|  |
| --- |
| 浙江工业大学 |
| 学生成绩管理系统 |
| 计算机0801王俊俏 |

|  |
| --- |
| 王俊俏  2010-6-1 |

一:程序设计思路

总的思路就是分管理模块和用户模块.管理员(比如教务处)负责导入新学生和新教师,并初始化他们的个人信息(主要就是登陆密码)和选课表(学生选了什么课).用户包括学生用户和教师用户,学生只能查看自己的以往成绩,教师可以登记学生的成绩,但是没有插入学生的权限.

具体情况就分前台和后台两部分来说:

前台是主程序的界面.首先是用户登录界面,有教师,学生和管理员三个选项.

1. 教师登录后可以进行学生成绩的更新操作
2. 学生登录后只能查看自己的成绩,不能进行数据修改.
3. 管理员可以进行学生的插入,教师的插入,课程的插入以及选课表的插入.可以对数据进行Excel导入导出,方便管理.也可以在Datagridview中修改数据然后存入数据库中.

后台主要是数据存储和权限控制.数据存储方面采用SQL数据库,数据库中存有4张表,分别是:

1. 教师表,Teachers(教师号Tno,教师姓名Tname,教师登录密码Tpsw).主键是Tno
2. 学生表,Students(学号Sno,学生姓名Sname,学生登录密码Spsw).主键是Sno
3. 课程表Courses(课程号Cno,课程名称Cname).主键是Cno
4. 选课表Reports(学号Sno,课程号Cno,得分Score),主键是(Sno,Cno)

权限控制方面是这样的:管理员有一个登陆密码,这个是固定的.其他教师学生的登陆密码是存储在数据库中的,在相应的表中有这个属性.具体登陆情况如下表:

|  |  |  |
| --- | --- | --- |
| 用户类型 | 登陆名 | 密码 |
| 管理员 | Admin | 给定的 |
| 学生 | 学号 | 学号(登录后可以改) |
| 教师 | 教师号 | 教师号(登陆后可以改) |

二:程序结构框架图



三:模块及类的设计

共有5个Form,分别是

1 : 登录界面FormLogin

2 : 管理员界面 FormAdm

3：学生界面 FormStu

4: 教师界面 FormTea

5 : 密码修改界面 FormUpdatePwd

他们的调用关系是: 登录界面可以根据不同的登录类型,调出管理员界面,学生界面和教师界面中的一个.然后对于学生和教师,由于需要修改密码,故他们可以调出密码修改界面.

在数据方面,这5个Form都跟SQL数据库有联系,登录界面需要校验登录名和密码,而登录名和密码都存放在数据库中.管理员可以添加学生,添加教师,添加课程和导入学生选课的表,这些都需要对数据库进行更新操作.学生界面主要是查看数据,并没有修改数据.教师可以给学生打分,也就是更新学生的成绩.密码修改界面可以更新密码.

在导入导出方面,FormAdm,FormStu,FormTea具有数据导出功能,可以导出到Excel中作后续操作(打印等).FormAdm可以从Excel中导入数据,这样完全完成了数据的内外交互.

四:主要算法流程图

工程中用到的算法不多,主要是SQL数据查询语句代劳了.下面以登录界面的流程为例做简单介绍,其他窗体都差不多.



五: 关键源代码及编程步骤

创建SQL数据库的代码为 (在SQL server 2008中执行的)

create table Teachers

(

Tno varchar(10) unique not null,

Tname varchar(20) not null,

Tpwd varchar(20)

constraint PK\_TEC primary key (Tno)

)

create table students

(

Sno varchar(10) unique not null,

Sname varchar(20) not null,

Spwd varchar(20)

constraint PK\_STU primary key (Sno)

)

create table Courses

(

Cno varchar(10) unique not null,

Cname varchar(20) not null,

Tno varchar(10) not null,

constraint PK\_COU primary key (Cno)

)

create table reports

(

Sno varchar(10) not null,

Cno varchar(10) not null,

Score int

constraint PK\_REP primary key(Sno,Cno),

foreign key (Sno) references students(Sno),

foreign key (Cno) references courses(Cno)

)

登录界面的主要代码就是密码验证这块:

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

{

if (comboBox1.Text == "学生")

{

sqlcom.CommandText = "select \* from students where sno='" +

textBox1.Text + "' and spwd='" + textBox2.Text+"'";

sqlcon.Open();

sqldr = sqlcom.ExecuteReader(CommandBehavior.CloseConnection);

if (sqldr.Read())//datareader中有记录说明

{

//MessageBox.Show("login successfully");

FormStu FS = new FormStu();

FS.father = this;

FS.sno = sqldr[0].ToString(); FS.sname = sqldr[1].ToString();

FS.Show();

this.Visible = false;

}

else

{

MessageBox.Show("用户名或密码不对");

}

sqldr.Close();

}

else if (comboBox1.Text == "教师")

{

sqlcom.CommandText = "select \* from Teachers where Tno='" +

textBox1.Text + "' and Tpwd='" + textBox2.Text+"'";

sqlcon.Open();

sqldr = sqlcom.ExecuteReader(CommandBehavior.CloseConnection);

if (sqldr.Read())//datareader中有记录说明

{

//MessageBox.Show("login successfully");

FormTea FT = new FormTea();

FT.father = this;

FT.Tno = sqldr[0].ToString();

FT.Tname = sqldr[1].ToString();

FT.Show();

this.Visible = false;

}

else

{

MessageBox.Show("用户名或密码不对");

}

sqlcon.Close();

}

else

{

if (textBox1.Text=="Admin" && textBox2.Text=="123456")

{

//MessageBox.Show("login successfully");

FormAdm FA = new FormAdm();

FA.father = this; FA.Show();

this.Visible = false;

}

else

{

MessageBox.Show("登录失败");

}

}

}

在FormAdm,FormStu,FormTea中都有数据显示功能,这部分代码都差不多,这里仅贴一例:

private void 显示学生信息ToolStripMenuItem\_Click(object sender, EventArgs e)

{

ds.Reset();

sqlcom.CommandText = "select sno 学号,sname 姓名,spwd 密码 from students";

sqlda.SelectCommand = sqlcom;

sqlcomb = new SqlCommandBuilder(sqlda);

sqlcon.Open();

sqlda.Fill(ds, "students");

sqlcon.Close();

dv = ds.Tables["students"].DefaultView;

dataGridView1.DataSource = dv;

//数据绑定,先取消之前的绑定

textBox1.DataBindings.Clear();

textBox1.DataBindings.Add(new Binding("Text", dv, "学号"));

textBox2.DataBindings.Clear();

textBox2.DataBindings.Add(new Binding("Text", dv, "姓名"));

textBox3.DataBindings.Clear();

textBox3.DataBindings.Add(new Binding("Text", dv, "密码"));

}

数据导出到Excel的代码为:

//将数据集中的数据写入电子表格

private void DsToWsheet(DataSet ds, Worksheet wsheet)

{

int C = 0, R = 1;

foreach (DataColumn dc in ds.Tables[0].Columns)

{

wsheet.Cells[1, ++C] = dc.ColumnName;

}

foreach (DataRow dr in ds.Tables[0].Rows)

{

++R;

for (int i = 1; i <= C; i++)

{

wsheet.Cells[R, i] = dr[i - 1];

}

}

}

private void 导出ExcelToolStripMenuItem1\_Click(object sender, EventArgs e)

{

string fileName;

saveFileDialog1.Filter = "Excel 2003 \*.xls|\*.xls|Excel 2007 \*.xlsx|\*.xlsx";

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

fileName = saveFileDialog1.FileName;

Excel.Application app =

new Microsoft.Office.Interop.Excel.Application();

app.Visible = false;

Workbook wBook = app.Workbooks.Add(true);

Worksheet wSheet = wBook.Worksheets[1] as Worksheet;

DsToWsheet(ds, wSheet);

//设置禁止弹出保存和覆盖的询问提示框

app.DisplayAlerts = false;

app.AlertBeforeOverwriting = false;

wBook.SaveAs(fileName, Missing.Value, Missing.Value, Missing.Value,

Missing.Value, Missing.Value, XlSaveAsAccessMode.xlExclusive, Missing.Value,

Missing.Value, Missing.Value, Missing.Value, Missing.Value);

app.Quit();

app = null;

}

}

从Excel中导入数据的代码为(使用的是OleDb连接)

private void 从ExcelToolStripMenuItem1\_Click(object sender, EventArgs e)

{

string fileName;

openFileDialog1.Filter = "Excel 2003 \*.xls|\*.xls|Excel 2007 \*.xlsx|\*.xlsx";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

fileName = openFileDialog1.FileName;

OleDbConnection OleDbcon = new OleDbConnection();

OleDbCommand OleDbcom = new OleDbCommand();

OleDbcon.ConnectionString =

"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +

fileName + ";Extended Properties=\"Excel 12.0 Xml;HDR=YES\"";

OleDbcom.Connection = OleDbcon;

OleDbcom.CommandText = "select \* from [sheet1$]";

OleDbDataAdapter OleDbda = new OleDbDataAdapter(OleDbcom);

OleDbCommandBuilder OleDbcomb = new OleDbCommandBuilder(OleDbda);

OleDbcon.Open();

ds.Reset();

OleDbda.Fill(ds, "[sheet1$]");

dv = ds.Tables[0].DefaultView;

dataGridView1.DataSource = dv;

}

}

将DataGridView中的数据导入到数据库: 仅以插入学生为例

private void 学生表ToolStripMenuItem\_Click(object sender, EventArgs e)

{

sqlcon.Open();

foreach(DataRow dr in ds.Tables[0].Rows)

{

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

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

string spwd = dr[2].ToString();

try

{

sqlcom.CommandText =

"insert into students values ('"+sno

+"','"+sname+"','"+spwd+"')";

sqlcom.ExecuteNonQuery();

}

catch

{

}

}

sqlcon.Close();

statusBarShow("导入数据库学生表成功");

}

直接在DataGridView中修改数据,然后更新的代码:

private void 更新学生信息ToolStripMenuItem\_Click(object sender, EventArgs e)

{

sqlda.Update(ds,"students" );

statusBarShow("学生信息更新成功");

}

学生或教师修改密码:

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

{

if (comboBox1.Text == "学生")

{

sqlcom.CommandText =

"select \* from students where sno='" + textBox1.Text + "'"

+ " and spwd='" + textBox2.Text + "'";

sqlcon.Open();

sqldr = sqlcom.ExecuteReader(CommandBehavior.CloseConnection);

if (sqldr.Read())

{

sqlcom.CommandText =

"update students set spwd='" + textBox3.Text + "' "

+ "where sno='" + textBox1.Text + "'";

sqlcon.Close();

sqlcon.Open();

sqlcom.ExecuteNonQuery();

statusBar1.Text = "密码更新成功";

Thread.Sleep(1000);

statusBar1.Text = "状态栏 无消息";

}

else

{

MessageBox.Show("登录名或原密码错误,更新失败");

}

sqldr.Close();

}

else if (comboBox1.Text == "教师")

{

sqlcom.CommandText =

"select \* from Teachers where Tno='" + textBox1.Text + "'"

+ " and Tpwd='" + textBox2.Text + "'";

sqlcon.Open();

sqldr = sqlcom.ExecuteReader(CommandBehavior.CloseConnection);

if (sqldr.Read())

{

sqlcom.CommandText =

"update Teachers set Tpwd='" + textBox3.Text + "' "

+ "where Tno='" + textBox1.Text + "'";

sqlcon.Close();

sqlcon.Open();

sqlcom.ExecuteNonQuery();

statusBar1.Text = "密码更新成功";

Thread.Sleep(1000);

statusBar1.Text = "状态栏 无消息";

}

else

{

MessageBox.Show("登录名或原密码错误,更新失败");

}

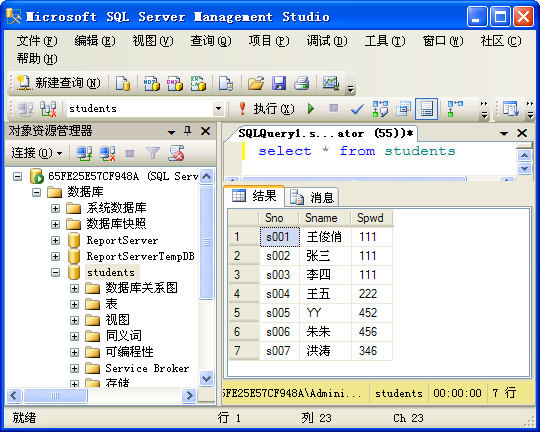
sqldr.Close();

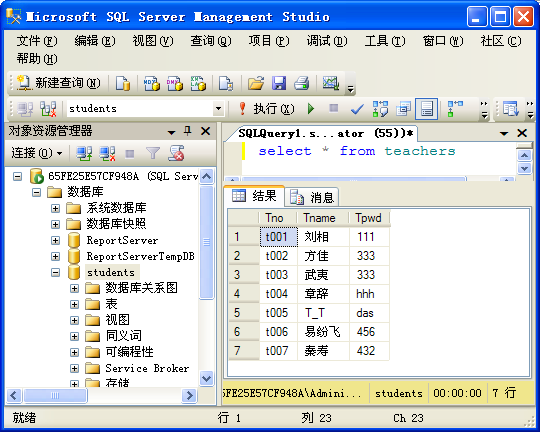
}

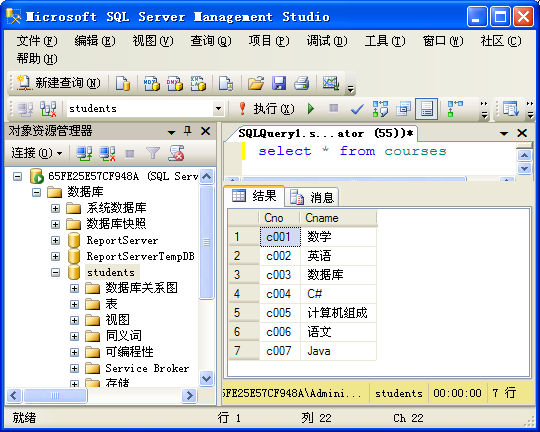
}

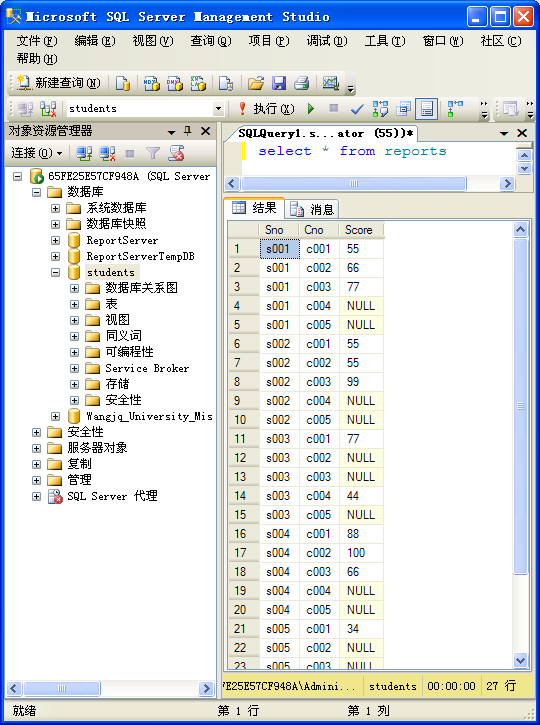
六:测试过程和运行结果分析

在后台SQL server 2008中查看所有的数据: 如下









C#工程的测试过程如下:

首先管理员登录

登录名密码为 Admin 123456



点击显示学生信息,截图如下:

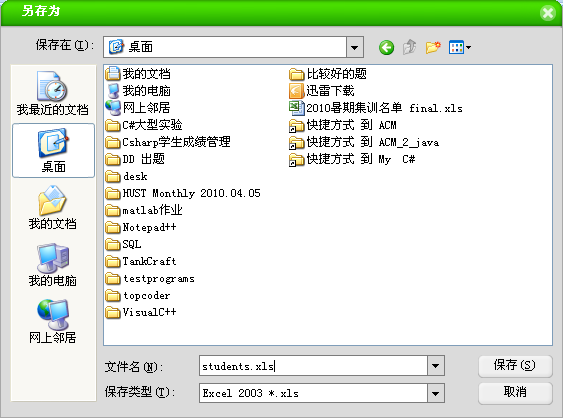


当我们在DataGridView中添加一个学生,并点击更新学生信.再次查询学生成绩,可以发现新增加的sxxx已经加进去了



其它教师项,课程项,学生成绩表项都一样测试..

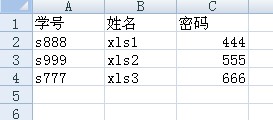
然后是对导出功能的测试:



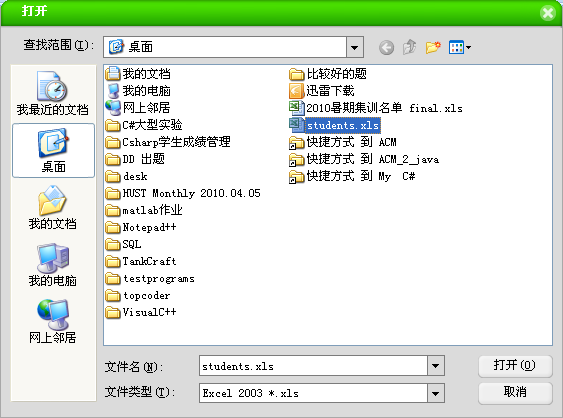
点击导入导出->导出Excel出现上面的对话框,输入文件名点保存就行了.桌面上就多了个students.xls,打开内容为:



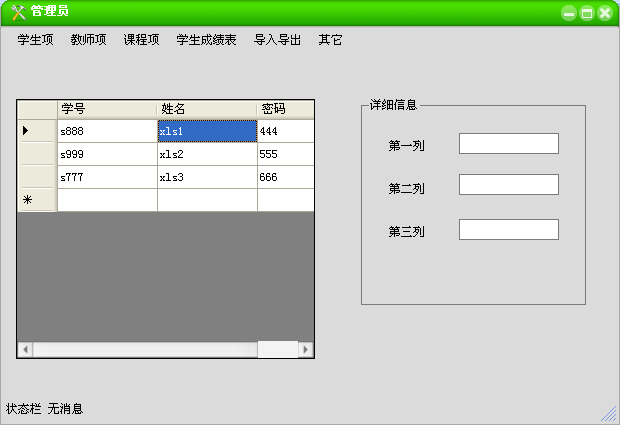
然后是导入测试,将这个students.xls的内容修改如下:



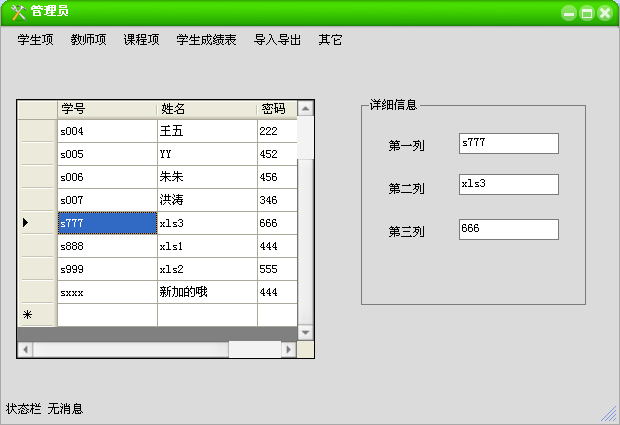
然后在FormAdm界面上点导入导出->从Excel导入,出现对话框:



在DataGridView中显示了students.xls中的内容:



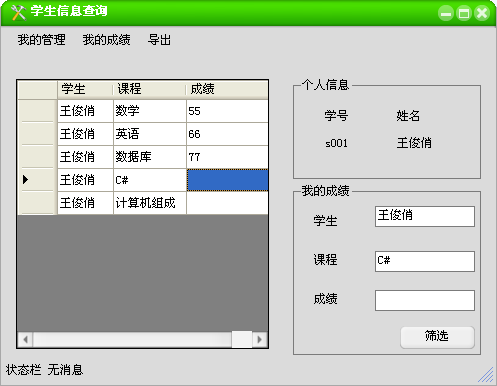
然后是将DataGridview中的数据导入到SQL数据库,点击导入导出->导入到数据库->学生表,完成导入,再点击显示学生信息,发现多了3个学生:



测试FormStu

以学号s001 密码 111登录

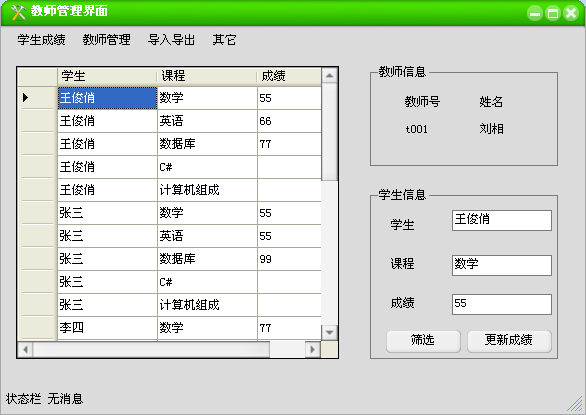
单击 我的成绩->显示所有 如下图:



筛选功能在FormTea中介绍.导出功能在FormAdm里已经讲过了.

测试FormTea,以教师号t001 密码111登录:

点击学生管理->显示选课表 如下:



更新成绩只需在成绩文本框中改,然后点击更新成绩即可.

筛选功能: 单击面板上的筛选按钮(这时筛选按钮就变成了显示),在上面三个文本框中输入筛选的条件,比如筛选选了C#课的:



现在给王俊俏的C#课打分,比如在成绩框输入99，然后更新,结果如下:



单击教师管理到密码修改界面:



只要选了类型,输入登录名,原密码,新密码,然后单击更新密码就可以了.

参考资料:

C#程序设计与案例教程

Visual Studio 2008 MSDN

<http://www.connectionstrings.com/>

<http://blog.csdn.net/gisfarmer/archive/2009/01/09/3738959.aspx>

<http://apps.hi.baidu.com/share/detail/6027706>

<http://support.microsoft.com/default.aspx?scid=kb;zh-cn;306023>