create table Student

(

Sno char(9) primary key,

Sname char(20) unique,

Ssex char (2),

Sage smallint,

Sdept char(20)

);

create table Course

(Cno char(4) primary key,

Cname char(40) not NULL,

Cpno char(4),

Ccredit smallint,

foreign key(Cpno) references Course(Cno)

);

create table SC

(Sno char(9),

Cno char(4),

Grade smallint,

primary key (Sno ,Cno),

foreign key(Sno)references Student (Sno),

foreign key(Cno)references Course(Cno)

);

select \* from Student

insert into Student

values('201215121','李勇','男','20','CS'),('201215122','刘晨','女','19','CS'),('201215123','王敏','女','18','MA'),('201215125','张立','男','19','IS');

insert into Course

values('1','数据库','5','4');--无先行课错误

insert into Course

values('2','数学',NULL,'2'),('6','数据处理',NULL,'2');

insert into Course

values('4','操作系统','6','3'),('7','PASCAL语言','6','4');

insert into Course

values('5','数据结构','7','4');

insert into Course

values('1','数据库','5','4');

insert into Course

values('3','信息系统','1','4');

insert into SC

values('201215121','1','92'),('201215121','2','85'),('201215121','3','88'),('201215122','2','90'),('201215122','3','80');

insert into Student

values('201215126','张伟','男','20','CS'),('201215127','王伟','男','20','CS'),('201215128','王芳','男','20','CS'),('201215129','王勇','男','20','CS'),

('201215131','李伟','男','20','CS'),('201215132','张敏','男','20','CS'),('201215133','张静','男','20','CS'),('201215134','张勇','男','20','CS'),

('201215135','王秀英','男','19','MA'),('201215136','李静','男','19','MA'),('201215137','李敏','男','19','MA'),('201215138','王艳','男','19','MA'),

('201215139','李秀英','男','19','MA'),('201215140','张丽','男','19','MA'),('201215142','李杰','男','19','MA'),

('201215143','李娜','女','18','IS'),('201215144','王静','女','18','IS'),('201215145','王磊','女','18','IS'),('201215146','王强','女','18','IS'),

('201215147','张秀英','女','18','IS'),('201215148','王丽','女','18','IS'),('201215149','李军','女','18','IS'),('201215150','王军','女','18','IS'),

('201215151','刘伟','女','18','IS'),('201215152','李强','女','18','IS'),('201215153','刘洋','女','18','IS'),('201215154','张杰','女','18','IS');

insert into Course

values('8','思想道德与法律基础',NULL,'3'),('9','中国近代史纲要',NULL,'3'),('10','马克思主义基本原理',NULL,'3'),('11','毛泽东思想与中国特色社会主义理论概述',NULL,'5'),

('12','形式与政策',NULL,'2'),('13','军事理论',NULL,'3'),('14','大学体育',NULL,'2'),('15','大学英语',NULL,'3'),

('16','大学语文',NULL,'4'),('17','大学生职业规划',NULL,'2'),('18','创业基础',NULL,'2'),('19','艺术导论',NULL,'2'),

('20','高等数学',NULL,'5'),('21','计算机科学导论',NULL,'3'),('22','普通物理',NULL,'3'),('23','普通物理实验',NULL,'1');

insert into Course values

('24','线性代数','20','4'),('25','离散数学','20','4'),('26','概率统计','20','4'),('27','数字电路与逻辑设计','22','3'),

('28','数字电路与逻辑设计实验','23','1'),('29','算法设计与分析','5','4'),('30','算法设计与分析实验','5','1');

insert into Course values('31','运筹学','24','3'),('32','通信原理',NULL,'3'),('33','',NULL,''),('34','计算机英语','15','3');

insert into Course values('35','数学模型','31','3');

insert into SC

values('201215126','1','92'),('201215126','8','93'),('201215126','9','96'),('201215126','10','62'),('201215126','11','77'),

('201215127','12','93'),('201215127','13','96'),('201215127','14','91'),('201215128','15','62'),('201215128','16','78');

insert into SC

values('201215128','1','92'),('201215128','2','93'),('201215128','3','96'),('201215128','4','62'),('201215128','5','77');

('201215129','1','92'),('201215129','2','94'),('201215129','3','91'),('201215129','4','92'),('201215129','5','57'),

('201215131','1','92'),('201215131','2','90'),('201215131','3','76'),('201215131','4','24'),('201215131','5','45'),

('201215132','7','88'),('201215132','8','88'),('201215132','9','83'),('201215132','10','84'),('201215132','1','85');

('201215133','7','89'),('201215133','8','89'),('201215133','9','88'),('201215133','10','85'),('201215133','1','83'),

('201215134','7','87'),('201215134','8','82'),('201215134','9','82'),('201215134','10','87'),('201215134','1','83'),

('201215135','7','31'),('201215135','8','96'),('201215135','9','92'),('201215135','10','91'),('201215135','1','32'),

('201215136','13','98'),('201215136','14','96'),('201215136','15','99'),('201215136','16','98'),('201215136','17','99'),

('201215137','13','92'),('201215137','14','97'),('201215137','15','94'),('201215137','16','96'),('201215137','17','92'),

('201215138','13','86'),('201215138','14','96'),('201215138','15','97'),('201215138','16','98'),('201215138','17','95'),

('201215139','13','96'),('201215139','14','95'),('201215139','15','94'),('201215139','16','96'),('201215139','17','99'),

('201215140','13','86'),('201215140','14','98'),('201215140','15','94'),('201215140','16','94'),('201215140','17','99');

insert into SC

values

('201215142','18','86'),('201215142','19','98'),('201215142','20','94'),('201215142','21','94'),('201215142','22','99'),

('201215143','18','86'),('201215143','19','98'),('201215143','20','94'),('201215143','21','94'),('201215143','22','99'),

('201215144','18','86'),('201215144','19','98'),('201215144','20','94'),('201215144','21','94'),('201215144','22','99');

insert into SC

values

('201215145','20','86'),('201215145','24','98'),('201215145','26','94'),('201215145','22','94'),('201215145','27','99'),

('201215146','20','86'),('201215146','24','98'),('201215146','26','94'),('201215146','22','94'),('201215146','27','99'),

('201215147','20','86'),('201215147','24','98'),('201215147','26','94'),('201215147','22','94'),('201215147','27','99'),

('201215148','20','86'),('201215148','24','98'),('201215148','26','94'),('201215148','22','94'),('201215148','27','99');

insert into SC

values

('201215149','5','86'),('201215149','23','98'),('201215149','28','94'),('201215149','29','94'),('201215149','30','99'),

('201215150','5','98'),('201215150','23','97'),('201215150','28','96'),('201215150','29','95'),('201215150','30','94'),

('201215151','5','81'),('201215151','23','91'),('201215151','28','92'),('201215151','29','94'),('201215151','30','94'),

('201215152','5','85'),('201215152','23','97'),('201215152','28','96'),('201215152','29','92'),('201215152','30','99');

--修改基本表

alter table Student add S\_entrance date;

alter table Student alter column Sage int;

alter table Course add unique(Cname);

--Student表建立视图

--报错：必须是批处理中仅有的语句 单独执行就行了

create view IS\_Student

as

select Sno,Sname,Sage

from Student

Where Sdept = 'IS';

--建立索引 默认升序

create unique index Stusno on Student(Sno);

create unique index Coucno on Course(Cno);

create unique index SCno on SC(Sno ASC,Cno DESC);

--书上错误

alter index SCno rename to SCSno;

select Sname, Sno, Sdept

from Student

--选择全部

select \*

from Student

--select 也可以是表达式

select Sname, 2014-Sage birth

from Student

--也可以是函数和字符串常量

select Sname, 'Year of Birth', 2014-Sage, lower(Sdept)

from Student

--distinct 去掉重复行 默认是all 保留重复

select distinct Sno

from SC

select distinct Sno

from SC

where Grade < 60

--between 包括边界

select Sname, Sdept, Sage

from Student

where Sage between 19 and 23

--确定集合

select Sname,Ssex

from Student

where Sdept in('CS','MA');

--%任意长度

select Sname

from Student

where Sname like'刘%';

-- \_一个字符

select Sname

from Student

where Sname like'\_洋%';

--is 不能用 = 代替

select Sno, Cno

from SC

where Grade is NULL；

--order by 排序

select \*

from Student

order by Sdept, Sage DESC

--聚集函数 where不能用 只有 having 和select可以

select count(\*)

from Student

select count(distinct Sno)

from SC

select max(Grade)

from SC

where Cno = '1'

select SUM(Ccredit)

from SC, Course

where Sno = '201215127' and SC.Cno = Course.Cno

--group by 细化聚集函数

select Cno, count(Cno)

from SC

group by Cno--没有会报错

select Sno

from SC

group by Sno

having count(\*)>3

select Sno, avg(Grade)

from SC

group by Sno

having avg(Grade) >= 90

-- where 同时连接和查询

select Student.Sno, Sname

from Student,SC

where Student.Sno = SC.Sno and

SC.Cno = '2' and SC.Grade > 90

--自身连接

select FIRST.Cno, SECOND.Cpno

from Course FIRST, Course SECOND

where FIRST.Cpno = SECOND.Cno

--左外连接

select Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade

from Student

left outer join SC on (Student.Sno = SC.Sno)

--多表连接

select Student.Sno, Sname, Cname, Grade

from Student, SC, Course

where Student.Sno = SC.Sno and SC.Cno = Course.Cno

--嵌套查询

select Sname

from Student

where Sno in(

select Sno

from SC

where Cno = '2'

);

select Sno, Sname, Sdept

from Student

where Sdept in

(select Sdept

from Student

where Sname = '刘晨'

);

--有些嵌套查询可以用连接运算代替

select Student.Sno, Sname

from Student, SC, Course

where Student.Sno = SC.Sno and

SC.Cno = Course.Cno and

Course.Cname = '信息系统'

--相关子查询

select Sno, Cno

from SC x

where Grade >=

(select AVG(Grade)

from SC y

where y.Sno = x.Sno

);

--带有any的子查询

select Sname,Sage

from Student

where Sage < ANY(select Sage

from Student

where Sdept = 'CS')

and Sdept<>'CS';

--可以用聚集函数max代替

select Sname,Sage

from Student

where Sage < (select max(Sage)

from Student

where Sdept = 'CS')

and Sdept<>'CS';

--exists查询 只返回真假

select Sname

from Student

where exists

(select \*

from SC

where Sno = Student.Sno and Cno = '1'

)

--选修所有课的学生

select Sname

from Student

where not exists

(select \*

from Course

where not exists

(select \*

from SC

where Sno = Student.Sno

and Cno = Course.Cno));

--无法理解的一段代码

select distinct Sno

from SC SCX

where not exists

(select \*

from SC SCY

where SCY.Sno = '201215122' and

not exists

(select \*

from SC SCZ

where SCZ.Sno = SCX.Sno and

SCZ.Cno = SCY.Cno));

--union

select \*

from Student

where Sdept = 'CS'

union

select \*

from Student

where Sage <= 19;

--intersect

select \*

from Student

where Sdept = 'CS'

intersect

select \*

from Student

where Sage <= 19

--插入子查询结果

create table Dept\_age

(Sdept char(15),

Avg\_age smallint);

insert into Dept\_age(Sdept, Avg\_age)

select Sdept,AVG(Sage)

from Student

group by Sdept;

select Sno from SC

where Cno = '1' and (Grade < 60 or grade is null);

--创建视图 不能省略属性名的情况

create view IS\_S1(Sno, Sname, Grade)

as

select Student.Sno, Sname, Grades

from Student, SC

where Sdept = 'IS' and Student.Sno = SC.Sno and SC.Cno = '1'

--分组视图

create view S\_G(Sno, Gavg)

as

select Sno, AVG(Grade)

from SC

group by Sno;

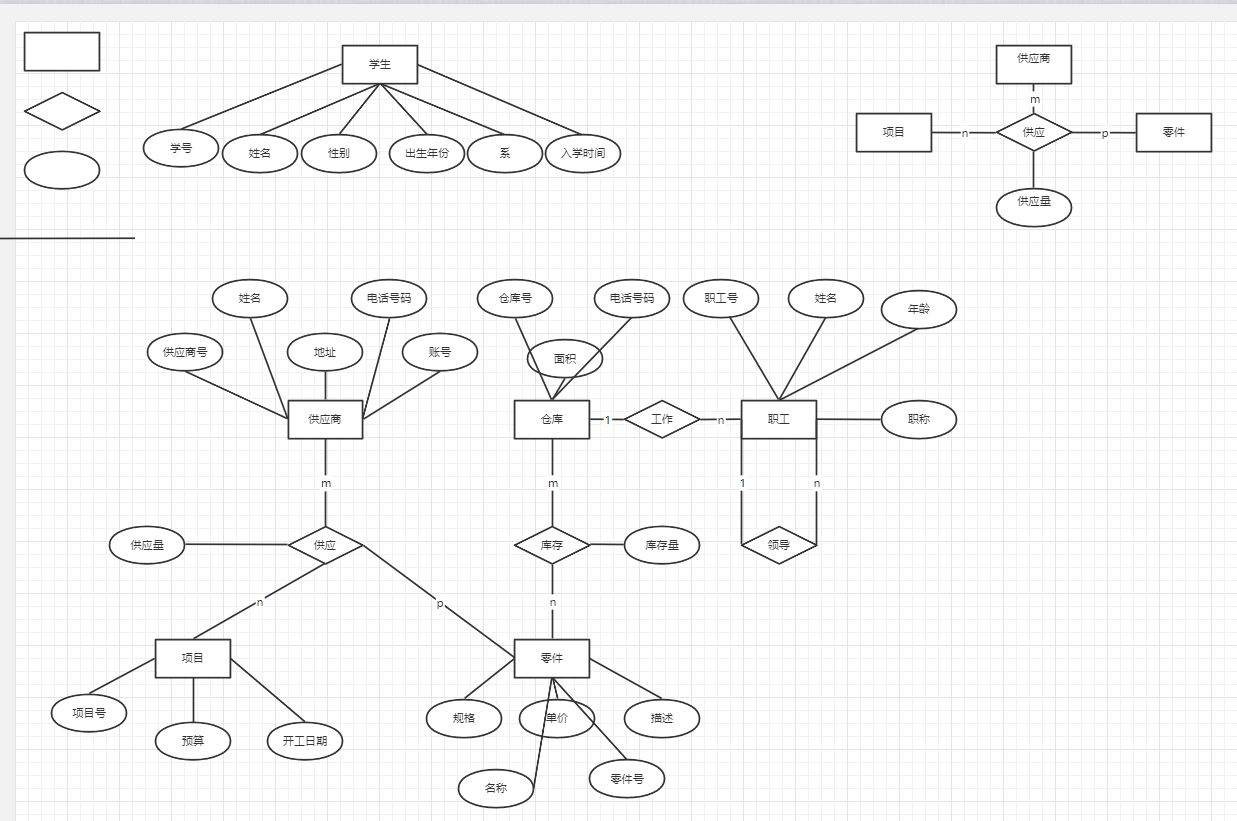
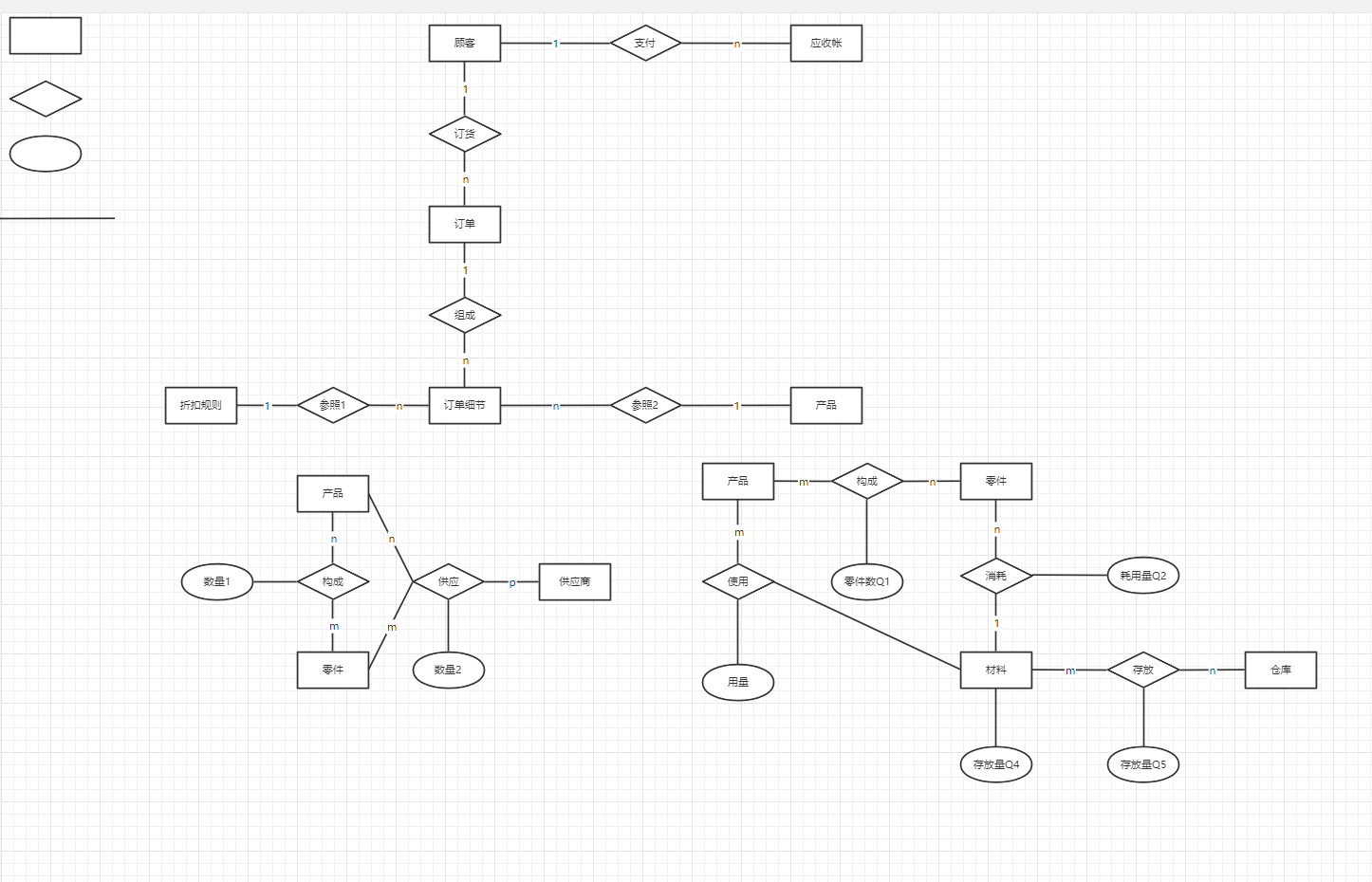
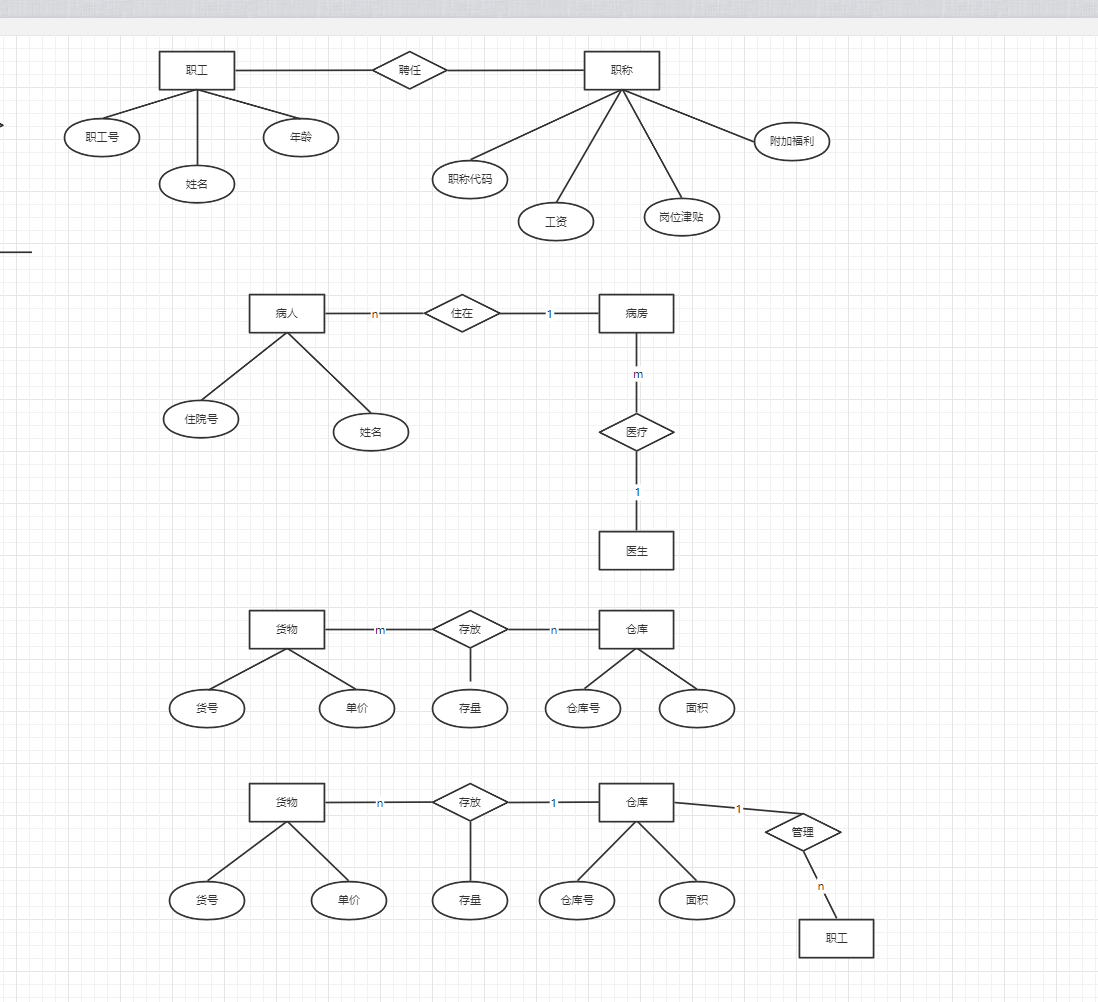
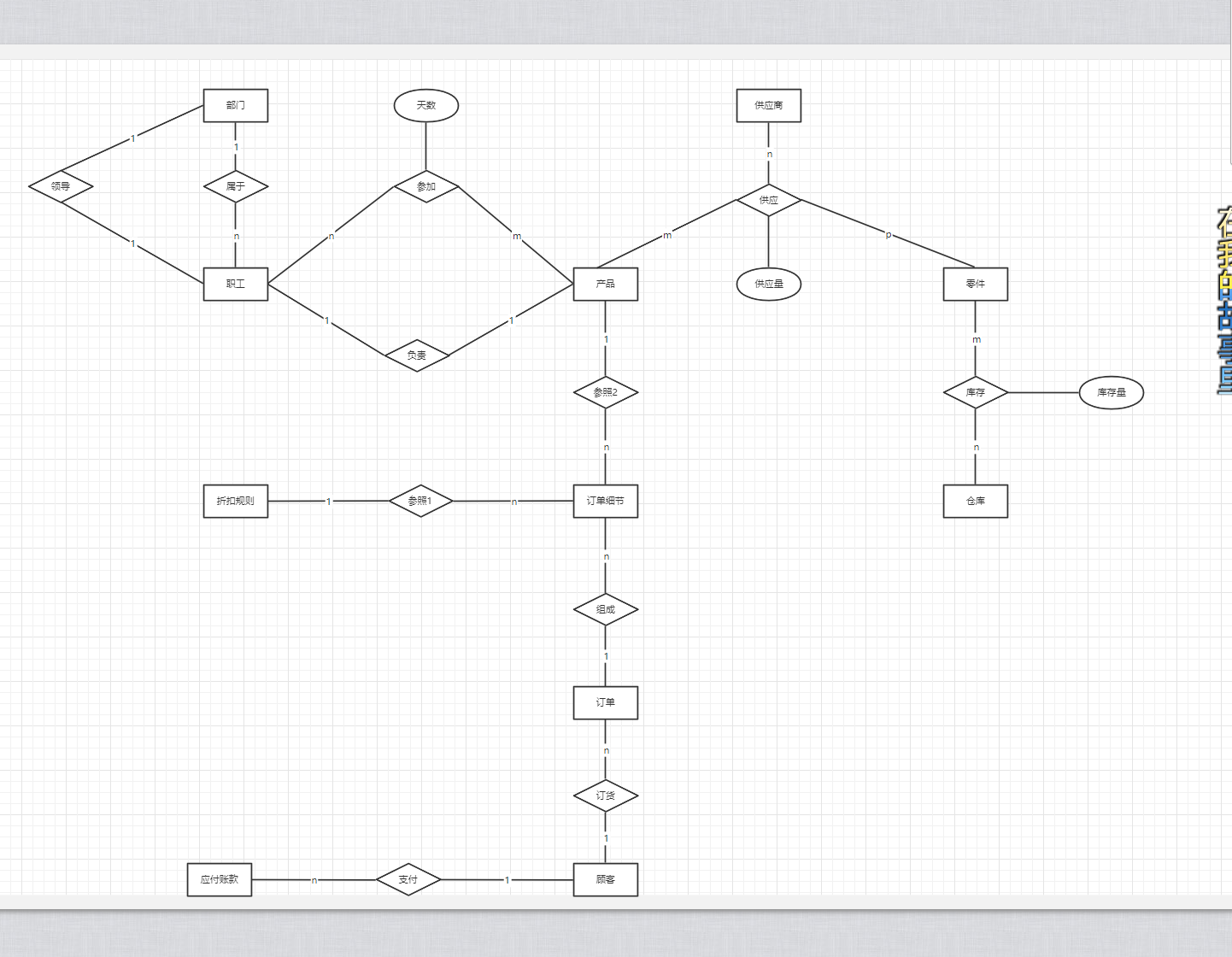
select \*

from (select Sno, AVG(Grade)

from SC

group by Sno) as S\_G(Sno, Gavg)

where Gavg >= 90;



**《数据库系统概论》大作业**

**实验报告书**

**河北大学网络空间安全与计算机学院**

|  |  |
| --- | --- |
| 学 号 | 20201202075 |
| 姓 名 | 马钰鹏 |
| 专 业 | 计算机科学与技术 |
| 年 级 | 20 |
| 指导教师 | 朱亮 |

|  |  |
| --- | --- |
| 分 数 |  |

2022年 6 月 10 日

摘 要

**摘要内容：**

当今时代是飞速发展的信息时代。在各行各业中离不开信息处理，这正是计算机被广泛应用于信息管理系统的环境。计算机的最大好处在于利用它能够进行信息管理。使用计算机进行信息控制，不仅提高了工作效率，而且大大的提高了其安全性。尤其对于复杂的信息管理，计算机能够充分发挥它的优越性。计算机进行信息管理与信息管理系统的开发密切相关，系统的开发是系统管理提目前随着各大高校的扩招，在校学生数量庞大。拥有一款好的选课管理系统软件，对于加强对在校生的选课管理起到积极作用。并且，可以为在校生随时查阅自己的选课信息、教师录入成绩等提供方便，为学校节省大量人力资源，本系统就是为了管理好学生选课信息而设计的。

**关键词：**SQL， C#， jetBrains Rider, .Net Framework

**目录**

[1 绪 论 3](#_Toc14928)

[1.1 概述 3](#_Toc7593)

[1.1.1 问题的提出 3](#_Toc23729)

[1.1.2 本课题的意义 3](#_Toc7375)

[1.2 开发环境与工具介绍 3](#_Toc31620)

[1.2.1 JetBrains Rider简介 3](#_Toc26139)

[1.2.2 Navicat简介 3](#_Toc14870)

[1.2.3 .Net Framework简介 4](#_Toc18533)

[2 系统需求分析与设计 4](#_Toc13733)

[2.1 用户需求分析 4](#_Toc16527)

[2.1.1 用户需求 4](#_Toc5701)

[2.1.2 系统功能需求 4](#_Toc9842)

[2.1.3 系统性能需求 4](#_Toc9196)

[2.2 功能模块图及分模块功能描述 5](#_Toc13649)

[2.2.1 系统的功能模块图 5](#_Toc29022)

[2.2.2 各模块简介 5](#_Toc16185)

[2.3 数据库设计 6](#_Toc29271)

[2.3.1 系统E-R图 6](#_Toc10764)

[2.3.2 数据库逻辑结构设计 6](#_Toc27249)

[3 系统实施 7](#_Toc10726)

[3.1 建立数据库 7](#_Toc1034)

[3.2 建立主程序 7](#_Toc7362)

[3.3 主要模块实施 8](#_Toc19234)

[3.4 系统测试 15](#_Toc32725)

[附录：部分源代码 31](#_Toc19491)

1 绪 论

1.1 概述

### 1.1.1 问题的提出

为方便学生和老师的管理，开发一个**学生信息管理系统。**

### 1.1.2 本课题的意义

在今天信息时代传统的管理方法必然会被计算机为基础的信息管理系统所代替，一个高效的学生选课管理系统可以存储历届的学生选课成绩档案,不需要大量的人力，只需要几名专门录入员即可操作系统，节省大量人力，可以迅速查到所需信息、高效、安全、学生在能方便的查看自己的成绩。

1.2 开发环境与工具介绍

本系统采用jetBrains Rider, Navicat, .Net Framework 开发。

### 1.2.1 jetBrains Rider简介

JetBrains Rider 是一款基于 IntelliJ 平台和 ReSharper 的跨平台 .NET IDE。支持多种 .NET 项目类型。Rider 支持 .NET 框架、新的跨平台框架 .NET Core 和基于 Mono 的项目。 功能丰富速度快，Rider 提供 2200 多种活跃代码检查，数百种来自 ReSharper 的上下文操作和重构，并与 IntelliJ 平台可靠的 IDE 功能相结合。 除了丰富的功能集，Rider 的设计宗旨是轻量级、响应式。跨平台，除了运行和调试多个运行时之外，Rider 本身还可以在多个平台上运行：Windows、macOS 和 Linux。

### 1.2.2 Navicat简介

“Navicat”是一套可创建多个连接的数据库管理工具，用以方便管理 [MySQL](https://baike.baidu.com/item/MySQL/471251)、[Oracle](https://baike.baidu.com/item/Oracle/301207" \t "https://baike.baidu.com/item/navicat/_blank)、[PostgreSQL](https://baike.baidu.com/item/PostgreSQL/530240" \t "https://baike.baidu.com/item/navicat/_blank)、[SQLite](https://baike.baidu.com/item/SQLite/375020" \t "https://baike.baidu.com/item/navicat/_blank)、[SQL Server](https://baike.baidu.com/item/SQL Server/245994" \t "https://baike.baidu.com/item/navicat/_blank)、[MariaDB](https://baike.baidu.com/item/MariaDB/6466119" \t "https://baike.baidu.com/item/navicat/_blank) 和 [MongoDB](https://baike.baidu.com/item/MongoDB/60411) 等不同类型的数据库，它与[阿里云](https://baike.baidu.com/item/%E9%98%BF%E9%87%8C%E4%BA%91/297128" \t "https://baike.baidu.com/item/navicat/_blank)、[腾讯云](https://baike.baidu.com/item/%E8%85%BE%E8%AE%AF%E4%BA%91/9905046)、[华为云](https://baike.baidu.com/item/%E5%8D%8E%E4%B8%BA%E4%BA%91/4572949)、Amazon RDS、Amazon Aurora、Amazon Redshift、Microsoft Azure、Oracle Cloud 和 MongoDB Atlas等云数据库兼容。你可以创建、管理和维护数据库。Navicat 的功能足以满足专业开发人员的所有需求，但是对数据库服务器初学者来说又简单易操作。Navicat 的用户界面 ([GUI](https://baike.baidu.com/item/GUI/479966)) 设计良好，让你以安全且简单的方法创建、组织、访问和共享信息。

### 1.2.3 .Net Framework简介

Microsoft .NET Framework是用于Windows的新托管代码编程模型。它将强大的功能与新技术结合起来，用于构建具有视觉上引人注目的用户体验的应用程序，实现跨技术边界的无缝通信，并且能支持各种业务流程。

2 系统需求分析与设计

2.1 用户需求分析

### 2.1.1 用户需求

要求不同的实体用户（一般用户，管理员）登录系统后，能够在合理范围内对数据库中的数据进行操作。

### 2.1.2 系统功能需求

1．一般用户：查询学生信息，课程信息，班级信息，成绩信息。

2. 管理员：对学生信息，课程信息，班级信息，成绩信息进行增删改查。

**2.1.3 系统性能需求**

1. 需要管理的数据有班级、学生、课程和选课数据；

2. 完成对班级信息、学生信息和课程信息的录入、修改、删除工作；

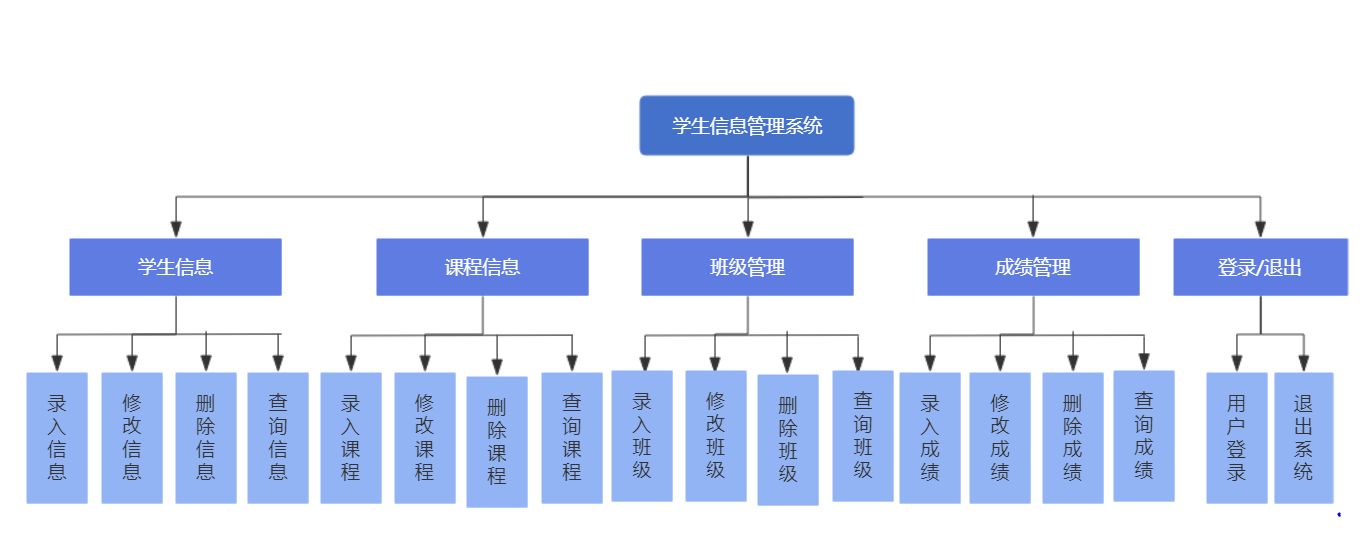
3. 能够完成对班级信息、学生信息和课程信息的查询；

4. 成绩录入要求能够自动生成成绩单；

5. 成绩单浏览。

2.2 功能模块图及分模块功能描述

### 2.2.1 系统的功能模块图



### 2.2.2 用户模块简介

1. 学生信息

对学生的信息进行管理，主要包括学生信息的录入、修改、删除、查询等工作。其中查询可以根据学号、姓名查询需要的信息；学生信息记录主要包括：学号、姓名、出生日期、性别、所在院系及专业等字段。

1. 课程信息

对课程的信息进行管理，主要包括课程信息的录入、修改、删除、查询等工作。其中查询可以根据课程号、课程名查询需要的信息；课程信息记录主要包括：课程号、课程名、所在班级、是否必修、学分、理论学时、实践学时等字段。

1. 班级管理

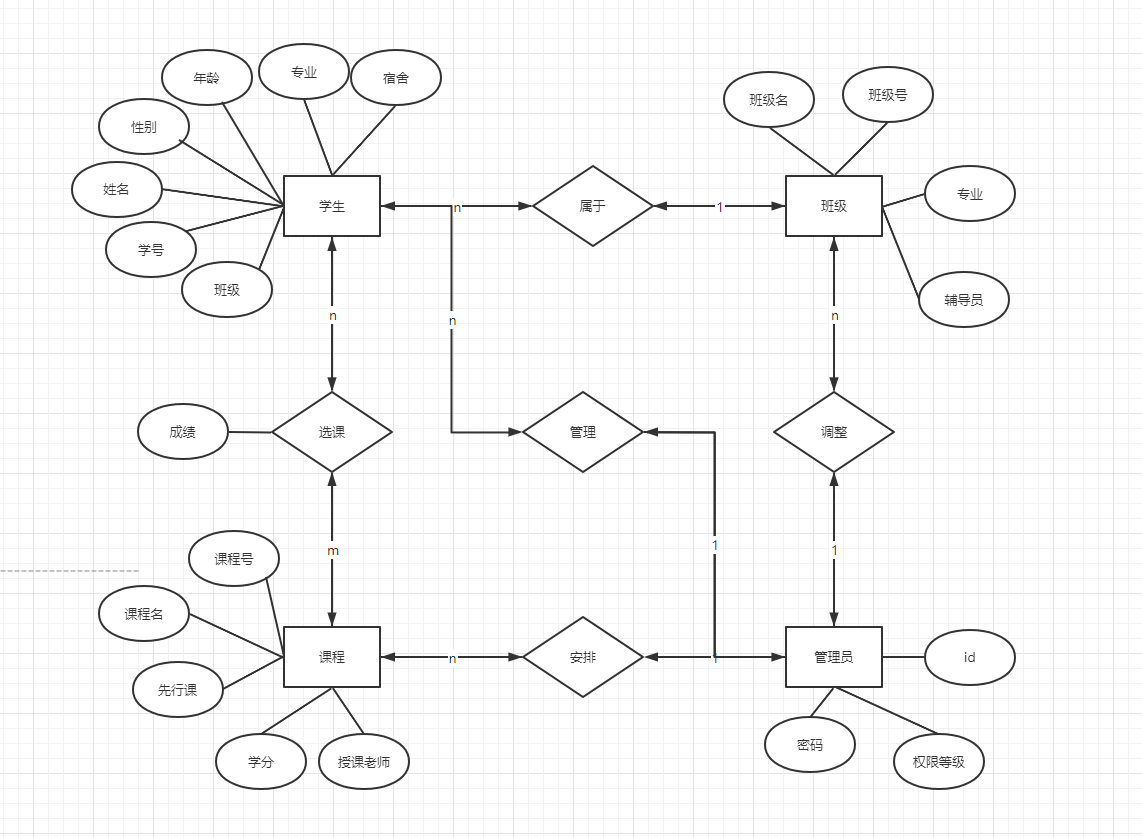
对班级的信息进行管理，主要包括班级信息的录入、修改、删除、查询等工作。其中查询可以根据班级号、班级名查询需要的信息。

1. 成绩管理

对各科成绩的信息进行管理，主要包括各科成绩的录入、修改、删除、查询等工作。其中查询可以根据学生学号、课程号查询需要的信息。

**2.3 数据库设计**

### 2.3.1 系统E-R图



### 2.3.2 数据库逻辑结构设计

管理表（id， 密码， 权限等级）

admin(user\_id, user\_password, user\_level);

学生表（名字，学号，性别，年龄，年级，专业，宿舍，班级）

Student(sname, snum, ssex, sage, sgrade，sdept, dormitory, class);

班级表（名字，班级人数，专业，辅导员）

Class(cname, stunum, sdept, cteacher);

课程表（课程号，课程名，先行课， 学分，授课老师）

Course(cno, cname, cpno,ccredit,cteacher);

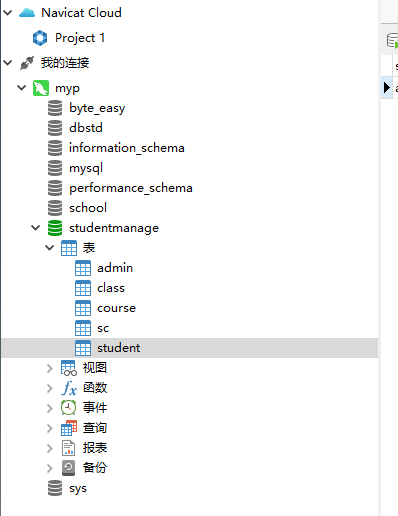
选课表（学号，课程号，分数）

SC（sno,cno,grade）;

3 系统实施

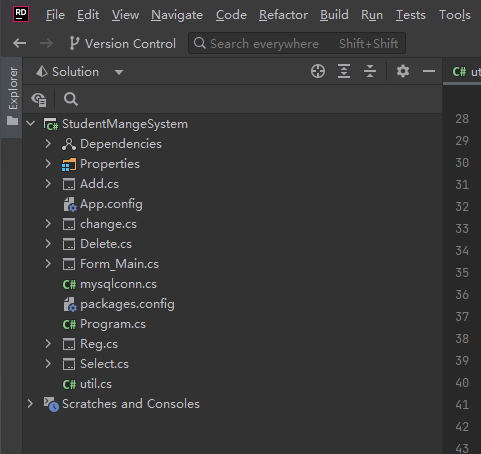
3.1 建立数据库

数据库由五个表组成，分别是admin(key:user\_id)，student(key:snum)，course(key: cno)，sc(key:sno,cno)，class(key:cname)，其中admin表存储了管理者信息和权限，student表存放了学生信息，course表存放了课程相关信息，sc表记录了学生选课信息,class记录了班级信息。建表代码详见附件1。



3.2 建立主程序

本程序由c#基于Microsoft .NET Framework 4.7.2编写，分为窗体部分和逻辑部分，Util类中包含增、删、改、查的方法，mysqlconn类提供一个与数据库的链接，其余类都是图形界面相关类。Form\_Main 为主窗口，以panel切换提供不同功能。主要供登录与注册，增删改查各功能之间的导航。Select、delete、change、add各窗口提供增删改查各个功能的集体实现。



3.3 主要模块实施

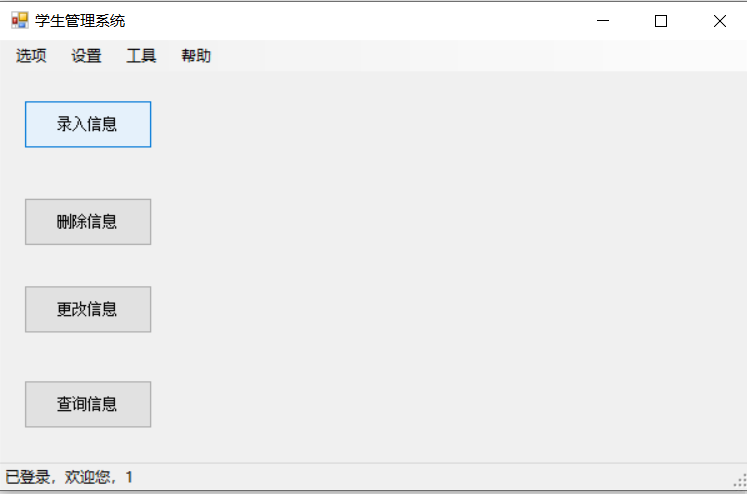
### 3.3.1 登录模块的开发

登录与注册

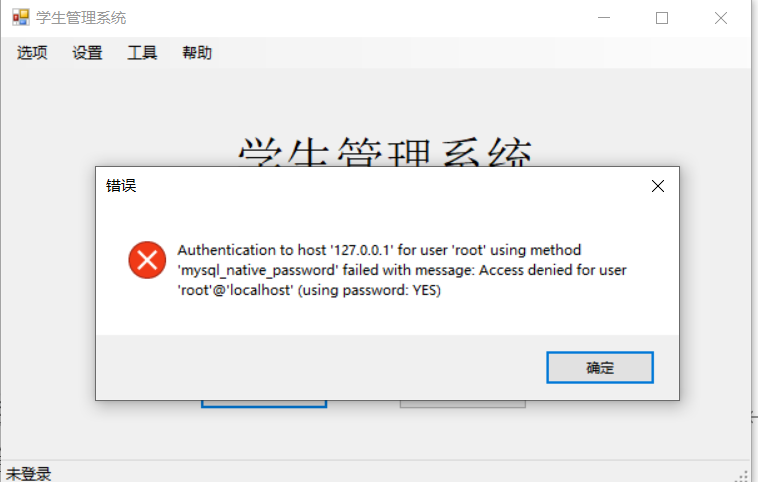
1． 登录



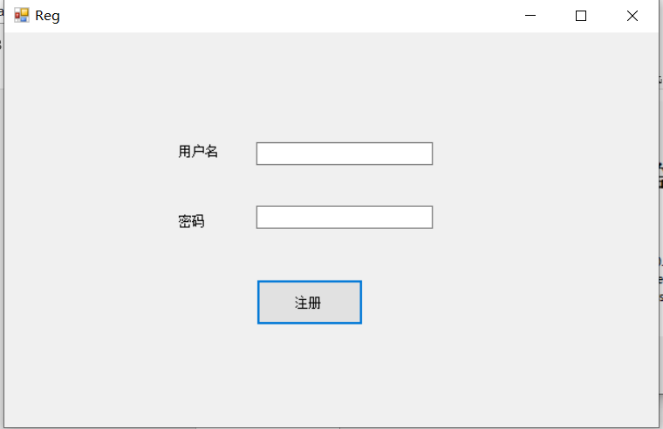
登陆成功时进入主页面



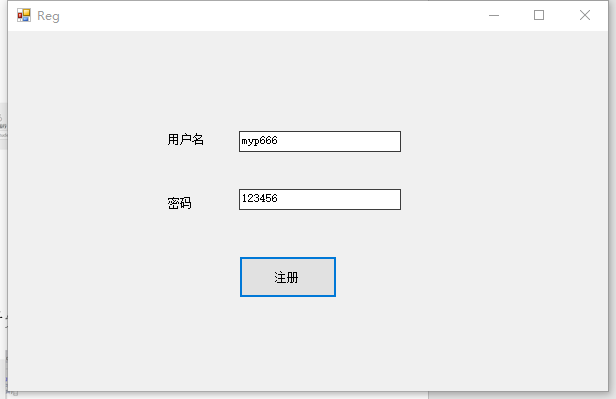
登陆失败提示错误信息



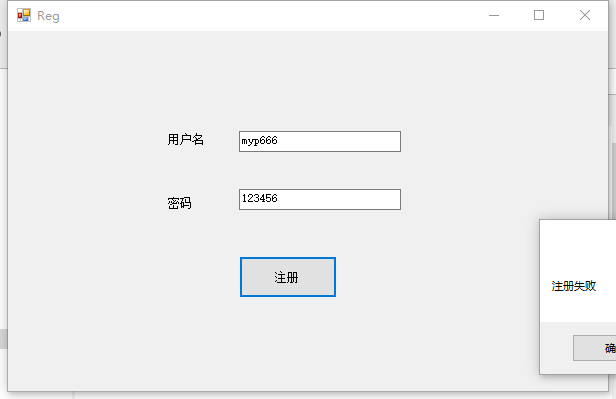
1. 注册

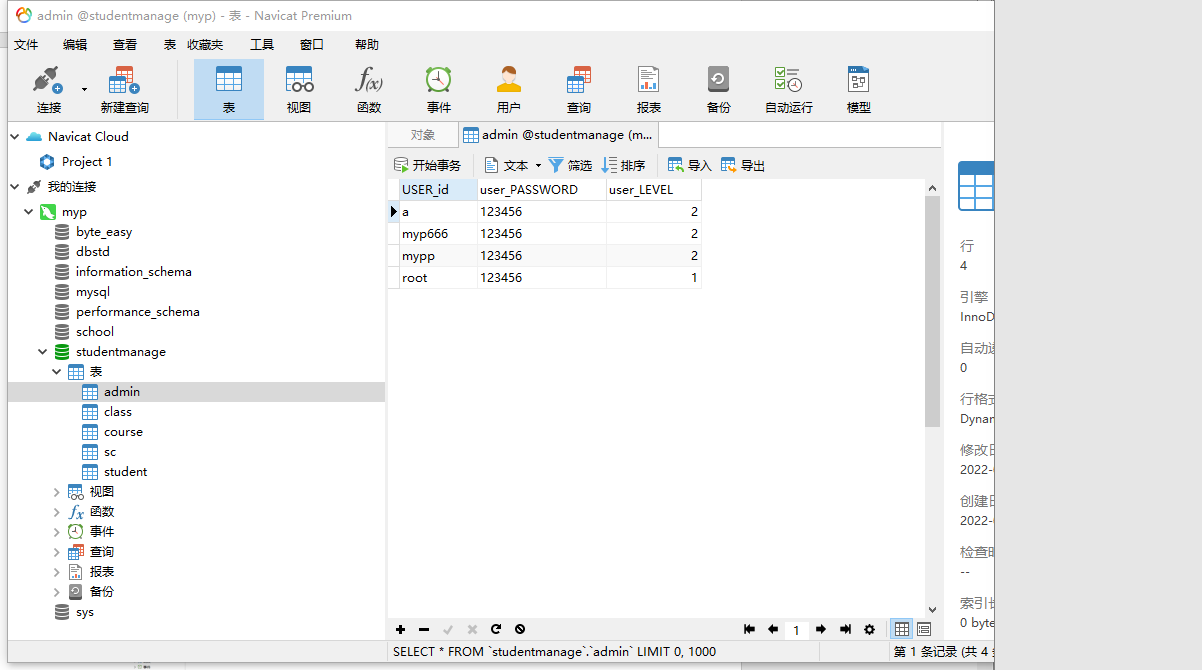


2.1注册成功提示成功，写入数据库，注册账户只有查询权限。



2.2若重名，则提示注册失败，撤销所有操作





1. 主界面

3.1 增加信息界面



3.2 删除信息界面



3.3 更改信息界面

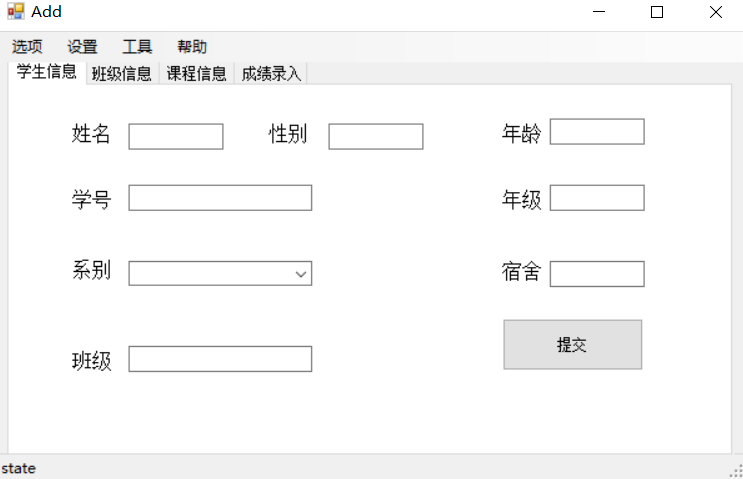


3.4 查询信息界面



3.4 系统测试

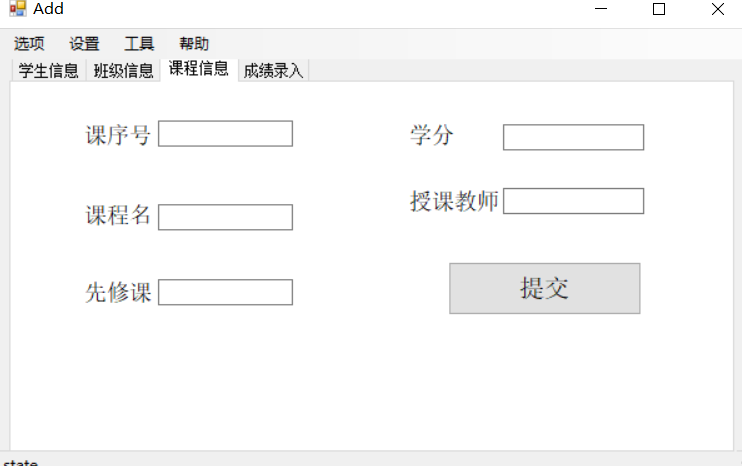
* 1. 增加学生信息



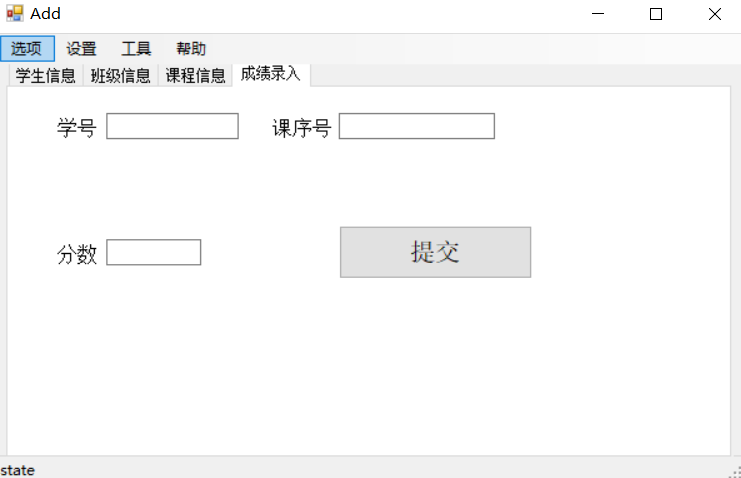
* 1. 增加班级信息



* 1. 增加课程信息

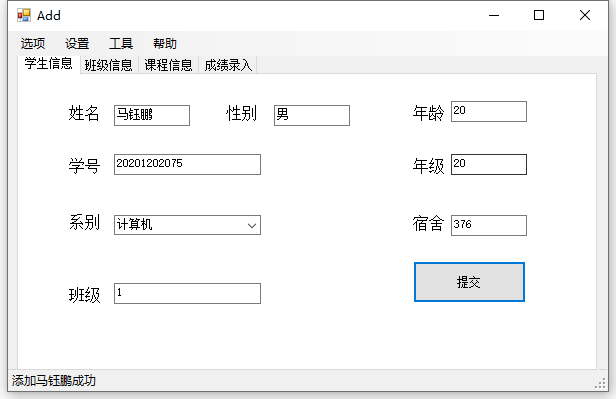


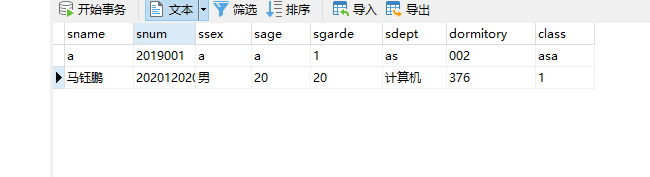
* 1. 增加成绩信息



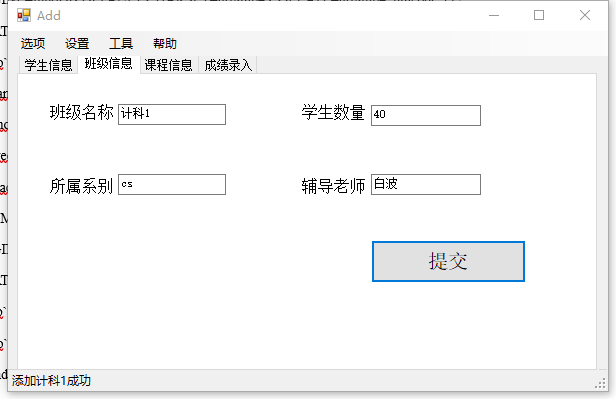
1. 增加成功演示

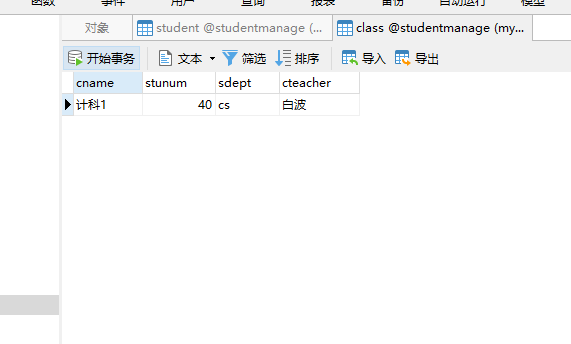
5.1学生信息



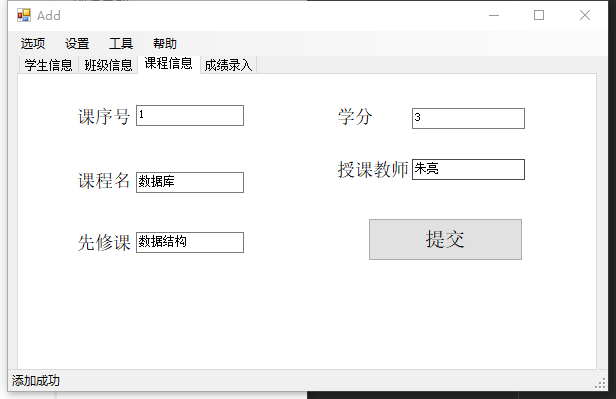


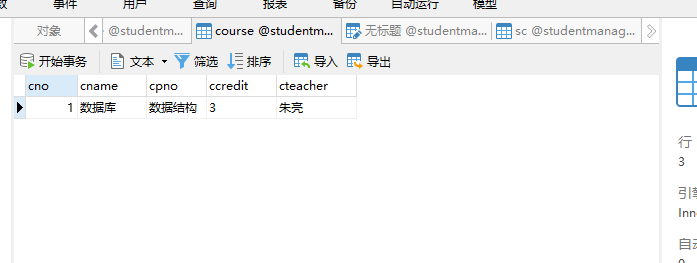
5.2增加班级信息



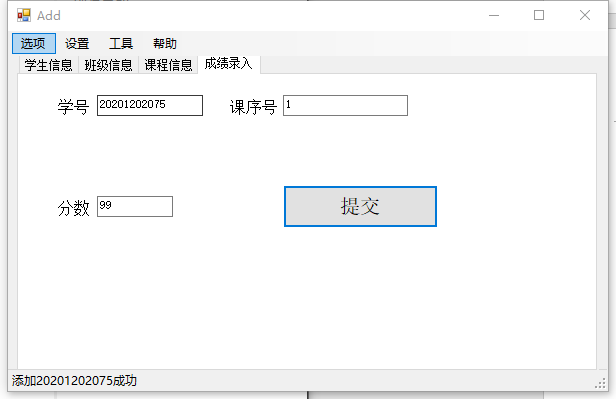


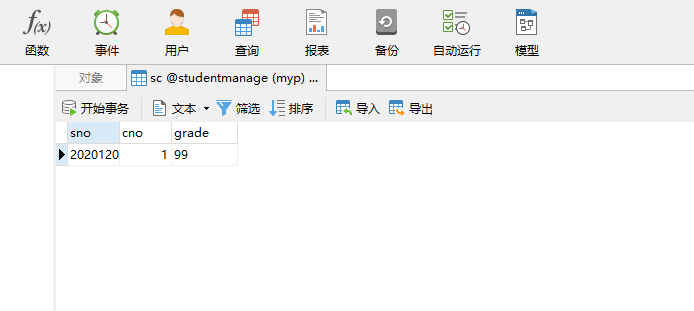
5.3增加课程信息





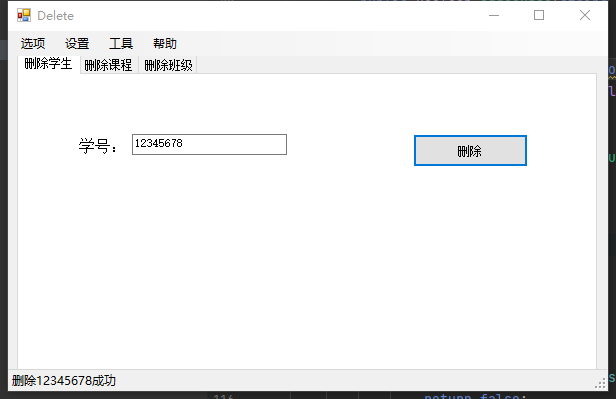
* 1. 成绩录入



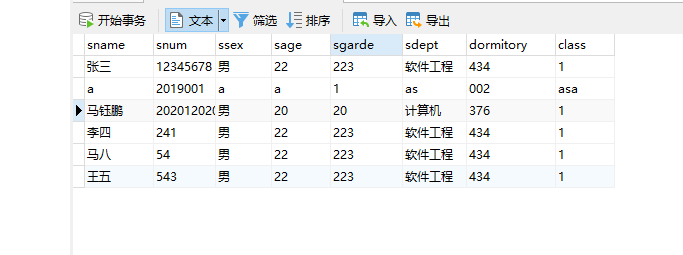


1. 删除信息

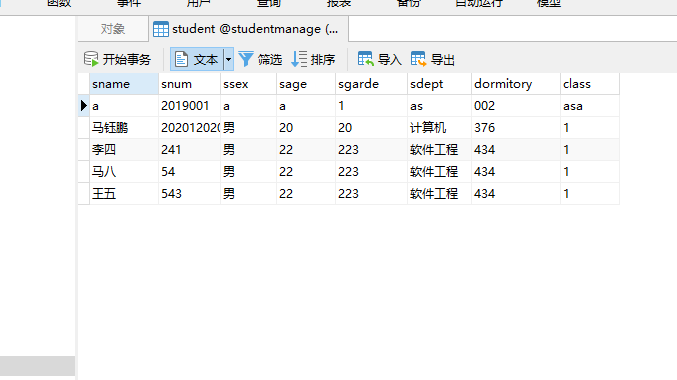
6.1 删除学生



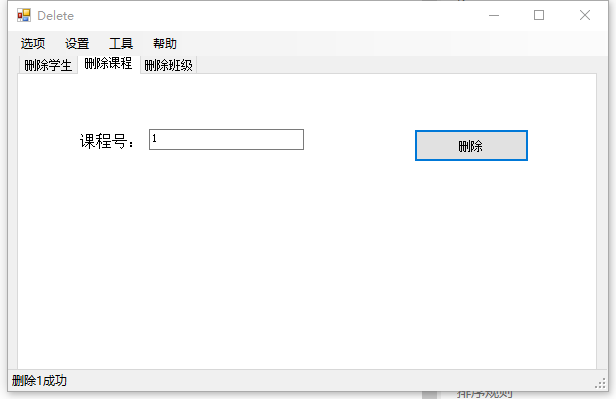
删除前

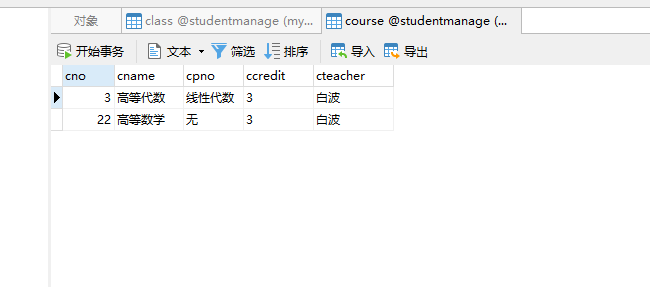


删除后

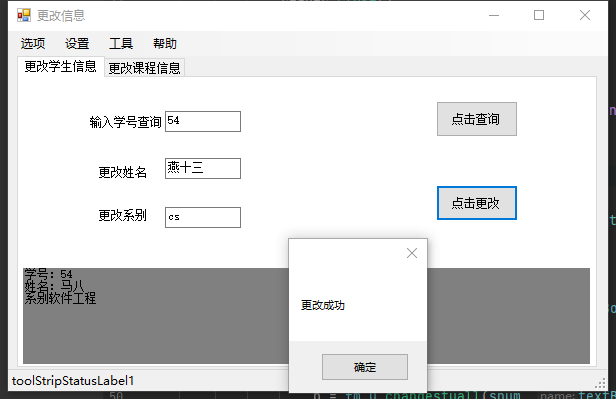


6.2 删除课程



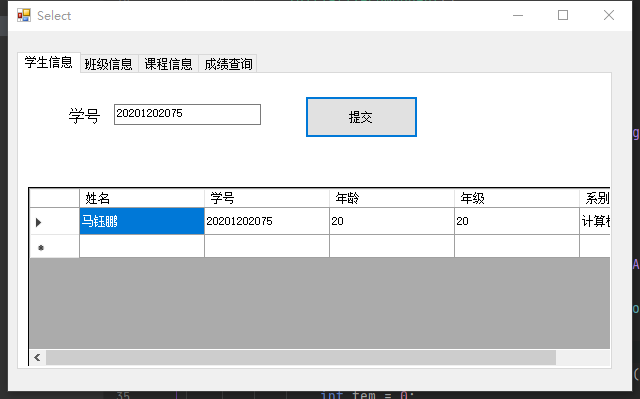


1. 更改信息



1. 查询信息

8.1 查询学生

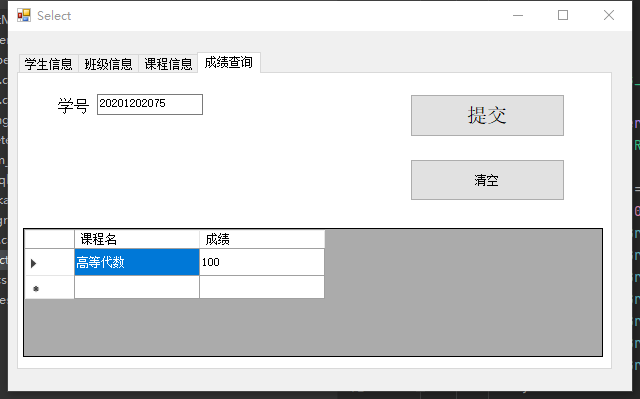


8.2 查询班级

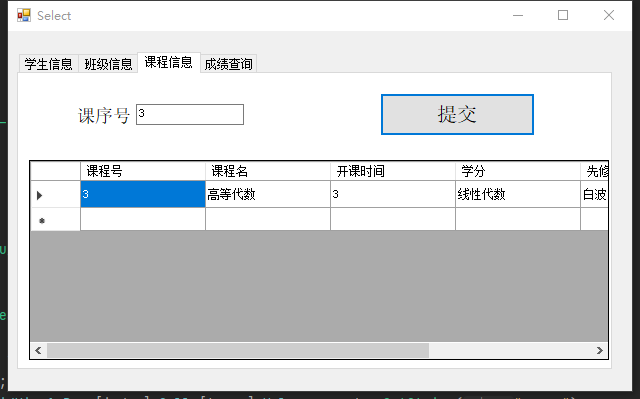
查询班级可以显示班级所有成员



8.3 查询个人成绩



8.4 查询课程信息



附录：部分源代码

建库代码：

SET FOREIGN\_KEY\_CHECKS=0;

DROP TABLE IF EXISTS `studentmanage`.`admin`;

DROP TABLE IF EXISTS `studentmanage`.`class`;

DROP TABLE IF EXISTS `studentmanage`.`course`;

DROP TABLE IF EXISTS `studentmanage`.`sc`;

DROP TABLE IF EXISTS `studentmanage`.`student`;

CREATE TABLE `admin` (

`USER\_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`user\_PASSWORD` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`user\_LEVEL` int(255) DEFAULT NULL,

PRIMARY KEY (`USER\_id`) USING BTREE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `class` (

`cname` varchar(255) DEFAULT NULL,

`stunum` int(30) DEFAULT NULL,

`sdept` varchar(255) DEFAULT NULL,

`cteacher` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `course` (

`cno` int(40) NOT NULL,

`cname` varchar(255) NOT NULL,

`cpno` varchar(10) DEFAULT NULL,

`ccredit` varchar(255) DEFAULT NULL,

`cteacher` varchar(255) DEFAULT NULL,

PRIMARY KEY (`cno`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `sc` (

`sno` varchar(30) NOT NULL,

`cno` int(30) NOT NULL,

`grade` varchar(255) DEFAULT NULL,

PRIMARY KEY (`sno`,`cno`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `student` (

`sname` varchar(255) DEFAULT NULL,

`snum` varchar(30) NOT NULL,

`ssex` varchar(255) DEFAULT NULL,

`sage` varchar(255) DEFAULT NULL,

`sgarde` varchar(255) DEFAULT NULL,

`sdept` varchar(255) DEFAULT NULL,

`dormitory` varchar(255) DEFAULT NULL,

`class` varchar(255) DEFAULT NULL,

PRIMARY KEY (`snum`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

BEGIN;

LOCK TABLES `studentmanage`.`admin` WRITE;

DELETE FROM `studentmanage`.`admin`;

INSERT INTO `studentmanage`.`admin` (`USER\_id`,`user\_PASSWORD`,`user\_LEVEL`) VALUES ('a', '123456', 2),('root', '123456', 1);

UNLOCK TABLES;

COMMIT;

BEGIN;

LOCK TABLES `studentmanage`.`class` WRITE;

DELETE FROM `studentmanage`.`class`;

UNLOCK TABLES;

COMMIT;

BEGIN;

LOCK TABLES `studentmanage`.`course` WRITE;

DELETE FROM `studentmanage`.`course`;

UNLOCK TABLES;

COMMIT;

BEGIN;

LOCK TABLES `studentmanage`.`sc` WRITE;

DELETE FROM `studentmanage`.`sc`;

UNLOCK TABLES;

COMMIT;

BEGIN;

LOCK TABLES `studentmanage`.`student` WRITE;

DELETE FROM `studentmanage`.`student`;

INSERT INTO `studentmanage`.`student` (`sname`,`snum`,`ssex`,`sage`,`sgarde`,`sdept`,`dormitory`,`class`) VALUES ('a', 2019001, 'a', 'a', '1', 'as', '002', 'asa');

UNLOCK TABLES;

COMMIT;

Form\_Main代码：

using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using MySql.Data.Common;  
using MySql.Data.MySqlClient;  
  
namespace StudentMangeSystem  
{  
 public partial class Form\_Main : Form  
 {  
 internal Boolean Login = false; *//登陆标记* internal MySqlConnection conn = null;  
 internal util u = null;  
  
 public Form\_Main()  
 {  
 InitializeComponent();  
 panel\_Mian.Visible = false;  
 Login\_panel.Visible = true;  
 }  
  
  
 private void Form\_Main\_Load(object sender, EventArgs e)  
 {  
 }  
  
 private void login\_button\_Click(object sender, EventArgs e)  
 {  
 conn = mysqlconn.getConn(textBox\_id.Text, textBox\_pw.Text);  
 try  
 {  
 conn.Open();  
 Login = true;  
 Login\_panel.Visible = false;  
 panel\_Mian.Visible = true;  
 u = new util(conn);  
 MySqlDataReader reader = u.selectLevel(textBox\_id.Text);  
 int level = 0;  
 if (reader.Read())  
 {  
 level = reader.GetInt32("user\_level");  
 }  
  
 reader.Close();  
 Login\_state.Text = $"已登录，欢迎您，{level}";  
 }  
 catch (MySqlException ex)  
 {  
 MessageBox.Show(ex.Message, "错误", MessageBoxButtons.**OK**, MessageBoxIcon.**Error**);  
 }  
 }  
  
 private void register\_Click(object sender, EventArgs e)  
 {  
 Reg reg = new Reg(this);  
 reg.Show();  
 }  
  
  
 private void panel\_Mian\_Paint(object sender, PaintEventArgs e)  
 {  
 }  
  
  
 private void Add\_Click(object sender, EventArgs e)  
 {  
 Add add = new Add(this);  
 add.Show();  
 }  
  
 private void Delete\_Click(object sender, EventArgs e)  
 {  
 Delete del = new Delete(this);  
 del.Show();  
 }  
  
 private void Change\_Click(object sender, EventArgs e)  
 {  
 change ch = new change(this);  
 ch.ShowDialog();  
 }  
  
 private void Select\_Click(object sender, EventArgs e)  
 {  
 Select sel = new Select(this);  
 sel.Show();  
 }  
  
  
 private void 退出登录ToolStripMenuItem\_Click(object sender, EventArgs e)  
 {  
 Login = false;  
 panel\_Mian.Visible = false;  
 Login\_panel.Visible = true;  
 Login\_state.Text = "未登录";  
 conn.Close();  
 }  
  
 private void 退出本系统ToolStripMenuItem\_Click(object sender, EventArgs e)  
 {  
 System.Environment.Exit(0);  
 }  
 }  
}

util类代码：

using MySql.Data.MySqlClient;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace StudentMangeSystem  
{  
 class util  
 {  
 MySqlConnection conn = null;  
 string sql;  
  
 public util(MySqlConnection conn)  
 {  
 this.conn = conn;  
 }  
  
 public MySqlDataReader selectLevel(string id)  
 {  
 sql = "select user\_level from admin where user\_id=@para1";  
 MySqlCommand cmd = new MySqlCommand(sql, conn);  
 cmd.Parameters.AddWithValue("para1", id);  
 return cmd.ExecuteReader();  
 }  
  
 public Boolean register(string id, string password)  
 {  
 MySqlTransaction transaction = conn.BeginTransaction();  
 try  
 {  
 string reg = $"create user '{id}'@'%' identified by '{password}'"; *//增加用户* string quanxian = $"grant select on \*.\* to {id} @'%'"; *//分配权限* string update = $"insert into admin(user\_id,user\_password,user\_level) values('{id}','{password}','2')";  
 MySqlCommand com1 = new MySqlCommand(reg, conn); *//增加用户* com1.ExecuteNonQuery();  
 MySqlCommand com2 = new MySqlCommand(quanxian, conn); *//分配权限* com2.ExecuteNonQuery();  
 MySqlCommand com3 = new MySqlCommand(update, conn); *//写表* com3.ExecuteNonQuery();  
 transaction.Commit();  
 return true;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 transaction.Rollback(); *//事务ExecuteNonQuery()执行失败报错，username被设置unique* return false;  
 }  
 }  
  
 public Boolean addstu(string name, string snum, string ssex, string age, string grade, string sdept, string dom,  
 string clasS)  
 {  
 string sql =  
 $"insert into student(Sname,snum,ssex,sage,sgarde,sdept,dormitory,class) values('{name}','{snum}','{ssex}','{age}','{grade}','{sdept}','{dom}','{clasS}')";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public Boolean addclass(string cname, string stunum, string sdept, string cteacher)  
 {  
 string sql =  
 $"insert into class(cname,stunum,sdept,cteacher) values('{cname}','{stunum}','{sdept}','{cteacher}')";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public Boolean addcourse(string cno, string cname, string cpno, string ccredit, string cteacher)  
 {  
 string sql =  
 $"insert into course(cno,cname,cpno,ccredit,cteacher) values('{cno}','{cname}','{cpno}','{ccredit}','{cteacher}')";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public Boolean addgrade(string sno, string cno, string grade)  
 {  
 string sql = $"insert into sc(sno,cno,grade) values('{sno}','{cno}','{grade}')";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public Boolean deletestu(string sno)  
 {  
 string sql = $"delete from student where snum={sno}";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public Boolean deletecourse(string cno)  
 {  
 MySqlTransaction transaction = conn.BeginTransaction();  
 try  
 {  
 string sql1 = $"delete from sc where cno={cno}";  
 string sql2 = $"delete from course where cno={cno}";  
 MySqlCommand com = new MySqlCommand(sql1, conn);  
 MySqlCommand com2 = new MySqlCommand(sql2, conn);  
 com.ExecuteNonQuery();  
 com2.ExecuteNonQuery();  
 transaction.Commit();  
 return true;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 transaction.Rollback(); *//事务ExecuteNonQuery()执行失败报错，username被设置unique* return false;  
 }  
 }  
  
 public Boolean deleteclass(string cname)  
 {  
 MySqlTransaction transaction = conn.BeginTransaction();  
 try  
 {  
 string sql1 = $"delete from student where class={cname}";  
 string sql2 = $"delete from class where cname={cname}";  
 MySqlCommand com = new MySqlCommand(sql1, conn);  
 MySqlCommand com2 = new MySqlCommand(sql2, conn);  
 com.ExecuteNonQuery();  
 com2.ExecuteNonQuery();  
 transaction.Commit();  
 return true;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 transaction.Rollback(); *//事务ExecuteNonQuery()执行失败报错，username被设置unique* return false;  
 }  
 }  
  
 public Boolean changestusdept(string sno, string sdept)  
 {  
 string sql = $"UPDATE student SET sdept = '{sdept}' WHERE snum = '{sno}' ";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 internal bool changestuname(string snum, string name)  
 {  
 string sql = $"UPDATE student SET sname = '{name}' WHERE snum = '{snum}' ";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 internal bool changestuall(string snum, string name, string sdept)  
 {  
 string sql = $"UPDATE student SET sname = '{name}',sdept='{sdept}' WHERE snum = '{snum}' ";  
 MySqlCommand com = new MySqlCommand(sql, conn);  
 try  
 {  
 int p = com.ExecuteNonQuery();  
 if (p == 1)  
 {  
 return true;  
 }  
  
 return false;  
 }  
 catch (MySqlException ex)  
 {  
 Console.WriteLine(ex.Message);  
 return false;  
 }  
 }  
  
 public MySqlDataReader SelectStu(string snum)  
 {  
 sql = "select \* from student where snum=@para1";  
 MySqlCommand cmd = new MySqlCommand(sql, conn);  
 cmd.Parameters.AddWithValue("para1", snum);  
 return cmd.ExecuteReader();  
 }  
  
 public MySqlDataReader Selectcla(string cname)  
 {  
 sql = "select \* from student where class=@para1";  
 MySqlCommand cmd = new MySqlCommand(sql, conn);  
 cmd.Parameters.AddWithValue("para1", cname);  
 return cmd.ExecuteReader();  
 }  
  
 public MySqlDataReader SelectCou(string cno)  
 {  
 sql = "select \* from course where cno=@para1";  
 MySqlCommand cmd = new MySqlCommand(sql, conn);  
 cmd.Parameters.AddWithValue("para1", cno);  
 return cmd.ExecuteReader();  
 }  
  
 public MySqlDataReader SelectGra(string sno)  
 {  
 sql = "select cname,grade from sc,course where sc.sno=@para1 and sc.Cno=course.Cno";  
 MySqlCommand cmd = new MySqlCommand(sql, conn);  
 cmd.Parameters.AddWithValue("para1", sno);  
 return cmd.ExecuteReader();  
 }  
 }  
}