《数据库原理及应用》

综合实验报告

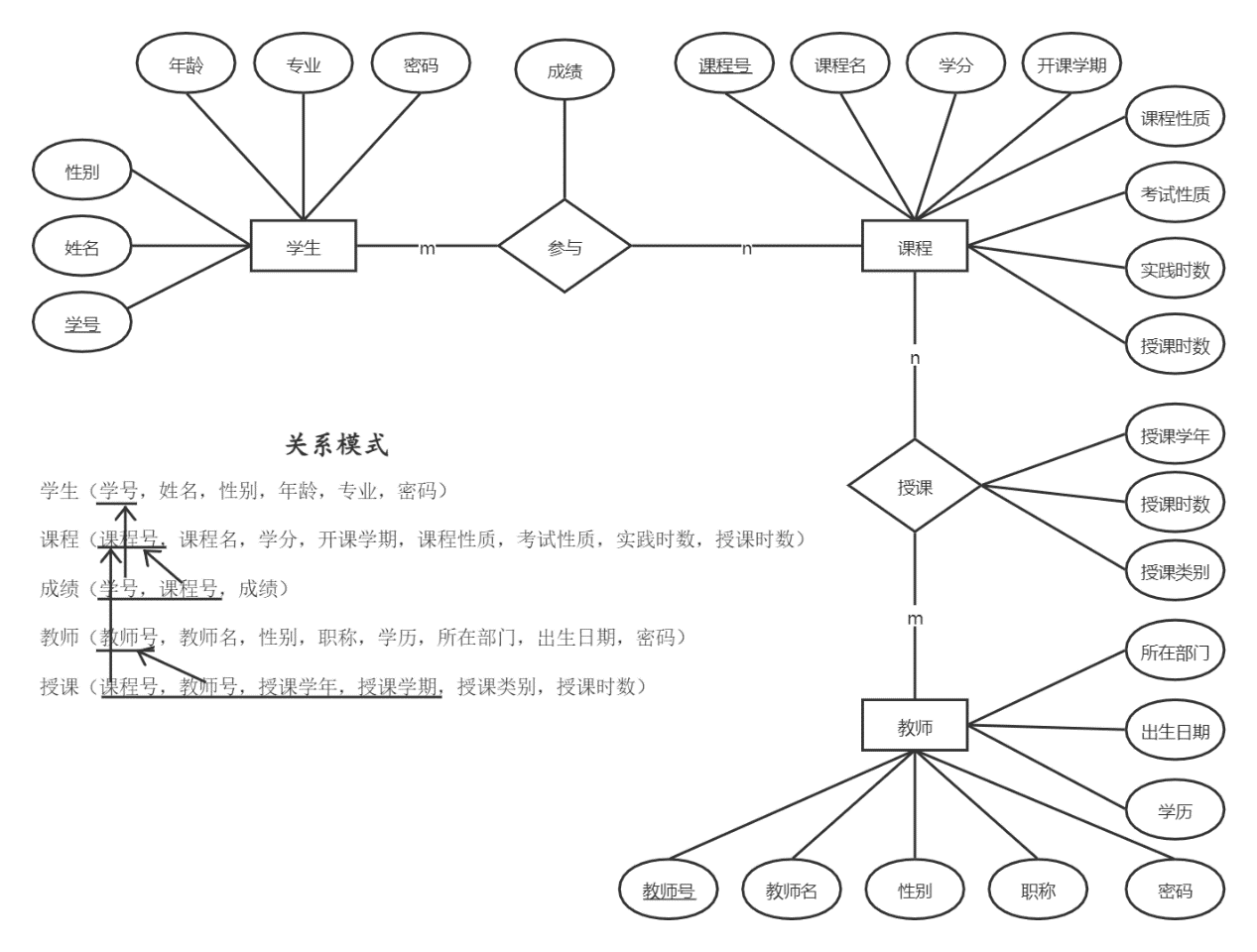


|  |  |  |
| --- | --- | --- |
| 题目 | 西柚学生信息管理系统 | |
| 专业年级 | 软件工程2020级 | |
| 姓名 | 齐嘉彬 |
| 学号 | 202031061303 |
| 指导教师 | 岳静 | |

1. 系统简介

学生信息管理系统是高校管理工作中一项非常重要的组成部分。用户的需求具体体现在各种信息的提供、保存、更新和查询方面，这就要求数据库结构能充分满足各种信息的输出和输入。收集基本数据、数据结构以及数据处理的流程，为后面的设计打下基础。该系统的设计开发主要为老师对学生进行统一管理提供信息化的方法。因此学生信息管理系统的用户包括教师和学生。主要涉及学生信息、课程信息以及选课情况和成绩等数据信息。

1. 数据库设计
2. **ER图**

****

1. **数据表设计**

（1）建立西柚学生信息系统数据库

CREATE DATABASE [西柚学生信息管理系统]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'西柚学生信息管理系统', FILENAME = N'D:\SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\西柚学生信息管理系统.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'西柚学生信息管理系统\_log', FILENAME = N'D:\SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\西柚学生信息管理系统\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

（2）建立课程表，授课表以及成绩表，其具体结构如下：

**表1 学生表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 学号 | 字符串，长度为7 | 主键，非空 |
| 姓名 | 字符串，长度为10 | 非空 |
| 性别 | 字符串，长度为2 | 取值‘男’，‘女’ |
| 年龄 | tinyint | 取值大于0 |
| 专业 | 字符串，长度为20 |  |
| 密码 | 可变字符串，最长长度为20 | 非空 |

**表2 课程表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 课程号 | 字符串，长度为8 | 主键 |
| 课程名 | 可变字符串，最长长度为30 | 非空，取值唯一 |
| 学分 | 整型 | 取值1-4 |
| 开课学期 | 整型 | 取值1-8 |
| 课程性质 | 字符串，长度为4 | 取值‘必修’，‘选修’，默认值为‘必修’ |
| 考试性质 | 字符串，长度为4 | 取值‘考试’，‘考查’，默认值为‘考试’ |
| 授课时数 | 整型 | 取值小于等于64 |
| 实践时数 | 整型 |  |

**表3 成绩表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 学号 | 字符串，长度为7 | 主键，非空，参照学生表中的学号 |
| 课程号 | 字符串，长度为8 | 主键，非空，参照课程表中的课程号 |
| 成绩 | 微整型 | 取值0-100 |

**表4 教师表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 教师号 | 字符串，长度为10 | 主键 |
| 教师名 | 字符串，长度为8 | 非空 |
| 性别 | 字符串，长度为2 | 取值‘男’，‘女’ |
| 职称 | 字符串，长度为6 | 取值‘助教’，‘讲师’，‘副教授’，‘教授’ |
| 学历 | 字符串，长度为6 | 取值‘本科’、‘硕士’、‘博士’、‘博士后’ |
| 出生日期 | smalldatatime |  |
| 所在部门 | 字符串，长度为20 |  |
| 密码 | 可变字符串，最长长度为20 | 非空 |

**表5 授课表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 课程号 | 字符串，长度为8 | 主键，非空，参照课程表中的课程号 |
| 教师号 | 字符串，长度为10 | 主键，非空，参照教师表中的教师号 |
| 授课学年 | 字符串，长度为4 | 主键 |
| 授课学期 | 整型 | 主键 |
| 授课类别 | 字符串，长度为6 | ‘主讲’，‘辅导’，‘带实验’ |
| 授课时数 | 整型 | 取值小于等于64 |

**表5 username表结构**

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| name | 字符串，长度为10 | 主键，非空 |

**SQL语句：**(考虑到排版，sql语句顺序与建表不同)

CREATE TABLE [dbo].[授课表](

[课程号] [char](8) NOT NULL,

[教师号] [char](10) NOT NULL,

[授课学年] [char](4) NOT NULL,

[授课学期] [int] NOT NULL,

[授课类别] [char](6) NULL,

[授课时数] [int] NULL,

PRIMARY KEY ([课程号],[教师号],[授课学年],[授课学期])

)

GO

ALTER TABLE [dbo].[授课表] ADD DEFAULT ('32') FOR [授课时数]

GO

ALTER TABLE [dbo].[授课表] WITH CHECK ADD CONSTRAINT [FK\_\_授课表\_\_教师号] FOREIGN KEY([教师号]) REFERENCES [dbo].[教师表] ([教师号])

GO

ALTER TABLE [dbo].[授课表] CHECK CONSTRAINT [FK\_\_授课表\_\_教师号]

GO

ALTER TABLE [dbo].[授课表] WITH CHECK ADD CONSTRAINT [FK\_\_授课表\_\_课程号] FOREIGN KEY([课程号]) REFERENCES [dbo].[课程表] ([课程号])

GO

ALTER TABLE [dbo].[授课表] CHECK CONSTRAINT [FK\_\_授课表\_\_课程号]

GO

ALTER TABLE [dbo].[授课表] WITH CHECK ADD CONSTRAINT [CK\_\_授课表\_\_授课类别] CHECK (([授课类别]='带实验' OR [授课类别]='辅导' OR [授课类别]='主讲'))

GO

ALTER TABLE [dbo].[授课表] CHECK CONSTRAINT [CK\_\_授课表\_\_授课类别]

GO

ALTER TABLE [dbo].[授课表] WITH CHECK ADD CONSTRAINT [CK\_\_授课表\_\_授课时数] CHECK (([授课时数]<=(64)))

GO

ALTER TABLE [dbo].[授课表] CHECK CONSTRAINT [CK\_\_授课表\_\_授课时数]

GO

CREATE TABLE [dbo].[课程表](

[课程号] [char](8) NOT NULL,

[课程名] [varchar](30) NOT NULL,

[学分] [int] NULL,

[开课学期] [int] NULL,

[课程性质] [char](4) NULL,

[考试性质] [char](4) NULL,

[授课时数] [int] NULL,

[实践时数] [int] NULL,

PRIMARY KEY([课程号])

)

GO

ALTER TABLE [dbo].[课程表] ADD CONSTRAINT [DF\_课程表\_课程性质] DEFAULT ('必修') FOR [课程性质]

GO

ALTER TABLE [dbo].[课程表] ADD CONSTRAINT [DF\_课程表\_考试性质] DEFAULT ('考试') FOR [考试性质]

GO

ALTER TABLE [dbo].[课程表] WITH CHECK ADD CHECK (([开课学期]>=(1) AND [开课学期]<=(12)))

GO

ALTER TABLE [dbo].[课程表] WITH CHECK ADD CHECK (([考试性质]='考查' OR [考试性质]='考试'))

GO

ALTER TABLE [dbo].[课程表] WITH CHECK ADD CHECK (([课程性质]='选修' OR [课程性质]='必修'))

GO

ALTER TABLE [dbo].[课程表] WITH CHECK ADD CHECK (([授课时数]<=(68)))

GO

ALTER TABLE [dbo].[课程表] WITH CHECK ADD CHECK (([学分]>=(1) AND [学分]<=(8)))

GO

CREATE TABLE [dbo].[学生表](

[学号] [char](7) NOT NULL,

[姓名] [char](10) NOT NULL,

[性别] [char](2) NULL,

[年龄] [tinyint] NULL,

[专业] [char](20) NULL,

[密码] [varchar](20) NOT NULL,

PRIMARY KEY([学号])

)

GO

ALTER TABLE [dbo].[学生表] WITH CHECK ADD CHECK (([性别]='女' OR [性别]='男'))

GO

CREATE TABLE [dbo].[教师表](

[教师号] [char](10) NOT NULL,

[教师名] [char](8) NOT NULL,

[性别] [char](2) NULL,

[职称] [char](6) NULL,

[学历] [char](6) NULL,

[出生日期] [smalldatetime] NULL,

[所在部门] [char](20) NULL,

[密码] [varchar](20) NOT NULL,

PRIMARY KEY([教师号])

)

GO

ALTER TABLE [dbo].[教师表] WITH CHECK ADD CONSTRAINT [CK\_\_教师表\_\_性别] CHECK (([性别]='女' OR [性别]='男'))

GO

ALTER TABLE [dbo].[教师表] CHECK CONSTRAINT [CK\_\_教师表\_\_性别]

GO

ALTER TABLE [dbo].[教师表] WITH CHECK ADD CONSTRAINT [CK\_\_教师表\_\_学历] CHECK (([学历]='博士后' OR [学历]='博士' OR [学历]='硕士' OR [学历]='本科'))

GO

ALTER TABLE [dbo].[教师表] CHECK CONSTRAINT [CK\_\_教师表\_\_学历]

GO

ALTER TABLE [dbo].[教师表] WITH CHECK ADD CONSTRAINT [CK\_\_教师表\_\_职称] CHECK (([职称]='教授' OR [职称]='副教授' OR [职称]='讲师' OR [职称]='助教'))

GO

ALTER TABLE [dbo].[教师表] CHECK CONSTRAINT [CK\_\_教师表\_\_职称]

GO

CREATE TABLE [dbo].[成绩表](

[学号] [char](7) NOT NULL,

[课程号] [char](8) NOT NULL,

[成绩] [smallint] NULL,

PRIMARY KEY ([学号],[课程号])

)

GO

ALTER TABLE [dbo].[成绩表] WITH CHECK ADD CONSTRAINT [FK\_成绩表\_课程表] FOREIGN KEY([课程号]) REFERENCES [dbo].[课程表] ([课程号])

GO

ALTER TABLE [dbo].[成绩表] CHECK CONSTRAINT [FK\_成绩表\_课程表]

GO

ALTER TABLE [dbo].[成绩表] WITH CHECK ADD CONSTRAINT [FK\_成绩表\_学生表] FOREIGN KEY([学号]) REFERENCES [dbo].[学生表] ([学号])

GO

ALTER TABLE [dbo].[成绩表] CHECK CONSTRAINT [FK\_成绩表\_学生表]

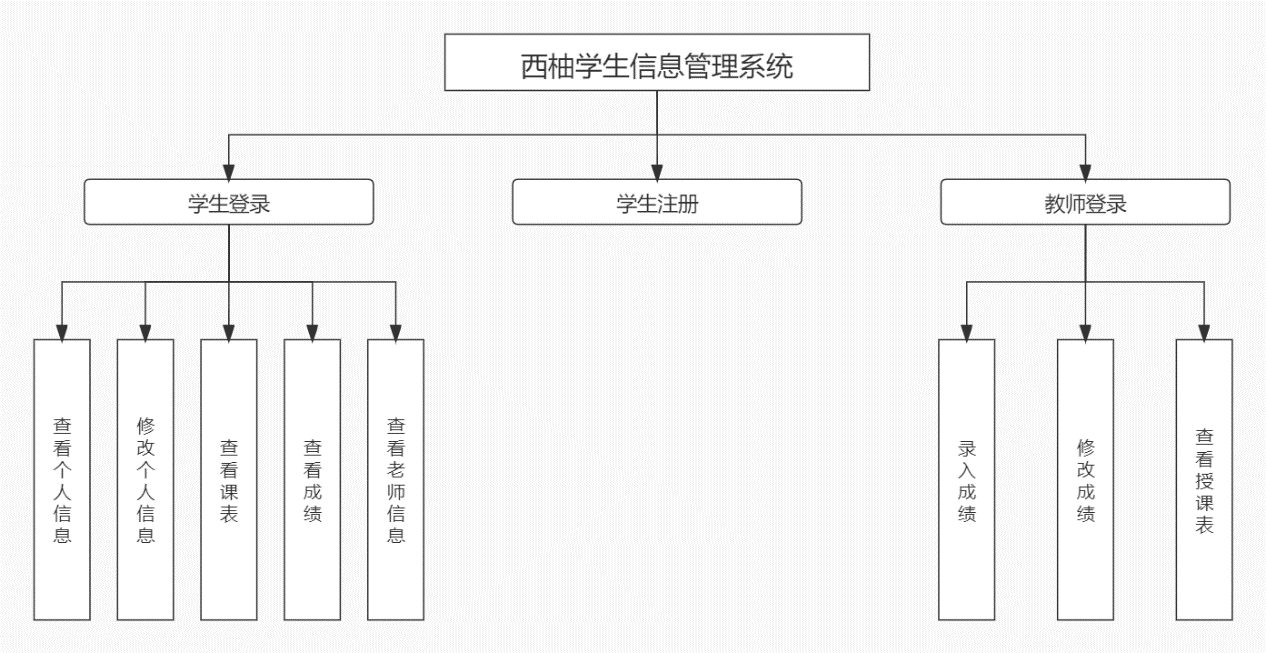
GO

ALTER TABLE [dbo].[成绩表] WITH CHECK ADD CHECK (([成绩]>=(0) AND [成绩]<=(100)))

GO

1. 系统实现

功能结构图

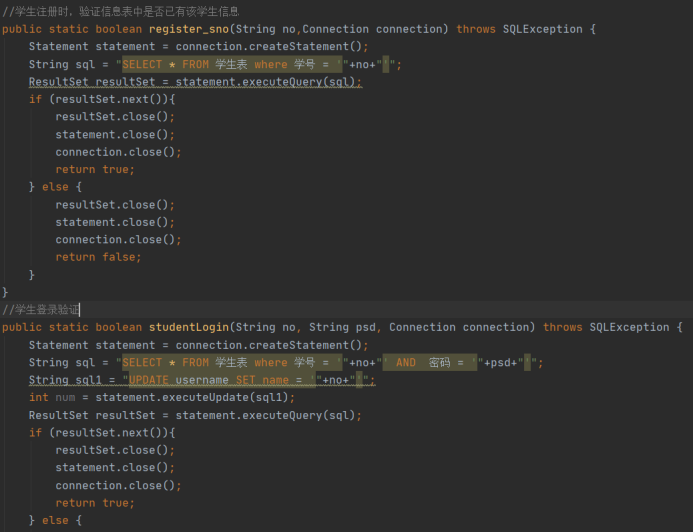


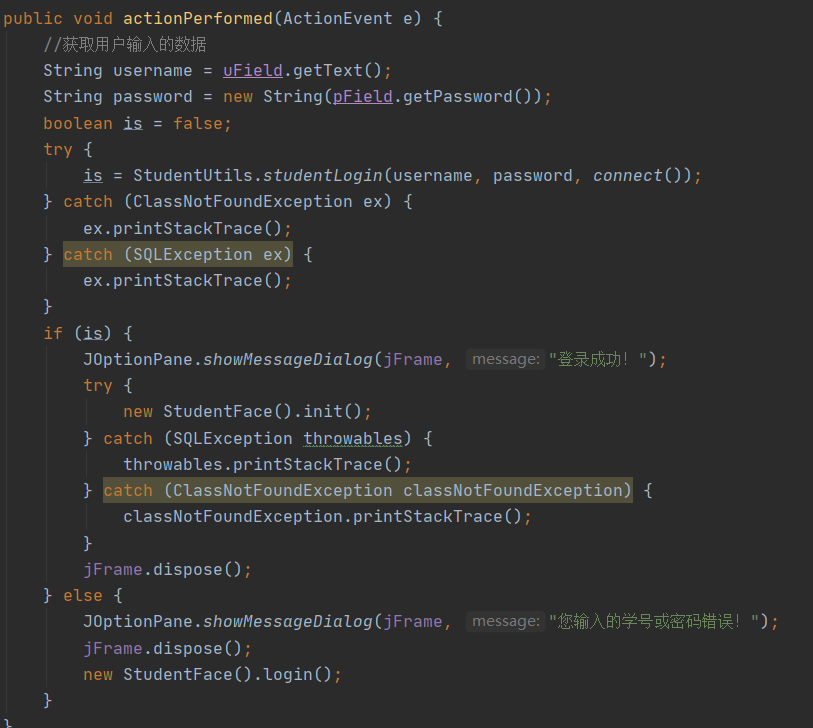
1. 学生登录（单表查询）

实现界面：



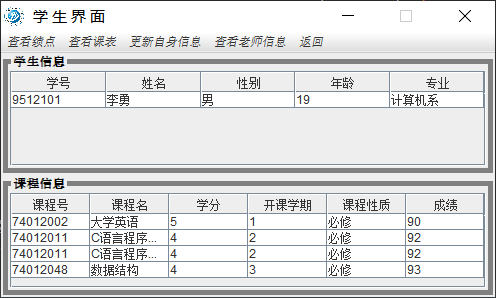
代码：



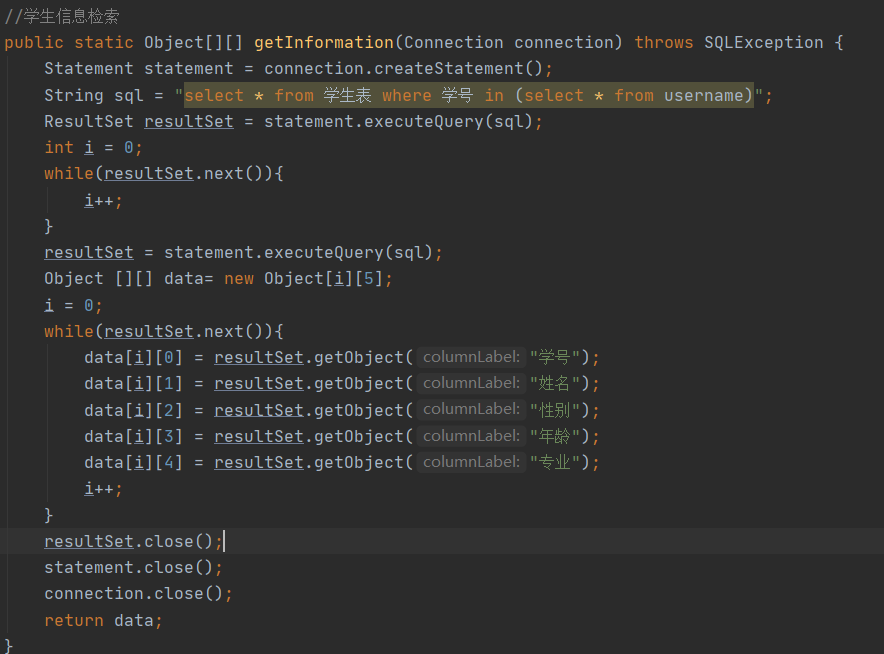


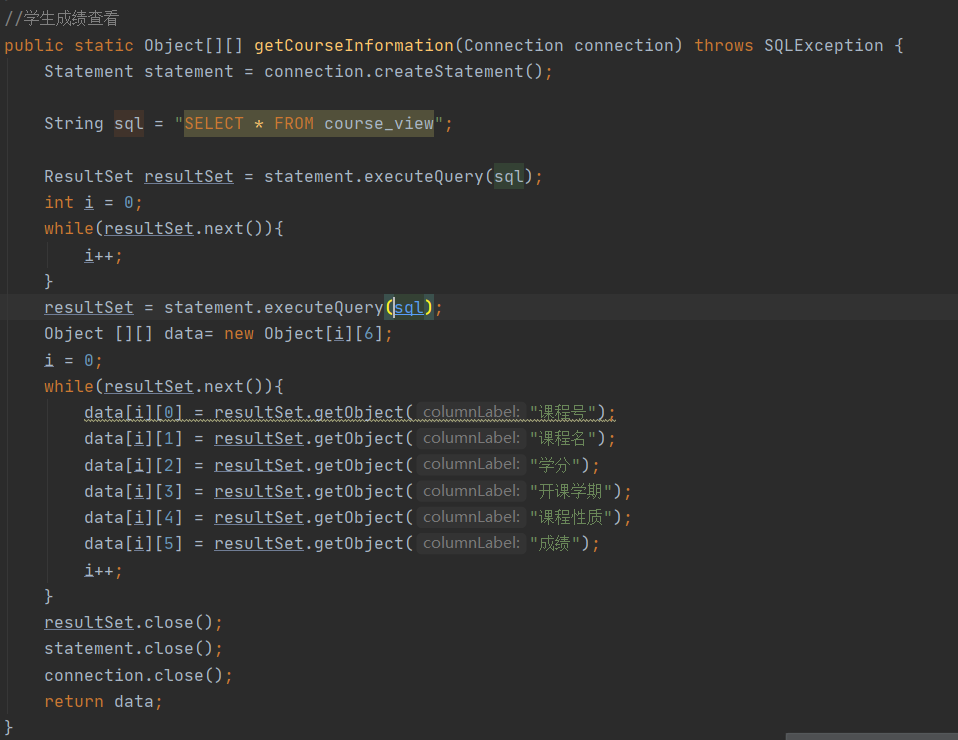
1. 查看个人信息及查看成绩（联表查询及视图）

实现界面：



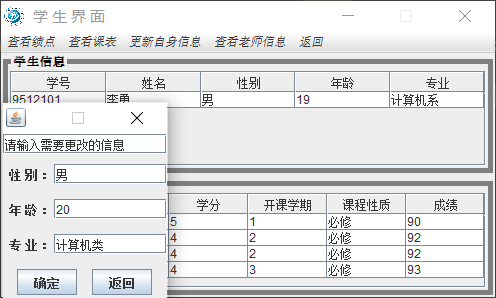
代码：



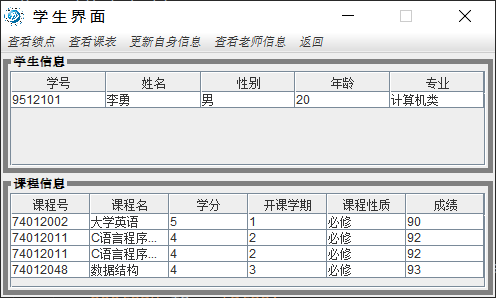


1. 修改个人信息

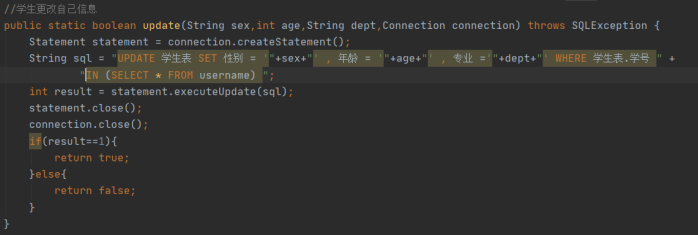
修改前：



修改后：

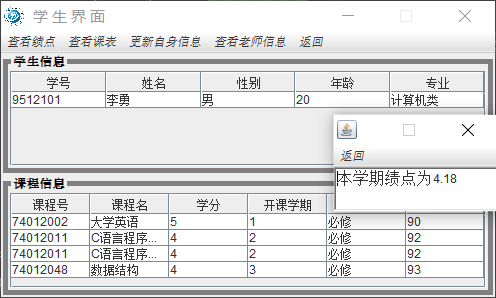


代码：

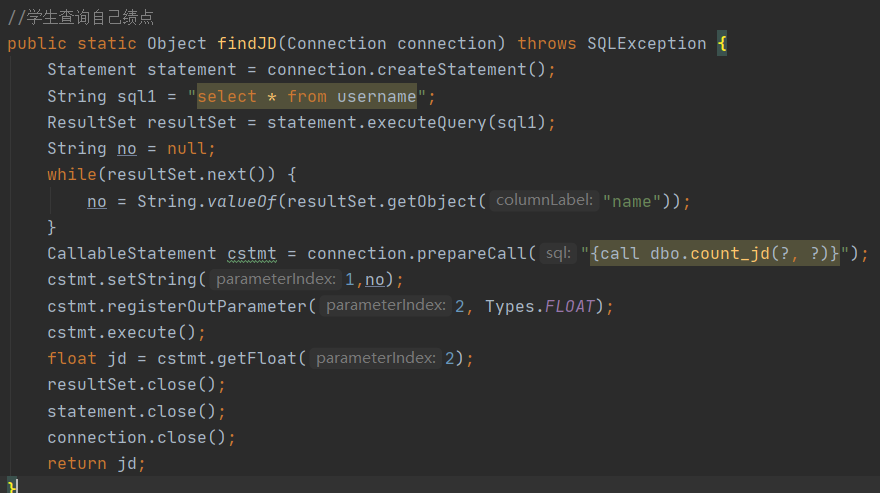


1. 查看绩点（调用存储过程）

实现界面：



代码：

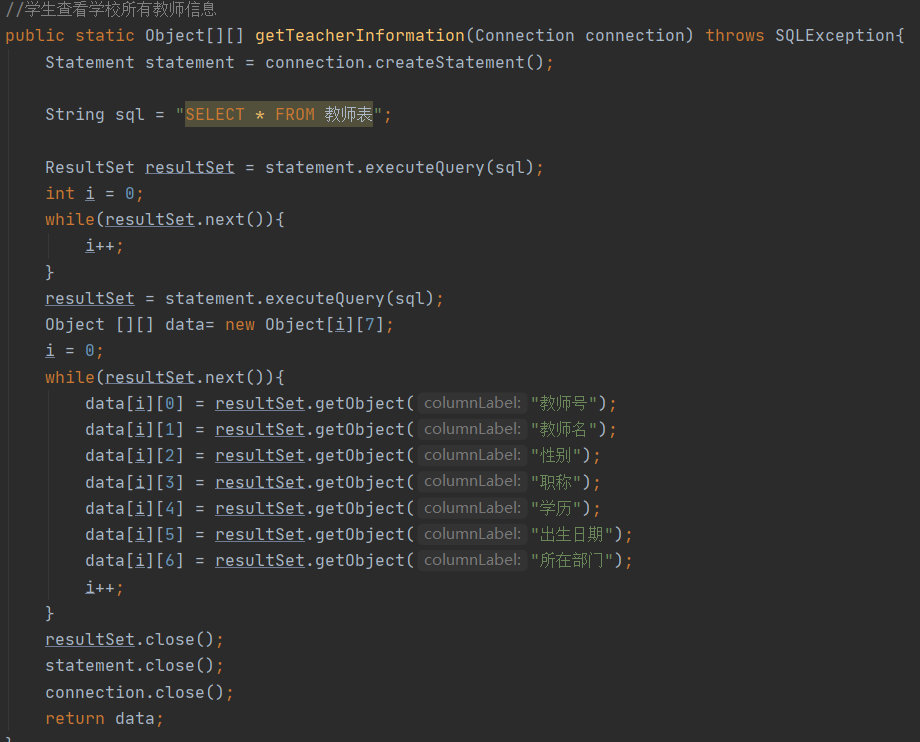


1. 查看老师信息（单表查询）

实现界面：

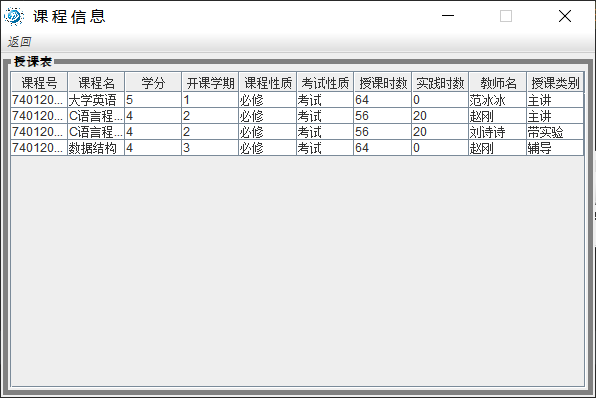


代码：

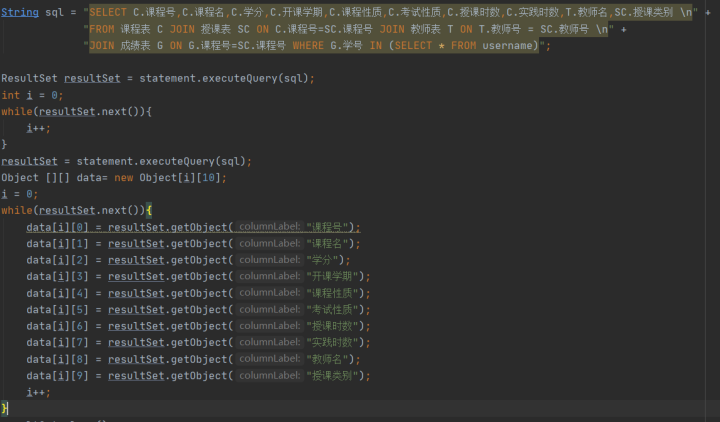


1. 查看课表

实现界面：



代码：

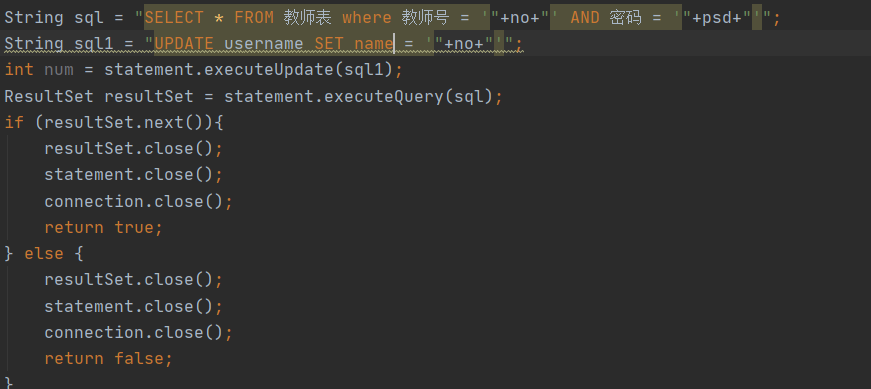


1. 教师登录

实现界面：

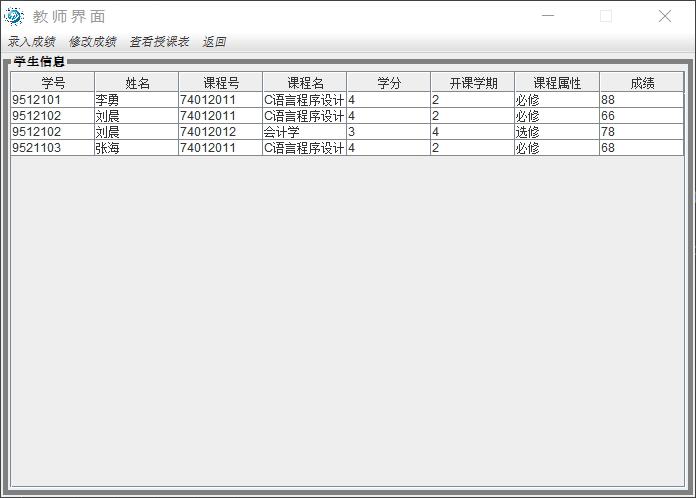


代码：



1. 查看选修自己课的学生的成绩

实现界面：

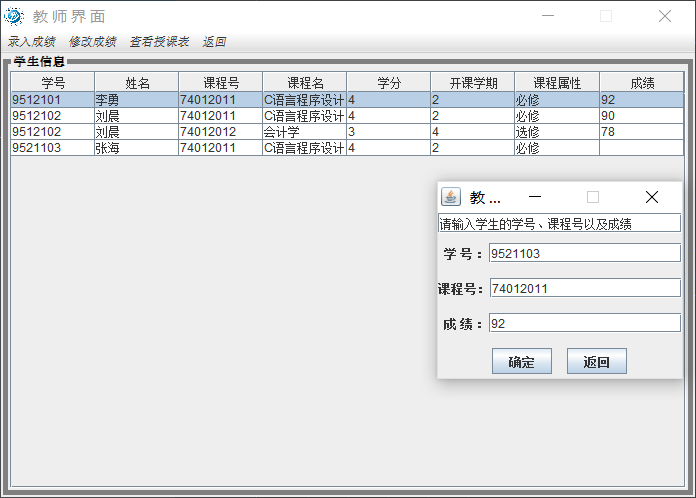


代码：

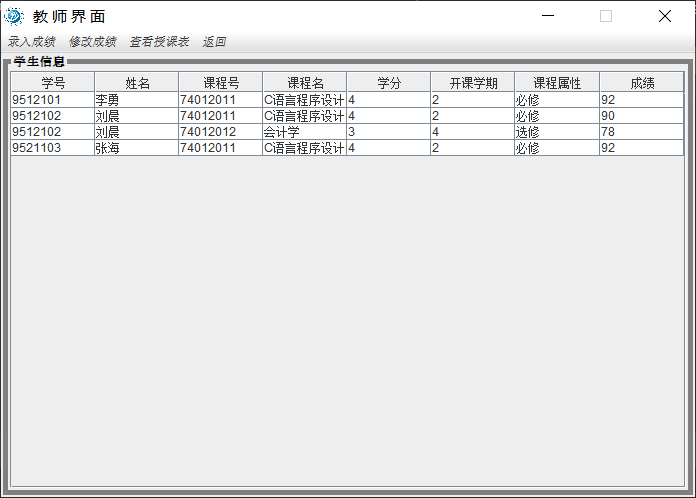


1. 录入成绩

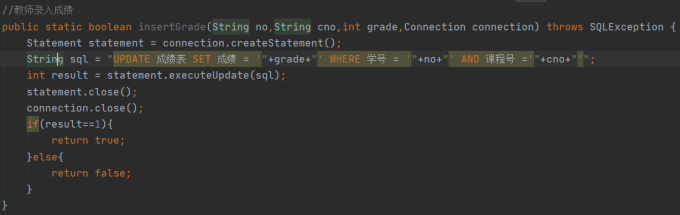
录入前：



录入后：

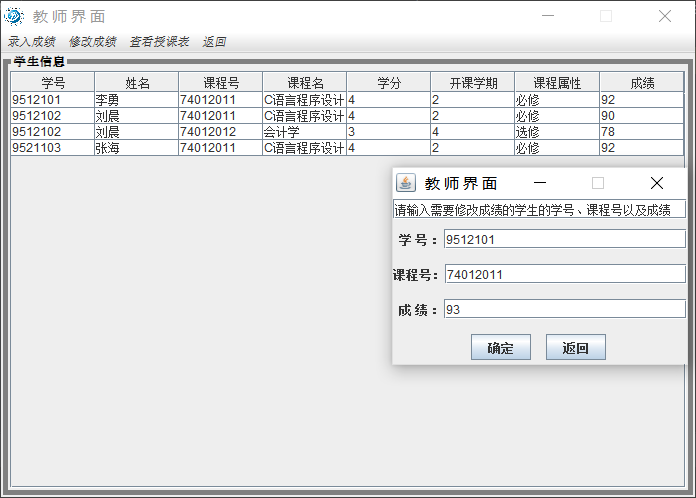


代码：

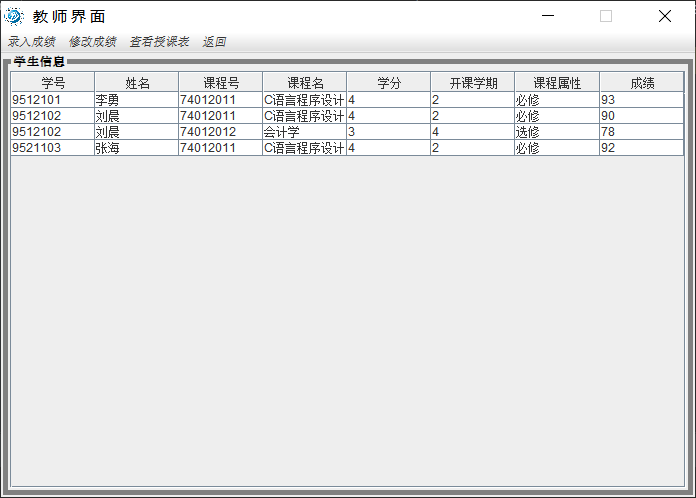


1. 修改成绩

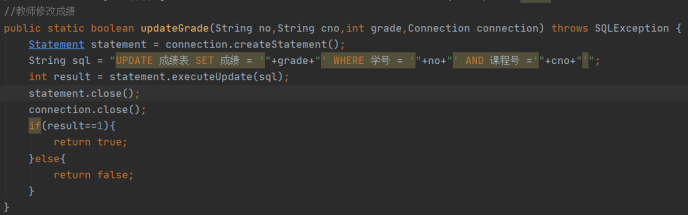
修改前：



修改后：

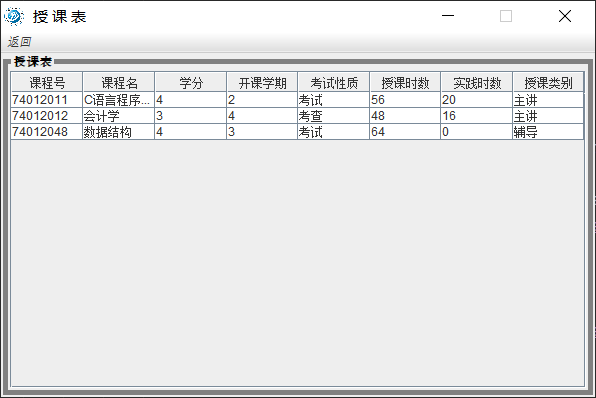


代码：

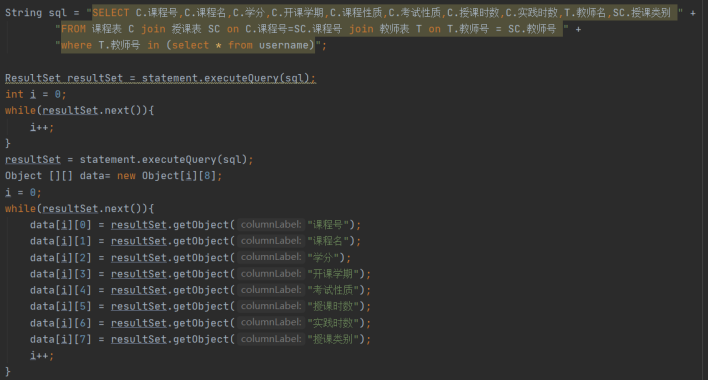


1. 查看授课表

实现界面

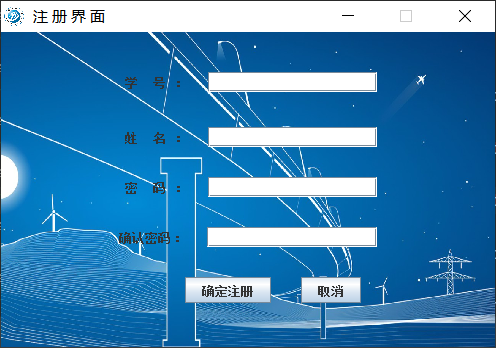


代码：

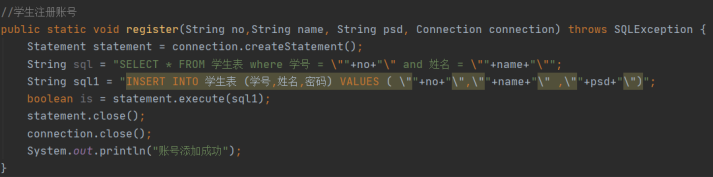


1. 学生注册

实现界面：



代码：



在实现这些功能的同时，加深了自己对sql中，增删改查四个基本功能的操作以及视图、存储过程的创建与运用。在计算绩点时，通过调用计算绩点的存储过程来实现，同时我也知道了应该在java调用存储过程的方法与sql中不同。

1. 总结

通过对西柚学生信息管理系统的开发，使我对系统开发有了很大的了解，需求分析和系统设计的质量往往是决定系统成功与否的决定性因素之一。从数据库的实现、系统功能的实现、系统关键技术的实现这三个角度对系统的部分实现进行分析。在数据库应用系统开发中，合理制定了存储过程，更加使系统中的查询灵活。

但是，由于自己在这方面知识较为不足，致使系统在功能方面还不完善，例如实现了用户注册功能，但是在新注册的用户登录之后并不能自行选课而导致无法看到相关信息等等，而且由于知识面的局限性，对于具体的系统方面的知识还有很大的欠缺，可能在设计数据库存在着不合理的地方，以至于影响到整个系统的功能以及性能。

总之，这次西柚学生信息管理系统的开发让我综合练习了SQL Server的使用和数据库的管理技术，综合熟悉并练习了SQL语言在编程中的使用，使我学到了很多书本上没有的知识，让我了解到只有将理论和实际相结合才能将学到的知识消化，同时也让我认识到了我自身的不足。系统设计留给我的经验对我以后的工作学习都是很有帮助的，而它给我留下的教训也足以让我在今后的学习中引以为戒。