

数据库实践课程设计简略版示例

教学管理系统

软件：SQLserver2012、Visual studio 2010

开发语言：SQL、C#

一、创建数据库概念模型

1、需求分析

1.1、系统管理员

系统管理员：

1) 维护学生的个人基本信息，实现对学生个人信息的添加、删除和更新等。学生信息包括学生的学号、姓名、性别、专业、院系、年龄、电话、EMAIL等。

2) 维护教师的个人基本信息，实现对教师个人信息的添加、删除和更新等。教师信息包括教师的工号、姓名、性别、职称、院系、年龄、电话、EMAIL等。

3) 维护课程信息，实现对课程信息的添加、删除和修改等。课程信息包括课程号、课程名、学分、课程性质等。

1.2、学生用户

学生用户：

1) 查询和修改学生的个人信息：如电话、EMAIL等。

2) 选择选修课和查看所有课程信息。包括查看选修课程的设置信息、选择选修课、退选已选课程、查看所有学习的课程等。

3) 学生可以查看自己所有课程的成绩信息。

1.3、教师用户

教师用户：

1) 查询和修改个人信息：如电话、EMAIL地址等。

2) 课程结束后，教师给所教授课程的选课学生进行成绩登记。

3) 教师可以查看自己的教学安排，包括讲授课程、学时数、以往所授课程的学生成绩等信息。

教学管理的基本规定如下：每门课程可以由多个教师讲授，不同的教师讲授的同名课程应加以区分；每个教师可以讲授多门课程；每个学生可以学习多门课程，每门课程有多个学生学习，每个学生学习每门课程都会获得一个成绩。

2、概念结构设计

2.1、抽象出系统的实体

根据分析，教学管理系统主要包含学生、教师和课程三个实体，画出三个实体的局部E-R图，并在图中标出实体的主键（加下划线的属性），分别如图2-1、图2-2和图2-3所示，其中学号是学生实体的主键，工号是教师实体的主键，课程号是课程的主键。



图2-1 学生实体及属性

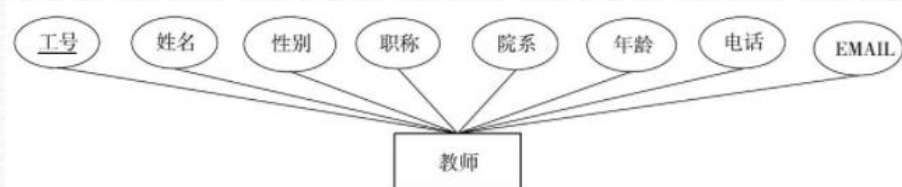


图2-2 教师实体及属性

2.2、设计分 E-R 图

教学管理系统共涉及三个实体集：学生、教师和课程，三者之间均存在联系。

学生与教师：一个教师可以教授多名学生，每个学生可以学习多个教师的课程。所以，学生与教师之间是通过教学进行联系的，并且二者之间的关系是多对多的联系。

学生与课程：一个学生可以学习多门课程，一门课程可以供多个学生学习。因此，学生与课程之间是多对多的联系。学生学习一门课程会有一个成绩。

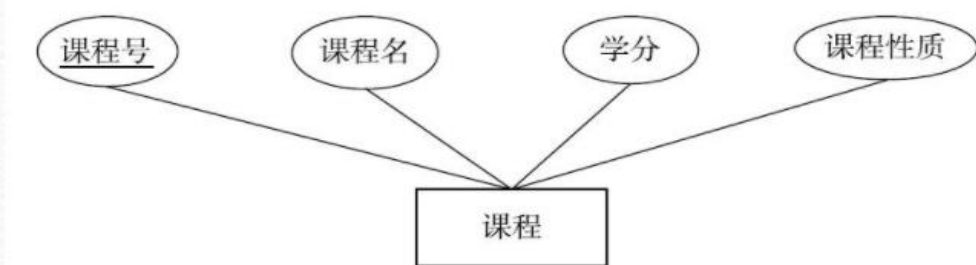
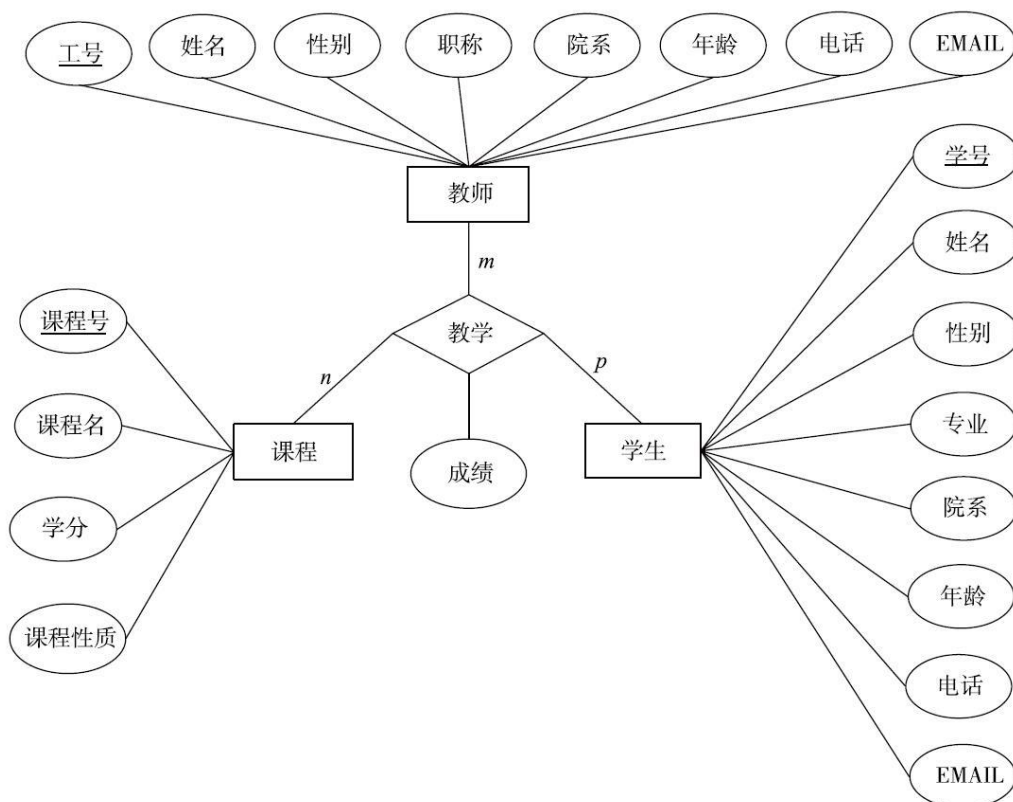


图2-3 课程实体及属性

教师与课程：一个教师可以教授多门课程，一门课程同时也可以被多名教师教授。因此，教师与课程之间的关系是多对多的联系。

2.3、合并各分 E-R 图，生成初步 E-R 图

2.4、全局 E-R 图



二、创建逻辑模型

3、逻辑结构设计

3.1、将 E-R 图转换成关系模型

学生 (学号, 姓名, 性别, 专业, 院系, 年龄, 电话, EMAIL) 为学生实体对应的关系模式, 其中学号是学生关系的主键。

教师 (工号, 姓名, 性别, 院系, 年龄, 职称, 电话, EMAIL) 为教师实体对应的关系模式, 其中工号是教师关系的主键。

课程 (课程号, 课程名, 学分, 课程性质) 为课程实体对应的关系模式, 其中课程号是课程关系的主键。

教学 (学号, 工号, 课程号, 成绩) 为联系 “教学” 对应的关系模式。因为教学是学生、课程和教师之间的多对多联系, 因此学生、教师和课程的主属性, 以及教学联系本身的属性 “成绩”, 共同构成了教学关系模式的属性, 其中学号、工号及课程号的组合是教学关系的主键。

三、创建物理模型

4、数据库物理设计与实施

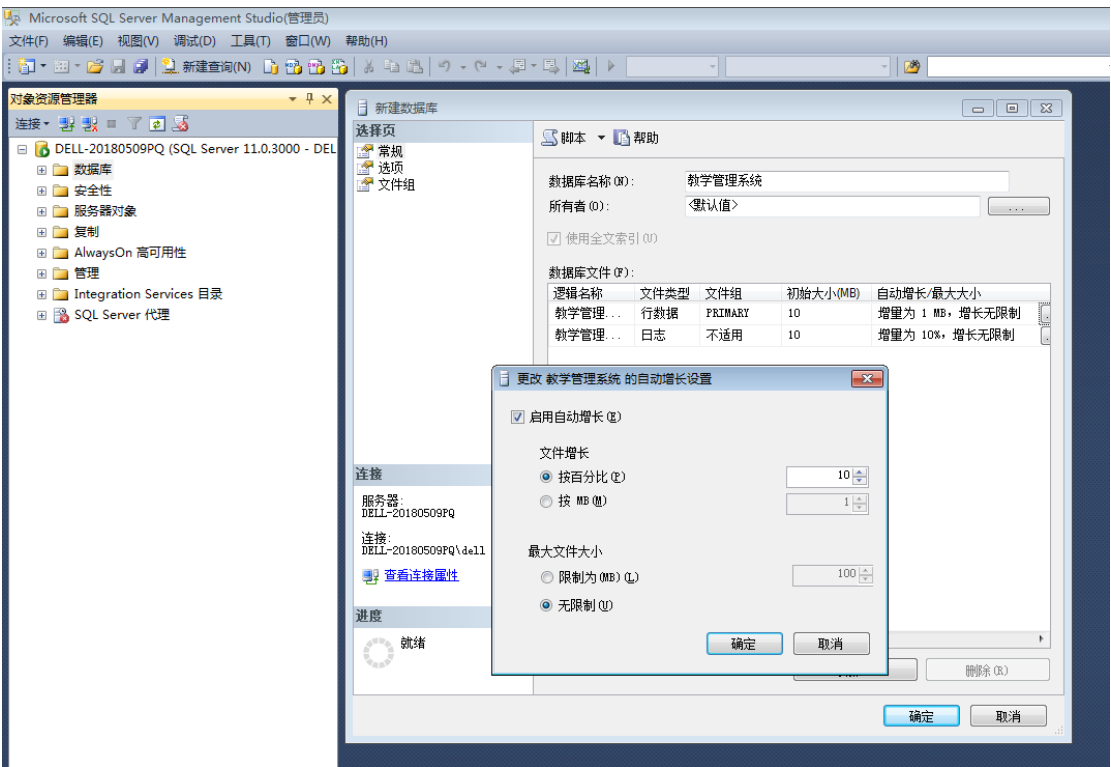
4.1、创建 “教学管理系统” 数据库

4.1.1、启动 SQL server

打开资源管理器 SSMS

4.1.2、创建数据库“教学管理系统”

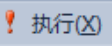
右键数据库》新建数据库》更改初始大小和增长方式》确定



4.1.3、（方法二）使用 SQL 建立数据库

打开资源管理器》单击新建查询》输入：

```
CREATE DATABASE
ON PRIMARY
(
NAME="教学管理系统_data",
FILENAME=" E:\SQL\DatawareNIE\MSSQL11.MSSQLSERVER\MSSQL\DATA\教学管理系统.mdf"
SIZE=10MB,
FILEGROWTH=10%
)
Log ON
(
NAME="教学管理系统_log",
FILENAME=" E:\SQL\DatawareNIE\MSSQL11.MSSQLSERVER\MSSQL\DATA\ 教 学 管 理 系 统_log.ldf",
SIZE=10MB,
MAXSIZE=100MB
FILEGROWTH=10%
)
```

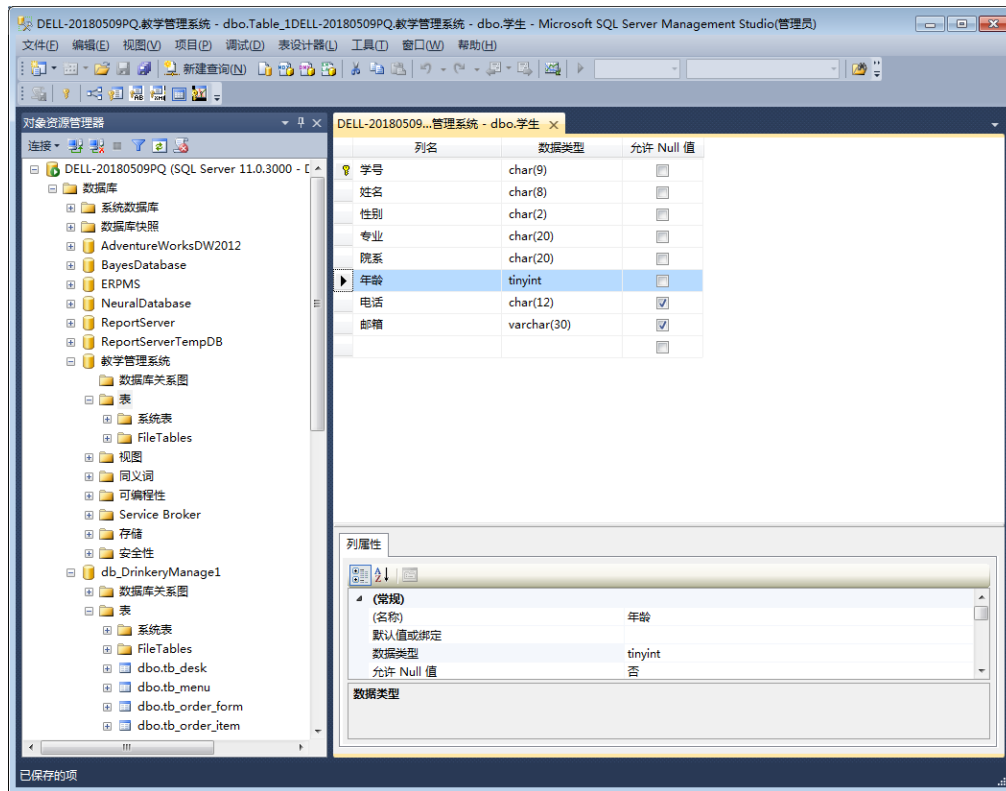
单击执行按钮  执行(X)，同样的效果。

4.2、建立和管理基本表

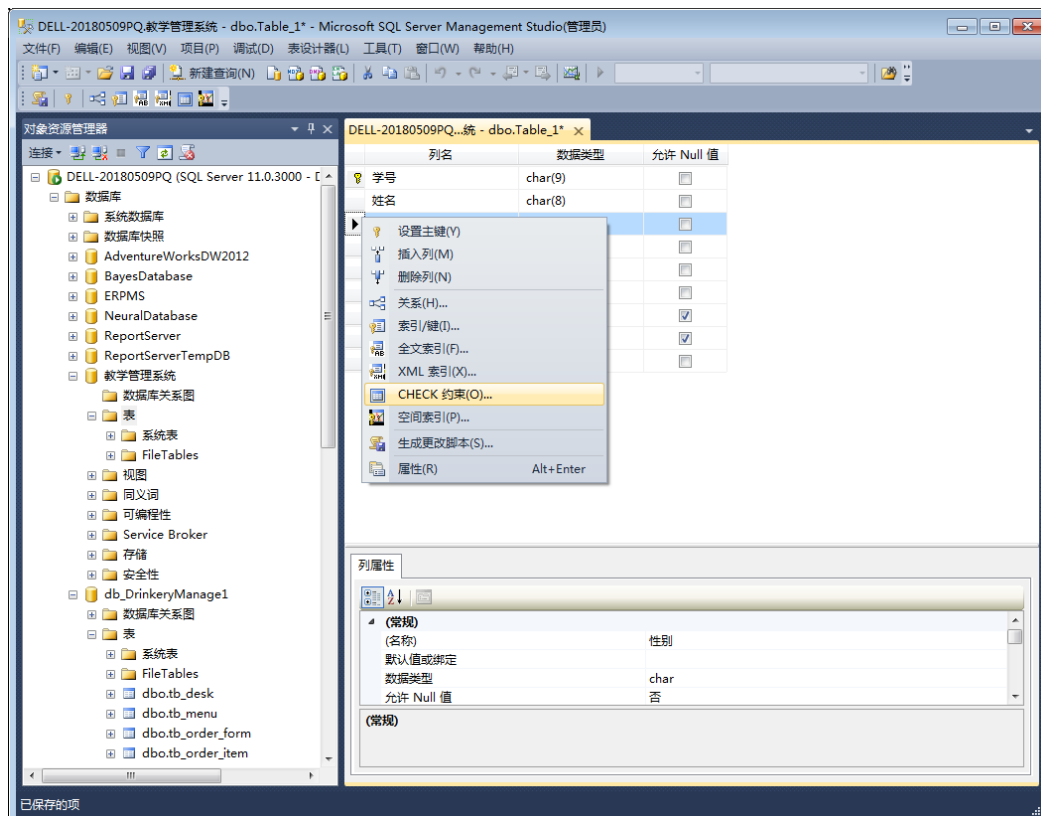
4.2.1、建立基本表

数据库建立学生、教师、课程和教学 4 张基本表。方式有两种：SQL 语句和 SSMS 图形工具建表。

i. 建立学生表



设置约束条件



CHECK 约束

选定的 CHECK 约束(S):

CK_学生_年龄
CK_学生_性别*

正在编辑新的 CHECK 约束 的属性。 需要先填充 “表达式” 属性, 然后才能接受新的 CHECK 约束。

(常规)	
表达式	([性别]='女' OR [性别]='男')
标识	
(名称)	CK_学生_性别
说明	
表设计器	
强制用于 INSERT 和 UPDAT	是
强制用于复制	是
在创建或重新启用时检查现有	是

添加(A) 删除(D) 关闭(C)

CHECK 约束

选定的 CHECK 约束(S):

CK_学生_年龄
CK_学生_性别*

正在编辑现有 CHECK 约束 的属性。

(常规)	
表达式	([年龄]>(1) AND [年龄]<(100))
标识	
(名称)	CK_学生_年龄
说明	
表设计器	
强制用于 INSERT 和 UPDAT	是
强制用于复制	是
在创建或重新启用时检查现有	是

添加(A) 删除(D) 关闭(C)

- 利用 SQL 语句在查询分析器中建表

```
CREATE TABLE 学生
(
    学号 CHAR(9) NOT NULL PRIMARY KEY,
    姓名 CHAR(8) NOT NULL,
    性别 CHAR(2) NOT NULL CHECK (性别 IN ('男','女')),
    专业 CHAR(20) NOT NULL,
    院系 CHAR(20) NOT NULL,
    年龄 TINYINT NOT NULL CHECK (年龄 BETWEEN 1 AND 100),
    电话 CHAR(12),
    EMAIL VARCHAR(30))
```

单击执行按钮。

ii. 建立教师表

教师表的属性信息见表2-2所示。

表2-2 教师基本表的属性信息

属性列	数据类型	是否为空 / 约束条件
工号	CHAR(5)	主键
姓名	CHAR(8)	否
性别	CHAR(2)	“男”，“女”
院系	CHAR(20)	否
年龄	TINYINT	在 1 ~ 100 之间取值
职称	CHAR(10)	“讲师”，“教授”，“副教授”
电话	CHAR(12)	是
EMAIL	VARCHAR(30)	是

创建教师表的 SQL 语句：

```
CREATE TABLE 教师
(
    工号 CHAR(5) NOT NULL PRIMARY KEY,
    姓名 CHAR(8) NOT NULL,
    性别 CHAR(2) NOT NULL CHECK (性别 IN ('男','女')),
    院系 CHAR(20) NOT NULL,
    年龄 TINYINT NOT NULL CHECK (年龄 BETWEEN 1 AND 100),
    职称 CHAR(10) NOT NULL CHECK (职称 IN ('讲师','教授','副教授')),
    电话 CHAR(12),
    EMAIL VARCHAR(30))
```

iii. 建立课程表

课程基本表的属性信息见表2-3所示。

表2-3 课程基本表的属性信息

属性列	数据类型	是否为空 / 约束条件
课程号	CHAR(7)	主键
课程名	CHAR(20)	否
学分	TINYINT	在 1 ~ 10 取值
课程性质	CHAR(4)	‘必修’,‘选修’

建立课程表的 SQL 语句:

```
CREATE TABLE 课程
(
    课程号 CHAR(7) NOT NULL PRIMARY KEY,
    课程名 CHAR(20) NOT NULL,
    学分 TINYINT NOT NULL CHECK (学分 BETWEEN 1 AND 10),
    课程性质 CHAR(4) NOT NULL CHECK( 课程性质 IN('必修','选修'))
)
```

iv. 建立教学表

教学基本表的属性信息见表2-4所示。

表2-4 教学基本表的属性信息

属性列	数据类型	是否为空 / 约束条件
学号	CHAR(9)	否
工号	CHAR(5)	否
课程号	CHAR(7)	否
成绩	NUMERIC(3)	允许为空, 如果不为空, 那么值应在 0 ~ 100 之间

建立教学表 SQL 语句:

```
CREATE TABLE 教学
(
    学号 CHAR(9) NOT NULL,
    工号 CHAR(5) NOT NULL,
    课程号 CHAR(7) NOT NULL,
    成绩 NUMERIC(3) CHECK (成绩 BETWEEN 0 AND 100)
    CONSTRAINT Student_Course_fkxh FOREIGN KEY (学号)
        REFERENCES 学生(学号),
    CONSTRAINT Student_Course_fkkch FOREIGN KEY (课程号)
        REFERENCES 课程(课程号),
    CONSTRAINT Student_Course_fkgh FOREIGN KEY (工号)
        REFERENCES 教师(工号),
    CONSTRAINT Student_Course_pk PRIMARY KEY (学号, 工号, 课程号))
```

注意: 在建立“教学”表时, 应该分别对“教学”表的属性“学号”“课程号”和“工号”增加外键约束。教学表的这三个属性信息分别来源于学生表、教师表和课程表中的相应字段, 如果没有学生表中“学号”、课程表中“课程号”或教师表中“工号”的属性值, 那么为教学表增加数据是不符合逻辑的, 违反了数据的参照完整性约束条件。新建表时可以利用子句“CONSTRAINT 约束名称 FOREIGN KEY (属性名) REFERENCES 表名 (属性名)”建立外键约束。

四、SQL 管理基本表

4.2.2、管理基本表

i. 向“学生”表中增加“入学时间”属性列, 数据类型为日期型 DATE

SQL 语句: ALTER TABLE * ADD * 语句

```
ALTER TABLE 学生 ADD 入学时间 DATE;
```


或者直接设计修改。

- ii. 将“学生”表中的“专业”属性列的数据类型改为字符型：VARCHAR(20).

SQL 语句：ALTER TABLE * ALTER COLUMN * *语句

```
ALTER TABLE 学生 ALTER COLUMN 专业 VARCHAR(20)
```

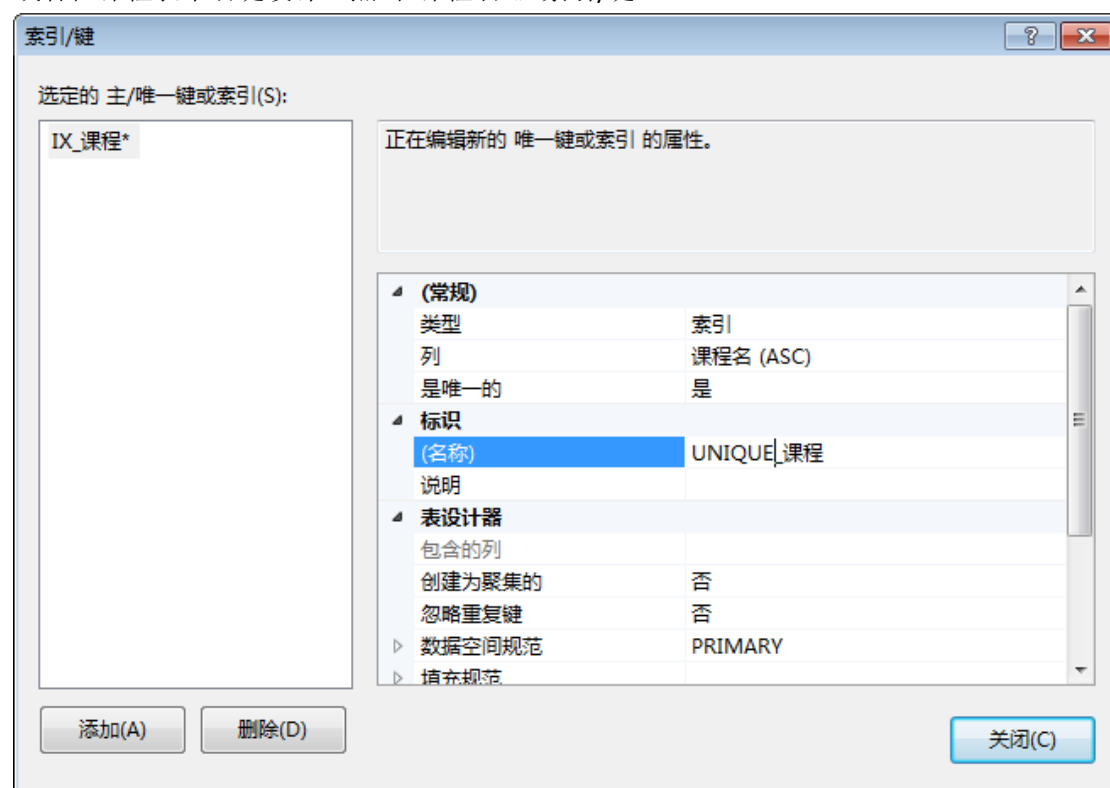
或者直接设计修改。

- iii. 增加课程名称必须取唯一值约束条件

SQL 语句：ALTER TABLE * ADD * 语句

```
ALTER TABLE 课程 ADD UNIQUE(课程名)
```

或者在课程表中右键设计，点击“课程名”》索引/键



- iv. 删除“学生”表中的“入学时间”列

SQL 语句：ALTER TABLE* DROP COLUMN*

```
ALTER TABLE 学生 DROP COLUMN 入学时间
```

- v. 删除“学生表”

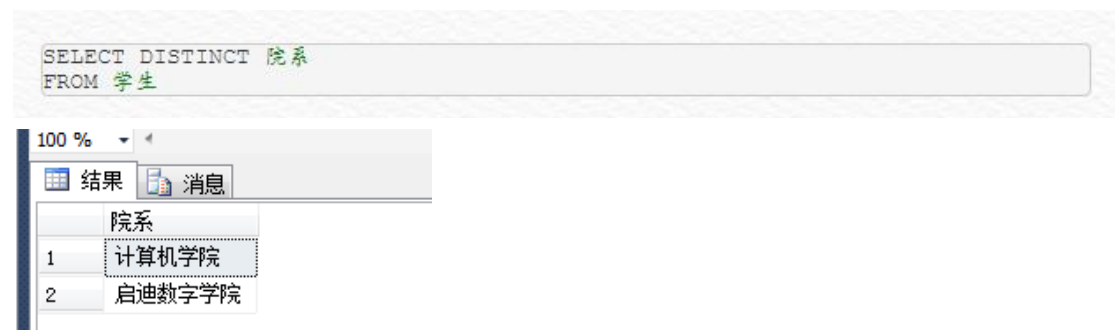
SQL 语句：DROP TABLE* CASCADE

```
DROP TABLE 学生 CASCADE
```

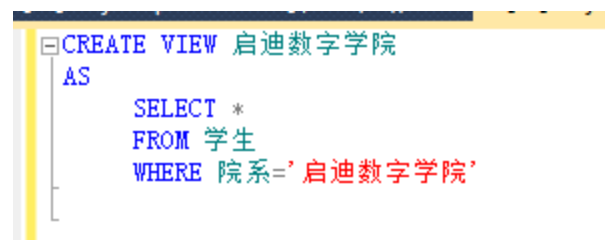
4.2.3、建立和管理视图

- i. 为每个院系分别建立一个学生视图

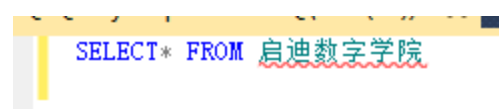
SQL 语句:



SQL 语句: 为启迪数字学院的学生建立视图



视图建好后, 新建查询窗口《输入查询语句



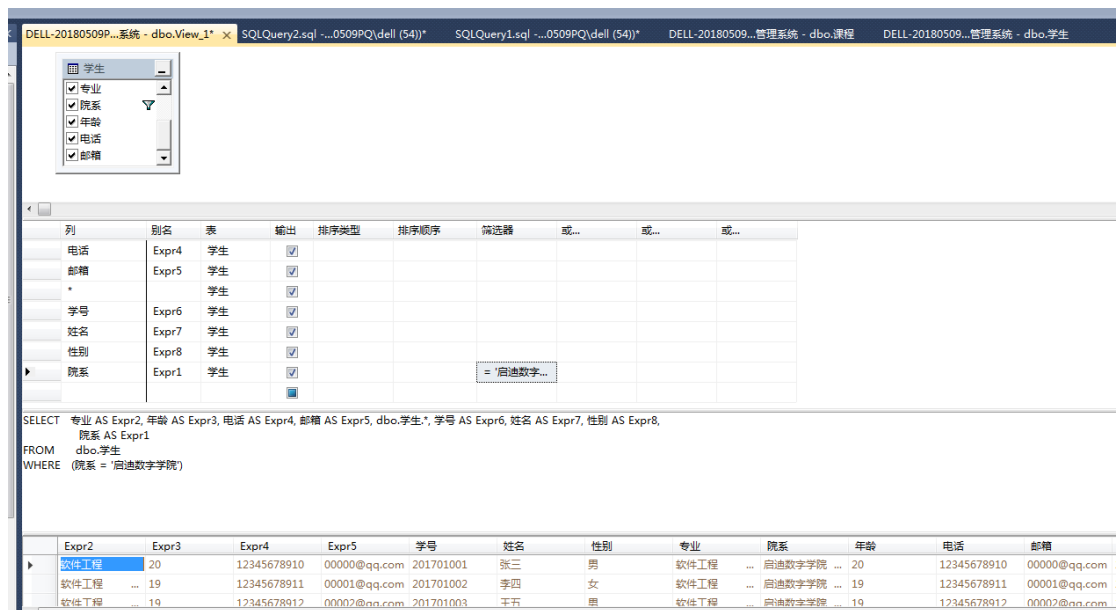
The screenshot shows a SQL query window with the following text:

```
SELECT * FROM 启迪数字学院
```

Below the query window, the results are displayed in a table:

	学号	姓名	性别	专业	院系	年龄	电话	邮箱
1	201701001	张三	男	软件工程	启迪数字学院	20	12345678910	00000@qq.com
2	201701002	李四	女	软件工程	启迪数字学院	19	12345678911	00001@qq.com
3	201701003	王五	男	软件工程	启迪数字学院	19	12345678912	00002@qq.com

第二种方式: 单击视图》右键视图》新建视图》在院系一行中的筛选器输入启迪数字学院》执行!



4.2.4、建立管理索引

- i. 在“教学”表中的列“学号”、“课程号”上建立索引

SQL 语句:

```
CREATE INDEX SC_XH_KCH
ON 教学(学号, 课程号)
```

或者在设计窗口中，右键》索引/键》添加》输入名称：SC_XH_KCH,选中“学号”、“课程号”



4.3、访问数据库（增删改查）

4.3.1、数据查询

- i. 简单查询

SQLQuery3.sql -...0509PQ\ dell (54))* x DELL-20180509...管理系统 - dbo.教学 DELL-20180509

```
SELECT *
FROM 学生
WHERE 学号='201701001'
```

100 %

结果 消息

	学号	姓名	性别	专业	院系	年龄	电话	邮箱
1	201701001	张三	男	软件工程	启迪数字学院	20	12345678910	00000@qq.com

SQLQuery3.sql -...0509PQ\ dell (54))* x DELL-20180509...管理系统 - dbo.教学 DELL-20180509

```
SELECT *
FROM 教师
WHERE 院系='启迪数字学院' AND 职称='讲师'
```

100 %

结果 消息

	工号	姓名	性别	院系	年龄	职称	电话	邮箱
1	03407	韩战豪	男	启迪数字学院	38	讲师	123456789100	NULL
2	03507	王磊	男	启迪数字学院	27	讲师	123456789109	11112@qq.com
3	03605	满天星	男	启迪数字学院	33	讲师	123456789108	11111@qq.com

SQLQuery3.sql -...0509PQ\ dell (54))* x DELL-20180509...管理系统 - dbo.教学 DELL-20180509...

```
SELECT 学生.学号, 学生.姓名, 教学.课程号, 课程.课程名
FROM 学生, 教学, 课程
WHERE 学生.学号=教学.学号 AND 教学.课程号=课程.课程号 AND 学生.姓名='张三'
```

100 %

结果 消息

	学号	姓名	课程号	课程名
1	201701001	张三	QD001	C语言

ii. 列出王磊老师所教授的所有信息

利用嵌套查询：

```
SELECT 课程.课程号, 课程.学分, 课程.课程性质, 课程.课程名, 教师.姓名 AS 任课教师
FROM 课程, 教师
WHERE 课程号 IN (SELECT DISTINCT 课程号
FROM 教学 WHERE 工号=(SELECT 工号
FROM 教师 WHERE 姓名='王磊'))
AND 教师.工号=(SELECT 工号 FROM 教师 WHERE 姓名='王磊')
```

100 %

结果 消息

	课程号	学分	课程性质	课程名	任课教师
1	QD001	3	必修	C语言	王磊

利用连接查询

```
SELECT 课程.课程号, 课程.学分, 课程.课程性质, 课程.课程名, 教师.姓名 AS 任课教师
FROM 课程, 教师, 教学
WHERE 教学.课程号=课程.课程号 AND 教学.工号=教师.工号 AND 教师.姓名= '王磊'
```

100 %

结果 消息

	课程号	学分	课程性质	课程名	任课教师
1	QD001	3	必修	C语言	王磊
2	QD001	3	必修	C语言	王磊

4.3.2、数据更新

通过编辑：选中当前表格》右键编辑前 200 行》即可

新增 SQL 语句：

```
INSERT INTO 课程 (课程号, 课程名, 学分, 课程性质)
VALUES ('ZH32012', '概率统计', '4', '必修')
```

修改 SQL 语句：

```
UPDATE 课程
SET 课程名='大学英语 (上)', 学分='3'
WHERE 课程名='大学英语'
```

另外，如果不知道相关的信息（学生“赵奇”的学号、教师“秦岚”的工号及课程“数据库原理”的课程号），可以利用下面的SQL语句，将信息查询到之后再行更新。

```
UPDATE 教学
SET 成绩='85'
WHERE 学号=(SELECT 学号 FROM 学生 WHERE 姓名='赵奇')
AND 工号=(SELECT 工号 FROM 教师 WHERE 姓名='秦岚')
AND 课程号=(SELECT 课程号 FROM 课程 WHERE 课程名='数据库原理')
```

```
UPDATE 教学
SET 成绩=成绩+10
WHERE 工号= (SELECT 工号 FROM 教师 WHERE 姓名='孟鑫')
AND 课程号= (SELECT 课程号 FROM 课程 WHERE 课程名='数学简史')
```

4.4、数据库维护

4.4.1、备份数据库

4.4.2、维护计划

4.4.3、创建触发器

- i. 输入学生年龄，若年龄<18，则自动修改为 18

```
CREATE TRIGGER Insert_Or_Update_学生
ON 学生
AFTER INSERT,UPDATE /*触发事件是插入或更新操作*/
AS /*定义触发动作*/
    UPDATE 学生
        SET 年龄=18
        FROM 学生 ,Inserted i
        WHERE 学号 =i.学号 AND i.年龄<18
```

- ii. 当删除学生学号时，自动删除“教学”中的学生信息

```
CREATE TRIGGER DELETE_学生
ON 学生
AFTER DELETE
AS
    DELETE FROM 教学
    WHERE 学号IN (SELECT 学号 FROM Deleted )
```

五、C#连接 SQL server

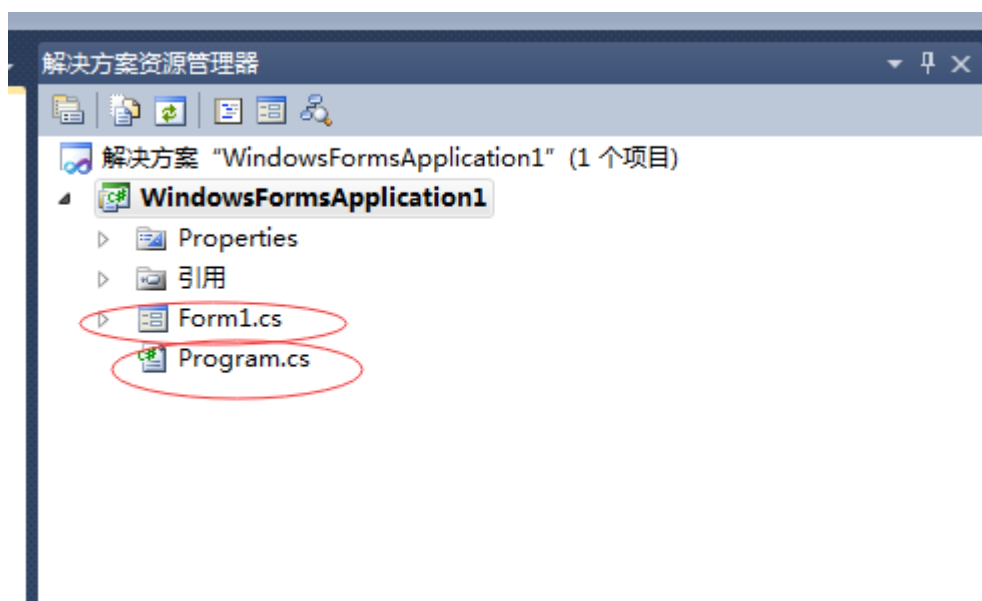
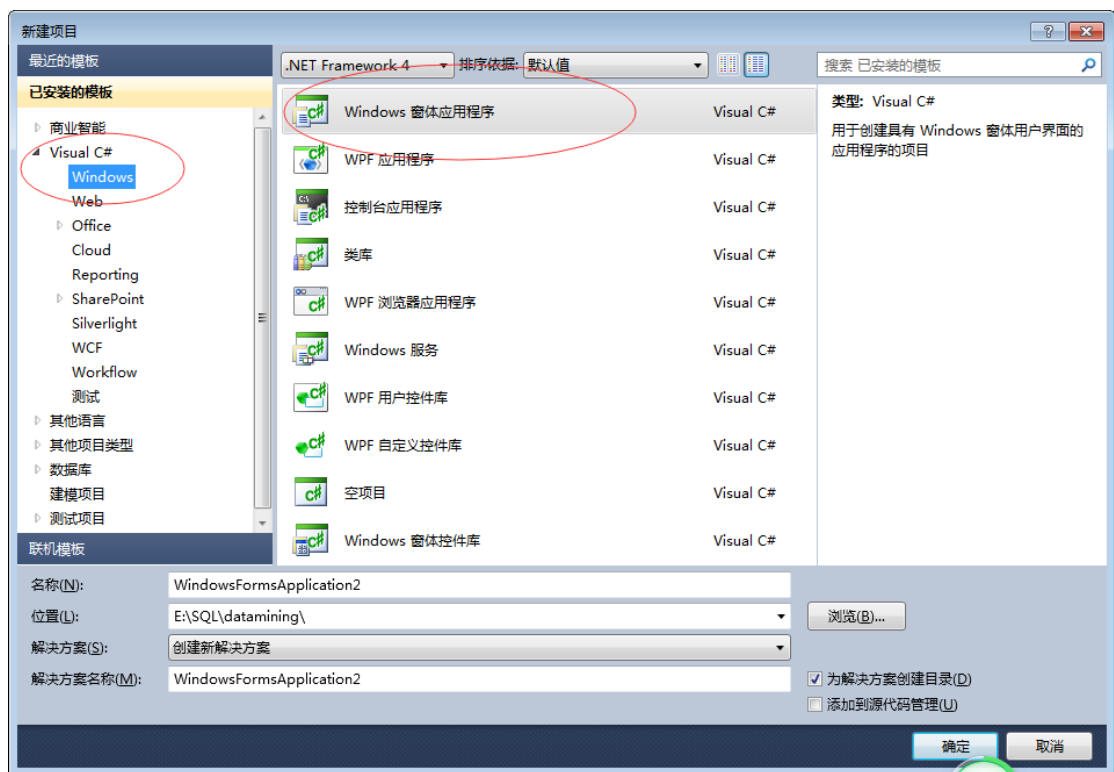
应用工具：visual studio 2010 官方旗舰版、SQL server2012

语言：C#

一、连接数据库

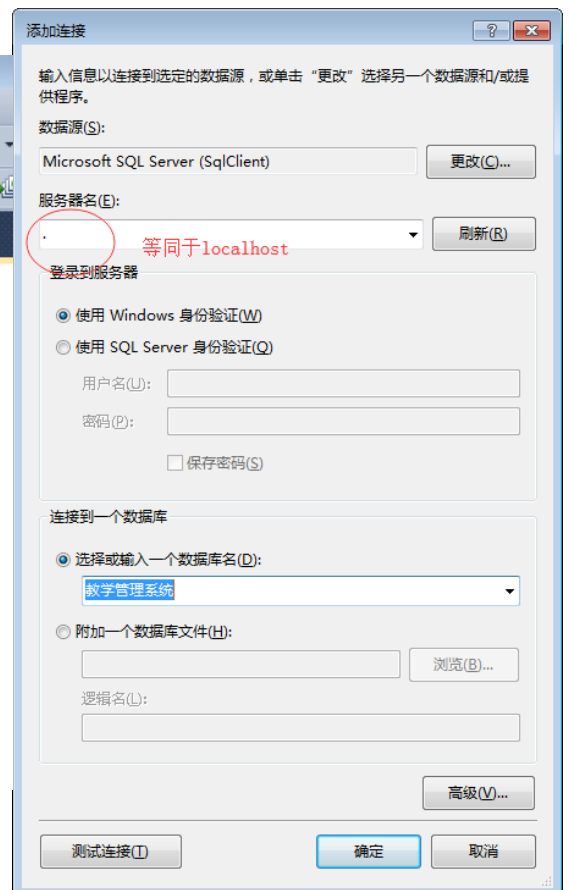
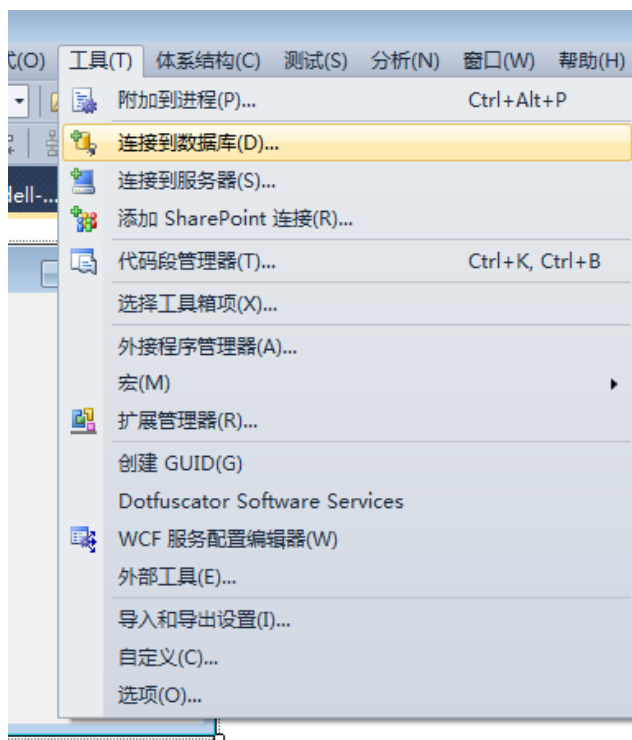
1、创建项目

创建 Windows 窗体应用程序

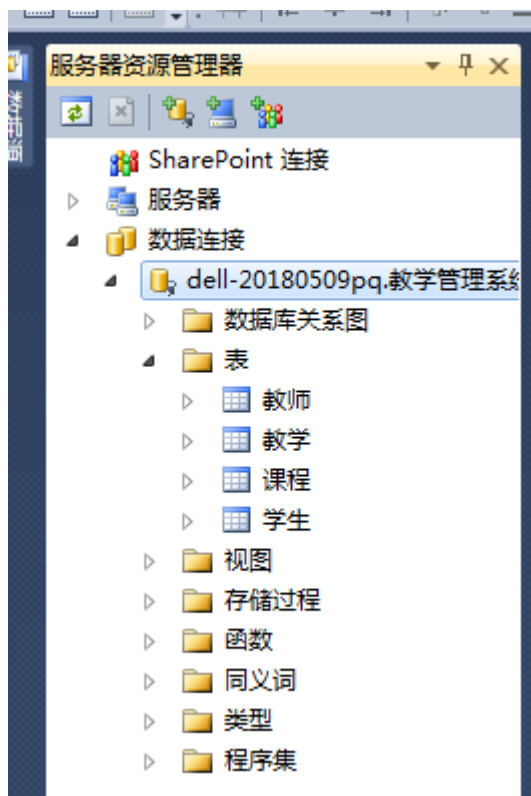


2、连接数据库

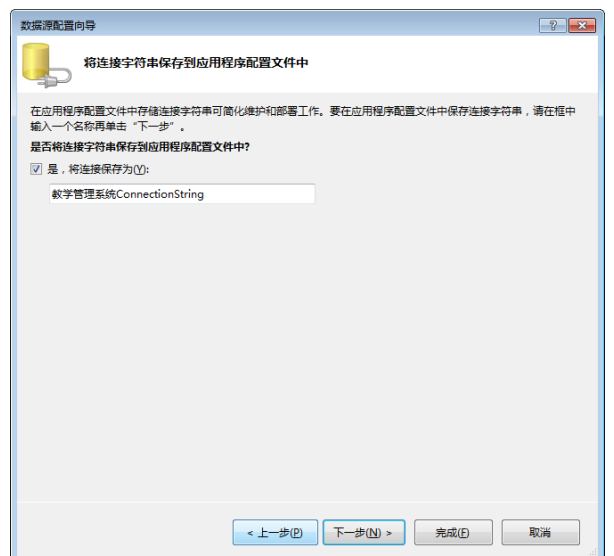
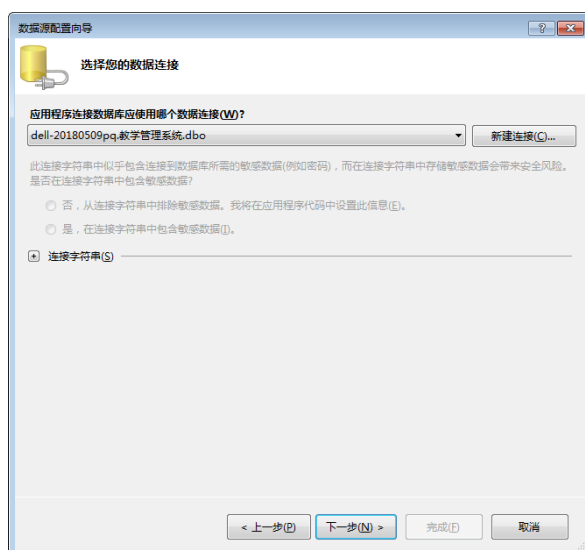
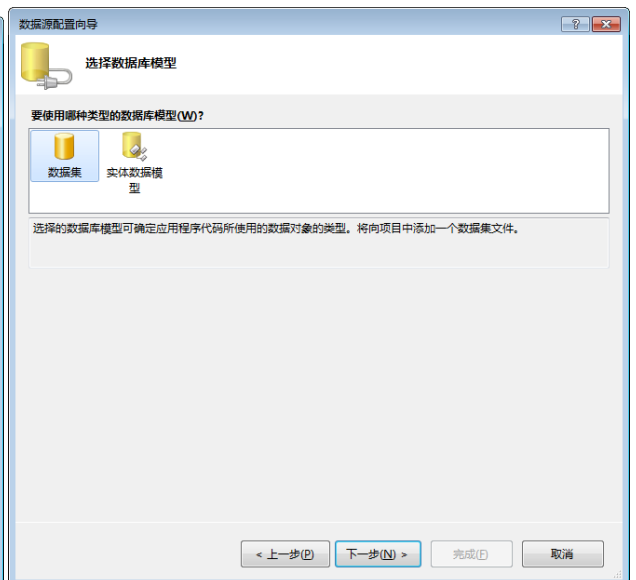
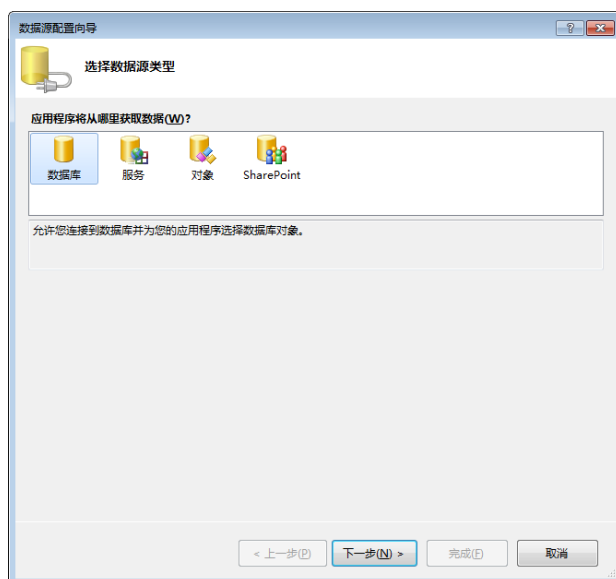
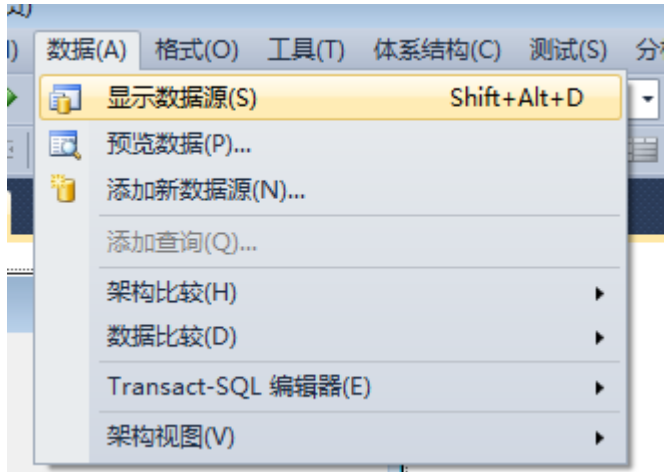
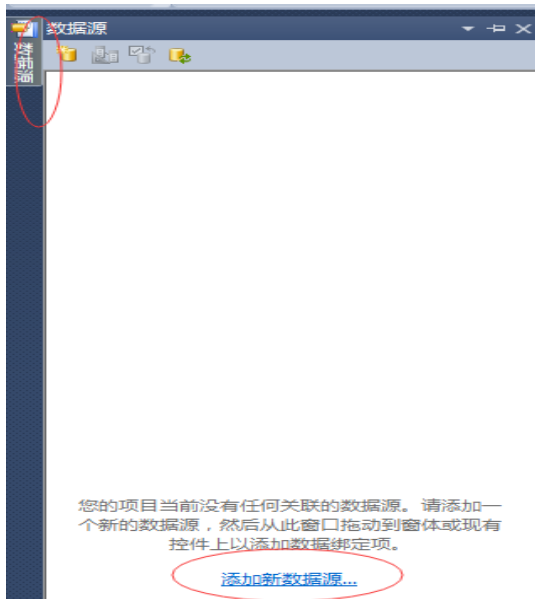
单击工具》连接到数据库

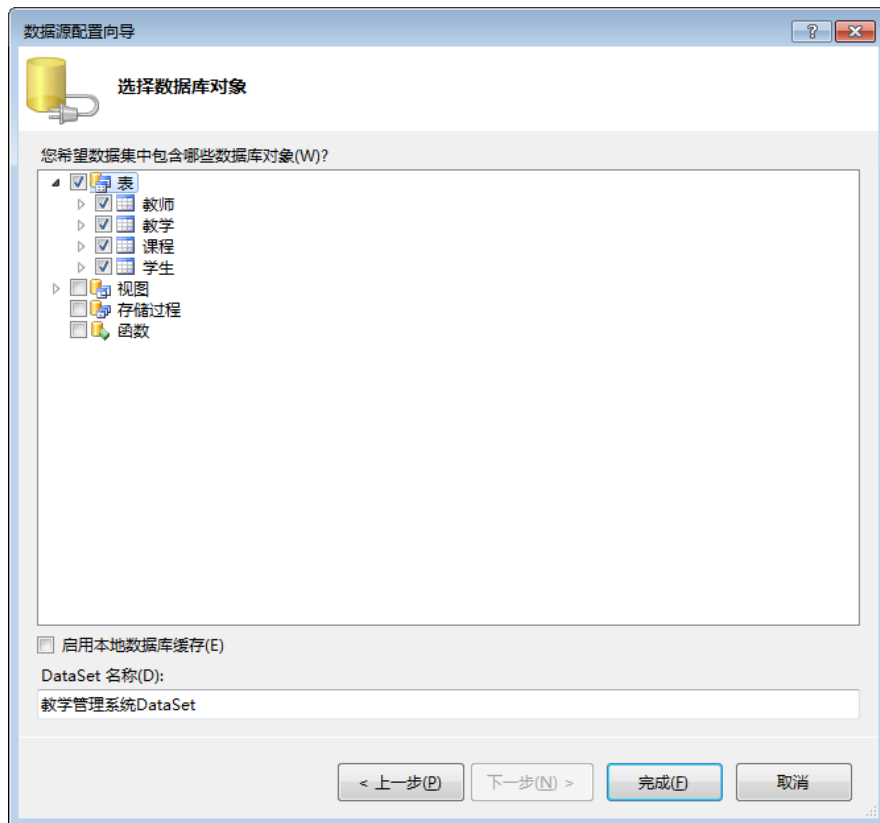


连接后的视图:



2.1、创建数据源



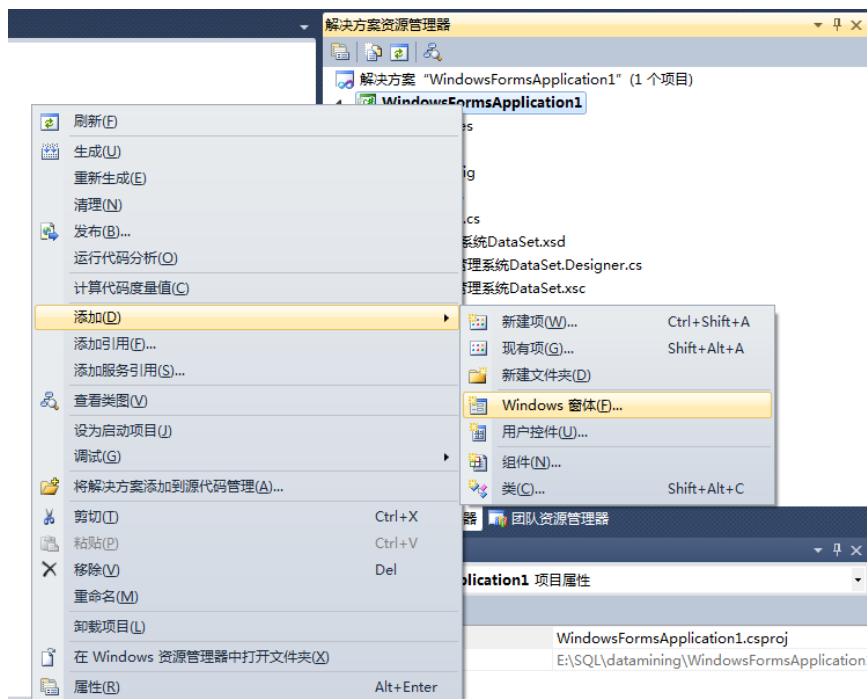


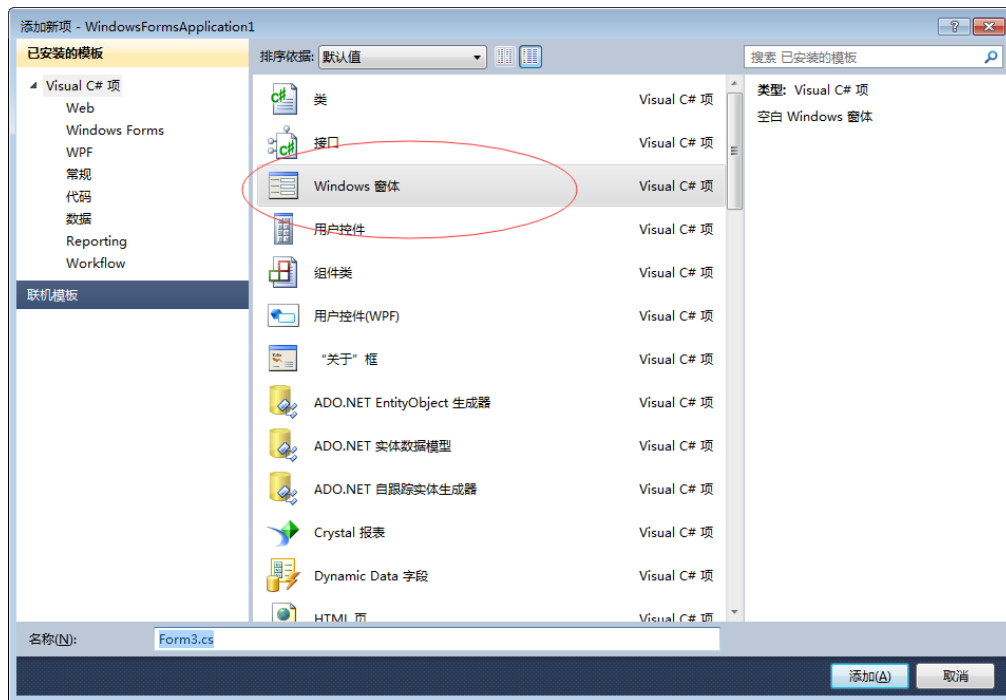
完成

3、设计窗体

因为要实现增删改查，**对于每个表格我们需要三个窗体**
具体步骤如下：

新建窗体方法：右键项目》添加》Windows 窗体》完成





Form1 代码:

引用部分

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Data.SqlClient;
//这几个命名空间是一定要有的;
```

开始主程序

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

更新模块

每次删除、修改、添加后若表格没有及时变化，可更新一下

```

private void button1_Click(object sender, EventArgs e) //按键一实现更新功能
{
    Form1_Load(sender, e);
}

private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        SqlConnection conn = new SqlConnection(); //SQL server数据库的一个打开链接
        string str = "Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True";//连接数据库
        conn.ConnectionString = str;
        conn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText = "select * from 学生";
        cmd.Connection = conn;
        SqlDataAdapter sda = new SqlDataAdapter(cmd);//重载，填充新实例
        DataSet ds = new DataSet();
        sda.Fill(ds, "学生");
        this.dataGridView1.DataSource = ds;
        this.dataGridView1.DataMember = "学生";
        this.dataGridView1.AutoGenerateColumns = true;
        for (int i = 1; i < this.dataGridView1.ColumnCount; i++)
        {
            this.dataGridView1.Columns[i].DefaultCellStyle.SelectionBackColor = Color.White;//设置选中颜色
            this.dataGridView1.Columns[i].DefaultCellStyle.SelectionForeColor = Color.Black;
            this.dataGridView1.Columns[i].ReadOnly = true;
        }
        conn.Close();
    }
    catch (Exception ee)
    {
        MessageBox.Show("错误：" + ee.Message, "错误");
    }
}
}

```

删除

```

private void button2_Click(object sender, EventArgs e)//删除
{
    if (this.dataGridView1.SelectedCells == null)
    {
        MessageBox.Show("请选择要删除的项！", "错误", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if (this.dataGridView1.CurrentCell.ColumnIndex == 0)
        {
            string st = this.dataGridView1[1, this.dataGridView1.CurrentCell.RowIndex].Value.ToString();//获取当前活动状态的单元格
            SqlConnection conn = new SqlConnection();
            string str = "Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True";//这部分其实没必要每次都重连，可以单独写一个连接函数
            conn.ConnectionString = str;
            conn.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "delete from 学生 where 姓名='" + st + "'";//对选中的行执行SQL删除语句
            cmd.Connection = conn;
            cmd.ExecuteNonQuery();
            conn.Close();
        }
    }
}

```

查询


```

private void button3_Click(object sender, EventArgs e)//查询
{
    if (this.textBox1.Text.Trim() == "")
    {
        MessageBox.Show("请输入要查询的姓名!", "提示");
    }
    else
    {
        SqlConnection conn = new SqlConnection();
        string str = "Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True";
        conn.ConnectionString = str;
        conn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText = "select * from 学生 where 姓名='" + this.textBox1.Text.Trim() + "'";//按姓名查询
        cmd.CommandText = "select * from 学生 where 学号='" + this.textBox1.Text.Trim() + "'";//按学号查询
        cmd.CommandText = "select * from 学生 where 专业='" + this.textBox1.Text.Trim() + "'";//按专业查询
        cmd.CommandText = "select * from 学生 where 院系='" + this.textBox1.Text.Trim() + "'";//按院系查询
        cmd.Connection = conn;
        SqlDataAdapter sda = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        sda.Fill(ds);
        this.dataGridView1.DataSource = ds;//获取数据源
        this.dataGridView1.DataMember = ds.Tables[0].ToString();//显示数据列表
        this.dataGridView1.AutoGenerateColumns = true;
        conn.Close();
    }
}

```

添加和修改

```

private void button4_Click(object sender, EventArgs e)//添加
{
    Form2 f = new Form2();//跳转添加窗体
    f.Show();
}

private void button5_Click(object sender, EventArgs e)//修改
{
    Form3 f = new Form3();//跳转修改窗体
    f.id = Convert.ToInt32(this.dataGridView1[0, this.dataGridView1.CurrentRow.Index].Value.ToString());//转换数据类型
    f.Show();
}

private void Form1_Load_1(object sender, EventArgs e)
{
    // TODO: 这行代码将数据加载到表“教学管理系统DataSet.学生”中。您可以根据需要移动或删除它。
    this.学生TableAdapter.Fill(this.教学管理系统DataSet.学生);
}

```

Form1 窗体设计:

学号	姓名	性别	专业	院系	年龄	电话	邮箱
*							

更新 删除 添加 修改

请输入您要查询的信息 查询

Form2 代码

引用部分

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading.Tasks;
using System.Data.SqlClient;
//这几个命名空间是一定要有的;

namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
    }
}
```

约束性别为男女

```
private void Form2_Load(object sender, EventArgs e)
{
    comboBox1.Items.Add("男");
    comboBox1.Items.Add("女");
    comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;
}
//... 其他代码
```

确定之前需判断输入是否符合约定

```
if (textBox1.Text.Trim() == "" || textBox2.Text.Trim() == "" || textBox3.Text.Trim() == "" ||
    textBox4.Text.Trim() == "" || textBox5.Text.Trim() == "" || comboBox1.Text.Trim() == "")
{
    //判断是否为空
}
```

```

private void button1_Click(object sender, EventArgs e) //确定键
{
    if (textBox1.Text.Trim() == "" || textBox2.Text.Trim() == "" || textBox3.Text.Trim() == "" || textBox4.Text.Trim() == "" || textBox5.Text.Trim() == "" || textBox6.Text.Trim() == "" || textBox7.Text.Trim() == "")
    {
        MessageBox.Show("请输入完整信息!", "错误", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        SqlConnection conn = new SqlConnection();
        string str = "Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True";
        conn.ConnectionString = str;
        conn.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText = "insert into 学生(学号, 姓名, 性别, 专业, 院系, 年龄, 电话, 邮箱)values(@学号,@姓名,@性别,@专业,@院系,@年龄,@电话,@邮箱)"; //添加
        cmd.Connection = conn;
        cmd.Parameters.Add("@学号", SqlDbType.Char, 9); //
        cmd.Parameters.Add("@姓名", SqlDbType.Char, 8); //
        cmd.Parameters.Add("@性别", SqlDbType.Char, 2); //
        cmd.Parameters.Add("@专业", SqlDbType.Char, 20); //
        cmd.Parameters.Add("@院系", SqlDbType.Char, 20); //约定输入数据类型
        cmd.Parameters.Add("@年龄", SqlDbType.Int); //
        cmd.Parameters.Add("@电话", SqlDbType.Char, 12); //
        cmd.Parameters.Add("@邮箱", SqlDbType.VarChar, 30); //
        cmd.Parameters["@学号"].Value = textBox1.Text.Trim(); //
        cmd.Parameters["@姓名"].Value = textBox2.Text.Trim(); //
        cmd.Parameters["@性别"].Value = comboBox1.Text.Trim(); //
        cmd.Parameters["@专业"].Value = textBox3.Text.Trim(); //约定输入位置
        cmd.Parameters["@院系"].Value = textBox4.Text.Trim(); //
        cmd.Parameters["@年龄"].Value = textBox5.Text.Trim(); //
        cmd.Parameters["@电话"].Value = textBox6.Text.Trim(); //
        cmd.Parameters["@邮箱"].Value = textBox7.Text.Trim(); //
        cmd.ExecuteNonQuery();
        conn.Close();
        this.Close();
    }
}

```

取消键，返回窗格 1

```

private void button2_Click(object sender, EventArgs e) //取消键
{
    this.Close();
}

```

Form2窗体设计：

The screenshot shows a Windows Form titled "Form2". It contains a vertical stack of labels and input controls:

- 学号 (Student ID): Text box
- 姓名 (Name): Text box
- 性别 (Gender): Dropdown menu
- 专业 (Major): Text box
- 院系 (Department): Text box
- 年龄 (Age): Text box
- 电话 (Phone): Text box
- 邮箱 (Email): Text box

At the bottom of the form, there are two buttons: "确定" (OK) on the left and "取消" (Cancel) on the right.

Form3代码:

学号为主键不可修改, 按相同学号进行修改信息的存储

```
public int id;

private void Form3_Load(object sender, EventArgs e)
{
    TextBox1.Text = id.ToString();
    SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True");
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    conn.Open();
    cmd.CommandText = "select * from 学生 where 学号=" + id.ToString(); //对相同的学号进行修改存储
    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    sda.Fill(ds);
    if (ds.Tables[0].Rows.Count == 1)
    {
        this.textBox2.Text = (ds.Tables[0].Rows[0]["姓名"].ToString());
        this.comboBox1.Text = (ds.Tables[0].Rows[0]["性别"].ToString());
        this.textBox3.Text = (ds.Tables[0].Rows[0]["专业"].ToString());
        this.textBox4.Text = (ds.Tables[0].Rows[0]["院系"].ToString());
        this.textBox5.Text = (ds.Tables[0].Rows[0]["年龄"].ToString());
        this.textBox6.Text = (ds.Tables[0].Rows[0]["电话"].ToString());
        this.textBox7.Text = (ds.Tables[0].Rows[0]["邮箱"].ToString());
        conn.Close();
    }
}
```

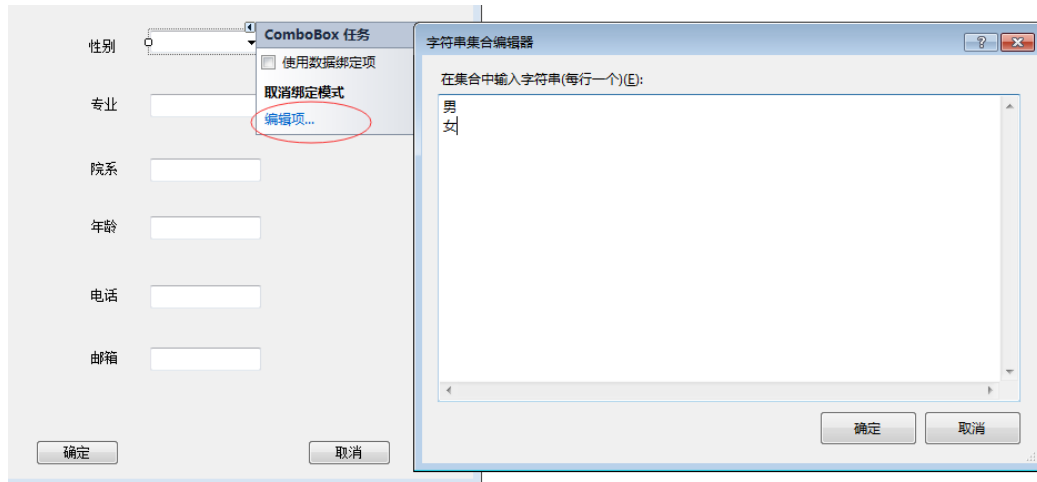
除了更新SQL语句和Form2的添加SQL语句不一样, 其他和Form2的确认代码一样

```
cmd.CommandText = "update 学生 set 学号=@学号, 姓名=@姓名, 性别=@性别, 专业=@专业, 院系=@院系, 年龄=@年龄, 电话=@电话, 邮箱=@邮箱 where 学号=@学号";
```

Form3窗体设计:

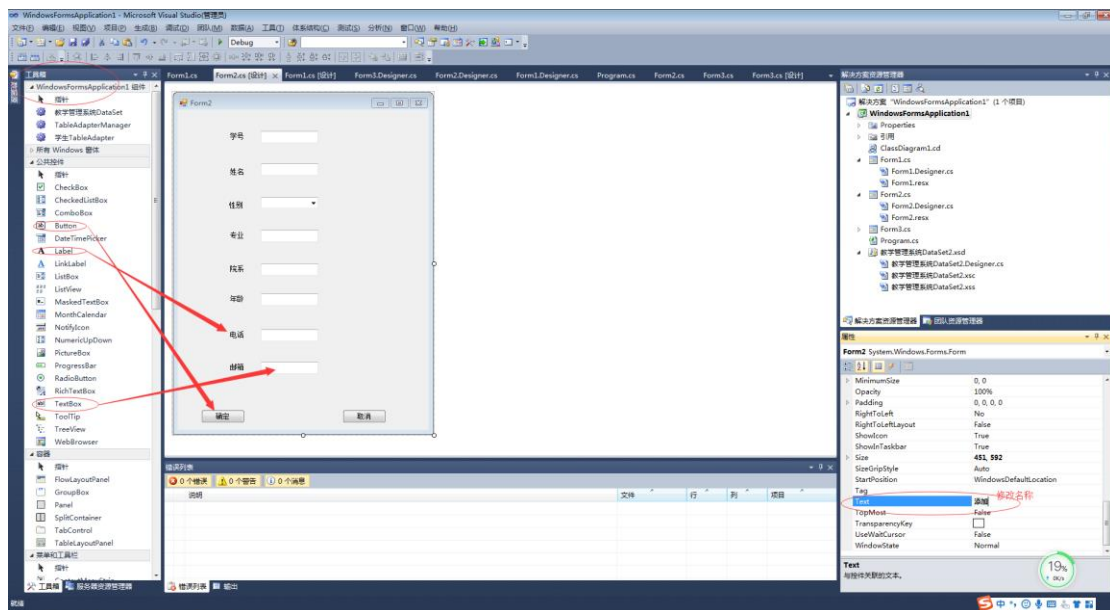
六、其他小问题

ComboBox 选项插入



窗体设计参考: https://blog.csdn.net/qq_36482772/article/details/78008531

比如: 将工具箱中的工具拖拽到窗口里即可



C#控件[1]DataGridView 控件 (一) 参考:

<https://jingyan.baidu.com/article/1709ad80a318c44635c4f041.html>

其他的 C#连接方式: <https://blog.csdn.net/www1501766557/article/details/89677358>

运行之后

Form1

	学号	姓名	性别	专业	院系	年龄	电话	邮箱
▶	201701001	张三	男	智能科学 ...	启迪数字学院 ...	20		
	201701002	李四	女	软件工程 ...	启迪数字学院 ...	19	12345678911	00001@qq.com
	201701003	王五	男	软件工程 ...	启迪数字学院 ...	19	12345678912	00002@qq.com
	201701004	步惊云	女	数据挖掘 ...	启迪 ...	33		
	201701005	肖雄	女	统计 ...	启迪 ...	19		
	201702001	张一	男	大数据 ...	计算机学院 ...	20	12345678913	00003@qq.com
	201702002	李一	女	大数据 ...	计算机学院 ...	18	12345678914	00004@qq.com
*								

更新 删除 添加 修改

请输入您要查询的信息 查询

添加

学号

姓名

性别

专业

院系

年龄

电话

邮箱

确定 取消

修改

学号

姓名

性别

专业

院系

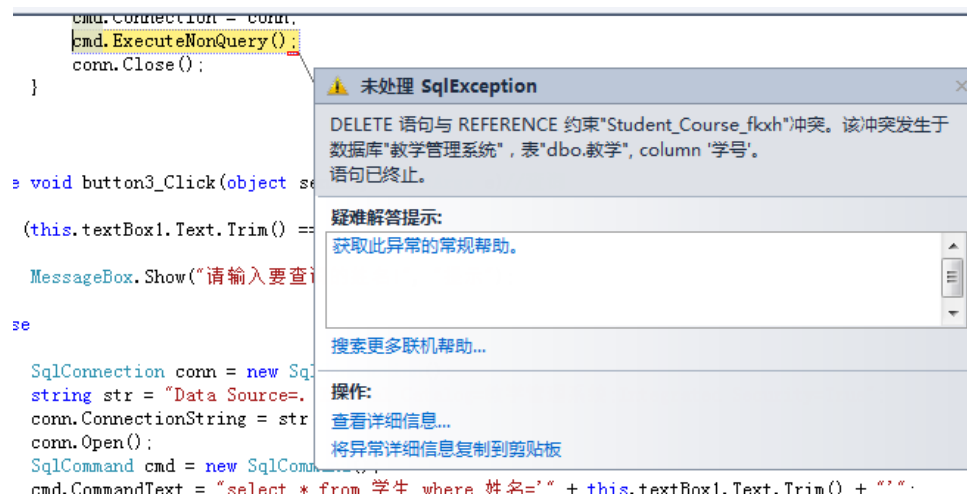
年龄

电话

邮箱

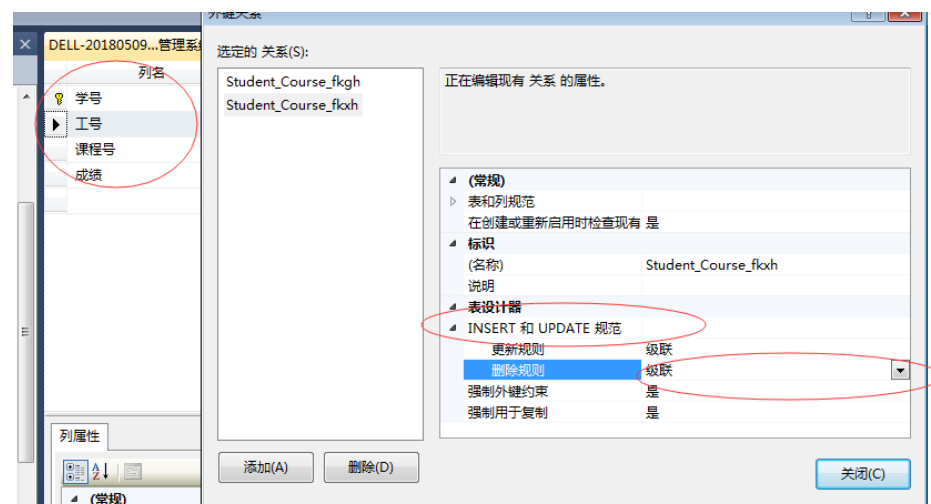
确定 取消

其中，删除出现了问题



解决方法 删除主键表记录时,你可以在建外键时选定外键记录一起级联删除还是拒绝删除。

单个数据源连接测试是不成功的!



注: "Data Source=.;Initial Catalog=教学管理系统;Integrated Security=True"

. 为服务器名, 等同于 localhost

教学管理系统 对应的数据库名称

实践报告要求是多个表格的增删改查, 以上只是一个表格的示范, 加油!