陆");  
JButton buttonForget = new JButton("修改密码");

（2）登陆界面的创建：

public windowsRegister() {  
 *//创建界面* frame1.setTitle("教务管理系统登陆界面");  
 frame1.setVisible(true);  
 frame1.setSize(450, 250);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame1.add(panelNorth,BorderLayout.*NORTH*);  
 frame1.add(panel, BorderLayout.*CENTER*);  
 panelNorth.add(labelName);  
 panel.setLayout(new GridLayout(3, 2));  
 panel.add(labelNo);panel.add(txtLabelNo);  
 panel.add(labelPassword);panel.add(txtLabelPassword);  
 panel.add(buttonRegister);  
 panel.add(buttonForget);  
 buttonRegister.addActionListener(this);  
 buttonForget.addActionListener(this);  
}

（3）登陆按钮的事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == buttonRegister) {  
 \* }  
 else if (e.getSource() == buttonForget) {  
 WindowForget forget = new WindowForget();  
 }  
}

（4）登陆界面向数据库获取信息并验证：

if (e.getSource() == buttonRegister) {  
 if (txtLabelNo.getText().trim().equals("")) {  
 JOptionPane.*showMessageDialog*(null, "请输入学号/教工号", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 } else if (txtLabelPassword.getText().trim().equals("")) {  
 JOptionPane.*showMessageDialog*(null, "密码不能为空", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 } else {  
 String userInput = null;  
 if(txtLabelNo.getText().length()<7){  
 userInput = "'" + txtLabelNo.getText() + "'";  
 }  
 else{  
 userInput = txtLabelNo.getText();  
 }  
 String passwordInput = txtLabelPassword.getText().toString();  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;database=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String sql="use Academic\_Affairs\_Management\_System\_20211576 select Position from Password\_20211576 where id = " + userInput;  
 String sql1="use Academic\_Affairs\_Management\_System\_20211576 select password from Password\_20211576 where id = " + userInput;  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (Exception em) {  
 em.printStackTrace();  
 }  
 Statement stml;  
 ResultSet rs,rs1;  
 try {  
 stml = dbConn.createStatement();  
 rs = stml.executeQuery(sql);*//SQL查询语句* while (rs.next()) {  
 String passwordCorrect;  
 String position = rs.getString("Position");  
 rs1 = stml.executeQuery(sql1);  
 while (rs1.next()) {  
 passwordCorrect = rs1.getString("password");  
 if(passwordInput.equals(passwordCorrect.toString().trim())) {  
 JOptionPane.*showMessageDialog*(null, "登陆成功");  
 *//关闭界面，打开新界面* if (position.toString().trim().equals("学生")){  
 *loginID* = userInput;  
 *loginName* = getStudentName(*loginID*);  
 System.*out*.println(*loginName*);  
 System.*out*.println(*loginID*);  
 WindowStudent student = new WindowStudent();  
 }else if (position.trim().equals("教务管理人员")){  
 WindowManager manager = new WindowManager();  
 }else{  
 *loginID* = userInput;  
 *loginName* = getTeacherName(*loginID*);  
 WindowTeacher teacher = new WindowTeacher();  
 }  
 frame1.dispose();  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "密码错误", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 }  
 }  
 }  
 dbConn.close();  
  
 } catch (SQLException ex) {  
 System.*out*.println(ex);  
 }  
 }  
}

public String getTeacherName(String loginID){\*}  
public String getStudentName(String loginID){\*}

**2、修改密码：关键技术：数据验证、数据库更新。**

1、在类中初始化各种组件，如JPanel、JLabel、JTextField、JButton等，设置它们的大小、位置、文本等属性。

JPanel panel = new JPanel();  
JPanel panel1 = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelWarning = new JLabel("注意：忘记密码请联系辅导员寻找教务管理人员修改！");  
JLabel labelNo = new JLabel("学号/工号");  
JTextField txtNo = new JTextField(10);  
JLabel labelOldPassword = new JLabel("原密码");  
JTextField txtOldPassword = new JTextField(10);  
JLabel labelNewPassword = new JLabel("新密码");  
JTextField txtNewPassword = new JTextField(10);  
JLabel labelNewPasswordTwice = new JLabel("新密码（确认一次）");  
JTextField txtNewPasswordTwice = new JTextField(10);  
JButton btnUpdate = new JButton("修改密码");

2、在构造方法中设置窗口标题、尺寸、位置、关闭方式等，并将组件添加到窗口中。

public WindowForget() {  
 frame1.setTitle("教务管理系统学生界面--修改密码");  
 frame1.setVisible(true);  
 frame1.setSize(400, 400);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panel, BorderLayout.*CENTER*);  
 panel.setLayout(new GridLayout(4,2));  
 panel1.setLayout(new GridLayout(2,1));  
 frame1.add(panel1, BorderLayout.*SOUTH*);  
 panel.add(labelNo);panel.add(txtNo);  
 panel.add(labelOldPassword);panel.add(txtOldPassword);  
 panel.add(labelNewPassword);panel.add(txtNewPassword);  
 panel.add(labelNewPasswordTwice);panel.add(txtNewPasswordTwice);  
 panel1.add(btnUpdate);panel1.add(labelWarning);  
 btnUpdate.addActionListener(this);  
}

3、实现actionPerformed方法，在按钮被点击时进行相应的处理。

4、获取四个输入框中的文本内容，如果有未填写的则弹出提示框警告用户。

if (e.getSource() == btnUpdate) {  
 if(txtNo.getText().trim().equals("") || txtOldPassword.getText().trim().equals("") || txtNewPassword.getText().trim().equals("") || txtNewPasswordTwice.getText().trim().equals("")){  
 JOptionPane.*showMessageDialog*(null, "学号/学工号，原密码，新密码等四项均需输入!", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 }

5、检查两次输入的新密码是否一致，如果不一致则弹出提示框并清空输入框，然后返回。

if(txtNewPassword.getText().equals(txtNewPasswordTwice.getText()) == false) {  
 JOptionPane.*showMessageDialog*(null, "两次输入的新密码不一致！");  
 txtNewPassword.setText("");  
 txtNewPasswordTwice.setText("");  
 return;  
}

6、创建数据库连接，并执行查询语句获取指定学号的职位和密码。

String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
String userName = "s20211576"; *// 默认用户名*String userPwd = "s20211576"; *// 密码*String sql = "use Academic\_Affairs\_Management\_System\_20211576 select Position from Password\_20211576 where id='" + txtNo.getText() + "'";  
String sql1 = "use Academic\_Affairs\_Management\_System\_20211576 select Password from Password\_20211576 where id='" + txtNo.getText() + "'";  
Connection dbConn = null;  
try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);

while(rs.next()){

//判断用户是否为教务管理人员，如果是则弹出提示框，否则继续执行。

if (position.toString().trim().equals("教务管理人员")) {  
 JOptionPane.*showMessageDialog*(null, "教务管理人员无法修改密码！请至数据库内修改！");  
}

//判断输入的旧密码是否与数据库中的密码匹配，如果匹配则执行更新密码的操作，并弹出提示框，否则弹出提示框。

if (position.toString().trim().equals("教务管理人员")) {  
 JOptionPane.*showMessageDialog*(null, "教务管理人员无法修改密码！请至数据库内修改！");  
}  
else {  
 ResultSet rs1 = stmt.executeQuery(sql1);  
 while(rs1.next()){  
 String passwordcorrect = rs1.getString("password");  
 if (passwordcorrect.toString().trim().equals(txtOldPassword.getText())) {  
 String sql2 = "use Academic\_Affairs\_Management\_System\_20211576 update Password\_20211576 set Password='" + txtNewPassword.getText() + "' where id='" + txtNo.getText() + "'";  
 System.*out*.println(sql2);  
 stmt.executeUpdate(sql2);  
 JOptionPane.*showMessageDialog*(null, "修改成功！");  
 } else {  
 JOptionPane.*showMessageDialog*(null, "原密码错误！");  
 }  
 }  
}

}  
 rs.close();  
 stmt.close();  
 dbConn.close();  
} catch (Exception em) {  
 em.printStackTrace();  
} finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
}

（二）教务管理系统管理员功能模块

1、维护学院、专业、班级、教师、学生信息：关键技术：数据库操作、表单验证。

学院：

（1）声明各类组件：

JPanel panel = new JPanel();  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelCenter = new JPanel();  
JPanel panelSouth = new JPanel();  
JLabel labelAcaNo = new JLabel("学院编号");  
JTextField textAcaNo = new JTextField(12);  
JLabel labelAcaName = new JLabel("学院名");  
JTextField textAcaName = new JTextField(12);  
JLabel labelAcaIntroduction = new JLabel("学院简介");  
JTextField textAcaIntroduction = new JTextField(12);  
JButton btnAdd = new JButton("添加");  
JButton btnSearch = new JButton("查询");  
JTable table = new JTable();  
JScrollPane scrollPane = new JScrollPane(table);  
String [] columnNames = {"学院编号","学院名","学院简介"};  
Object[][] data = new Object[getColumns()][10];

（2）登陆界面的创建：

public WindowManagerInformationUpdateAcademy() {  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-学院信息维护");  
 System.*out*.println(getColumns());  
 frame1.setVisible(true);  
 frame1.setSize(720,660);  
 frame1.setLocation(10,10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 table.setSize(350,400);  
 scrollPane.setBounds(0,0,352,250);  
 frame1.add(panel);  
 panel.add(panelNorth, BorderLayout.*NORTH*);  
 panel.add(panelCenter, BorderLayout.*CENTER*);  
 panel.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(4,2));  
 panelNorth.add(labelAcaNo);panelNorth.add(textAcaNo);  
 panelNorth.add(labelAcaName);panelNorth.add(textAcaName);  
 panelNorth.add(labelAcaIntroduction);panelNorth.add(textAcaIntroduction);  
 panelNorth.add(btnAdd);panelNorth.add(btnSearch);  
 panelCenter.add(scrollPane);  
 scrollPane.setViewportView(table);  
 btnAdd.addActionListener(this);  
 btnSearch.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd){  
 if(textAcaNo.getText().trim().equals("") == false  
 && textAcaName.getText().trim().equals("") == false  
 && textAcaIntroduction.getText().trim().equals("") == false){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_no ='"+textAcaNo.getText()+"'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Academy\_20211576(Aca\_no,Aca\_name,Aca\_introduction) " +  
 "values('"+textAcaNo.getText()+"','"+textAcaName.getText()+"','"+textAcaIntroduction.getText()+"')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Aca\_no = "";  
 while(rs0.next()){  
 Aca\_no += rs0.getString("Aca\_no");  
 }  
 if(Aca\_no.equals("") == false){  
 JOptionPane.*showMessageDialog*(null, "名字为"+textAcaName.getText()+"的学院信息已存在！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "名字为"+textAcaName.getText()+"的学院信息添加成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null,"请填写完整信息!");  
 }  
 }  
 if (e.getSource() == btnSearch){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textAcaNo.getText().toString().equals("") == true  
 && textAcaName.getText().toString().equals("") == true  
 && textAcaIntroduction.getText().toString().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textAcaNo中内容进行查询，当textAcaNo中为空时方可对textAcaName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textAcaNo.getText().toString().equals("") == false) {  
 Input = textAcaNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_no = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textAcaName.getText().toString().equals("") == false) {  
 Input = textAcaName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_name = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textAcaIntroduction.getText().toString().equals("") == false) {  
 Input = textAcaIntroduction.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view where Aca\_introduction = " + "'" +Input + "'";  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while(rs.next()){  
 data[count][0] = rs.getString("Aca\_no");  
 data[count][1] = rs.getString("Aca\_name");  
 data[count][2] = rs.getString("Aca\_introduction");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
}

（4）有关操作的函数:

public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Academy\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}

专业：

（1）声明各类控件：  
 JPanel panel = new JPanel();  
 JFrame frame1 = new JFrame();  
 JPanel panelNorth = new JPanel();  
 JPanel panelCenter = new JPanel();  
 JPanel panelSouth = new JPanel();  
 JLabel labelSdeNo = new JLabel("专业号");  
 JTextField textSdeNo = new JTextField(12);  
 JLabel labelSdeName = new JLabel("专业名");  
 JTextField textSdeName = new JTextField(12);  
 JLabel labelSdeIntroduction = new JLabel("专业简介");  
 JTextField textSdeIntroduction = new JTextField(12);  
 JLabel labelAcaNo = new JLabel("学院编号");  
 JTextField textAcaNo = new JTextField(12);  
 JLabel labelAcaName = new JLabel("学院名");  
 JTextField textAcaName = new JTextField(12);  
 JButton btnAdd = new JButton("添加");  
 JButton btnSearch = new JButton("查询");  
 JTable table = new JTable();  
 JScrollPane scrollPane = new JScrollPane(table);  
 String[] columnNames = {"专业号", "专业名", "专业简介", "学院编号", "学院名"};  
 Object[][] data = new Object[getColumns()][10];  
（2）创建面板：  
 public WindowManagerInformationUpdateMajor() {  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-专业信息维护");  
 System.*out*.println(getColumns());  
 frame1.setVisible(true);  
 frame1.setSize(760, 500);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 table.setSize(350, 400);  
 scrollPane.setBounds(0, 0, 352, 250);  
 frame1.add(panel);  
 panel.add(panelNorth, BorderLayout.*NORTH*);  
 panel.add(panelCenter, BorderLayout.*CENTER*);  
 panel.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(6, 2));  
 panelNorth.add(labelSdeNo);  
 panelNorth.add(textSdeNo);  
 panelNorth.add(labelSdeName);  
 panelNorth.add(textSdeName);  
 panelNorth.add(labelSdeIntroduction);  
 panelNorth.add(textSdeIntroduction);  
 panelNorth.add(labelAcaNo);  
 panelNorth.add(textAcaNo);  
 panelNorth.add(labelAcaName);  
 panelNorth.add(textAcaName);  
 panelNorth.add(btnAdd);  
 panelNorth.add(btnSearch);  
 panelCenter.add(scrollPane);  
 scrollPane.setViewportView(table);  
 btnAdd.addActionListener(this);  
 btnSearch.addActionListener(this);  
 }  
（3）按钮事件：

@Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if (textSdeNo.getText().trim().equals("") == false  
 && textSdeName.getText().trim().equals("") == false  
 && textSdeIntroduction.getText().trim().equals("") == false  
 && textAcaNo.getText().trim().equals("") == false  
 && textAcaName.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Sde\_no ='" + textSdeNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Sdept\_20211576(Sde\_no,Sde\_name,Sde\_introduction,Aca\_no,Aca\_name) values('" + textSdeNo.getText() + "','" + textSdeName.getText() + "','" + textSdeIntroduction.getText() + "','" + textAcaNo.getText() + "','" + textAcaName.getText() + "')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Sde\_no");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "专业号为" + textSdeNo.getText() + "的专业信息已存在！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "专业号为" + textSdeNo.getText() + "的专业信息添加成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
 }  
 if (e.getSource() == btnSearch) {  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textSdeNo.getText().toString().equals("") == true  
 && textSdeName.getText().toString().equals("") == true  
 && textSdeIntroduction.getText().toString().equals("") == true  
 && textAcaNo.getText().toString().equals("") == true  
 && textAcaName.getText().toString().equals("") == true) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 *// Get the number of columns in the ResultSet* ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textSdeNo.getText().toString().equals("") == false) {  
 Input = textSdeNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Sde\_no = " + "'" + Input + "'";  
 flag = 1;  
 } else if (textSdeName.getText().toString().equals("") == false) {  
 Input = textSdeName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Sde\_name = " + "'" + Input + "'";  
 flag = 2;  
 } else if (textSdeIntroduction.getText().toString().equals("") == false) {  
 Input = textSdeIntroduction.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Sde\_introduction = " + "'" + Input + "'";  
 flag = 3;  
 } else if (textAcaNo.getText().toString().equals("") == false) {  
 Input = textAcaNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Aca\_no = " + "'" + Input + "'";  
 flag = 4;  
 } else if (textAcaName.getText().toString().equals("") == false) {  
 Input = textAcaName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view where Aca\_name = " + "'" + Input + "'";  
 flag = 5;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while (rs.next()) {  
 data[count][0] = rs.getString("Sde\_no");  
 data[count][1] = rs.getString("Sde\_name");  
 data[count][2] = rs.getString("Sde\_introduction");  
 data[count][3] = rs.getString("Aca\_no");  
 data[count][4] = rs.getString("Aca\_name");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
 }  
（4）其他功能的函数：  
 public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Sdept\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
 }

班级：

（1）声明各类组件：  
 JPanel panel = new JPanel();  
 JFrame frame1 = new JFrame();  
 JPanel panelNorth = new JPanel();  
 JPanel panelCenter = new JPanel();  
 JPanel panelSouth = new JPanel();  
 JLabel labelClassNo = new JLabel("班级号");  
 JTextField textClassNo = new JTextField(12);  
 JLabel labelSdeNo = new JLabel("所属系号");  
 JTextField textSdeNo = new JTextField(12);  
 JLabel labelEntranceDate = new JLabel("入学时间");  
 JTextField textEntranceDate = new JTextField(12);  
 JLabel labelNum = new JLabel("学生人数");  
 JTextField textNum = new JTextField(12);  
 JButton btnAdd = new JButton("添加");  
 JButton btnUpdate = new JButton("修改");  
 JButton btnSearch = new JButton("查询");  
 JTable table = new JTable();  
 JScrollPane scrollPane = new JScrollPane(table);  
 String[] columnNames = {"班级号", "所属系号", "入学时间", "学生人数"};  
 Object[][] data = new Object[getColumns()][10];  
（2）创建界面：  
 public WindowManagerInformationUpdateClass() {  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-班级信息维护");  
 System.*out*.println(getColumns());  
 frame1.setVisible(true);  
 frame1.setSize(660, 500);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 table.setSize(350, 400);  
 scrollPane.setBounds(0, 0, 352, 250);  
 frame1.add(panel);  
 panel.add(panelNorth, BorderLayout.*NORTH*);  
 panel.add(panelCenter, BorderLayout.*CENTER*);  
 panel.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(11, 1));  
 panelNorth.add(labelClassNo);  
 panelNorth.add(textClassNo);  
 panelNorth.add(labelSdeNo);  
 panelNorth.add(textSdeNo);  
 panelNorth.add(labelEntranceDate);  
 panelNorth.add(textEntranceDate);  
 panelNorth.add(labelNum);  
 panelNorth.add(textNum);  
 panelNorth.add(btnAdd);  
 panelNorth.add(btnUpdate);  
 panelNorth.add(btnSearch);  
 panelCenter.add(scrollPane);  
 scrollPane.setViewportView(table);  
 btnAdd.addActionListener(this);  
 btnUpdate.addActionListener(this);  
 btnSearch.addActionListener(this);  
 }  
（3）按钮事件：

@Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if (textClassNo.getText().trim().equals("") == false  
 && textSdeNo.getText().trim().equals("") == false  
 && textEntranceDate.getText().trim().equals("") == false  
 && textNum.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Class\_no ='" + textClassNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576(Class\_no,Sde\_no,Entrance\_date,Num) values('" + textClassNo.getText() + "','" + textSdeNo.getText() + "','" + textEntranceDate.getText() + "'," + textNum.getText() + ")";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Class\_no");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "班级号为" + textClassNo.getText() + "的班级信息已存在！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "班级号为" + textClassNo.getText() + "的班级信息添加成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
 }  
 if (e.getSource() == btnUpdate) {  
 if (textClassNo.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = null;  
 if (textSdeNo.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Sde\_no = '" + textSdeNo.getText() + "' where Class\_no = '" + textClassNo.getText() + "'";  
 else if (textEntranceDate.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Entrance\_date = '" + textEntranceDate.getText() + "' where Class\_no = '" + textClassNo.getText() + "'";  
 else if (textNum.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Class\_20211576 set Num = " + textNum.getText() + " where Class\_no = '" + textClassNo.getText() + "'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql); *// 如果连接成功 控制台输出* JOptionPane.*showMessageDialog*(null, "班级号为" + textClassNo.getText() + "的班级信息修改成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "班级号和（所属系号/入学时间/学生人数中至少有一项）不为空!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 }  
 if (e.getSource() == btnSearch) {  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textClassNo.getText().equals("") == true  
 && textSdeNo.getText().equals("") == true  
 && textEntranceDate.getText().equals("") == true  
 && textNum.getText().equals("") == true) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 if (col == 2) {  
 data[row][col] = rs.getString(col + 1).substring(0, 10);  
 } else {  
 data[row][col] = rs.getString(col + 1);  
 }  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textClassNo.getText().equals("") == false) {  
 Input = textClassNo.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Class\_no = " + "'" + Input + "'";  
 flag = 1;  
 } else if (textSdeNo.getText().equals("") == false) {  
 Input = textSdeNo.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Sde\_no = " + "'" + Input + "'";  
 flag = 2;  
 } else if (textEntranceDate.getText().equals("") == false) {  
 Input = textEntranceDate.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Entrance\_date = " + "'" + Input + "'";  
 } else if (textNum.getText().equals("") == false) {  
 Input = textNum.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view where Num = " + Input;  
 flag = 4;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while (rs.next()) {  
 data[count][0] = rs.getString("Class\_no");  
 data[count][1] = rs.getString("Sde\_no");  
 data[count][2] = rs.getString("Entrance\_date");  
 data[count][3] = rs.getString("Num");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
 }

（4）其他需要的函数：  
 public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e1) {  
 throw new RuntimeException(e1);  
 } catch (SQLException e2) {  
 throw new RuntimeException(e2);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Class\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e3) {  
 throw new RuntimeException(e3);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e4) {  
 throw new RuntimeException(e4);  
 }  
 }  
 return columnCount;  
 }

教师：

（1）声明各类组件：  
 JPanel panel = new JPanel();  
 JFrame frame1 = new JFrame();  
 JPanel panelNorth = new JPanel();  
 JLabel labelNo = new JLabel("教工号");  
 JTextField textNo = new JTextField(12);  
 JLabel labelName = new JLabel("姓名");  
 JTextField textName = new JTextField(12);  
 JLabel labelSex = new JLabel("性别");  
 JTextField textSex = new JTextField(12);  
 JLabel labelSdept = new JLabel("所属系别编号");  
 JTextField textSdept = new JTextField(12);  
 JLabel labelPosition = new JLabel("职位");  
 JTextField textPosition = new JTextField(12);  
 JButton btnAdd = new JButton("添加");  
 JButton btnUpdate = new JButton("修改");  
 JButton btnSearch = new JButton("查询");  
 JTable table = new JTable();  
 JScrollPane scrollPane = new JScrollPane(table);  
 String [] columnNames = {"教工号","姓名","性别","所属系别编号","职位"};  
 Object[][] data = new Object[getColumns()][10];

（2）创建面板：  
 public WindowManagerInformationUpdateTeacher() {  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-教师信息维护");  
 System.*out*.println(getColumns());  
 frame1.setVisible(true);  
 frame1.setSize(540,640);  
 frame1.setLocation(10,10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 table.setSize(350,400);  
 scrollPane.setBounds(0,0,352,250);  
 frame1.add(panel);  
 panel.add(panelNorth, BorderLayout.*NORTH*);  
 panelNorth.setLayout(new GridLayout(11,2));  
 panelNorth.add(labelNo);panelNorth.add(textNo);  
 panelNorth.add(labelName);panelNorth.add(textName);  
 panelNorth.add(labelSex);panelNorth.add(textSex);  
 panelNorth.add(labelSdept);panelNorth.add(textSdept);  
 panelNorth.add(labelPosition);panelNorth.add(textPosition);  
 panelNorth.add(btnAdd);  
 panelNorth.add(btnUpdate);panelNorth.add(btnSearch);  
 frame1.add(scrollPane, BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 btnAdd.addActionListener(this);  
 btnUpdate.addActionListener(this);  
 btnSearch.addActionListener(this);  
 }  
（3）按钮事件：  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if(textNo.getText().trim().equals("") == false  
 && textName.getText().trim().equals("") == false  
 && textSex.getText().trim().equals("") == false  
 && textSdept.getText().trim().equals("") == false  
 && textPosition.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_no ='" + textNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576(Emp\_no,Emp\_name,Sex,Sde\_no,Position) values('"+textNo.getText()+"','"+textName.getText()+"','"+textSex.getText()+"','"+textSdept.getText()+"','"+textPosition.getText()+"')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Emp\_no");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "工号为"+textNo.getText()+"的教师信息已存在!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "工号为"+textNo.getText()+"的教师信息添加成功!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
 }  
 if (e.getSource() == btnUpdate){  
 if(textNo.getText().trim().equals("") == false  
 &&(textName.getText().trim().equals("") == true  
 || textSex.getText().trim().equals("") == true  
 || textSdept.getText().trim().equals("") == true  
 || textPosition.getText().trim().equals("") == true)){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = null;  
 if(textName.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Emp\_name = '"+textName.getText()+"' where Emp\_no = '"+textNo.getText()+"'";  
 else if(textSex.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Sex = "+textSex.getText()+" where Emp\_no = '"+textNo.getText()+"'";  
 else if(textSdept.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Sdept = '"+textSdept.getText()+"' where Emp\_no = '"+textNo.getText()+"'";  
 else if(textPosition.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Teacher\_20211576 set Position = '"+textPosition.getText()+"' where Emp\_no = '"+textNo.getText()+"'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql); *// 如果连接成功 控制台输出* JOptionPane.*showMessageDialog*(null, "工号为"+textNo.getText()+"的教师信息修改成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "工号和（姓名，性别，所属系别编号职位中至少有一项）不为空!");  
 }  
 }  
 if (e.getSource() == btnSearch){  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textNo.getText().toString().equals("") == true  
 && textName.getText().toString().equals("") == true  
 && textSex.getText().toString().equals("") == true  
 && textSdept.getText().toString().equals("") == true  
 && textPosition.getText().toString().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 *// Get the number of columns in the ResultSet* ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textNo中内容进行查询，当textNo中为空时方可对textName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textNo.getText().toString().equals("") == false) {  
 Input = textNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_no = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textName.getText().toString().equals("") == false) {  
 Input = textName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_name = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textSex.getText().toString().equals("") == false) {  
 Input = textSex.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Sex = " + "'" +Input + "'";  
 } else if (textSdept.getText().toString().equals("") == false) {  
 Input = textSdept.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Sde\_no = " +"''"+ Input + "''";  
 flag = 4;  
 } else if (textPosition.getText().toString().equals("") == false) {  
 Input = textPosition.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Position = " + "'" +Input + "'";  
 flag = 5;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while(rs.next()){  
 data[count][0] = rs.getString("Emp\_no");  
 data[count][1] = rs.getString("Emp\_name");  
 data[count][2] = rs.getString("Sex");  
 data[count][3] = rs.getString("Sde\_no");  
 data[count][4] = rs.getString("Position");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
 }

（4）其他需要的函数：  
 public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 *// 如果连接成功 控制台输出* } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
 }

学生：

（1）各类控件的声明：  
 JPanel panel = new JPanel();  
 JFrame frame1 = new JFrame();  
 JPanel panelNorth = new JPanel();  
 JPanel panelCenter = new JPanel();  
 JPanel panelSouth = new JPanel();  
 JLabel labelNo = new JLabel("学号");  
 JTextField textNo = new JTextField(12);  
 JLabel labelName = new JLabel("姓名");  
 JTextField textName = new JTextField(12);  
 JLabel labelSex = new JLabel("性别");  
 JTextField textSex = new JTextField(12);  
 JLabel labelBirth = new JLabel("出生日期");  
 JTextField textBirth = new JTextField(12);  
 JLabel labelEntrancedate = new JLabel("入学日期");  
 JTextField textEntrancedate = new JTextField(12);  
 JLabel labelClassno = new JLabel("班级号");  
 JTextField textClassno = new JTextField(12);  
 JLabel labelHomeaddress = new JLabel("家庭住址");  
 JTextField textHomeaddress = new JTextField(12);  
 JLabel labelSdept = new JLabel("所在系");  
 JTextField textSdept = new JTextField(12);  
 JLabel labelPostcode = new JLabel("邮政编码");  
 JTextField textPostcode = new JTextField(12);JButton btnAdd = new JButton("添加");  
 JButton btnUpdate = new JButton("修改");  
 JButton btnSearch = new JButton("查询");  
 JLabel labelWarning1 = new JLabel("提示：当上方所有项为空时查询键为查询所有信息，否则为查询指定信息" );  
 JLabel labelWarning2 = new JLabel("出生日期/入学日期格式为yyyy-mm-dd/(yyyy/mm/dd)");  
 JTable table = new JTable();  
 JScrollPane scrollPane = new JScrollPane(table);  
 String [] columnNames = {"学号","姓名","性别","出生日期","入学日期","班级号","家庭住址","所在系","邮政编码","就读状态"};  
 Object[][] data = new Object[getColumns()][10];

（2）创建面板：  
 public WindowManagerInformationUpdateStudent() {  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-学生信息维护");  
 System.*out*.println(getColumns());  
 frame1.setVisible(true);  
 frame1.setSize(880,660);  
 frame1.setLocation(10,10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 table.setSize(350,400);  
 scrollPane.setBounds(0,0,352,250);  
 frame1.add(panel);  
 panel.add(panelNorth, BorderLayout.*NORTH*);  
 panel.add(panelCenter, BorderLayout.*CENTER*);  
 panel.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(1,2));  
 panelNorth.setLayout(new GridLayout(9,2));  
 panelNorth.add(labelNo);panelNorth.add(textNo);  
 panelNorth.add(labelName);panelNorth.add(textName);  
 panelNorth.add(labelSex);panelNorth.add(textSex);  
 panelNorth.add(labelBirth);panelNorth.add(textBirth);  
 panelNorth.add(labelEntrancedate);panelNorth.add(textEntrancedate);  
 panelNorth.add(labelClassno);panelNorth.add(textClassno);  
 panelNorth.add(labelHomeaddress);panelNorth.add(textHomeaddress);  
 panelNorth.add(labelSdept);panelNorth.add(textSdept);  
 panelNorth.add(labelPostcode);panelNorth.add(textPostcode);  
 labelNo.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelName.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelSex.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelBirth.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelEntrancedate.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelClassno.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelHomeaddress.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelSdept.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 labelPostcode.setHorizontalAlignment(SwingConstants.*RIGHT*);  
 panelCenter.setLayout(new GridLayout(6,1));  
 panelCenter.add(btnAdd);  
 panelCenter.add(btnUpdate);panelCenter.add(btnSearch);  
 panelCenter.add(labelWarning1);panelCenter.add(labelWarning2);  
 frame1.add(scrollPane,BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 btnAdd.addActionListener(this);  
 btnUpdate.addActionListener(this);  
 btnSearch.addActionListener(this);  
 }  
 （3）按钮事件：

@Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if(textNo.getText().trim().equals("") == false  
 && textName.getText().trim().equals("") == false  
 && textSex.getText().trim().equals("") == false  
 && textBirth.getText().trim().equals("") == false  
 && textEntrancedate.getText().trim().equals("") == false  
 && textClassno.getText().trim().equals("") == false  
 && textHomeaddress.getText().trim().equals("") == false  
 && textSdept.getText().trim().equals("") == false) {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql0 = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sde\_no ='" + textNo.getText() + "'";  
 String sql = "insert into Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576(Sno,Sname,Sex,Birth,Entrance\_date,Class\_no,Home\_addr,Sde\_no,Postcode) values('"+textNo.getText()+"','"+textName.getText()+"','"+textSex.getText()+"',"+textBirth.getText()+","+textEntrancedate.getText()+",'"+textClassno.getText()+"','"+textHomeaddress.getText()+"','"+textSdept.getText()+"','"+textPostcode.getText()+"')";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql0); *// 如果连接成功 控制台输出* System.*out*.println(sql);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs0 = stmt.executeQuery(sql0);  
 String Class\_no = "";  
 while (rs0.next()) {  
 Class\_no += rs0.getString("Sno");  
 }  
 if (Class\_no.equals("") == false) {  
 JOptionPane.*showMessageDialog*(null, "学号为"+textNo.getText()+"的同学信息已存在!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "学号为"+textNo.getText()+"的同学信息添加成功!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 }  
 rs0.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!");  
 }  
 }  
 if (e.getSource() == btnUpdate){  
 if(textNo.getText().trim().equals("") == false  
 &&(textName.getText().trim().equals("") == false  
 || textSex.getText().trim().equals("") == false  
 || textBirth.getText().trim().equals("") == false  
 || textEntrancedate.getText().trim().equals("") == false  
 || textClassno.getText().trim().equals("") == false  
 || textHomeaddress.getText().trim().equals("") == false  
 || textSdept.getText().trim().equals("") == false)){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = null;  
 if(textName.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sname = '"+textName.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textSex.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sex = '"+textSex.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textBirth.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Birth = "+textBirth.getText()+" where Sno = '"+textNo.getText()+"'";  
 else if(textEntrancedate.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Entrance\_date = '"+textEntrancedate.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textClassno.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Class\_no = '"+textClassno.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textHomeaddress.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Home\_addr = '"+textHomeaddress.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textSdept.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Sde\_no = '"+textSdept.getText()+"' where Sno = '"+textNo.getText()+"'";  
 else if(textPostcode.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Student\_20211576 set Postcode = '"+textPostcode.getText()+"' where Sno = '"+textNo.getText()+"'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql); *// 如果连接成功 控制台输出* JOptionPane.*showMessageDialog*(null, "学号为"+textNo.getText()+"的同学信息修改成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "学号和（姓名/性别/出生日期/入学日期/班级号/家庭住址/所在系/邮政编码中至少有一项）不为空!");  
 }  
 }  
 if (e.getSource() == btnSearch){  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textNo.getText().equals("") == true  
 && textName.getText().equals("") == true  
 && textSex.getText().equals("") == true  
 && textBirth.getText().equals("") == true  
 && textEntrancedate.getText().equals("") == true  
 && textClassno.getText().equals("") == true  
 && textHomeaddress.getText().equals("") == true  
 && textSdept.getText().equals("") == true  
 && textPostcode.getText().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 *// Get the number of columns in the ResultSet* ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 if(col == 3||col == 4){  
 data[row][col] = rs.getString(col + 1).substring(0,10);  
 }  
 else{  
 data[row][col] = rs.getString(col + 1);  
 }  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textNo中内容进行查询，当textNo中为空时方可对textName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textNo.getText().equals("") == false) {  
 Input = textNo.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sno = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textName.getText().equals("") == false) {  
 Input = textName.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sname = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textSex.getText().equals("") == false) {  
 Input = textSex.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sex = " + "'" +Input + "'";  
 } else if (textBirth.getText().equals("") == false) {  
 Input = textBirth.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Birth BETWEEN '" + Input + "' AND dateadd(second,-1,dateadd(day,1,'" + Input + "'))";  
 flag = 4;  
 } else if (textEntrancedate.getText().equals("") == false) {  
 Input = textEntrancedate.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Entrance\_date BETWEEN '" + Input + "' AND dateadd(second,-1,dateadd(day,1,'" + Input + "'))";  
 flag = 5;  
 } else if (textClassno.getText().equals("") == false) {  
 Input = textClassno.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Class\_no = " + "'" +Input + "'";  
 flag = 6;  
 } else if (textHomeaddress.getText().equals("") == false) {  
 Input = textHomeaddress.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Home\_addr = " + "'" +Input + "'";  
 flag = 7;  
 } else if (textSdept.getText().equals("") == false) {  
 Input = textSdept.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sde\_no = " + "'" +Input + "'";  
 flag = 8;  
 } else if (textPostcode.getText().equals("") == false) {  
 Input = textPostcode.getText();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Postcode = " + "'" +Input + "'";  
 flag = 9;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while(rs.next()){  
 data[count][0] = rs.getString("Sno");  
 data[count][1] = rs.getString("Sname");  
 data[count][2] = rs.getString("Sex");  
 data[count][3] = rs.getString("Birth");  
 data[count][4] = rs.getString("Entrance\_date");  
 data[count][5] = rs.getString("Class\_no");  
 data[count][6] = rs.getString("Home\_addr");  
 data[count][7] = rs.getString("Sde\_no");  
 data[count][8] = rs.getString("Postcode");  
 data[count][9] = rs.getString("AttendanceStatus");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
 }

（4）其他需要的函数：  
 public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
 }

2、维护教师授课信息：关键技术：数据库操作、表单验证。

（1）声明各类控件：

JPanel panel = new JPanel();  
JPanel panel1 = new JPanel();  
JPanel panel2 = new JPanel();  
JPanel panel4 = new JPanel();  
JPanel panel5 = new JPanel();  
JPanel panel6 = new JPanel();  
JPanel panel7 = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelSemester = new JLabel("学期");  
JComboBox comlabelSearchSemester = new JComboBox();  
JLabel labelCno = new JLabel("课程号");  
JPasswordField txtCno = new JPasswordField(12);  
JLabel labelEmp\_no = new JLabel("教工号");  
JTextField txtEmp\_no = new JTextField(12);  
JLabel labelCourseOfferedSdeNo = new JLabel("允许选课院系");  
JTextField txtCourseOfferedSdeNo = new JTextField(12);  
JButton btnAdd = new JButton("添加/修改");  
JLabel labelSearchSemester = new JLabel("学期");  
JComboBox comSearchSemester = new JComboBox();  
JLabel labelSearchCname = new JLabel("课程名");  
JTextField txtSearchCname = new JTextField(12);  
JLabel labelSearchName = new JLabel("授课教师");  
JTextField txtSearchName = new JTextField(12);  
JButton btnSearchSemester = new JButton("查询当前学期教师授课情况");  
JButton btnSearchCname = new JButton("查询当前课程教师授课情况");  
JButton btnSearchName = new JButton("查询当前教师授课情况");  
JButton btnSearchCnameAndName = new JButton("查询当前学期当前课程当前教师授课情况");  
JButton btnList = new JButton("显示所有学期所有教师授课情况");  
String initialText = null;  
JTable table = new JTable();  
JScrollPane scrollPane = new JScrollPane(table);  
String[] columnNames = {"学期", "可选课系号", "可选课系名", "课程号", "课程名", "授课教师", "学分", "总学时"};  
Object[][] data = new Object[getColumns()][8];

（2）创建面板：

public WindowManagerTeacherCourse() {  
 frame1.setTitle("教务管理系统管理员界面--教师授课管理");  
 frame1.setVisible(true);  
 frame1.setSize(620, 680);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panel, BorderLayout.*NORTH*);  
 frame1.add(panel4, BorderLayout.*CENTER*);  
 frame1.add(panel5, BorderLayout.*SOUTH*);  
 panel.add(panel1);  
 panel.add(panel2);  
 panel1.setLayout(new GridLayout(4, 2));  
 panel2.setLayout(new GridLayout(4, 1));  
 panel1.add(labelSemester);  
 panel1.add(comlabelSearchSemester);  
 panel1.add(labelCno);  
 panel1.add(txtCno);  
 panel1.add(labelEmp\_no);  
 panel1.add(txtEmp\_no);  
 panel1.add(labelCourseOfferedSdeNo);  
 panel1.add(txtCourseOfferedSdeNo);  
 panel2.add(btnAdd);  
 panel2.add(btnList);  
 panel4.add(scrollPane);  
 scrollPane.setViewportView(table);  
 panel5.add(panel6, BorderLayout.*EAST*);  
 panel5.add(panel7, BorderLayout.*WEST*);  
 panel7.setLayout(new GridLayout(4, 2));  
 panel6.setLayout(new GridLayout(4, 1));  
 panel7.add(labelSearchSemester);  
 panel7.add(comSearchSemester);  
 panel7.add(labelSearchCname);  
 panel7.add(txtSearchCname);  
 panel7.add(labelSearchName);  
 panel7.add(txtSearchName);  
 panel6.add(btnSearchSemester);  
 panel6.add(btnSearchCname);  
 panel6.add(btnSearchName);  
 panel6.add(btnSearchCnameAndName);  
 int date = Calendar.*getInstance*().get(Calendar.*YEAR*);  
 int i = date - 10;  
 for (; i < date + 1; i++) {  
 comSearchSemester.addItem(i + "-" + (i + 1) + "年第1学期");  
 comlabelSearchSemester.addItem(i + "-" + (i + 1) + "年第1学期");  
 comSearchSemester.addItem(i + "-" + (i + 1) + "年第2学期");  
 comlabelSearchSemester.addItem(i + "-" + (i + 1) + "年第2学期");  
 }  
 comSearchSemester.setSelectedIndex(18);  
 comlabelSearchSemester.setSelectedIndex(18);  
 initialText = (String) comlabelSearchSemester.getSelectedItem();  
 btnAdd.addActionListener(this);  
 btnList.addActionListener(this);  
 btnSearchSemester.addActionListener(this);  
 btnSearchCname.addActionListener(this);  
 btnSearchName.addActionListener(this);  
 btnSearchCnameAndName.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 WindowManagerTeacherCourseAdd wmtca = new WindowManagerTeacherCourseAdd();  
 }  
 if (e.getSource() == btnList) {  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view";  
 Connect(sql);  
 }  
 if (e.getSource() == btnSearchSemester) {  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Semester='" + comSearchSemester.getSelectedItem() + "'";  
 Connect(sql);  
 }  
 if (e.getSource() == btnSearchCname) {  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Cname='" + txtSearchCname.getText() + "'";  
 Connect(sql);  
 }  
 if (e.getSource() == btnSearchName) {  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Emp\_name='" + txtSearchName.getText() + "'";  
 Connect(sql);  
 }  
 if (e.getSource() == btnSearchCnameAndName) {  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Cname='" + txtSearchCname.getText() + "' and Emp\_name='" + txtSearchName.getText() + "'";  
 Connect(sql);  
 }  
}

（4）其他需要的函数：

public int getColumns () {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}  
public void Connect(String sql){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 8; j++) {  
 data[i][j] = "";  
 }  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}

（5）添加/修改窗口：

((1)) 声明各类控件：

Random rand = new Random();  
JPanel panel = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelSemester = new JLabel("学期");  
JComboBox comSearchSemester = new JComboBox();  
JLabel labelEmp\_no = new JLabel("教工号");  
JTextField txtEmp\_no = new JTextField(12);  
JLabel labelCno = new JLabel("课程号");  
JTextField txtCno = new JTextField(12);  
JLabel labelCourseOfferedSdeNo = new JLabel("允许选课院系");  
JTextField txtCourseOfferedSdeNo = new JTextField(12);  
JLabel labelCname = new JLabel("课程名");  
JTextField txtCname = new JTextField(12);  
JLabel labelCcredit = new JLabel("学分");  
JTextField txtCcredit = new JTextField(12);  
JLabel labelStuotal = new JLabel("可选学生数");  
JTextField txtStuotal = new JTextField(12);  
JButton btnAdd = new JButton("添加");  
JButton btnUpdate = new JButton("修改");

(（2）)创建面板：

public WindowManagerTeacherCourseAdd() {  
 frame1.setTitle("教务管理系统-管理员界面-教师授课管理-添加课程信息");  
 frame1.setVisible(true);  
 frame1.setSize(380,400);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panel, BorderLayout.*CENTER*);  
 panel.setLayout(new GridLayout(8,2));  
 panel.add(labelSemester);panel.add(comSearchSemester);  
 panel.add(labelEmp\_no);panel.add(txtEmp\_no);  
 panel.add(labelCno);panel.add(txtCno);  
 panel.add(labelCourseOfferedSdeNo);panel.add(txtCourseOfferedSdeNo);  
 panel.add(labelCname);panel.add(txtCname);  
 panel.add(labelCcredit);panel.add(txtCcredit);  
 panel.add(labelStuotal);panel.add(txtStuotal);  
 panel.add(btnAdd);panel.add(btnUpdate);  
 int date = Calendar.*getInstance*().get(Calendar.*YEAR*);  
 int i = date - 10;  
 for (; i < date + 1; i++) {  
 comSearchSemester.addItem(i + "-" + (i + 1) + "年第1学期");  
 comSearchSemester.addItem(i + "-" + (i + 1) + "年第2学期");  
 }  
 btnAdd.addActionListener(this);  
 btnUpdate.addActionListener(this);  
}

(（3）)按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd){  
 if(comSearchSemester.toString().equals("") == false  
 &&txtEmp\_no.getText().trim().equals("") == false  
 && txtCno.getText().trim().equals("") == false  
 && txtCourseOfferedSdeNo.getText().trim().equals("") == false  
 && txtCname.getText().trim().equals("") == false  
 && txtCcredit.getText().trim().equals("") == false  
 && txtStuotal.getText().trim().equals("") == false){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "insert into Course\_Available\_20211576(Semester,Emp\_no,Cno,Course\_Offered\_Sde\_no,Cname,Credit,Stu\_Total) values ('" + comSearchSemester.getSelectedItem() + "','" + txtEmp\_no.getText() + "','" + txtCno.getText() + "','" + txtCourseOfferedSdeNo.getText() + "','" + txtCname.getText() + "','" + txtCcredit.getText() + "',37)";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 JOptionPane.*showMessageDialog*(null, "添加成功,请重新打开界面！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 frame1.dispose();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 frame1.dispose();  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 }  
 if(e.getSource() == btnUpdate){  
 if(comSearchSemester.toString().equals("") == false  
 &&txtEmp\_no.getText().trim().equals("") == false  
 && txtCno.getText().trim().equals("") == false  
 && txtCourseOfferedSdeNo.getText().trim().equals("") == false  
 && (txtCname.getText().trim().equals("") == true  
 || txtCcredit.getText().trim().equals("") == true  
 || txtStuotal.getText().trim().equals("") == true)){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = null;  
 if(txtCname.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Course\_Available\_20211576 set Cname = '"+txtCname.getText()+"' where Emp\_no = '"+txtEmp\_no.getText()+"' and Semester = '"+comSearchSemester.getSelectedItem()+"' and Cno = '"+txtCno.getText()+"' and Course\_Offered\_Sde\_no = '"+txtCourseOfferedSdeNo.getText()+"'";  
 else if(txtCcredit.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Course\_Available\_20211576 set Credit = '"+txtCcredit.getText()+"' where Emp\_no = '"+txtEmp\_no.getText()+"' and Semester = '"+comSearchSemester.getSelectedItem()+"' and Cno = '"+txtCno.getText()+"' and Course\_Offered\_Sde\_no = '"+txtCourseOfferedSdeNo.getText()+"'";  
 else if(txtStuotal.getText().trim().equals("") == false)  
 sql = "update Academic\_Affairs\_Management\_System\_20211576.dbo.Course\_Available\_20211576 set Stu\_Total = '"+txtStuotal.getText()+"' where Emp\_no = '"+txtEmp\_no.getText()+"' and Semester = '"+comSearchSemester.getSelectedItem()+"' and Cno = '"+txtCno.getText()+"' and Course\_Offered\_Sde\_no = '"+txtCourseOfferedSdeNo.getText()+"'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println(sql); *// 如果连接成功 控制台输出* JOptionPane.*showMessageDialog*(null, "授课信息修改成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "工号和（姓名，性别，所属系别编号职位中至少有一项）不为空!");  
 }  
 }  
}

3、维护学生可选课程的信息：关键技术：数据库操作、表单验证。

（1）声明各类控件：

JPanel panelEast = new JPanel();  
JPanel panelNorth = new JPanel();  
JPanel panelCenter = new JPanel();  
JPanel panelNorthEast = new JPanel();  
JPanel panelNorthWest = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelSemester = new JLabel("学期");  
JComboBox comSemester = new JComboBox();  
JLabel labelSdept = new JLabel("系");  
JComboBox comSdept = new JComboBox();  
JLabel labelCno = new JLabel("课程号");  
JTextField txtCno = new JTextField(12);  
JLabel labelCname = new JLabel("课程名");  
JTextField txtCname = new JTextField(12);  
JLabel labelEmp\_Name = new JLabel("教师名");  
JTextField txtEmp\_Name = new JTextField(12);  
JLabel labelCredit = new JLabel("学分");  
JTextField txtCredit = new JTextField(12);  
JButton btnList = new JButton("信息一览");  
JButton btnSemestersEnquire = new JButton("当前学期内可选课程查询");  
JButton btnSdeptEnquire = new JButton("当前系可选课程查询");  
JButton btnSemesterandSdeptEnquire = new JButton("当前学期内当前系可选课程查询");  
JButton btnCnoEnquire = new JButton("课程号查询");  
JButton btnTeacherEnquire = new JButton("授课教师查询");  
JButton btnCreditEnquire = new JButton("学分对应课程查询");  
JButton btnCnameEnquire = new JButton("课程名查询");  
JTable table = new JTable();  
String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
Connection dbConn = null;  
String userName = "s20211576"; *// 默认用户名*String userPwd = "s20211576"; *// 密码*String[] columnNames = {"学期","可选修系","可选修专业","课程号","课程名","教师名","学分","选课人数"};  
String[][] data = new String[getColumns()][8];

（2）创建面板：

public WindowManagerCourseAvailable() {  
 frame1.setTitle("教务管理系统管理员界面-学生可选课程管理");  
 frame1.setVisible(true);  
 frame1.setSize(900,800);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);  
 panelNorth.add(panelNorthEast,BorderLayout.*EAST*);  
 panelNorth.add(panelNorthWest,BorderLayout.*WEST*);  
 panelNorthWest.setLayout(new GridLayout(6,2));  
 panelNorthEast.setLayout(new GridLayout(4,2));  
 panelNorthWest.add(labelSemester);panelNorthWest.add(comSemester);  
 int date = Calendar.*getInstance*().get(Calendar.*YEAR*);  
 int i=date-10;  
 for(;i<date+1;i++){  
 comSemester.addItem(i+"-"+(i+1)+"年第1学期");  
 comSemester.addItem(i+"-"+(i+1)+"年第2学期");  
 }  
 comSemester.setSelectedIndex(17);  
 panelNorthWest.add(labelSdept);panelNorthWest.add(comSdept);  
 *//通过数据库内Sdept\_20211576表内的数据添加系* try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
  
 Statement stml = dbConn.createStatement();  
 String sql = "SELECT Sde\_name FROM Sdept\_20211576";  
 ResultSet rs = stml.executeQuery(sql);  
  
 while (rs.next()) {  
 comSdept.addItem(rs.getString("Sde\_name"));  
 }  
  
 rs.close();  
 stml.close();  
 } catch (ClassNotFoundException | SQLException ex) {  
 ex.printStackTrace();  
 }  
 comSdept.setSelectedItem("软件工程");  
 panelNorthWest.add(labelCno);panelNorthWest.add(txtCno);  
 panelNorthWest.add(labelCname);panelNorthWest.add(txtCname);  
 panelNorthWest.add(labelEmp\_Name);panelNorthWest.add(txtEmp\_Name);  
 panelNorthWest.add(labelCredit);panelNorthWest.add(txtCredit);  
 panelNorthEast.add(btnList);panelNorthEast.add(btnSemestersEnquire);  
 panelNorthEast.add(btnSdeptEnquire);panelNorthEast.add(btnSemesterandSdeptEnquire);  
 panelNorthEast.add(btnCnoEnquire);panelNorthEast.add(btnCnameEnquire);  
 panelNorthEast.add(btnTeacherEnquire);panelNorthEast.add(btnCreditEnquire);  
 JScrollPane scrollPane = new JScrollPane(table);  
 panelCenter.add(scrollPane);  
 scrollPane.setViewportView(table);  
 btnList.addActionListener(this);  
 btnSemestersEnquire.addActionListener(this);  
 btnSdeptEnquire.addActionListener(this);  
 btnCnoEnquire.addActionListener(this);  
 btnCnameEnquire.addActionListener(this);  
 btnTeacherEnquire.addActionListener(this);  
 btnCreditEnquire.addActionListener(this);  
 btnSemesterandSdeptEnquire.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnList) {*//数据一览* String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view";  
 Connect(sql);  
 }else if (e.getSource() == btnSemestersEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 }else if (e.getSource() == btnSdeptEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Sde\_name='"+comSdept.getSelectedItem().toString().trim()+"'";  
 Connect(sql);  
 }else if (e.getSource() == btnSemesterandSdeptEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Semester='"+comSemester.getSelectedItem()+"' and Sde\_name='"+comSdept.getSelectedItem().toString().trim()+"'";  
 Connect(sql);  
 }else if (e.getSource() == btnCnoEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cno='" + txtCno.getText().trim() + "'";  
 Connect(sql);  
 } else if (e.getSource() == btnCnameEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cname='" + txtCname.getText().trim() + "'";  
 Connect(sql);  
 } else if (e.getSource() == btnTeacherEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Emp\_Name='" + txtEmp\_Name.getText().trim() + "'";  
 Connect(sql);  
 } else if (e.getSource() == btnCreditEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Credit='" + txtCredit.getText().trim() + "'";  
 Connect(sql);  
 }  
}

（4）其他需要的函数：

public int getColumns () {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}  
public void Connect(String sql){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 8; j++) {  
 data[i][j] = "";  
 }  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}

4、查询教师、学生的各项信息：关键技术：数据库查询、查询条件构造。

教师：（1）声明各类控件：

JPanel panel = new JPanel();  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelSouth = new JPanel();  
JLabel labelNo = new JLabel("工号");  
JTextField textNo = new JTextField(12);  
JLabel labelName = new JLabel("姓名");  
JTextField textName = new JTextField(12);  
JLabel labelSex = new JLabel("性别");  
JTextField textSex = new JTextField(12);  
JLabel labelSdept = new JLabel("所属专业编号");  
JTextField textSdept = new JTextField(12);  
JLabel labelPosition = new JLabel("职称");  
JTextField textPosition = new JTextField(12);  
JButton btnEnquiry = new JButton("查询/一览");  
JLabel labelStatus = new JLabel("上方空格均不填即查询所有记录，否则按照填写的信息查询");  
JTable table = new JTable();  
JScrollPane scrollPane = new JScrollPane(table);  
  
String[] columnNames = {"工号","姓名","性别","所属专业编号","职称"};  
Object[][] data = new Object[getColumns()][10];

（2）创建面板：

public WindowManagerEnquiryTeacher() {  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-教师信息查询");  
 frame1.setVisible(true);  
 frame1.setSize(800, 680);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelSouth, BorderLayout.*CENTER*);  
 panelNorth.setLayout(new GridLayout(9, 2));  
 panelNorth.add(labelNo);panelNorth.add(textNo);  
 panelNorth.add(labelName);panelNorth.add(textName);  
 panelNorth.add(labelSex);panelNorth.add(textSex);  
 panelNorth.add(labelSdept);panelNorth.add(textSdept);  
 panelNorth.add(labelPosition);panelNorth.add(textPosition);  
 panelSouth.add(btnEnquiry, BorderLayout.*NORTH*);  
 panelSouth.add(labelStatus, BorderLayout.*NORTH*);  
 panelSouth.add(scrollPane, BorderLayout.*CENTER*);  
 scrollPane.setViewportView(table);  
 btnEnquiry.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnEnquiry) {  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textNo.getText().toString().equals("") == true  
 && textName.getText().toString().equals("") == true  
 && textSex.getText().toString().equals("") == true  
 && textSdept.getText().toString().equals("") == true  
 && textPosition.getText().toString().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 *// Get the number of columns in the ResultSet* ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
  
 *// Retrieve all rows of data from the ResultSet and store in the data array* int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textNo中内容进行查询，当textNo中为空时方可对textName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textNo.getText().toString().equals("") == false) {  
 Input = textNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_no = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textName.getText().toString().equals("") == false) {  
 Input = textName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Emp\_name = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textSex.getText().toString().equals("") == false) {  
 Input = textSex.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Sex = " + "'" +Input + "'";  
 flag = 3;  
 } else if (textSdept.getText().toString().equals("") == false) {  
 Input = textSdept.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Sde\_no = " + "'" +Input + "'";  
 flag = 4;  
 } else if (textPosition.getText().toString().equals("") == false) {  
 Input = textPosition.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view where Position = " + "'" +Input + "'";  
 flag = 5;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 int count = 0;  
 while(rs.next()){  
 data[count][0] = rs.getString("Emp\_no");  
 data[count][1] = rs.getString("Emp\_name");  
 data[count][2] = rs.getString("Sex");  
 data[count][3] = rs.getString("Sde\_no");  
 data[count][4] = rs.getString("Position");  
 count++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 *// Create a new DefaultTableModel with the data and column names* DefaultTableModel model = new DefaultTableModel(data, columnNames);  
  
 *// Set the model for the JTable* table.setModel(model);  
 }  
}

（4）其他需要的函数：

public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 System.*out*.println("Connection Successful!"); *// 如果连接成功 控制台输出* } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Teacher\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}

学生：（1）声明各类控件：

JPanel panel = new JPanel();  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelSouth = new JPanel();  
JLabel labelNo = new JLabel("学号");  
JTextField textNo = new JTextField(12);  
JLabel labelName = new JLabel("姓名");  
JTextField textName = new JTextField(12);  
JLabel labelSex = new JLabel("性别");  
JTextField textSex = new JTextField(12);  
JLabel labelBirth = new JLabel("出生日期");  
JTextField textBirth = new JTextField(12);  
JLabel labelEntrancedate = new JLabel("入学日期");  
JTextField textEntrancedate = new JTextField(12);  
JLabel labelClassno = new JLabel("班级号");  
JTextField textClassno = new JTextField(12);  
JLabel labelHomeaddress = new JLabel("家庭住址");  
JTextField textHomeaddress = new JTextField(50);  
JLabel labelSdept = new JLabel("所在系");  
JTextField textSdept = new JTextField(12);  
JLabel labelPostcode = new JLabel("邮政编码");  
JTextField textPostcode = new JTextField(12);  
JButton btnEnquiry = new JButton("查询/一览");  
JLabel labelStatus = new JLabel("上方空格均不填即查询所有记录，否则按照填写的信息查询");  
JTable table = new JTable();  
JScrollPane scrollPane = new JScrollPane(table);  
  
String[] columnNames = {"学号", "姓名", "性别", "出生日期", "入学日期", "班级号", "家庭住址", "所在系", "邮政编码","就读状态"};  
Object[][] data = new Object[getColumns()][10];

（2）创建面板：

public WindowManagerEnquiryStudent() {  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统管理员界面-信息维护功能-学生信息查询");  
 frame1.setVisible(true);  
 frame1.setSize(800, 680);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelSouth, BorderLayout.*CENTER*);  
 panelNorth.setLayout(new GridLayout(9, 2));  
 panelNorth.add(labelNo);  
 panelNorth.add(textNo);  
 panelNorth.add(labelName);  
 panelNorth.add(textName);  
 panelNorth.add(labelSex);  
 panelNorth.add(textSex);  
 panelNorth.add(labelBirth);  
 panelNorth.add(textBirth);  
 panelNorth.add(labelEntrancedate);  
 panelNorth.add(textEntrancedate);  
 panelNorth.add(labelClassno);  
 panelNorth.add(textClassno);  
 panelNorth.add(labelHomeaddress);  
 panelNorth.add(textHomeaddress);  
 panelNorth.add(labelSdept);  
 panelNorth.add(textSdept);  
 panelNorth.add(labelPostcode);  
 panelNorth.add(textPostcode);  
 panelSouth.add(btnEnquiry, BorderLayout.*NORTH*);  
 panelSouth.add(labelStatus, BorderLayout.*NORTH*);  
 panelSouth.add(scrollPane, BorderLayout.*CENTER*);  
 scrollPane.setViewportView(table);  
 btnEnquiry.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnEnquiry) {  
 *//清空表格* for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 10; j++) {  
 data[i][j] = "";  
 }  
 }  
 if (textNo.getText().toString().equals("") == true  
 && textName.getText().toString().equals("") == true  
 && textSex.getText().toString().equals("") == true  
 && textBirth.getText().toString().equals("") == true  
 && textEntrancedate.getText().toString().equals("") == true  
 && textClassno.getText().toString().equals("") == true  
 && textHomeaddress.getText().toString().equals("") == true  
 && textSdept.getText().toString().equals("") == true  
 && textPostcode.getText().toString().equals("") == true){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 *// Get the number of columns in the ResultSet* ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 else {  
 *//在数据库中进行查询,先对textNo中内容进行查询，当textNo中为空时方可对textName进行查询,往后以此类推* int flag = 0;  
 String Input = null;  
 String sql = null;  
 if (textNo.getText().toString().equals("") == false) {  
 Input = textNo.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sno = " + "'" +Input + "'";  
 flag = 1;  
 } else if (textName.getText().toString().equals("") == false) {  
 Input = textName.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select password from Manager\_Student\_view where Sname = " + "'" +Input + "'";  
 flag = 2;  
 } else if (textSex.getText().toString().equals("") == false) {  
 Input = textSex.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select password from Manager\_Student\_view where Sex = " + "'" +Input + "'";  
 } else if (textBirth.getText().toString().equals("") == false) {  
 Input = textBirth.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Birth BETWEEN '" + Input + "' AND dateadd(second,-1,dateadd(day,1,'" + Input + "'))";  
 flag = 4;  
 } else if (textEntrancedate.getText().toString().equals("") == false) {  
 Input = textEntrancedate.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Entrance\_date BETWEEN '" + Input + "' AND dateadd(second,-1,dateadd(day,1,'" + Input + "'))";  
 flag = 5;  
 } else if (textClassno.getText().toString().equals("") == false) {  
 Input = textClassno.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Class\_no = " + "'" +Input + "'";  
 flag = 6;  
 } else if (textHomeaddress.getText().toString().equals("") == false) {  
 Input = textHomeaddress.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Home\_addr = " + "'" +Input + "'";  
 flag = 7;  
 } else if (textSdept.getText().toString().equals("") == false) {  
 Input = textSdept.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Sde\_no = " + "'" +Input + "'";  
 flag = 8;  
 } else if (textPostcode.getText().toString().equals("") == false) {  
 Input = textPostcode.getText().toString();  
 sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view where Postcode = " + "'" +Input + "'";  
 flag = 9;  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 while(rs.next()){  
 data[0][0] = rs.getString("Sno");  
 data[0][1] = rs.getString("Sname");  
 data[0][2] = rs.getString("Sex");  
 data[0][3] = rs.getString("Birth");  
 data[0][4] = rs.getString("Entrance\_date");  
 data[0][5] = rs.getString("Class\_no");  
 data[0][6] = rs.getString("Home\_addr");  
 data[0][7] = rs.getString("Sde\_no");  
 data[0][8] = rs.getString("Postcode");  
 data[0][9] = rs.getString("AttendanceStatus");  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
}

（4）其他需要的函数：

public int getColumns() {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Student\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}

5、添加教务管理系统管理员用户：关键技术：数据库操作、表单验证。

（1）声明各类控件：

JPanel panel = new JPanel();  
JPanel panel1 = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelEmp\_No = new JLabel("教师工号");  
JTextField txtEmp\_No = new JTextField(10);  
JLabel labelEmp\_name = new JLabel("教师姓名");  
JTextField txtEmp\_name = new JTextField(10);  
JLabel labelSex = new JLabel("性别");  
JTextField txtSex = new JTextField(10);  
JLabel labelEmp\_SdeptNo = new JLabel("所属院系编号");  
JTextField txtEmp\_SdeptNo = new JTextField(10);  
JButton btnAdd = new JButton("添加");

（2）创建面板：

public WindowManagerAddTeacherPassword() {  
 frame1.setTitle("教务管理系统管理员界面--添加管理员用户");  
 frame1.setVisible(true);  
 frame1.setSize(400, 400);  
 frame1.setLocation(300, 249);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panel, BorderLayout.*CENTER*);  
 panel.setLayout(new GridLayout(4,2));  
 frame1.add(panel1, BorderLayout.*SOUTH*);  
 panel.add(labelEmp\_No);panel.add(txtEmp\_No);  
 panel.add(labelEmp\_name);panel.add(txtEmp\_name);  
 panel.add(labelSex);panel.add(txtSex);  
 panel.add(labelEmp\_SdeptNo);panel.add(txtEmp\_SdeptNo);  
 panel1.add(btnAdd);  
 btnAdd.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnAdd) {  
 if (txtEmp\_No.getText().equals("") || txtEmp\_name.getText().equals("") || txtSex.getText().equals("") || txtEmp\_SdeptNo.getText().equals("")) {  
 JOptionPane.*showMessageDialog*(null, "请填写完整信息", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 return;  
 }  
 else{  
 frame1.dispose();  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql1 = "use Academic\_Affairs\_Management\_System\_20211576 insert into Teacher\_20211576 values ('" + txtEmp\_No.getText().toString() + "','" + txtEmp\_name.getText().toString() + "','" + txtSex.getText().toString() + "','" + txtEmp\_SdeptNo.getText().toString() + "','教务管理人员')"; *// SQL语句* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 JOptionPane.*showMessageDialog*(null, "教师表添加成功！密码表添加成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 ResultSet rs = stmt.executeQuery(sql1);  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 }  
 }  
}

（三）教务管理系统学生功能模块

1、查看个人信息：关键技术：数据库查询、数据展示。

（1）声明各类控件：

String loginID = windowsRegister.*loginID*;  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelSouth = new JPanel();  
JPanel panelCenter = new JPanel();  
JLabel labelSno = new JLabel("学号");  
JLabel txtSno;  
JLabel labelSname = new JLabel("姓名");  
JLabel txtSname;  
JLabel labelSex = new JLabel("性别");  
JLabel txtSex;  
JLabel labelBirth = new JLabel("出生日期");  
JLabel txtBirth;  
JLabel labelEntranceDate = new JLabel("入学日期");  
JLabel txtEntranceDate;  
JLabel labelClass = new JLabel("班级号");  
JLabel txtClass;  
JLabel labelAddress = new JLabel("家庭住址");  
JLabel txtAddress;  
JLabel labelSdeNo = new JLabel("系号");  
JLabel txtSdeNo;  
JLabel labelSdeName = new JLabel("所在系");  
JLabel txtSdeName;  
JLabel labelPostcode = new JLabel("邮政编码");  
JLabel txtPostcode;  
JLabel labelStatus = new JLabel("就读状态");  
JLabel txtStatus;

（2）创建面板：

public WindowStudentEnquiry() {  
 System.*out*.println(loginID);  
 Object[] StudentRelated = new Object[11];  
 getInformation(StudentRelated);  
 txtSno = new JLabel((String) StudentRelated[0]);  
 txtSname = new JLabel((String) StudentRelated[1]);  
 txtSex = new JLabel((String) StudentRelated[2]);  
 txtBirth = new JLabel((String) StudentRelated[3]);  
 txtEntranceDate = new JLabel((String) StudentRelated[4]);  
 txtClass = new JLabel((String) StudentRelated[5]);  
 txtAddress = new JLabel((String) StudentRelated[6]);  
 txtSdeNo = new JLabel((String) StudentRelated[7]);  
 txtSdeName = new JLabel((String) StudentRelated[8]);  
 txtPostcode = new JLabel((String) StudentRelated[9]);  
 txtStatus = new JLabel((String) StudentRelated[10]);  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统学生界面-学生信息查询");  
 frame1.setVisible(true);  
 frame1.setSize(680, 600);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);  
 frame1.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(6, 4));  
 panelNorth.add(labelSno);panelNorth.add(txtSno);  
 panelNorth.add(labelSname);panelNorth.add(txtSname);  
 panelNorth.add(labelSex);panelNorth.add(txtSex);  
 panelNorth.add(labelBirth);panelNorth.add(txtBirth);  
 panelNorth.add(labelEntranceDate);panelNorth.add(txtEntranceDate);  
 panelNorth.add(labelClass);panelNorth.add(txtClass);  
 panelNorth.add(labelAddress);panelNorth.add(txtAddress);  
 panelNorth.add(labelSdeNo);panelNorth.add(txtSdeNo);  
 panelNorth.add(labelSdeName);panelNorth.add(txtSdeName);  
 panelNorth.add(labelPostcode);panelNorth.add(txtPostcode);  
 panelNorth.add(labelStatus);panelNorth.add(txtStatus);   
}

（3）按钮事件：

无

（4）其他需要的函数：

public void getInformation(Object[] StudentRelated){  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_view where Sno = " + loginID.trim();  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 StudentRelated[0] = rs.getString("Sno");  
 StudentRelated[1] = rs.getString("Sname");  
 StudentRelated[2] = rs.getString("Sex");  
 StudentRelated[3] = rs.getString("Birth").substring(0,10);  
 StudentRelated[4] = rs.getString("Entrance\_date").substring(0,10);  
 StudentRelated[5] = rs.getString("Class\_no");  
 StudentRelated[6] = rs.getString("Home\_addr");  
 StudentRelated[7] = rs.getString("Sde\_no");  
 StudentRelated[8] = rs.getString("Sde\_name");  
 StudentRelated[9] = rs.getString("Postcode");  
 StudentRelated[10] = rs.getString("AttendanceStatus");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}

2、查看已选课程：关键技术：数据库查询、数据展示。

（1）声明各类控件：

JButton btnEnquiry = new JButton("已选课程列表");  
JTable table = new JTable();  
String[] columnNames = {"学期", "课程号", "课程名", "任课教师工号", "学分"};  
Object[][] data = new Object[getColumns()][5];

（2）创建面板：

public WindowStudentEnquiry() {  
 System.*out*.println(loginID);  
 Object[] StudentRelated = new Object[11];  
 getInformation(StudentRelated);  
 txtSno = new JLabel((String) StudentRelated[0]);  
 txtSname = new JLabel((String) StudentRelated[1]);  
 txtSex = new JLabel((String) StudentRelated[2]);  
 txtBirth = new JLabel((String) StudentRelated[3]);  
 txtEntranceDate = new JLabel((String) StudentRelated[4]);  
 txtClass = new JLabel((String) StudentRelated[5]);  
 txtAddress = new JLabel((String) StudentRelated[6]);  
 txtSdeNo = new JLabel((String) StudentRelated[7]);  
 txtSdeName = new JLabel((String) StudentRelated[8]);  
 txtPostcode = new JLabel((String) StudentRelated[9]);  
 txtStatus = new JLabel((String) StudentRelated[10]);  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统学生界面-学生信息查询");  
 frame1.setVisible(true);  
 frame1.setSize(680, 600);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);   
 panelCenter.add(btnEnquiry);  
 frame1.add(scrollPane,BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 btnEnquiry.addActionListener(this);  
}

（3）按钮事件：

public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnEnquiry) {  
 String sqlEnquiryStudent = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_Enquiry\_view where Sno = '" + loginID.toString() + "'";  
 btnEnquiry(sqlEnquiryStudent);  
 }  
}

（4）其他需要的函数：

public void btnEnquiry(String sql){  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 int columnCount = 0;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 data[0][0] = rs.getString("Semester");  
 data[0][1] = rs.getString("Cno");  
 data[0][2] = rs.getString("Cname");  
 data[0][3] = rs.getString("Emp\_name");  
 data[0][4] = rs.getString("Credit");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}  
public int getColumns () {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_Enquiry\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}

3、选课：关键技术：表单验证、数据库操作。

（1）声明各类控件：

String loginID = windowsRegister.*loginID*;  
JPanel panelNorth = new JPanel();  
JPanel panelCenter = new JPanel();  
JPanel panelNorthEast = new JPanel();  
JPanel panelNorthWest = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelSemester = new JLabel("学期");  
JComboBox comSemester = new JComboBox();  
JLabel labelSdept = new JLabel("系");  
JLabel comSdept = new JLabel(getSdept());  
JLabel labelCno = new JLabel("课程号");  
JTextField txtCno = new JTextField(12);  
JLabel labelCname = new JLabel("课程名");  
JTextField txtCname = new JTextField(12);  
JLabel labelEmp\_Name = new JLabel("教师名");  
JTextField txtEmp\_Name = new JTextField(12);  
JLabel labelCredit = new JLabel("学分");  
JTextField txtCredit = new JTextField(12);  
JButton btnSemestersEnquire = new JButton("当前学期内可选课程查询");  
JButton btnCnoEnquire = new JButton("课程号查询");  
JButton btnTeacherEnquire = new JButton("授课教师查询");  
JButton btnCreditEnquire = new JButton("学分对应课程查询");  
JButton btnCnameEnquire = new JButton("课程名查询");  
JButton btnChoose = new JButton("选课");  
JTable table = new JTable();  
String sqlInsert = "";  
String sqlCheck = "";  
String[] columnNames = {"学期","可选修系","可选修专业","课程号","课程名","教师名","学分","可选课人数"};  
String[][] data = new String[getColumns()][8];

（2）创建面板：

public WindowStudentCourseChoose() {  
 frame1.setTitle("教务管理系统学生界面-学生选课界面");  
 frame1.setVisible(true);  
 frame1.setSize(800,540);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 panelNorth.add(panelNorthEast,BorderLayout.*EAST*);  
 panelNorth.add(panelNorthWest,BorderLayout.*WEST*);  
 panelNorth.add(panelCenter,BorderLayout.*CENTER*);  
 panelNorthWest.setLayout(new GridLayout(6,2));  
 panelNorthEast.setLayout(new GridLayout(4,2));  
 panelCenter.setLayout(new GridLayout(2,1));  
 panelNorthEast.add(btnSemestersEnquire);  
 panelNorthEast.add(btnCnoEnquire);panelNorthEast.add(btnCnameEnquire);  
 panelNorthEast.add(btnTeacherEnquire);panelNorthEast.add(btnCreditEnquire);  
 panelNorthEast.add(btnChoose);  
 btnChoose.setPreferredSize(new Dimension(25,30));  
  
 panelNorthWest.add(labelSemester);panelNorthWest.add(comSemester);  
 int date = Calendar.*getInstance*().get(Calendar.*YEAR*);  
 LocalDate dateStart = LocalDate.*of*(date, 1, 1);  
 LocalDate dateSpring = LocalDate.*of*(date, 3, 1);  
 LocalDate dateAutumn = LocalDate.*of*(date, 9, 1);  
 LocalDate today = LocalDate.*now*();  
 long dateSpringChoose = dateSpring.until(dateStart, ChronoUnit.*DAYS*);  
 long dateAutumnChoose = dateAutumn.until(dateStart, ChronoUnit.*DAYS*);  
 long todayChoose = today.until(dateStart, ChronoUnit.*DAYS*);  
 System.*out*.print(today);  
 if(todayChoose<=dateSpringChoose){  
 comSemester.addItem((date-1)+"-"+date+"年第1学期");  
 comSemester.addItem((date-1)+"-"+date+"年第2学期");  
 comSemester.addItem(date+"-"+(date+1)+"年第1学期");  
 comSemester.setSelectedIndex(2);  
 }  
 else if(todayChoose<=dateAutumnChoose&&todayChoose>dateSpringChoose){  
 comSemester.addItem((date-1)+"-"+date+"年第2学期");  
 comSemester.addItem(date+"-"+(date+1)+"年第1学期");  
 comSemester.addItem(date+"-"+(date+1)+"年第2学期");  
 comSemester.setSelectedIndex(2);  
 }  
 else if(todayChoose>dateAutumnChoose){  
 comSemester.addItem(date+"-"+(date+1)+"年第1学期");  
 comSemester.addItem(date+"-"+(date+1)+"年第2学期");  
 comSemester.addItem((date+1)+"-"+(date+2)+"年第1学期");  
 comSemester.setSelectedIndex(2);  
 }  
 panelNorthWest.add(labelSdept);panelNorthWest.add(comSdept);  
 panelNorthWest.add(labelCno);panelNorthWest.add(txtCno);  
 panelNorthWest.add(labelCname);panelNorthWest.add(txtCname);  
 panelNorthWest.add(labelEmp\_Name);panelNorthWest.add(txtEmp\_Name);  
 panelNorthWest.add(labelCredit);panelNorthWest.add(txtCredit);  
 JScrollPane scrollPane = new JScrollPane(table);  
 frame1.add(scrollPane,BorderLayout.*CENTER*);  
 scrollPane.setViewportView(table);  
 btnSemestersEnquire.addActionListener(this);  
 btnCnoEnquire.addActionListener(this);  
 btnCnameEnquire.addActionListener(this);  
 btnTeacherEnquire.addActionListener(this);  
 btnCreditEnquire.addActionListener(this);  
 btnChoose.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnSemestersEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Semester='"+comSemester.getSelectedItem()+"' and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 }else if (e.getSource() == btnCnoEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cno='" + txtCno.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 } else if (e.getSource() == btnCnameEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Cname='" + txtCname.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 } else if (e.getSource() == btnTeacherEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Emp\_Name='" + txtEmp\_Name.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 } else if (e.getSource() == btnCreditEnquire) {  
 String sql = "SELECT \* FROM Manager\_Course\_Teacher\_manage\_view where Credit='" + txtCredit.getText().trim() + "'and Sde\_name='"+comSdept.getText().trim()+"' and Semester='"+comSemester.getSelectedItem()+"'";  
 Connect(sql);  
 } else if (e.getSource() == btnChoose) {  
 if(sqlInsert.equals("")) {  
 JOptionPane.*showMessageDialog*(null, "请先选择课程！", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 }else {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ec) {  
 throw new RuntimeException(ec);  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println("sqlInsert:"+sqlInsert);  
 if(checkInsert(sqlCheck) == true){*//checkInsert:检查该生是否已选过这门课* JOptionPane.*showMessageDialog*(null, "该课程已选！", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 dbConn.close();  
 return;  
 }  
 else{  
 JOptionPane.*showMessageDialog*(null, "选课成功！", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 ResultSet rs = stmt.executeQuery(sqlInsert);  
  
 } catch (SQLException ee) {  
 throw new RuntimeException(ee);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
 }  
 }  
 *// 添加监听器，监听表格的行选中事件* table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
 public void valueChanged(ListSelectionEvent event) {  
 *// 获取当前选中行的索引* int row = table.getSelectedRow();  
 if (row >= 0) {  
 *// 获取选中行的各列的数据* String insertSemester = table.getValueAt(row, 0).toString();  
 String insertSno = loginID.toString();  
 String insertCourseOfferedSdeNo = table.getValueAt(row, 1).toString();  
 String insertCno = table.getValueAt(row, 3).toString();  
 String insertCname = table.getValueAt(row, 4).toString();  
 *//中间变量insertEmpName* String insertEmpName = table.getValueAt(row, 5).toString();  
 String insertEmpNo = getEmpNo(insertEmpName);  
 String insertCredit = table.getValueAt(row, 6).toString();  
 String insertStuTotal = table.getValueAt(row, 7).toString();  
 sqlInsert = "INSERT INTO Course\_chosen\_20211576(Semester,Sno,Cno,Cname,Emp\_no,Credit,Course\_Offered\_Sde\_no) VALUES ('" + insertSemester.trim() + "','" + insertSno.trim() + "','" + insertCno.trim() + "','" + insertCname.trim() + "','" + insertEmpNo.trim() + "'," + insertCredit + ",'" + insertCourseOfferedSdeNo.trim() + "')";  
 sqlCheck = "SELECT \* FROM Course\_chosen\_20211576 where Sno='" + insertSno.trim() + "' and Cno='" + insertCno.trim() + "' and Semester='" + insertSemester.trim() + "' and Course\_Offered\_Sde\_no='" + insertCourseOfferedSdeNo.trim() + "' and Emp\_no='" + insertEmpNo.trim() + "'";  
 }  
 }  
 });  
}

（4）其他需要的函数：

private boolean checkInsert(String sqlCheck) {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* boolean flag = false;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sqlCheck);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 flag = true;  
 break;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return flag;  
}  
private String getEmpNo(String insertEmpName) {  
 String insertEmpNo = null;  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Teacher\_view where Emp\_Name='" + insertEmpName + "'");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 insertEmpNo = rs.getString("Emp\_no");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return insertEmpNo;  
}  
  
public int getColumns () {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ex) {  
 throw new RuntimeException(ex);  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}  
public void Connect(String sql){  
 for (int i = 0; i < getColumns(); i++) {  
 for (int j = 0; j < 8; j++) {  
 data[i][j] = "";  
 }  
 }  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 Statement stmt = dbConn.createStatement();  
 System.*out*.println(sql);  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int numColumns = rsmd.getColumnCount();  
 int row = 0;  
 while (rs.next()) {  
 for (int col = 0; col < numColumns; col++) {  
 data[row][col] = rs.getString(col + 1);  
 }  
 row++;  
 }  
 rs.close();  
 stmt.close();  
 dbConn.close();  
 } catch (Exception em) {  
 em.printStackTrace();  
 } finally {  
 try {  
 } catch (Exception ec) {  
 ec.printStackTrace();  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}  
public String getSdept(){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String txtSdept = null;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
  
 Statement stml = dbConn.createStatement();  
 String sql = "SELECT \* FROM Student\_view where Sno = '"+loginID+"'";  
 ResultSet rs = stml.executeQuery(sql);  
  
 while (rs.next()) {  
 txtSdept = rs.getString("Sde\_name");  
 }  
  
 rs.close();  
 stml.close();  
 } catch (ClassNotFoundException | SQLException ex) {  
 ex.printStackTrace();  
 }  
 return txtSdept;  
}

4、查看课程成绩：关键技术：数据库查询、数据展示。

（1）声明各类控件：

String loginID = windowsRegister.*loginID*;  
String loginName = windowsRegister.*loginName*;  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelSouth = new JPanel();  
JPanel panelCenter = new JPanel();  
JLabel labelSno = new JLabel("学号");  
JLabel txtSno;  
JLabel labelSname = new JLabel("姓名");  
JLabel txtSname;  
JLabel labelEntranceDate = new JLabel("入学日期");  
JLabel txtEntranceDate;  
JLabel labelClass = new JLabel("班级号");  
JLabel txtClass;  
JLabel labelSdeNo = new JLabel("系号");  
JLabel txtSdeNo;  
JLabel labelSdeName = new JLabel("所在系");  
JLabel txtSdeName;  
JLabel labelStatus = new JLabel("就读状态");  
JLabel txtStatus;  
JButton btnEnquiry = new JButton("成绩查询");  
JLabel btnWarning = new JLabel("分数为空白或0表示该课程尚未结课或教师尚未赋分，请稍后查询!");  
JTable table = new JTable();  
String[] columnNames = {"学期","课程号","课程名","授课教师","成绩"};  
Object[][] data = new Object[getColumns()][5];

（2）创建面板：

public WindowStudentScoreEnquiry() {  
 System.*out*.println(loginID);  
 System.*out*.println(getColumns());  
 Object[] StudentRelated = new Object[11];  
 getInformation(StudentRelated);  
 txtSno = new JLabel(loginID);  
 txtSname = new JLabel(loginName);  
 txtEntranceDate = new JLabel((String) StudentRelated[4]);  
 txtClass = new JLabel((String) StudentRelated[5]);  
 txtSdeNo = new JLabel((String) StudentRelated[7]);  
 txtSdeName = new JLabel((String) StudentRelated[8]);  
 txtStatus = new JLabel((String) StudentRelated[10]);  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统学生界面-学生成绩查询");  
 frame1.setVisible(true);  
 frame1.setSize(680, 600);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);  
 frame1.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(4, 4));  
 panelNorth.add(labelSno);panelNorth.add(txtSno);  
 panelNorth.add(labelSname);panelNorth.add(txtSname);  
 panelNorth.add(labelEntranceDate);panelNorth.add(txtEntranceDate);  
 panelNorth.add(labelClass);panelNorth.add(txtClass);  
 panelNorth.add(labelSdeNo);panelNorth.add(txtSdeNo);  
 panelNorth.add(labelSdeName);panelNorth.add(txtSdeName);  
 panelNorth.add(labelStatus);panelNorth.add(txtStatus);  
 JScrollPane scrollPane = new JScrollPane(table);  
 panelCenter.setLayout(new GridLayout(2, 1));  
 panelCenter.add(btnEnquiry);panelCenter.add(btnWarning);  
 frame1.add(scrollPane,BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 btnEnquiry.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnEnquiry) {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sqlCourseEnquiry = "select \* from Student\_Score\_Enquiry\_view where Sno = '" + loginID + "' and Sname = '" + loginName + "' order by Semester asc";  
 System.*out*.println(sqlCourseEnquiry);  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e1) {  
 throw new RuntimeException(e1);  
 } catch (SQLException e2) {  
 throw new RuntimeException(e2);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sqlCourseEnquiry);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 int count = 0;  
 while (rs.next()&&count<getColumns()) {  
 data[count][0] = rs.getString("Semester");  
 data[count][1] = rs.getString("Cno");  
 data[count][2] = rs.getString("Cname");  
 data[count][3] = rs.getString("Emp\_name");  
 data[count][4] = rs.getString("Grade");  
 count++;  
 }  
 } catch (SQLException e3) {  
 throw new RuntimeException(e3);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e4) {  
 throw new RuntimeException(e4);  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 }  
}

（4）其他需要的函数：

public int getColumns () {  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_Score\_Enquiry\_view");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}  
*//保留*public void getInformation(Object[] StudentRelated){  
 String Driver2 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Student\_view where Sno = '" + loginID.trim() + "'";  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver2);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 StudentRelated[0] = rs.getString("Sno");  
 StudentRelated[1] = rs.getString("Sname");  
 StudentRelated[2] = rs.getString("Sex");  
 StudentRelated[3] = rs.getString("Birth").substring(0,10);  
 StudentRelated[4] = rs.getString("Entrance\_date").substring(0,10);  
 StudentRelated[5] = rs.getString("Class\_no");  
 StudentRelated[6] = rs.getString("Home\_addr");  
 StudentRelated[7] = rs.getString("Sde\_no");  
 StudentRelated[8] = rs.getString("Sde\_name");  
 StudentRelated[9] = rs.getString("Postcode");  
 StudentRelated[10] = rs.getString("AttendanceStatus");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}

（四）教务管理系统教师功能模块

1、查询所授课程：关键技术：数据库查询、数据展示。

（1）声明各类控件：

String loginID = windowsRegister.*loginID*;  
String loginName = windowsRegister.*loginName*;  
static String *sqlScoreManage* = "";  
static String *sqlScore* = "";  
JFrame frame1 = new JFrame();  
JPanel panelNorth = new JPanel();  
JPanel panelSouth = new JPanel();  
JPanel panelCenter = new JPanel();  
JLabel labelNo = new JLabel("工号");  
JLabel txtNo;  
JLabel labelName = new JLabel("姓名");  
JLabel txtName;  
JLabel labelSex = new JLabel("性别");  
JLabel txtSex;  
JLabel labelSdeNo = new JLabel("所在系");  
JLabel txtSdeNo;  
JLabel labelSdeName = new JLabel("系名");  
JLabel txtSdeName;  
JLabel labelPosition = new JLabel("职称");  
JLabel txtPosition;  
JButton btnEnquiry = new JButton("授课列表");  
JButton btnScoreManage = new JButton("成绩管理");  
JTable table = new JTable();  
String[] columnNames = {"学期","允许选课系","系名","课程号","课程名","学分","可选课人数"};  
Object[][] data = new Object[getColumns()][7];  
Object[] TeacherRelated = new Object[6];

（2）创建面板：

public WindowTeacher() {  
 System.*out*.println(loginID);  
 System.*out*.println(loginName);  
 getInformation(TeacherRelated);  
 txtNo = new JLabel((String)TeacherRelated[0]);  
 txtName = new JLabel((String)TeacherRelated[1]);  
 txtSex = new JLabel((String)TeacherRelated[2]);  
 txtSdeNo = new JLabel((String)TeacherRelated[3]);  
 txtSdeName = new JLabel((String)TeacherRelated[4]);  
 txtPosition = new JLabel((String)TeacherRelated[5]);  
 table = new JTable(data, columnNames);  
 frame1.setTitle("教务管理系统-教师界面");  
 frame1.setVisible(true);  
 frame1.setSize(680, 540);  
 frame1.setLocation(10, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);  
 frame1.add(panelSouth, BorderLayout.*SOUTH*);  
 panelNorth.setLayout(new GridLayout(3, 4));  
 panelNorth.add(labelNo);panelNorth.add(txtNo);  
 panelNorth.add(labelName);panelNorth.add(txtName);  
 panelNorth.add(labelSex);panelNorth.add(txtSex);  
 panelNorth.add(labelSdeNo);panelNorth.add(txtSdeNo);  
 panelNorth.add(labelSdeName);panelNorth.add(txtSdeName);  
 panelNorth.add(labelPosition);panelNorth.add(txtPosition);  
 panelCenter.add(btnEnquiry);panelCenter.add(btnScoreManage);  
 JScrollPane scrollPane = new JScrollPane(table);  
 frame1.add(scrollPane, BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 btnEnquiry.addActionListener(this);  
 btnScoreManage.addActionListener(this);  
 *// 添加监听器，监听表格的行选中事件* }

（3）点击表格事件：

table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {  
 public void valueChanged(ListSelectionEvent event) {  
 *// 获取当前选中行的索引* int row = table.getSelectedRow();  
 if (row >= 0) {  
 *// 获取选中行的各列的数据  
 //String[] columnNames = {"学期","允许选课系","系名","课程号","课程名","学分","可选课人数"};* String insertSemester = table.getValueAt(row, 0).toString();  
 String insertSdeNo = table.getValueAt(row, 1).toString();  
 String insertSdeName = table.getValueAt(row, 2).toString();  
 String insertCno = table.getValueAt(row, 3).toString();  
 String insertCname = table.getValueAt(row, 4).toString();  
 String insertCredit = table.getValueAt(row, 5).toString();  
 String insertStuTotal = table.getValueAt(row, 6).toString();  
 *sqlScoreManage* = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Teacher\_Score\_Manage\_view where Emp\_no = '" + loginID.trim() + "' and Semester = '" + insertSemester.trim() + "' and Cno = '" + insertCno.trim() + "'";  
 *sqlScore* = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Teacher\_Manage\_SC\_view where Emp\_name = '" + loginName.trim() + "' and Semester = '" + insertSemester.trim() + "' and Cno = '" + insertCno.trim() + "'";  
 }  
 }  
 });

（4）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == btnEnquiry) {  
 Enquiry();  
 }  
 if (e.getSource() == btnScoreManage){  
 *//如果没有选中行，则弹出提示框* if (table.getSelectedRow() == -1) {  
 JOptionPane.*showMessageDialog*(null, "请先选中课程再进行此操作！", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 return;  
 }  
  
 if(getInformationSC(TeacherRelated,*sqlScore*) == true){  
 JOptionPane.*showMessageDialog*(null, "该课程无人选择，无法进行赋分操作!", "提示", JOptionPane.*ERROR\_MESSAGE*);  
 return;  
 }  
 else{  
 WindowTeacherScoreManage windowTeacherScoreManage = new WindowTeacherScoreManage();  
 }  
 }  
}

（5）其他需要的函数：

public void Enquiry(){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Emp\_name = '" + loginName.trim() + "'";  
 System.*out*.println(sql);  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 int columnCount = 0;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 int i=0;  
 while (rs.next()) {  
 data[i][0] = rs.getString("Semester");  
 data[i][1] = rs.getString("Course\_Offered\_Sde\_no");  
 data[i][2] = rs.getString("Sde\_name");  
 data[i][3] = rs.getString("Cno");  
 data[i][4] = rs.getString("Cname");  
 data[i][5] = rs.getString("Credit");  
 data[i][6] = rs.getString("Stu\_total");  
 i++;  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
}  
public int getColumns () {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery("use Academic\_Affairs\_Management\_System\_20211576 select \* from Manager\_Course\_Teacher\_manage\_view where Emp\_name = '" + loginName.trim() + "'");  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}  
public void getInformation(Object[] TeacherRelated){  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = "use Academic\_Affairs\_Management\_System\_20211576 select \* from Teacher\_view where Emp\_no = '" + loginID.trim() + "'";  
 System.*out*.println(sql);  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 TeacherRelated[0] = rs.getString("Emp\_no");  
 TeacherRelated[1] = rs.getString("Emp\_name");  
 TeacherRelated[2] = rs.getString("Sex");  
 TeacherRelated[3] = rs.getString("Sde\_no");  
 TeacherRelated[4] = rs.getString("Sde\_name");  
 TeacherRelated[5] = rs.getString("Position");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}  
public boolean getInformationSC(Object[] TeacherRelated, String sql){  
 String Driver4 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* sql = *sqlScore*;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver4);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 TeacherRelated[0] = rs.getString("Semester");  
 TeacherRelated[1] = rs.getString("Cno");  
 TeacherRelated[2] = rs.getString("Cname");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
 int flag = 0;  
 for(int i=0;i<3;i++){  
 if(TeacherRelated[i].equals("")){  
 flag++;  
 }  
 }  
 if(flag==3){  
 return true;  
 }  
 else{  
 return false;  
 }  
}

2、给有学生选修的课程进行赋分：关键技术：表单验证、数据库操作。

（1）声明各类控件：

String loginID = windowsRegister.*loginID*;  
String loginName = windowsRegister.*loginName*;  
JPanel panelNorth = new JPanel();  
JPanel panelCenter = new JPanel();  
JPanel panelSouth = new JPanel();  
JFrame frame1 = new JFrame();  
JLabel labelSemester = new JLabel("学期");  
JLabel txtLabelSemester;  
JLabel labelCno = new JLabel("课程号");  
JLabel txtLabelCno;  
JLabel labelCname = new JLabel("课程名");  
JLabel txtLabelCname;  
JLabel labelEmpname = new JLabel("教师姓名");  
JLabel txtLabelEmpname = new JLabel(loginName);  
JLabel labelScore = new JLabel("分数");  
JTextField txtScore = new JTextField(12);  
JButton btnInsertScore = new JButton("成绩录入");  
JTable table = new JTable();  
String[] columnNames = {"学期","课程号","课程名","学号","分数"};  
Object[][] data = new Object[getColumns()][5];  
Object[] TeacherRelated = new Object[3];

（2）创建面板：

public WindowTeacherScoreManage() {  
 System.*out*.println("100");  
 getInformation(TeacherRelated);  
 txtLabelSemester = new JLabel((String)TeacherRelated[0]);  
 txtLabelCno = new JLabel((String)TeacherRelated[1]);  
 txtLabelCname = new JLabel((String)TeacherRelated[2]);  
 System.*out*.println("1");  
 frame1.setTitle("教务管理系统-教师界面-"+txtLabelCname.getText()+" 课程成绩管理界面");  
 frame1.setVisible(true);  
 frame1.setSize(800,600);  
 frame1.setLocation(300, 10);  
 frame1.setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 frame1.add(panelNorth, BorderLayout.*NORTH*);  
 frame1.add(panelCenter, BorderLayout.*CENTER*);  
 frame1.add(panelSouth, BorderLayout.*SOUTH*);  
 System.*out*.println("11");  
 panelNorth.setLayout(new GridLayout(5,2));  
 panelNorth.add(labelSemester);panelNorth.add(txtLabelSemester);  
 panelNorth.add(labelCno);panelNorth.add(txtLabelCno);  
 panelNorth.add(labelCname);panelNorth.add(txtLabelCname);  
 panelNorth.add(labelEmpname);panelNorth.add(txtLabelEmpname);  
 panelNorth.add(labelScore);panelNorth.add(txtScore);  
 panelCenter.add(btnInsertScore, BorderLayout.*CENTER*);  
 JScrollPane scrollPane = new JScrollPane(table);  
 frame1.add(scrollPane,BorderLayout.*SOUTH*);  
 scrollPane.setViewportView(table);  
 if(getColumns()==0){  
 frame1.dispose();  
 JOptionPane.*showMessageDialog*(null, "该课程暂无学生选修!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 else{  
 getTable();  
 }  
 btnInsertScore.addActionListener(this);  
}

（3）按钮事件：

@Override  
public void actionPerformed(ActionEvent e) {  
 if(e.getSource() == btnInsertScore){  
 *//如果没有选中行，弹窗提示* if(table.getSelectedRow() == -1){  
 JOptionPane.*showMessageDialog*(null, "请选中学生后再进行赋分!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 else{  
 *//如果选中行，若有成绩则弹窗无法修改，若无成绩则直接录入* if(data[table.getSelectedRow()][4] != null){  
 JOptionPane.*showMessageDialog*(null, "该学生已有成绩，无法修改!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 else{  
 String Sno = (String) data[table.getSelectedRow()][3];  
 String Score = txtScore.getText();  
 String sql = "update SC\_20211576 set Score = '"+txtScore+"' where Sno = '"+Sno+"' and Cno = '"+txtLabelCno.getText()+"' and Semester = '"+txtLabelSemester.getText()+"'";  
 GiveGrade(sql);  
 JOptionPane.*showMessageDialog*(null, "录入成功!", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 getTable();  
 }  
 }  
 }  
}

（4）其他需要的函数：

public void GiveGrade(String sql){  
 String Driver3 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* System.*out*.println(sql);  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver3);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 stmt.executeUpdate(sql);  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}  
public void getInformation(Object[] TeacherRelated){  
 String Driver4 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = WindowTeacher.*sqlScoreManage*;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver4);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 while (rs.next()) {  
 TeacherRelated[0] = rs.getString("Semester");  
 TeacherRelated[1] = rs.getString("Cno");  
 TeacherRelated[2] = rs.getString("Cname");  
 }  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}  
public void getTable(){  
 String Driver5 = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = WindowTeacher.*sqlScore*;  
 System.*out*.println("getTable.sql:"+sql);  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver5);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException ea) {  
 throw new RuntimeException(ea);  
 } catch (SQLException eb) {  
 throw new RuntimeException(eb);  
 }  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 int i = 0;  
 while (rs.next()) {  
 *//String[] columnNames = {"学期","课程号","课程名","学号","分数"};* data[i][0] = rs.getString("Semester");  
 data[i][1] = rs.getString("Cno");  
 data[i][2] = rs.getString("Cname");  
 data[i][3] = rs.getString("Sno");  
 data[i][4] = rs.getString("Grade");  
 i++;  
 }  
 DefaultTableModel model = new DefaultTableModel(data, columnNames);  
 table.setModel(model);  
 } catch (SQLException ex) {  
 throw new RuntimeException(ex);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException exc) {  
 throw new RuntimeException(exc);  
 }  
 }  
}  
public int getColumns () {  
 String Driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 String url = "jdbc:sqlserver://localhost:1433;DatabaseName=Academic\_Affairs\_Management\_System\_20211576;encrypt=false";  
 String userName = "s20211576"; *// 默认用户名* String userPwd = "s20211576"; *// 密码* String sql = WindowTeacher.*sqlScoreManage*;  
 Connection dbConn = null;  
 try {  
 Class.*forName*(Driver);  
 dbConn = DriverManager.*getConnection*(url, userName, userPwd);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 int columnCount;  
 try {  
 Statement stmt = dbConn.createStatement();  
 ResultSet rs = stmt.executeQuery(sql);  
 ResultSetMetaData rsmd = rs.getMetaData();  
 columnCount = 0;  
 while (rs.next()) {  
 *//统计表中共有多少行数据* columnCount++;  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } finally {  
 try {  
 dbConn.close();  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 return columnCount;  
}

七、总结

本系统的设计基于SQL Server和Java swing，主要难点在给对SQL server表中数据的查询及修改，此系统基本可以实现预设的各种功能，供学校师生共同使用。

在教务系统的权限管理设计过程中，由于时间关系，有的模块中功能不是很完善，但总体达到了预期效果，该系统可正常运行，此系统由开始的需求分析到总体设计再到最后的各个功能的实现，每个环节都需要认真仔细地设计编写，系统界面简洁、便于操作，很容易使用户熟悉操作提高工作效率，但由于缺乏开发经验，系统中必定会存在着未发现的错误，希望在之后的学习中能够发现并解决这些看不到的隐患，不断地优化完善做出更好的系统。

八、后期待修复的问题（未在上述报告中体现）

存在的问题：1、可以多重组合条件进行查询。

2、查询汇总出某学生某学期所选修的全部课程的学分、学时以及成绩、已经获得的学分统计

3、查询汇总出某班某学期所有学生按学号排序、按平均绩点排序的学号、姓名以及平均绩点。

4、期末时批量录入学生的成绩，录入结束后能自动计算获取的绩点；

5、设计一个触发器，当修改学生成绩时，该成绩对应的绩点自动改变

6、设计一个存储过程，按指定学期统计每位教师的授课数和学时数。

7、当教师批量录入成绩完成，可以自动触发触发器来计算学生获得的绩点。