

DataBase_exp5

邓语苏 2021级本科生
计算机科学与技术
2024/5/21

实验目的

了解用C#开发基于SQL Server数据库的窗体应用程序的过程。

实验步骤

环境搭建

该项目采用C#开发，因此在编写程序之前需要搭建环境。打开Visual Studio 2019，点击“工具——获取工具和功能”，勾选“.NET桌面开发”进行安装。

创建项目

选择Windows窗体控件库（.NET Framework）



绘制界面

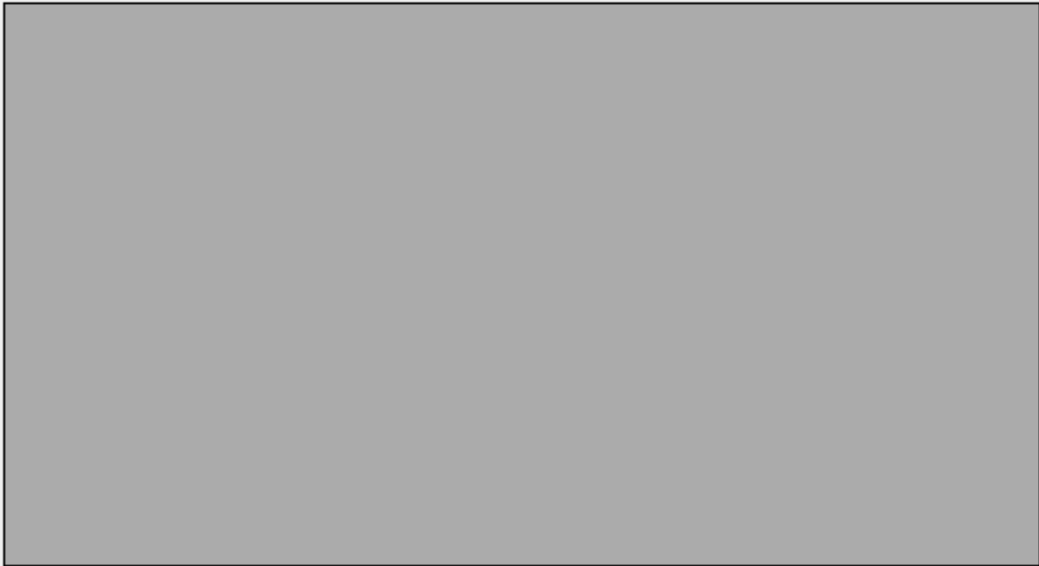
打开工具箱，拖拽控件至主窗体Form1。

数据库实验

检索

学号: 姓名:

列表



选中控件，在属性——Text可以更改控件上的文字。

属性

label2 System.Windows.Forms.Label

Image ☐ (无)

ImageAlign MiddleCenter

ImageIndex ☐ (无)

ImageKey ☐ (无)

ImageList (无)

RightToLeft No

Text 姓名:

TextAlign TopLeft

UseMnemonic True

UseWaitCursor False

行为

AllowDrop False

AutoEllipsis False

ContextMenuStrip (无)

Enabled True

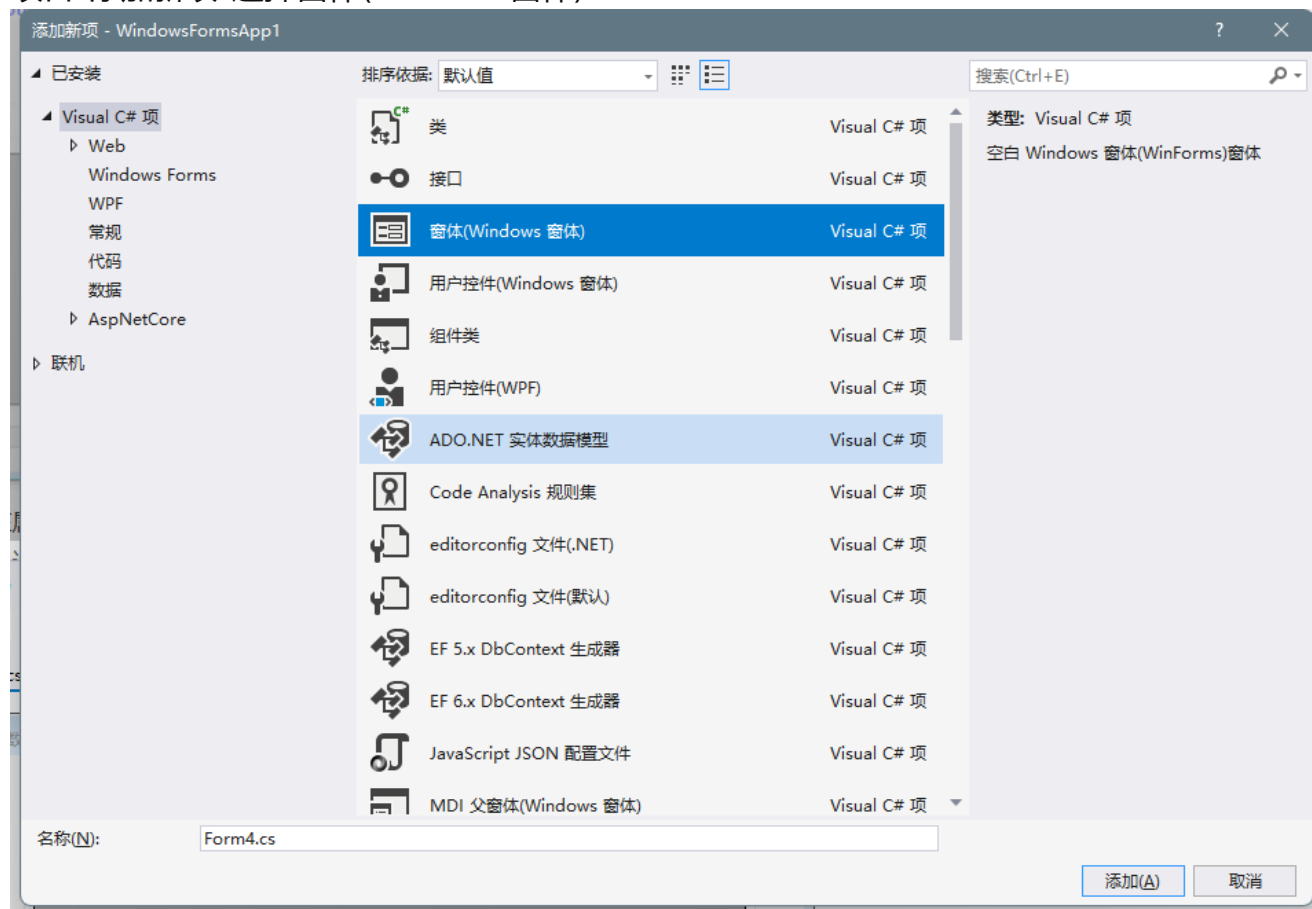
TabIndex 7

UseCompatibleTextRender False

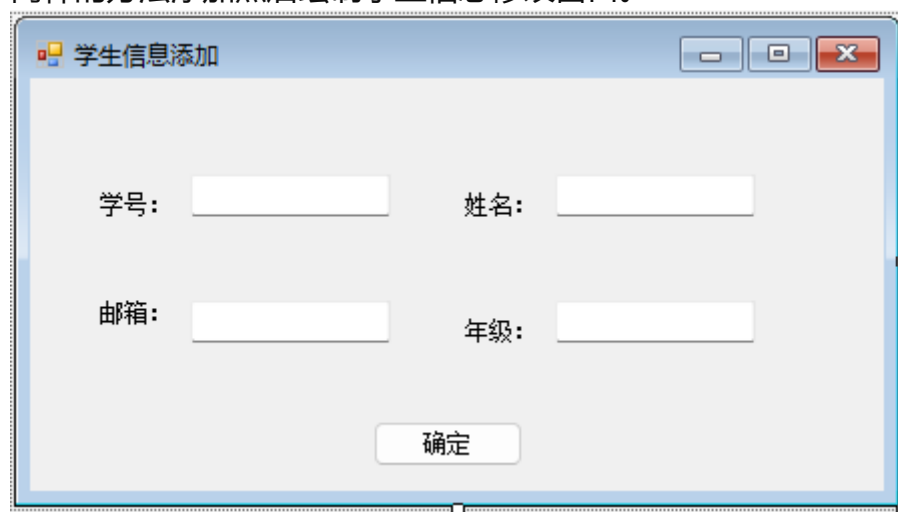
Visible True

添加窗体

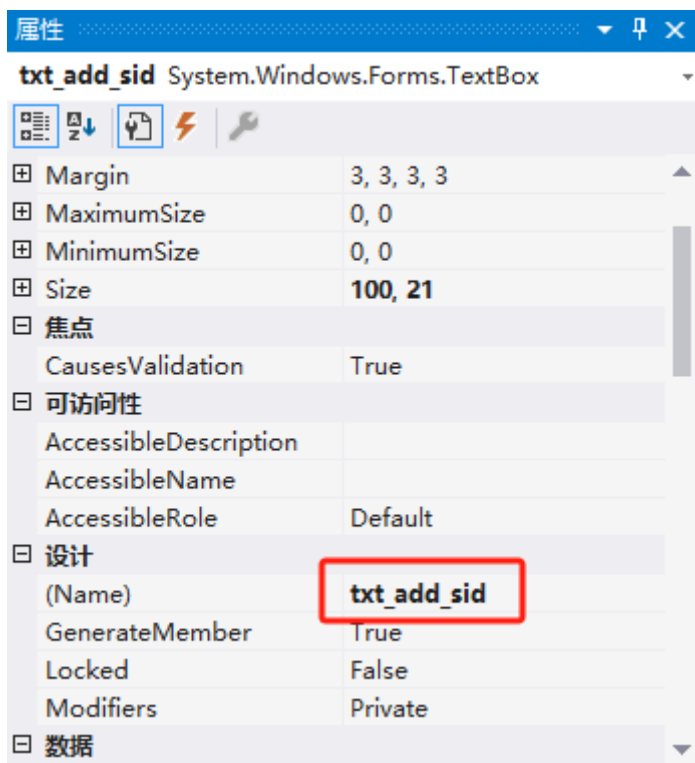
项目-添加新项-选择窗体(Windows窗体)



同样的方法添加然后绘制学生信息修改窗口。



选中学号编辑框，将其Name改为txt_sid，如下图所示：



用同样的方法按照下表给其他控件改名

主窗口	(Name)	新增窗口	(Name)	添加窗口	(Name)
学号输入框	txt_sid	学号输入框	txt_add_sid	学号输入框	txt_update_sid
姓名输入框	txt_sname	姓名输入框	txt_add_sname	姓名输入框	txt_update_sname
查询按钮	btn_find	邮箱输入框	txt_add_email	邮箱输入框	txt_update_email
新增按钮	btn_add	年级输入框	txt_add_grade	年级输入框	txt_update_grade
删除按钮	btn_delete	确定按钮	btn_confirm_add	确定按钮	btn_update_confirm_C
编辑按钮	btn_update				
DataGridView	studentsList				

如果你在没给控件改名之前不小心双击了控件，可以点击自动生成的代码上方的“1个引用”，双击跳转到引用它的位置，删除引用它的这行代码，然后再回来删除自动生成的函数。

配置数据库

修改App.config文件，添加以下选中的代码：

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration>
3   <startup>
4     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5   </startup>
6   <appSettings>
7     <add key="connectionString" value="server=localhost;database=School;uid=sa;pwd=123456"/>
8   </appSettings>
9
10 </configuration>

```

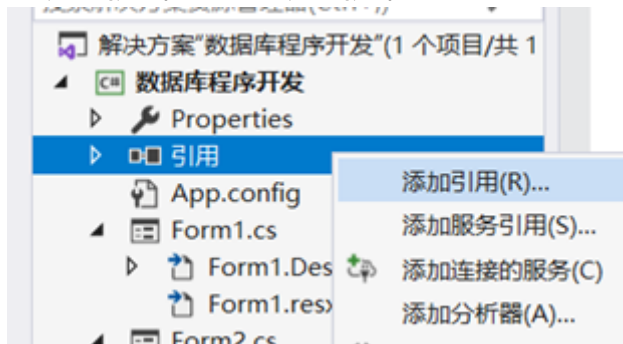
C#

```

<appSettings>
<add key="connectionString"
value="server=localhost;database=School;uid=sa;pwd=123456"/>
</appSettings>

```

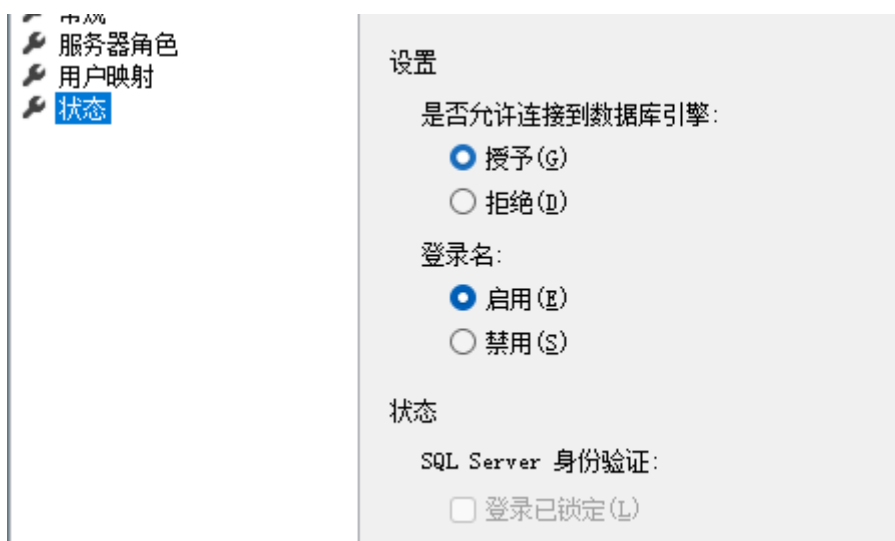
右键“引用”，选择添加引用，



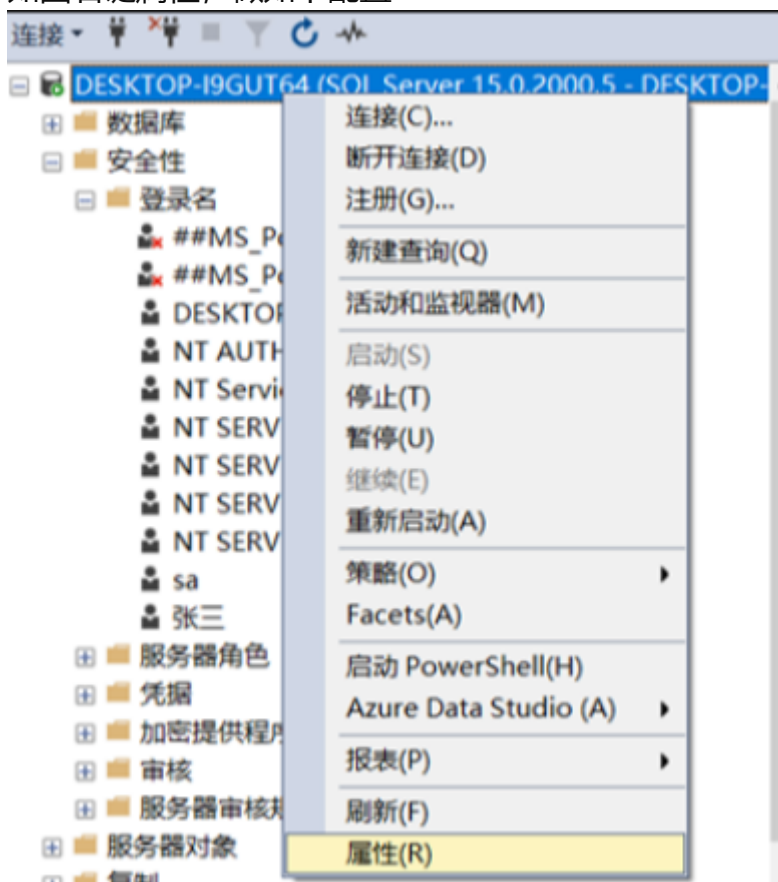
勾选

<input type="checkbox"/>	System.ComponentModel.DataAnnotations	4.0.0.0
<input checked="" type="checkbox"/>	System.Configuration	4.0.0.0
<input type="checkbox"/>	System.Configuration.Install	4.0.0.0
<input checked="" type="checkbox"/>	System.Core	4.0.0.0
<input checked="" type="checkbox"/>	System.Data	4.0.0.0
<input checked="" type="checkbox"/>	System.Data.DataSetExtensions	4.0.0.0

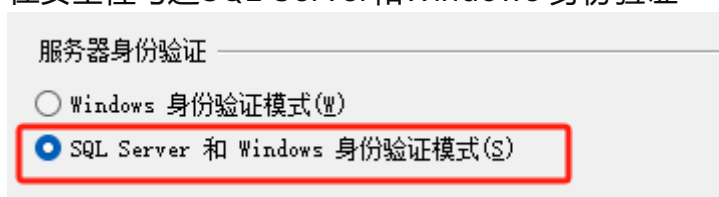
打开SQL Server Management，展开“安全性”，展开“登录名”，右键单击“sa”，打开属性，把密码改为123456。点击“状态”做如下设置：



如图右键属性，做如下配置：



在安全性勾选SQL Server和Windows 身份验证



重启SQL Server服务

编写代码

在Form1.cs的如下图所示的位置添加该代码：

5 个引用

```
public partial class Form1 : Form
```

```
{
```

1 个引用

```
public Form1()
```

```
{
```

```
    InitializeComponent();
```

```
}
```

```
static string connectionString = System.Configuration.ConfigurationManager.AppSettings["connectionString"];
```

1 个引用

```
public static DataSet Query(String sql)
```

```
{
```

```
    SqlConnection con = new SqlConnection(connectionString);
```

```
    SqlDataAdapter sda = new SqlDataAdapter(sql, con);
```

```
    DataSet ds = new DataSet();
```

```
    try
```

```
    {
```

```
        con.Open();
```

```
        sda.Fill(ds, "students");
```

```
        return ds;
```

```
    }
```

```
    catch (SqlException e)
```

```
    {
```

```
        throw new Exception(e.Message);
```

```
    }
```

```
    finally
```

```
    {
```

```
        sda.Dispose();
```

```
        con.Close();
```

```
    }
```

```
}
```

3 个引用

```
public static int ExecuteSql(String sql)
```

```
{
```

```
    SqlConnection con = new SqlConnection(connectionString);
```

```
    SqlCommand cmd = new SqlCommand(sql, con);
```

```
    try
```

```
    {
```

```
        con.Open();
```

```
        int rows = cmd.ExecuteNonQuery();
```

```
        return rows;
```

```
    }
```

```
    catch (SqlException e)
```

```
    {
```

```
        throw new Exception(e.Message);
```

```
    }
```

```
    finally
```

```
    {
```

```
        cmd.Dispose();
```

```
        con.Close();
```

```
    }
```

```
}
```

```
static string connectionString
=System.Configuration.ConfigurationManager.AppSettings["connectionS
tring"];

    public static DataSet Query(String sql)
    {
        SqlConnection con = new
SqlConnection(connectionString);
        SqlDataAdapter sda = new SqlDataAdapter(sql, con);
        DataSet ds = new DataSet();
        try
        {
            con.Open();
            sda.Fill(ds, "students");
            return ds;
        }
        catch (SqlException e)
        {
            throw new Exception(e.Message);
        }
        finally
        {
            sda.Dispose();
            con.Close();
        }
    }

    public static int ExecuteSql(String sql)
    {
        SqlConnection con = new
SqlConnection(connectionString);
        SqlCommand cmd = new SqlCommand(sql, con);
        try
        {
            con.Open();
            int rows = cmd.ExecuteNonQuery();
            return rows;
        }
        catch(SqlException e)
        {
            throw new Exception(e.Message);
        }
    }
```



```

        finally
        {
            cmd.Dispose();
            con.Close();
        }
    }
}

```

在Form3.cs中，对Form3函数做如下修改：

```

C#
public Form3(string sid,string sname,string email,string grade)
{
    InitializeComponent();
    txt_update_sid.Text = sid;
    txt_update_sname.Text = sname;
    txt_update_email.Text = email;
    txt_update_grade.Text = grade;
}

```

在主窗口双击“查询”按钮，在生成的函数中填入如下代码，表示点击这个控件要进行的操作。

```

C#
string sid = txt_sid.Text.Trim();
string sname = txt_sname.Text.Trim();
this.studentsList.DataSource = Query("select * from students where
sid like '%" + sid + "%' and sname like '%" + sname + "%'").Tables["students"];

```

在主窗口双击“新增”按钮，在生成的函数中填入如下代码：

```

C#
Form2 childrenForm = new Form2();
childrenForm.Owner = this;
childrenForm.Show();

```

在主窗口双击“删除”按钮，在生成的函数中填入如下代码：

C#

```

int a = studentsList.CurrentRow.Index;
string sid = studentsList.Rows[a].Cells[0].Value.ToString().Trim();
string sql = "delete from students where sid='" + sid + "'";

if(ExecuteSql(sql)>0)
{
    MessageBox.Show("删除成功");
}

```

在主窗口双击“更新”按钮，在生成的函数中填入如下代码：

C#

```

int a = studentsList.CurrentRow.Index;
string[] str = new string[studentsList.Rows.Count];
for (int i=0; i < studentsList.Columns.Count; i++)
{
    str[i] = studentsList.Rows[a].Cells[i].Value.ToString();
}
Form3 childrenForm = new Form3(str[0], str[1], str[2], str[3]);
childrenForm.Owner = this;
childrenForm.Show();

```

在Form2(学生信息添加窗口)双击“确定”按钮，在自动生成的函数中添加如下代码：

C#

```

string sid = txt_add_sid.Text.Trim();
string sname = txt_add_sname.Text.Trim();
string email = txt_add_email.Text.Trim();
string grade = txt_add_grade.Text.Trim();
string sql = "insert into students values('" + sid + "','" + sname
+ "','" + email + "','" + grade + "')";
Form1.ExecuteSql(sql);
this.Close();

```

在Form3(学生信息修改窗口)双击“确定”按钮，在自动生成的函数中添加如下代码：

```

string sid = txt_update_sid.Text.Trim();
string sname = txt_update_sname.Text.Trim();
string email = txt_update_email.Text.Trim();
string grade = txt_update_grade.Text.Trim();
string sql = "update students set
sname='"+sname+"',email='"+email+"',grade='"+grade+"' where
sid='"+sid+"'";
Form1.ExecuteNonQuery(sql);
this.Close();

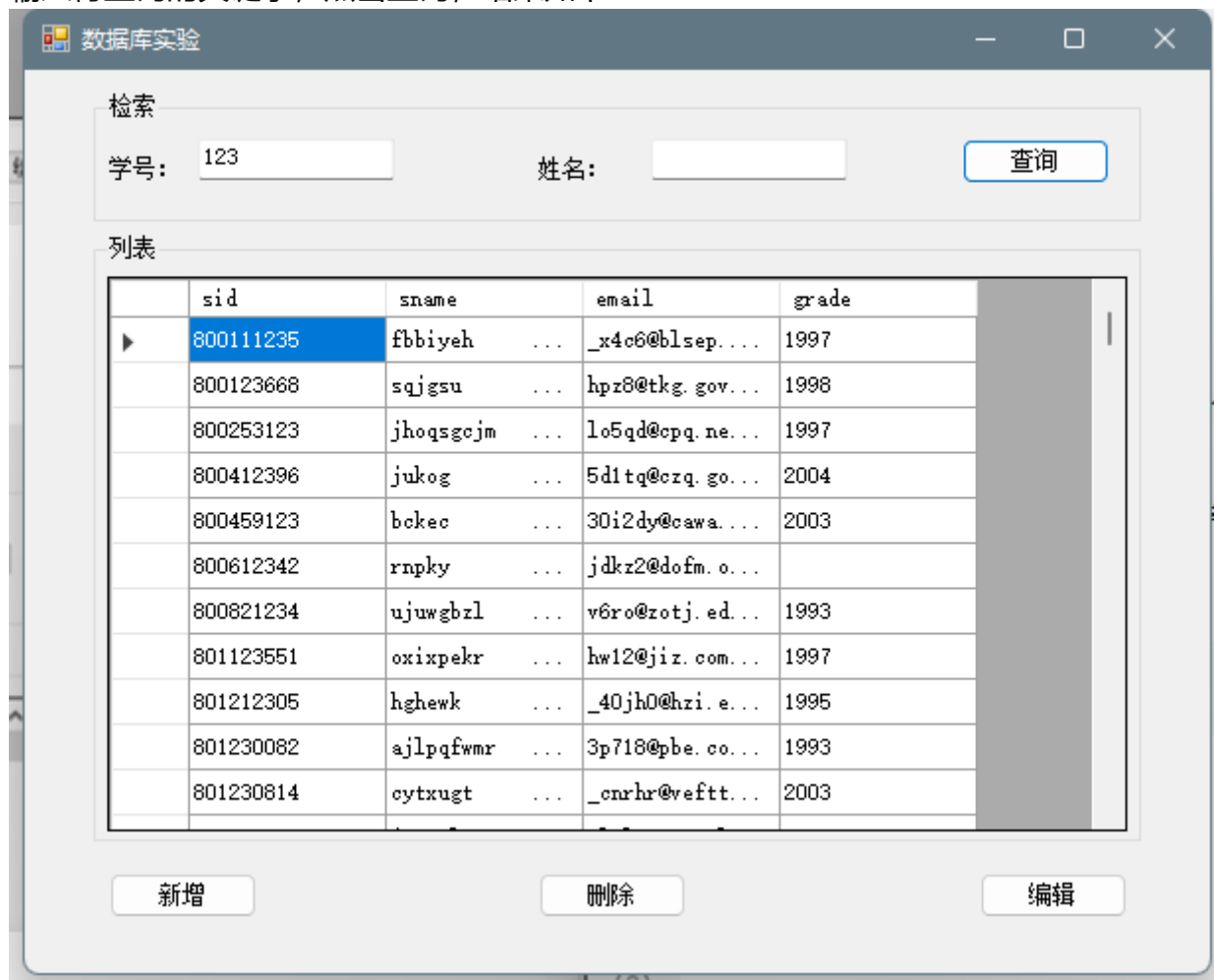
```

测试程序

运行之前，确保你本地的SQL server数据库中有名为School的数据库，并有名为student，包含sid,sname,email,grade四个属性的表。

数据查询

输入待查询的关键字，点击查询，结果如下：



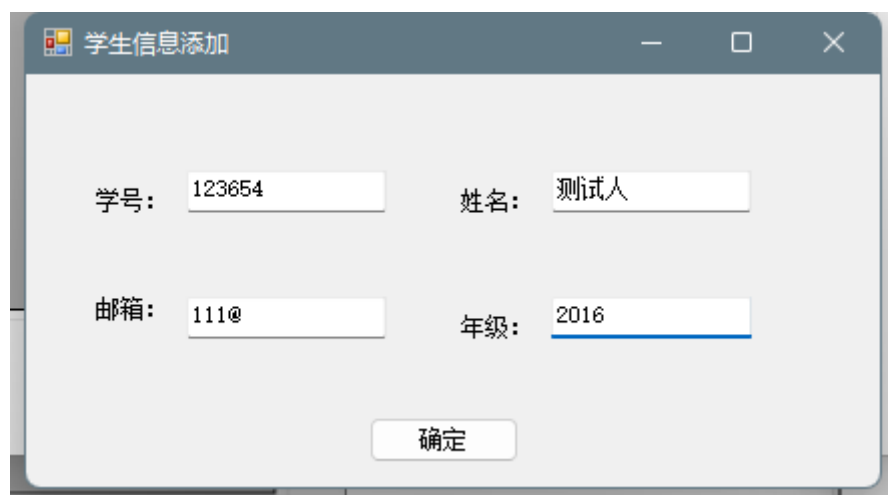
检索

学号: 姓名:

列表

	sid	sname	email	grade
▶	800111235	fbbiyeh	_x4c6@blsep...	1997
	800123668	sqjgsu	hpz8@tkg.gov...	1998
	800253123	jhoqsgcjm	lo5qd@cpq.ne...	1997
	800412396	jukog	5dltq@czq.go...	2004
	800459123	bckec	30i2dy@cawa...	2003
	800612342	rnpky	jdkz2@dofm.o...	
	800821234	ujuwgbzl	v6ro@zotj.ed...	1993
	801123551	oxixpekr	hw12@jiz.com...	1997
	801212305	hghewk	_40jh0@hzi.e...	1995
	801230082	ajlpqfwmr	3p718@pbe.co...	1993
	801230814	cytxugt	_cnrhr@veftt...	2003

数据添加

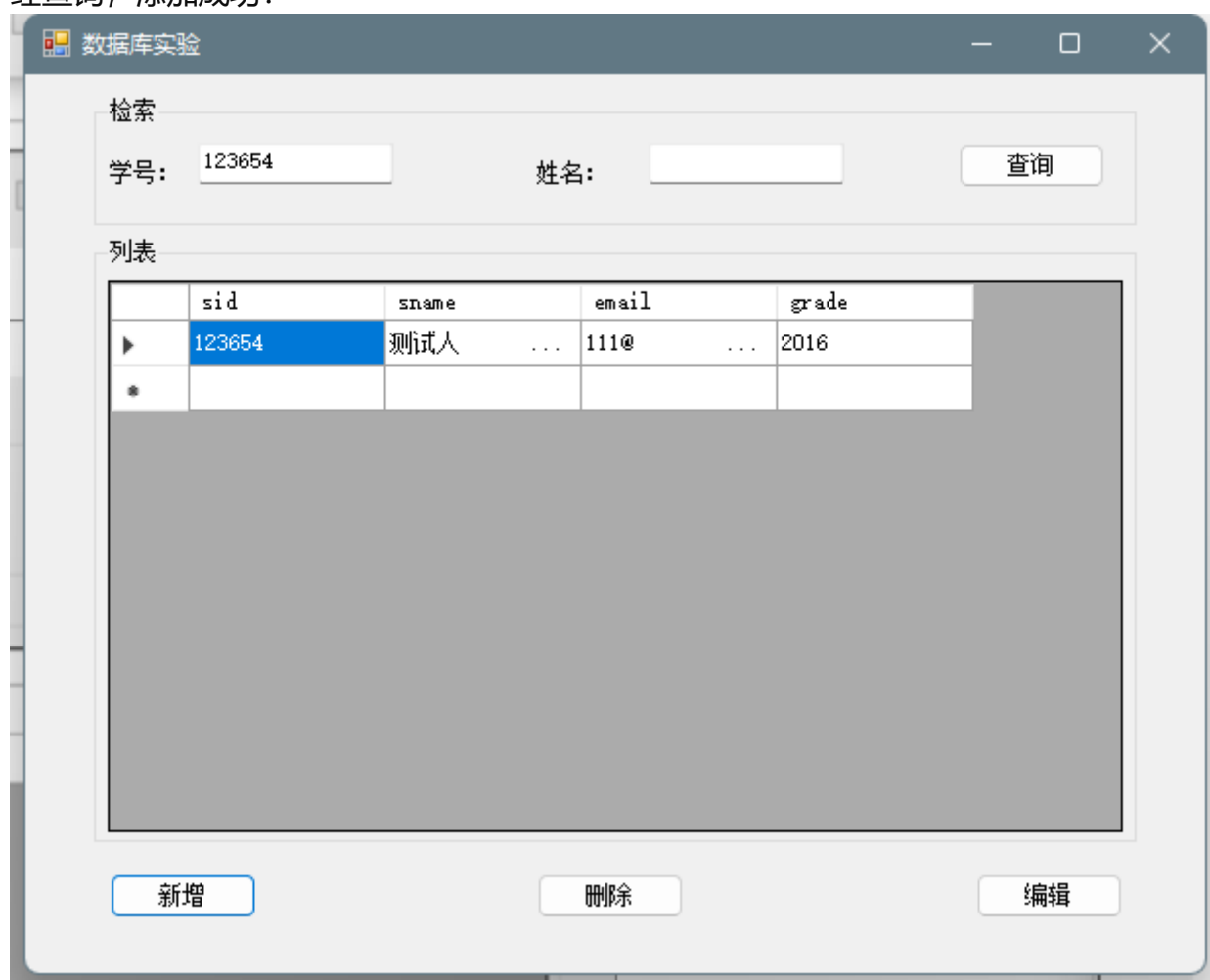


学生信息添加窗口，包含以下输入框：

- 学号: 123654
- 姓名: 测试人
- 邮箱: 111@
- 年级: 2016

底部有一个“确定”按钮。

经查询，添加成功：



数据库实验窗口，包含检索和列表功能。

检索部分：

- 学号: 123654
- 姓名:
- 查询按钮

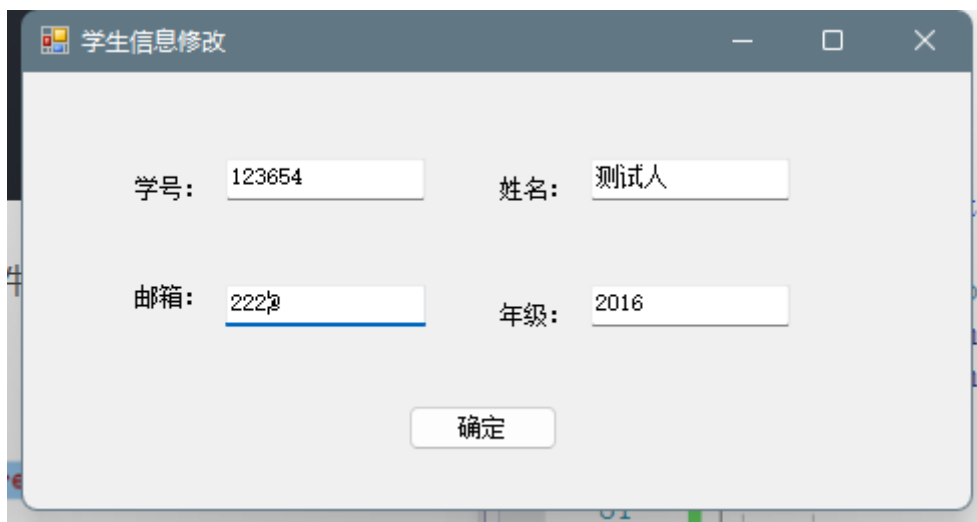
列表部分：

	sid	sname	email	grade
▶	123654	测试人	...	111@ ... 2016
*				

底部有三个按钮：新增、删除、编辑。

数据更新

比如要修改测试人的email，从111@修改为222@，

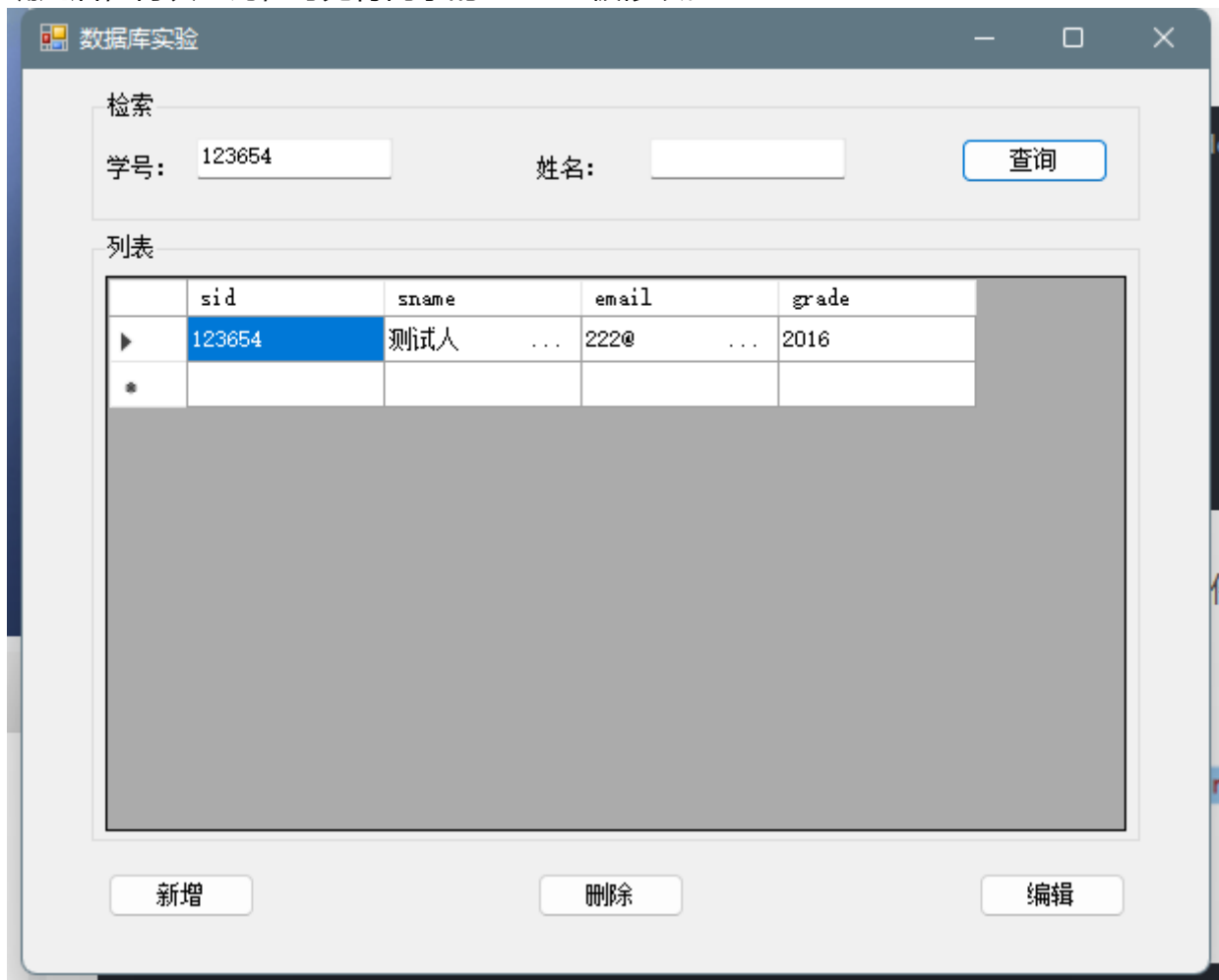


A dialog box titled "学生信息修改" (Student Information Modification) with a standard Windows window frame. It contains four text input fields arranged in a 2x2 grid. The top row has "学号:" (Student ID) with the value "123654" and "姓名:" (Name) with the value "测试人" (Test Person). The bottom row has "邮箱:" (Email) with the value "222@" and "年级:" (Grade) with the value "2016". Below the fields is a single "确定" (Confirm) button.

学号:	123654	姓名:	测试人
邮箱:	222@	年级:	2016

确定

确定后，再次查询，可见付同学的email已被修改。



A window titled "数据库实验" (Database Experiment) with a search section and a results table. The search section has "学号:" (Student ID) with "123654" and "姓名:" (Name) with an empty field, followed by a "查询" (Query) button. Below is a "列表" (List) section containing a table with 5 columns: an action column, "sid", "sname", "email", and "grade". The first row is selected, showing "123654", "测试人", "222@", and "2016". Below the table are three buttons: "新增" (Add), "删除" (Delete), and "编辑" (Edit).

检索

学号: 123654 姓名: 查询

列表

	sid	sname	email	grade
▶	123654	测试人	222@	2016
*				

新增 删除 编辑

数据删除

删除付同学的记录。
选中，点击删除。



再次查询已经没有这条记录了。

