第二部分 学籍管理系统的设计和实现

姓名： 闫宇辰 学号： 18180100062 班级： 1818039

一、需求分析（系统数据和功能）

1、系统实现功能如下：

(1)录入学生基本信息，如学号、姓名、性别、出生日期等（其它如课程、教师等信息可以在建库时输入）；

(2)按学号、姓名查询学生基本信息；

(3)录入学生成绩；

(4)查询学生所选修的课程及成绩，并给出必修课平均成绩、所有课程平均成绩；

(5)查某一个学生被哪些教师教过课；

(6)查询快要被开除的学生。

2、 数据库的有关语义如下：

(1).每个专业每年都制订了教学计划，学生每年必须按照教学计划修完一定学分的课程（必修课、限选课和任选课），如大三上学期必修课22学分，限选课10学分，任选课6学分；

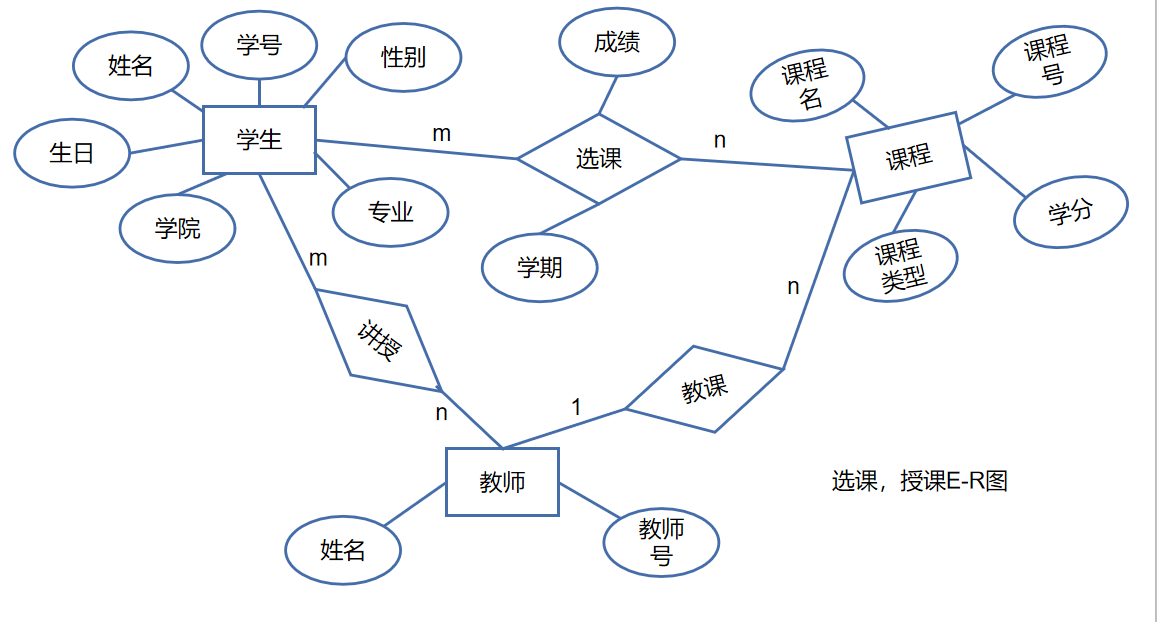
(2).教师可以给多个班带课，但是不能给一个班带多门课程；

(3)一门课程最多允许学生一次补考，学生达到如下条件之一的被开除：一学期不及格的必修课学分超过10个；不及格必修课学分累计超过30个；不及格选修课学分累计超过20个。

由此分析，为实现系统功能，数据库要储存学生，老师，课程的信息，以及学生选课情况及成绩，老师的授课情况 等相关数据信息。

二、概念结构设计（E-R图）

学生选课，教师授课。学生，教师，课程之间的E-R图如下：



三、逻辑结构设计（关系模式）

此阶段的任务是把概念结构转换成SQL Server 2014数据库管理系统能处理的数据模型。在进行这种转换时，是把实体类型和联系类型分别转换成一个个关系模式，这中间存在着多种可能组合，必须从中选择一个性能好的关系模式集作为关系数据库的模式。

为实现系统功能，我们设计了六个表：

(1)学生信息表：学号 ，姓名 ，性别 ，生日，专业 ，学院;

(2)课程信息表：课程号 ，课程名 ，学分 , 课程类型;

(3)老师信息表：教师号， 姓名; 主键为教师号;

(4)选课信息表：学期 , 学号，课程号，成绩，课程类型;

(5)授课信息表：教师号 , 课程号

(6)学分要求表：学期，必修学分，限修学分，任选学分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生信息表 student | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 学号 | sno | char (11) | × | 主键 |
| 姓名 | sname | cbar (20) | × |  |
| 性别 | ssex | char (4) | × | 只能为“男”或“女” |
| 生日 | sbirth | date | √ |  |
| 专业 | smajor | char (20) | √ |  |
| 学院 | scollege | cbar (20) | √ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程信息表 course | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 课程号 | cno | char(10) | × | 主码 |
| 课程名 | cname | char(20) | × |  |
| 学分 | credit | int | × |  |
| 课程类型 | ctype | int | × | 1-3 1:必修 2:限修 3:任选 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 选课信息表 sc | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 学期 | term | int | × | 1-8 依次为大一上，大一下 |
| 课程类型 | ctype | char(1) | √ |  |
| 学号 | sno | char(11) | × | 外码 |
| 课程号 | cno | char(10) | × | 外码 |
| 成绩 | grade | int | √ | 0-100 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教师信息表 teacher | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 教师号 | tno | char(11) | × | 主码 |
| 教师名 | tname | char(10) | × |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 授课信息表 tc | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 教师号 | tno | char(11) | × | 外码 |
| 课程号 | cno | char(10) | × | 外码 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学分要求表 command | | | | |
| 名称 | 字段 | 类型 | 允许空值 | 备注 |
| 学期 | term | int | × | 1-8 |
| 必修 | bx | int | × |  |
| 限修 | xx | int | × |  |
| 任选 | rx | int | × |  |

四、程序开发环境及应用环境

Windows10，visual studio 2019，

C# winform + SQL server2014 ， .NET Framework 4.7.2

注意；本程序只在本地测试通过，如果想使用本程序，请先在sql server中建立名为Student\_Status\_Management\_System的数据库，并将 ‘测试语句’文件夹中的sql语句执行后，再运行本程序。

Student\_Status\_Management\_System文件夹中是源码，Student\_Status\_Management\_System/bin/debug中有可执行文件，

五、应用程序设计中遇到的问题及解决方法

1. 计算平均成绩时，因为grade为整型，所以使用avg函数只能得到整数：

先求得sum 和count，然后在c#中，转化类型为float再相除。

2. 查询快被开除的学生：

先从学生表中读取所有学生的信息，然后依次查询每个学生的成绩情况，若满足被开除的三个条件中任意一条，就将其信息放在列表里，遍历完所有学生后返回列表并显示。

3。 Winform 窗体与sqlserver的交互：

使用 System.Data.SqlClient 命名空间中的 SqlConnection类来连接数据库，用SqlCommand类执行sql语句；用SqlDataReader 类 读取select到的信息，

使用该类的ExecuteReader()方法可以查看插入数据的影响行数。

六、总结

设计了一个很简陋的数据库，仅仅满足了系统要求实现的六个功能，在功能中没有涉及到的 比如学院与老师的所属关系或者学生的班级情况，并未在数据库中设计。

通过对本程序的开发制作，我对SQL软件有了进一步的了解，对该软件的功能有了一定的认识。通过winform与SQLsever的开发，我熟悉了c# 相关语法，更加熟练运用SQL语句。此次作业激起了我对数据库的兴趣，我还会继续深入学习数据库相关知识，。再次感谢一直辛辛苦苦地教我们数据库课程的指导老师。

七、建立数据库和应用程序的核心代码

这里先列举了导入数据库测试数据的部分sql语句

然后针对程序要求的六个功能，列出了其对应的窗口截图，相关的sql语句和核心代码。

1.数据库部分： 建表并导入测试数据

所有sql语句都在文件夹’测试语句’中，这里只举部分例子。

(1)建表：

在 create.sql中，

建立学生表：

create table student(

sno char(11) not null,

sname char(20) not null,

ssex char(4) not null,

sbirth date,

smajor char(20),

scollege char (20),

primary key(sno),

check (ssex in ('男','女'))

)

(2) 录入课程数据

在 insert\_course.sql中

导入课程号为 111 课程名为 必修1-1 学分为3 的必修课（ctype 为 1）:

insert into course (cno,cname , credit , ctype) values('111','必修1-1',3,1)

(3) 录入学生数据:

在 insert\_students.sql中

insert into student values('18180100062','闫宇辰','男','2000-11-12','网络空间安全','网信院')

insert into student values('18180100086','李苏','男','1999-11-08','网络空间安全','网信院')

(4) 导入选课信息：

在 insert\_sc1.sql 和 insert\_sc2.sql 中

第一学期中，学号为18180100086的学生选了课程号为111的课：

insert into sc(term,sno,cno) values(1,'18180100086','111')

(5) 更新课程表和选课表

在 update\_course\_sc.sql中

为了便于测试 ‘查询快被开除学生’ 功能， 将学号为18180100086学生的成绩全部设为40分，不及格，并将所有课学分设为10分。

update sc set grade=40 where sno = '18180100086'

update course set credit = 20

(6) 导入教师信息

在insert\_teacher.sql中

录入教师号为t111,姓名为必1-1的老师:

insert into teacher values('t111','必1-1')

(7) 导入授课信息

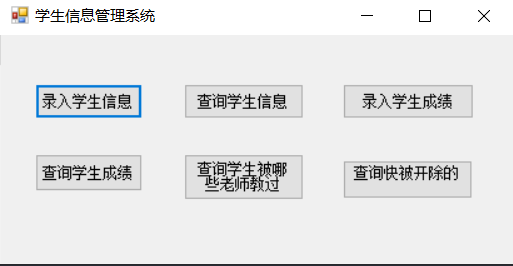
在 insert\_tc.sql中

教师号为t111的老师教授课程号为111的 课程:

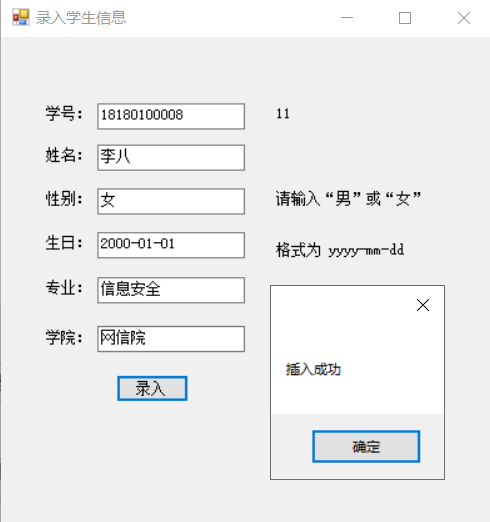
insert into tc values('t111','111')

2.前台程序

总界面：



（1）功能一：录入学生信息:



相关sql语句：

insert into student values('18180100008','李八','女','2000-01-01','信息安全','网信院')

核心代码部分：

private void button1\_Click(object sender, EventArgs e)

{

try

{

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

SqlConnection mycon = new SqlConnection(con);

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

student stu = new student();

stu.sno = txt\_sno.Text;

stu.sname = txt\_sname.Text;

stu.ssex = txt\_ssex.Text;

stu.sbirth = txt\_sbirth.Text;

stu.smajor = txt\_smajor.Text;

stu.scollege = txt\_scollege.Text;

string sql = "insert into student (sno,sname,ssex,sbirth,smajor,scollege) values('" + stu.sno + "','" + stu.sname + "','" + stu.ssex + "','" + stu.sbirth + "','" + stu.smajor + "','" + stu.scollege + "' )";

//Console.WriteLine(sql);

cmd.CommandText = sql;

int row = cmd.ExecuteNonQuery();

if (row == 1)

{

MessageBox.Show("插入成功");

}

else

{

MessageBox.Show("插入失败");

}

mycon.Close();

}

catch

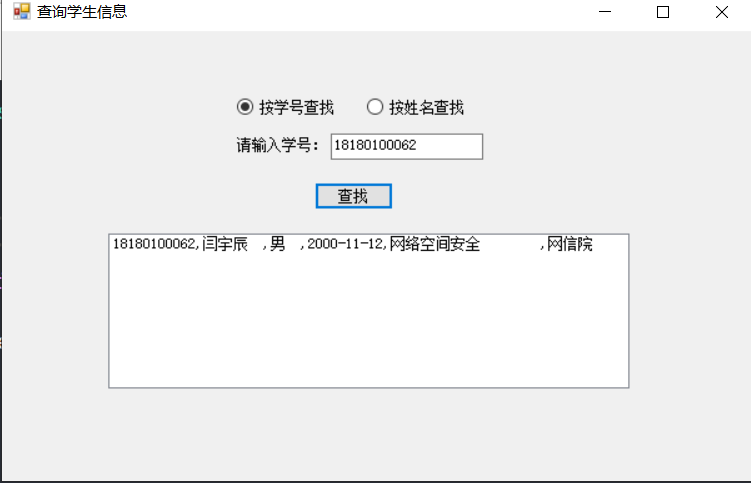
{

MessageBox.Show("插入失败");

}

}

(2)功能二:按学号，姓名查询学生信息：



相关sql语句：

select \* from student where sno=’18180100062’

Select \* from student where sname = ’闫宇辰’

核心代码：

private void button1\_Click(object sender, EventArgs e)

{

stuinfo.Items.Clear();

List<student> stulist = Get\_Stuinfo(radioButton1.Checked, radioButton2.Checked, textBox1.Text);

foreach (student stud in stulist)

{

stuinfo.Items.Add(stud.sno + ',' + stud.sname.Substring(0, 5) + ',' + stud.ssex + ',' + stud.sbirth + ',' + stud.smajor + ',' + stud.scollege);

}

}

private List<student> Get\_Stuinfo(Boolean rad1,Boolean rad2,string txt)

{

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

SqlConnection mycon = new SqlConnection(con);

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

if (rad1)//学号查找

{

cmd.CommandText = "select \* from student where sno = '" + textBox1.Text + "'";

}

else if (rad2)

{

cmd.CommandText = "select \* from student where sname = '" + textBox1.Text + "'";

}

SqlDataReader dr = cmd.ExecuteReader();

student stu = null;

List<student> stulist = new List<student>();

while (dr.Read())

{

stu = new student();

stu.sno = dr[0].ToString();

stu.sname = dr[1].ToString();

stu.ssex = dr[2].ToString();

DateTime t1 = Convert.ToDateTime(dr[3].ToString());

stu.sbirth = t1.ToString("yyyy-MM-dd");

stu.smajor = dr[4].ToString();

stu.scollege = dr[5].ToString();

stulist.Add(stu);

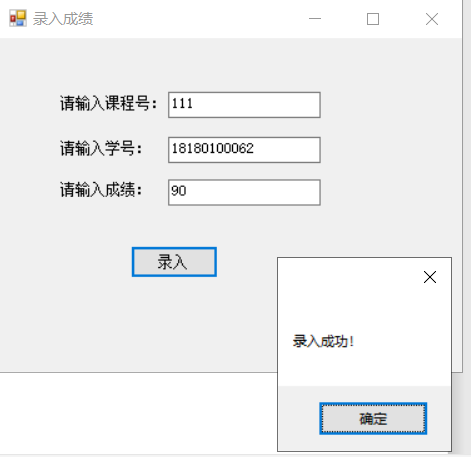
}

mycon.Close();

return stulist;

}

(3)功能三：录入成绩



相关sql语句

update sc set grade=90 where sno = ‘18180100062’ and cno=’111’

相关代码：

private void button1\_Click(object sender, EventArgs e)

{

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using (SqlConnection mycon = new SqlConnection(con))

{

try

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "update sc set grade = " + txt\_grade.Text + " where sno = '" + txt\_sno.Text + "' and cno = '" + txt\_cno.Text + "'";

Console.WriteLine(cmd.CommandText);

int row = cmd.ExecuteNonQuery();

if (row == 1)

{

MessageBox.Show("录入成功！");

}

else

{

MessageBox.Show("录入失败！");

}

mycon.Close();

}

catch

{

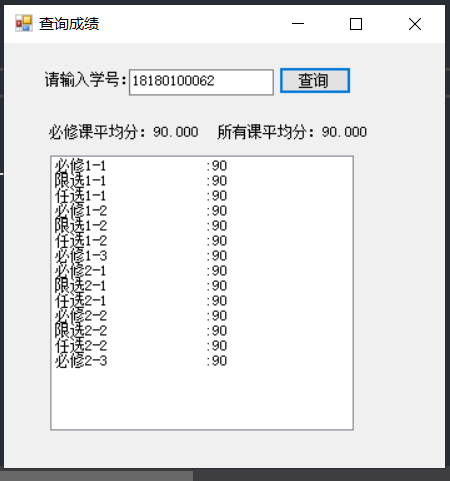
MessageBox.Show("录入失败！");

}

}

}

(4)功能四：查询成绩



相关sql语句：

select cname,grade from sc,course where sno ='18180100062' and sc.cno = course.cno

核心代码：

private void button1\_Click(object sender, EventArgs e)

{

try

{

scinfo.Items.Clear();

List<sc> scs = Get\_Grades(txt\_sno.Text);

lbl\_avg\_b.Text = "必修课平均分：" + Get\_Average\_Bixiu(txt\_sno.Text);

lbl\_avg\_all.Text = "所有课平均分：" + Get\_Average\_All(txt\_sno.Text);

foreach (sc sc in scs)

{

scinfo.Items.Add(sc.cname + ":" + sc.grade.ToString());

}

}

catch

{

MessageBox.Show("成绩未录入！");

}

}

private List<sc> Get\_Grades(string sno)

{

List<sc> scs = new List<sc>();

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using (SqlConnection mycon = new SqlConnection(con))

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "select cname,grade from sc,course where sno ='" + txt\_sno.Text + "' and sc.cno = course.cno";

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

sc sc1 = new sc();

sc1.cname = dr[0].ToString();

sc1.grade = int.Parse(dr[1].ToString());

scs.Add(sc1);

}

}

return scs;

}

private string Get\_Average\_Bixiu(string sno)

{

string average\_bixiu = "";

int count, sum;

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using (SqlConnection mycon = new SqlConnection(con))

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "select sum(grade),count(grade) from sc where sno ='" + txt\_sno.Text + "' and ctype = 1";

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

Console.WriteLine(dr[0].ToString());

sum = int.Parse(dr[0].ToString());

count = int.Parse(dr[1].ToString());

}

average\_bixiu = ((double)sum / count).ToString("#0.000");

return average\_bixiu;

}

private string Get\_Average\_All(string sno)

{

string average\_all = "";

int count, sum;

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using (SqlConnection mycon = new SqlConnection(con))

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "select sum(grade),count(grade) from sc where sno ='" + txt\_sno.Text + "' ";

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

sum = int.Parse(dr[0].ToString());

count = int.Parse(dr[1].ToString());

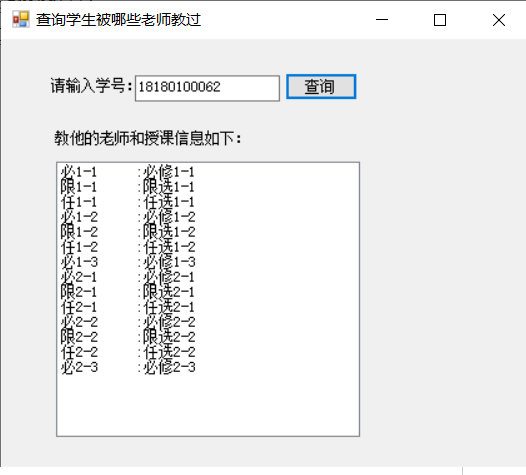
}

average\_all = ((double)sum / count).ToString("#0.000");

return average\_all;

}

(5)功能五：查询学生被哪些老师教过：



相关sql语句：

select tname,cname from tc,course,teacher,sc where sno ='18180100062' and sc.cno = course.cno and sc.cno = tc.cno and teacher.tno = tc.tno

核心代码：

private void button1\_Click(object sender, EventArgs e)

{

tinfo.Items.Clear();

List<teacher> teachers = Get\_Teachers(txt\_sno.Text);

foreach (teacher teacher in teachers)

{

tinfo.Items.Add(teacher.tname+":"+teacher.cname);

}

}

private List<teacher> Get\_Teachers(string sno)

{

List<teacher> teachers = new List<teacher>();

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using (SqlConnection mycon = new SqlConnection(con))

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "select tname,cname from tc,course,teacher,sc where sno ='" + txt\_sno.Text + "' and sc.cno = course.cno and sc.cno = tc.cno and teacher.tno = tc.tno";

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

teacher tea = new teacher();

tea.tname = dr[0].ToString();

tea.cname = dr[1].ToString();

teachers.Add(tea);

}

}

return teachers;

}

(6)功能6：查询快被开除的学生：



相关sql语句：

1: 一学期不及格的必修课学分超过10个；

查询每学期必修挂科的学分数 select term, sum(credit) from sc, course where sno = '" + stu.sno + "' and grade<60 and sc.cno = course.cno and sc.ctype = 1 group by term

2: 不及格必修课学分累计超过30个；

查询必修课挂科的总学分数select sum(credit) from sc, course where sno = '"+ stu.sno +"' and grade<60 and sc.ctype = 1 and sc.cno = course.cno

3: 不及格选修课学分累计超过20个。

查询任选课挂科的总学分数select sum(credit) from sc, course where sno = '" + stu.sno + "' and grade<60 and sc.ctype = 3 and sc.cno = course.cno

核心代码：

private void Form7\_select\_expel\_Load(object sender, EventArgs e)

{

this.Text = "查询快被开除的学生";

listBox1.Items.Clear();

List<student> expelled\_students = new List<student>();

List<student> all\_students = Get\_All\_Student();

foreach(student stu in all\_students)

{

if (expel\_checked(stu))

{

expelled\_students.Add(stu);

}

}

foreach(student stu in expelled\_students)

{

listBox1.Items.Add(stu.sname);

}

}

private List<student> Get\_All\_Student()

{

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

SqlConnection mycon = new SqlConnection(con);

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = "select \* from student";

SqlDataReader dr = cmd.ExecuteReader();

student stu = null;

List<student> stulist = new List<student>();

while (dr.Read())

{

stu = new student();

stu.sno = dr[0].ToString();

stu.sname = dr[1].ToString();

stu.ssex = dr[2].ToString();

DateTime t1 = Convert.ToDateTime(dr[3].ToString());

stu.sbirth = t1.ToString("yyyy-MM-dd");

stu.smajor = dr[4].ToString();

stu.scollege = dr[5].ToString();

stulist.Add(stu);

}

mycon.Close();

return stulist;

}

private Boolean expel\_checked(student stu)

{ // 1: 一学期不及格的必修课学分超过10个；

// 2: 不及格必修课学分累计超过30个；

// 3: 不及格选修课学分累计超过20个。

if (checked1(stu) || checked2(stu) || checked3(stu))

{

return true;

}

else

{

return false;

}

}

private Boolean checked1(student stu)

{

string sql = "select term, sum(credit) from sc, course where sno = '" + stu.sno + "' and grade<60 and sc.cno = course.cno and sc.ctype = 1 group by term";

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

SqlConnection mycon = new SqlConnection(con);

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

int sum;

if (int.TryParse(dr[1].ToString(),out sum))

{

if (sum > 10)

{

return true;

}

}

}

mycon.Close();

return false;

}

private Boolean checked2(student stu)

{

string sql = "select sum(credit) from sc, course where sno = '"+ stu.sno +"' and grade<60 and sc.ctype = 1 and sc.cno = course.cno ";

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

using(SqlConnection mycon = new SqlConnection(con))

{

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

if (dr.IsDBNull(0))

{

return false;

}

else

{

int sum = int.Parse(dr[0].ToString());

if (sum <= 30)

{

return false;

}

else

{

return true;

}

}

}

}

private Boolean checked3(student stu)

{

string sql = "select sum(credit) from sc, course where sno = '" + stu.sno + "' and grade<60 and sc.ctype = 3 and sc.cno = course.cno ";

string con = "Server=.;Database=Student\_Status\_Management\_System;Trusted\_Connection=SSPI";

SqlConnection mycon = new SqlConnection(con);

mycon.Open();

SqlCommand cmd = mycon.CreateCommand();

cmd.CommandText = sql;

SqlDataReader dr = cmd.ExecuteReader();

dr.Read();

if (dr.IsDBNull(0))

{

return false;

}

else

{

int sum = int.Parse(dr[0].ToString());

if (sum <= 20)

{

return false;

}

else

{

return true;

}

}

}