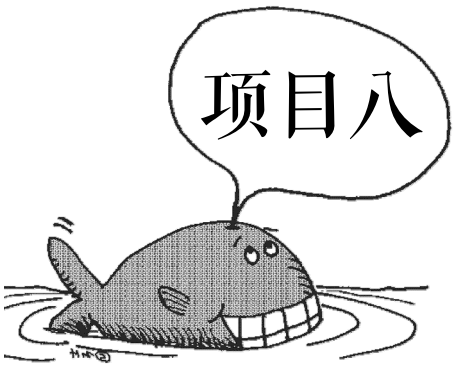

校园网管理系统的构建

项目背景

本项目要求设计一个完整的校园网管理系统，该系统是一个现代校园网管理系统的雏形。该校园网管理系统前台用 ASP.NET 设计，后台用 SQL Server 2008 数据库，将前面所设计数据库知识全面融合到系统中，完成校园网管理系统的功能。

项目分析

本项目完成一个具体的校园网管理系统。通过本项目的完成，要求读者对数据库作全面的认识，具备数据库开发的能力。

项目目标

【知识目标】

1．全面掌握系统开发的流程及步骤；

2．学会运用所学知识开发系统；

3．综合运用相关开发工具开发系统。

【能力目标】

1．具备设计前台的能力；

2．掌握后台数据库的设计和管理；

3．具备数据库管理的能力。

【情感目标】

1．培养良好的抗压能力；

2．培养沟通的能力并通过沟通获取关键信息；
3．培养团队的合作精神；
4．培养实现客户利益最大化的理念；
5．培养事物发展是渐进增长的认知。

任务一　数据库系统的设计

任务说明

设计一个系统，首先要考虑到其功能的完整性，其次考虑其延展性。一个好的系统结构是非常清晰的，每个模板都有一些独立的功能，各模板组合起来又能完成更加复杂的功能，所以设计好系统结构是非常重要的。

在此，校园网管理系统中有两类用户，分别是管理员和普通用户。管理员的操作主要包括学生管理、教师管理、课程管理、班级管理、选课管理和成绩管理等功能；普通用户的对象主要是学生，具体操作包括修改密码、课程信息查询、选课、课程查询和成绩查询等功能。

模块图中的基本模块的功能可以具体描述出来，如图 8.1.1 所示。

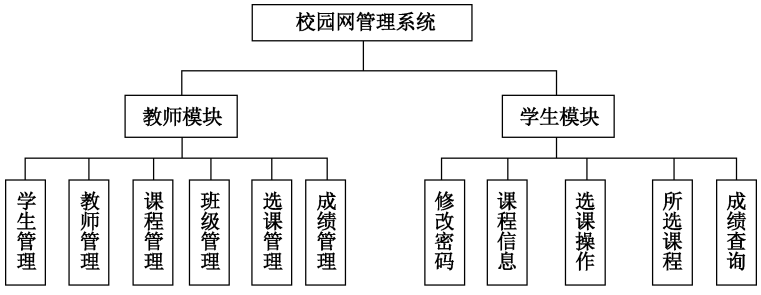

图 8.1.1　系统模块图

1．教师模块

该模块主要由 6 个子模块构成，主要负责学生、教师、课程和班级等相关信息的管理。

（1）学生管理：负责管理所有在校注册学生的个人信息，其主要功能包括添加、删除、修改和查找学生信息。每个学生有唯一的学号，管理员添加新生后，新生即可登录此系统浏览个人信息，登录此系统的用户名和密码默认是此学生的学号。

（2）教师管理：负责管理系统管理员的信息。其主要功能是将本校教师的权限设为管理员。管理员可添加新教师信息，每个教师有唯一的编号，之后通过把教师设为管理员，使此教师拥有管理员的权限，从而使此教师可登录系统进行管理员的相关操作。

（3）课程管理：负责管理所有的课程的信息。其主要功能包括添加、删除、修改和查找课程信息。只有管理员才具有对班级管理信息进行维护的权限。课程管理模块是选课管理模块的基础，只有在课程管理中添加课程的信息，学生才能进行选课。

（4）班级管理：负责班级的管理。其主要功能包括添加、删除和修改班级信息，以及对班级信息的查询。只有管理员具有对班级管理信息进行维护的权限。学生信息的添加建立在班级信息维护的基础上，每个学生必须属于特定的班级。当管理员对学生成绩进行统计时，可以统

计各个班级的平均分、最高分等。

（5）选课管理：负责选课的管理。其主要功能包括删除、统计学生选课信息。它以课程管理模块中维护好的信息作为基础，既可以对选修课程进行管理，又可以统计选修人数，也可以对超过选课规定人数进行删除。

（6）成绩管理：学生选修的每一门课程都要有成绩，查询的内容包括课程名称、某位学生的成绩等。只有管理员能录入学生每一门课程的成绩，并可以进行修改，也可以计算某个班级的某门课程的最高分，平均分，计算优秀和不及格人数等。学生只能查询自己所修课程的成绩。

2．学生模块

学生只能进入此模块，该模块主要有 5 方面的功能。可操作个人相关信息，如修改个人的登录密码、浏览相关的课程信息、进行选课操作、查看自己已经选修的课程、查询自己的成绩等。

任务分析

根据前面设计的系统功能模块结构，本任务要设计若干数据表，要求尽量减少数据冗余。可以在系统中创建 9 个表，除学生、班级、教师、课程等基本表外，为了便于系统管理员管理，还设计了用户表，记录用户登录系统时的用户名、密码和权限。此外，可以在过程中创建临时的数据表，这样更有利于系统的实现。

实施步骤 START

第 1 步：使用 Microsoft SQL Server 2008 建立数据库，数据库名为 xywglxt。

首先是表 users，用于存储校园网管理系统中所有参与人员的信息，包括管理员登录信息、学生登录信息，这样做的目的是方便系统判断用户登录的类型，以及对用户类型的统一管理。用户表中主要包括用户名、用户密码和用户类型，具体定义如表 8.1.1 所示。

表 8.1.1　users

字 段 名	类　型	约　束	备　注
user_is	varchar（20）	主键	用户名
user_Password	varchar（20）		用户密码
user_Power	int（4）		用户类型

本系统中最重要的对象是学生，表 student 就是用于存储所有学生信息的，具体定义如表 8.1.2 所示。

表 8.1.2　student

字 段 名	类　型	约　束	备　注
sno	char（10）	主键	学号
sname	char（10）	非空	姓名
ssex	char（2）	只取男、女	性别
sbirthday	datetime（8）		出生日期
sscore	numeric（18，0）		入学成绩
classno	char（8）	与表 class 中的 classno 外键关联	班级编号

学生所在班级信息相对独立，系统用 class 记录所有班级信息，具体定义如表 8.1.3 所示。

表 8.1.3 class

字 段 名	类 型	约 束	备 注
classno	char（8）	主键	班级编号
classname	char（10）	非空	班级名称
pno	char（4）	与 professional 中 pno 外键关联	专业编号

构建 teacher 来存储本校所有教师信息，教师表给出一个较为简单的结构，具体定义如表 8.1.4 所示。

表 8.1.4 teacher

字 段 名	类 型	约 束	备 注
tno	char（4）	主键	教师编号
tname	char（10）	非空	教师姓名
tsex	char（2）	只取男、女	性别
tbirthday	datetime（8）		出生日期
ttitle	char（10）		职称

每一个教师教授的课程存储在 teaching。具体定义如表 8.1.5 所示。

表 8.1.5 teaching

字 段 名	类 型	约 束	备 注
tno	char（4）	主键，与 teacher 中 tno 外键关联，级联删除	教师编号
cno	char（7）	主键，与 course 中 cno 外键关联	课程编号

设计了 course，用于存储本校所有课程信息，其中包括课程名称和学分，具体定义如表 8.1.6 所示。

表 8.1.6 course

字 段 名	类 型	约 束	备 注
cno	char（7）	主键	课程编号
cname	char（30）	非空	课程名称
credits	real（4）	非空	学分

学生所学课程都会有成绩，并且每个学生每一门课只有一个成绩。系统设计了 choice，用于存储本校所有学生所学课程信息，具体定义如表 8.1.7 所示。

表 8.1.7 choice

字 段 名	类 型	约 束	备 注
sno	char（10）	主键，与 student 中 sno 外键关联，级联删除	学分
cno	chasr（7）	主键，与 course 中 cno 外键关联	课程编号
grade	real（4）		成绩

学生所属专业情况记录在 professional 中，具体定义如表 8.1.8 所示。

表 8.1.8　professional

字　段　名	类　　型	约　　束	备　　注
pno	char（4）	主键	专业编号
pname	char（30）	非空	专业名称
deptname	char（2）	与 department 中 deptno 外键关联	系部编号

专业所在系部情况记录在 department 中，具体定义如表 8.1.9 所示。

表 8.1.9　department

字　段　名	类　　型	约　　束	备　　注
deptno	char（2）	主键	系部编号
deptname	char（20）	非空	系部名称

第 2 步：利用存储过程，可以完成一些较为综合的功能。

（1）SELECT 存储过程的创建。

下面是具体的程序代码。

```
CREATE PROCEDYRE [selest - student - 1]
 (@sno [varchar] (50) )
AS
Select *
From student
Where Sno=@Sno
```

该存储过程用于从 student 中查询特定的学生信息，具体内容包括学生的学号、姓名、性别、出生日期、入学成绩等信息。存储过程中涉及表中各字段的含义都已描述过。在本系统中，很多情况下需要判断学生信息的有效性，即此学生是否已注册，调用此存储过程可方便地根据学号判断学生信息的有效性。此存储过程还可在学生浏览个人信息时使用，调用它将快速地返回学生的基本信息。

（2）INSERT 存储过程的创建。

下面是具体的程序代码。

```
CREATE PRPCEDURE [INSERT_student_1]
 (@Sno [char] (10) ,
 (@Snama [char] (10) ,
 (@Ssex [char] (2) ,
 (@Sbirthday[datetime],
 (@Classno[char] (8) )
AS INSERT INTO [Student_Class].[do].[student]
 ( [sno]
[sname],
[ssex],
[sbirthday],
[sscore],
[classno])
```

```
VALUES
 ( @Student id,
@Sname,
@Ssex,
@Sbirthday,
@Sscore,
@Classno
)
```

通过该存储过程向 student 中添加新的学生基本信息，具体内容包括学号、姓名、性别、出生年月、入学成绩等。该存储过程在系统注册学生信息时被调用，每个学生有唯一的学号，在添加信息时，输入的学号要保证唯一性，否则系统会提示出错。

实操练习

1．创建向 course 中添加新课程信息的存储过程：insert_course_1。
2．创建向 class 中添加班级信息的存储过程：insert_class_1。
3．创建更新 student 中特定的学生信息情况的存储过程：update_student_1。

任务二　首页与管理员页面代码的编写

任务说明

本任务主要使读者掌握控件的使用方法，掌握数据库连接的一般方法，掌握判断用户登录的一般方法，掌握 DataGrid 等数据控件的使用方法，理解 DataSet 的作用和原理，掌握数据绑定的方法，掌握账务数据库编程在程序中的应用等。

任务分析

要完成本任务，主要实现以下操作。
（1）主页面（登录）代码编写。
（2）管理员操作模块中的学生信息管理主页面代码编写。
（3）管理员操作模块中的课程信息管理主页面代码编写。
（4）管理员操作模块中的成绩信息管理主页面代码编写。
（5）管理员操作模块中的学生选课管理主页面代码编写。

实施步骤　START

第 1 步：主页面（登录）代码编写。

编写如图 8.2.1 所示的系统登录页面，做好页面静态设计和控件设计，并要求登录有权利限制。登录页面通过下拉菜单进行用户识别，不同用户登录将根据其不同的身份进入不同的功能页面，系统用户包括管理员和学生，在用户身份验证通过后，系统利用较方便的 GET 方式将用户名和用户身份等信息存储在临时变量中，再分别进入管理员操作模块和学生操作模块，并伴随用户对系统进行操作的整个生命周期。

图 8.2.1　登录界面

以下给出学生课程管理系统首页（default.aspx.cs）的后台支持类主要代码，前台脚本代码（default.aspx）可以通过使用 .NET 集成开发环境，依照所给界面设计方案来完成。

登录页面主要代码如下。

在“登录”按钮 Buttonl_Click 的单击事件下进行编码。

```
       Protected void Buttonl_Click（object sender，EventArgse）
       SqlConnection con = new
       Sqlconnection
（sytem.Configuration.configuratioManager.AppSettings["dsn"].ToString()）；
       //创建连接数据库的字符串，具体连接放在Web.Config文件中
       Con.Open()；//打开连接
       If（this.DropDownListl.SelectedItem.Value.Equals（"l"））//判断登录的用户类型
       {
       SqlCommand com = new SqlCommand（"select count（*）from student where
sname='"+this.
       TexBoxl.text + "and sno='"+this.TextBox2.Text +'"，con）；//查找数据库中是否含
有此记录
       Int n = Convert.Tolnt32（com.ExecuteScalar()）；
       If（n>0）//利用返回记录的个数来判断是否存在，若存在，则转入相应的功能页面
       {
       Response.Redirect（"studentcheck.aspx?
S_na="+this.TexBoxl.Text+"&s_no="+this.TexBox2.Text）；
       }
       Else
       {
       This.Labell.Text="输入学生用户名或密码错误！"；
       }
       }
       Else
       {
       Sqlcommand com = new Sqlcommand（"select count（*）from user where User_id
='"+this.TextBoxl.Text + '"and User_password='"+this.TextBox2.Text+"'"，.con）；
       Int n = convert Tolnt32（com.ExecuteScalar()）；
       If（n>0）
       {
```

```
        Response.Redirect（"admin-student.aspx?no="+this.TextBoxl.Text
+"&psw="+this.TextBox2.Text）;
        }
        Else
        {
        This.Labell.text="您输入的管理用户名或者密码错误"
        }
        }
        Con.Close();
        }
```

登录页面中利用了 ADO.NET 的一些数据库连接对象，如 SqlConnection（创建数据库），SqlCommand（获取操作命令），SqlDataReader（读取记录）等。一般利用 ADO.NET 中的这些对象来获取和修改数据库中的数据。

第 2 步：管理员操作模块中的学生信息管理页面代码编写。

学生信息管理页面如图 8.2.2 所示，其所属的学生信息管理模块是学生课程管理系统中管理学生学籍的部分。学生信息管理页面主要负责所有学生信息的浏览，以及到其他管理页面的链接，页面采用 DataGrid 控件的管理 student 与 DataSet 数据集的绑定并返回所有学生信息，分页显示，可以对学生信息进行添加、修改、查找或删除操作。

学号	姓名	性别	出生日期	入学成绩	班级		
c14F1701	刘备	男	1988/6/4	123	电子商务	编辑	删除
c14F1702	杨贵妃	女	1987/6/10	234	电子商务	编辑	删除
c14F1703	张飞	男	1989/2/11	345	电子商务	编辑	删除
c14F1704	关羽	男	1988/2/16	456	电子商务	编辑	删除
c14F1705	赵龙	男	1987/1/23	567	电子商务	编辑	删除
c14F1706	啊芬	男	1987/1/28	555	电子商务	编辑	删除
c14F1707	杨博斯	男	1987/2/2	124	电子商务	编辑	删除
c14F1708	小王	男	1987/2/7	456	电子商务	编辑	删除
c14F1301	李艳	男	1987/1/24	345	计算机网络技术	编辑	删除
c14F1302	杨海艳	男	1987/1/29	342	计算机网络技术	编辑	删除
c14F1303	秦齐忠	男	1987/2/3	345	计算机网络技术	编辑	删除
c14F1401	月梅	男	1987/2/4	432	计算机应用技术	编辑	删除
c14F1402	可春	男	1987/1/25	340	计算机应用技术	编辑	删除
c14F1403	刘芬	男	1987/1/30	356	计算机应用技术	编辑	删除
c14F1501	杨延华	男	1987/1/26	333	计算机维修	编辑	删除
1 2							

图 8.2.2　学生信息管理页面

此页面中，“查询学生”按钮的 Click 事件把 Panel 的 Visible 属性重设为 True，以显示输入查询条件的表格。根据提示，用户输入查询条件时，“确定”按钮的 Click 事件通过生成 SQL 语句实现查询功能，查询的结果最终显示在 DataGrid 控件 Dgd_student 中，在该控件中设置了“编辑”和“删除”列，提供数据的修改、删除操作。在“显示所有信息”控件的 Click 事件 Btn_all_Click()中，完成 DataGrid 控件 Dgd_student 的数据绑定操作，使其显示所有学生的信息。同时，令容纳查询条件的 Panel 控件的 Visible 属性设为 False，因为此时系统部接收查询条件，只有当触发“查询学生”按钮的 Click 事件后，才能重新显示查询条件。

学生信息管理页面的后台支持类（student.aspx.cs）的主要代码如下。

在页面载入事件中进行数据绑定。

```
        Protected void Page_Load（object sender, EventArgs e）
```

```
        {
        if (! IsPostBack)
        {
        SqlConnection con = new SqlConnection (System. Configuration.
ConfigurationManager.AppSettings["dsn"].ToString()) ;
        con.Open();
        SqlCommand com = new SqlCommand ("select s.sno, s.sname, s.ssex, s.sbirthday,
s.sscore, c.classname from student s left outer join class c on s.classno=c.classno",
con) ;
        SqIDataAdapter sda=new SqIDataAdapter();
        Sda.SelectCommand=com;
        DataSet ds = new DataSet();
        Sda.Fill (ds, "t1") ;
        this.stu_dg1.DataKeyField = "sno";//要添加此语句，才可以查找控件
        this.stu_dg1.DataSource = ds.Tables["t1"].DefaultView;
        this.stu_dg1.DataBind();
        con.Close();
        this.Panel1.Visible = false;//存放在PSOTBACK中，表示第一次执行有效
        this.Panel2.Visible = false;//
        }//DataGrid中的数据要用样式表固定
        }
```

“添加新生”按钮单击事件的代码如下。

```
        Protected boid Buttonl_Click (object sender,EventArgs e)
         {
        this.Panel2.Visible = false;
        this.Panel1.Visible = true;
        }
```

“编辑”记录事件需要重新绑定。

```
        Protected boid stu_dg1_EditCommand (object source, DataGridCommandEvenArgs
e)
        {
        this.stu_dg1.EditItemIndex = e.Item.ItemIndex;
        SqlConnection con=new SqlConnection
 (System.Configuration.ConfigurationManager.
        AppSettings["dsn"].ToString()) ;
        Con.Open();
        SqlCommand com = new SqlCommand ("select s.sno, s.sname, s.ssex, s.sbirthday,
s.sscore, c.classname from student s left outer join class c on s.classno=c.classno",
con) ;
        SqIDataAdapter sda = new SqIDataAdapter();
        sda.SelectCommand = com;
        DataSet ds = new DataSet();
        sda.Fill (ds,"t1") ;
        this.stu_dg1.DataSource = ds.Tables["t1"].DefaultView;
```

```
        this.stu_dg1.DataBind();
        con.Close();
        }
```

“取消”按钮单击事件的代码如下。

```
        Protected void stu_dg1_CancelCommand（object source,
DataGridCommandEventArgs e)
        {
        this.stu_dg1.EditItemIndex = -1;
        //控件再绑定
        }
```

“分页”事件的代码如下。

```
        Protected void stu_dg1_PageIndexChanged（object source,
DataGriPageChangedEventArgs e)
        {
        This.stu_dg1.CurrentPageIndex = e.NewPageIndex;
        //控件再绑定
        }
```

更新记录事件的代码如下。

```
        protected void stu_dg1_UpadateCommand（object source,
DataGridCommandEventArgs e)
        {
        string name, sex, bir, score, cla;
        string key = this.stu_dg1.DataKeys[e.Item.ItemIndex].ToString();
        TextBox tb;
        tb = （TextBox）e.Item.Cells[1].Controls[1];
        name = tb.Text.Trim();
        tb = （TextBox）e.Item.Cells[2].Controls[1];
        sex = tb.Text.Trim();
        tb = （TextBox）e.Item.Cells[3].Controls[1];
        bir = tb.Text.Trim();
        tb = （TextBox）e.Item.Cells[4].Controls[1];
        score = tb.Text.Trim();
        tb = （TextBox）e.Item.Cells[5].Controls[1];
        cla = tb.Text.Trim();
        SqlConnection con = new
        SqlConnection
（System.Configuration.ConfigurationManager.AppSettings["dsn"].ToString()）;
        Con.Open();
        SqlCommand com = new SqlCommand（"upadte student set sname='"+name+"',
ssex='"+sex+"', sbirthday='"+bir+"', sscore='"+scre+"'where sno='"+key+"'", con）;
        Com.ExecuteNonQuery();
        This.stu_dg1.EditItemIndex = -1;
        Com = new SqlCommand（"select s.sno, s.sname, s.ssex, s.sbirthday, s.sscore,
c.classnamefrom student slefy outer join class c on s.classno=c.classno", con）;
```

```
        SqlIDataAdapter sda = new SqlDataAdpter();
        sda.SelectCommand = com;
        DataSet ds = new DataSet();
        sda.Fill (ds, "t1") ;
        this.stu_dg1.DataSource = ds.Tables["t1"].DefaultView;
        this.stu_dg1.DataBind();
        con.Close();
        }
```

“删除”事件的代码如下。

```
        Protected void stu_dg1_DeleteCommand (object source,
DataGridCommandEventArgs e)
        {
        steing key = this.stu_dg1.DataKeys[e.Item.ItemIndex].Tostring();
        SqlConnection con = new SqlConnection
(System.Configuration.ConfigurationManager.)
        AppSettings[“dsn”].ToString()) ;
        con.Open();
        SqlCommand com = new SqlCommand ("delete from student where sno = '"+key+"'",
con) ;
        com.ExecuteNonQuery();
        com = new SqlCommand ("select s.sno, sname, s.ssex, s.sbirthday, s.sscore,
c.classname from student s lefy outer join class c on s.classno=c.classno", con) ;
        //控件再绑定
        }
```

“添加学生”按钮单击事件的代码如下。

```
        Protected void Button4_Click (object sender, EventArgs e)
        {
        SqlConnection con = new SqlConnection (System. Configuration.
ConfigurationManager.)
        AppSettings["dsn"].ToString()) ;
        con.Open();
        SqlCommand com1 = new SqlCommand ("insert into student (sno, sname, sbirthday,
ssex, sscore, classno) values (@sno, @sname, @sbirthday, @ssex, @sscore, @classno)
", con) ;//写入数据库
        SqlParameter sq1 = new SqlParameter ("@sname", SqlDbType.VarChar) ;
        SqlParameter sq2 = new SqlParameter ("@sno", SqlDbType.VarChar) ;
        SqlParameter sq3 = new SqlParameter ("@ssex", SqlDbType.VarChar) ;
        SqlParameter sq4 = new SqlParameter ("@sbirthday", SqlDbType.VarChar) ;
        SqlParameter sq5 = new SqlParameter ("@sscore", SqlDbType.VarChar) ;
        SqlParameter sq6 = new SqlParameter ("@classno", SqlDbType.VarChar) ;
        sp1.Value = this.TextBox1.Text;
        sp2.Value = this.TextBox2.Text;
        sp3.Value = this.TextBox3.Text;
        sp4.Value = this.TextBox4.Text;
```

```
       sp5.Value = this.TextBox5.Text;
       sp6.Value = this.TextBox6.Text;
       com1.Parameters.Add（sp1）;
       com1.Parameters.Add（sp2）;
       com1.Parameters.Add（sp3）;
       com1.Parameters.Add（sp4）;
       com1.Parameters.Add（sp5）;
       com1.Parameters.Add（sp6）;
       SqlCommand com2 = new SqlCommand（"select count（*）from student where
sno='"+this.TextBox2.Text+"'", con）;
       Int n = Convert.ToInt32（com2.ExecuteScalar()）;
       If（n>0）
       {
       this.Label2.Text="学生编号不能重复！";
       This.TextBox2.Text="      {else}
       Coml.ExecuteNonQuery();
       This.Label2.Text="插入记录成功！"
       SqlCommand com=new SqlCommang（"select,s,ssex,s,sbirthday,s,sscore,c,
classname from student s left outer join class c on s,classno=c,classno",con）;//
控件再绑定
       Protected void Button5 Click （objectsender,EventArgs e）
       This.Panell.Visible.=false;
```

“查询学生”按钮单击事件的代码如下。

```
       Protected void Button6 Click （object sender e ）
       string ck=this.TextBox7.Text.Trim9（
       ）;
       SqConnection con =new Sqonnection （System .Configuration.
ConfiguigurationManager.AppSettings["dsn"].ToString9()）;
       Con.Open();
       SqCommand com =new SqCommand （"select count9"（*）form student where
sname='"+ck+"'', con）;
       Int n =Convert.ToInt32（com.Execute  Scalar()）;
       If9（n》）
       This.Labe14 .Text='查找到'+n+'条记录！"

       com’=new SqlCommand （'selext s.sname，s.sname，s.ssex，s.sbirt
       hdayby，s.sscorn，c.classname from student s left outer join class c on
s.classno=c，classno=c，classno=c，classno where s.sname=''+ck+'', con）;
       //控件再绑定
       ……
       Else
       this.Labe14，Text=''查找到0条记录!''
       }
       }
```

第 3 步：管理员操作模块中的课程信息管理主页面代码编写。

课程信息管理页面如图 8.2.3 所示，它和学生信息管理页面非常相似。在页面初始加载时，就进行 DataGricd 控件 Dgd-course 的绑定操作，完成课程信息的显示，Dgd-course 控件第 0 列——“授课信息”中链接信息指向与此课程相关内容的显示页面，如任课教师的信息等。管理员也可以对课程信息进行编辑和删除。

图 8.2.3　课程信息管理页面

管理员可以分页浏览所有课程信息，也可以单击第一列的“课程遍号”按钮浏览为课程分配的教师情况，该页面的显示采用-blank，即再不覆盖。

```
       This.course_dgl.CurrenPageIndex=e.NewPageIndex;
       SqlConnection con =SqlConnerction（System.Configuration.
ConfigurationManager.Appssetting ["dsn"].ToString()）;
       Con.Open();
       SqlCommandcom=new sqlCmmand（"select cno，credits from couser"，con）};
       Protect void Buttonl_Cick（object） sender EvenArgs e）//跳转到相关教师信息页面}
       Response. Redirect（"shwodetaisl.aspx"）;
```

“添加课程”按钮单击事件的代码如下。

```
       Protect vido Button_Cick（object sener，EvenArgs e）
       SqiConnecttion con=new SqlConnection（System
Configuration.ConfigurationManager.AppSettings["dsn"].ToString()）; con Open();
       Sqlcommand.coml.=new splcommand（"insert into course（cno，came,credits）
values（@cno，@came，@credits）",con）;
       Splcparameter sp1= new sqlcparameter（@cno） sqldbtype.varchar）
       Sqlparameters sp2=new sqlparameter（@cname sqldbtype varchar）
       Sqlparameters sp3=new sqlparameter（@credits sqldptype varchar）
       Spl value=this textbox1 text
       Sp2value=this textbox2 text
       Sp3 value=this textbox3 text
       Coml. Parameters.add（sp1）
       Coml..parameters.add（sp2）
       Coml. Parameters.add（sp3）
       Sqlcommand com2=new sqlcommand（"select count（*）from course where
cno='"+this.textboxl text+"'",con）;
```

```
      Int n =convert toint32（com2 executescalar()）;
      If（n＞0）
      {
      This labe14 text=课程编号不能重复！";
      This textbox2 text="""
      }
      Else
      }
      Coml. Executenonquery()
      This labe14 text="插入记录成功！"
      Sqlpcommand com = new sqlpcmmnad （"select cno canme credits from course",
con）;
```

第 4 步：管理员操作模块中的成绩信息管理主页面代码编写。

成绩信息管理页面如图 8.2.4 所示，该页面完成的功能较多，包括按选定的条件进行成绩查询，根据成绩范围对包含在该范围中的学生成绩进行统计，具体统计这门课程的平均分、最高分、优秀人数和不及格人数等。此页面实现的关键在于根据条件生成 SQL 语句。当“查询”、“统计”操作被触发时，系统将完成对数据库中多个表的操作。

课程号	课程名	学分		
0101001	平面设计	5.5	编辑	删除
0101002	小型企业网络	6	编辑	删除
0101003	艺术欣赏	4	编辑	删除
0101004	数码产品维修	3	编辑	删除
0101005	美术基础	3	编辑	删除
0101006	3dsmax	3.5	编辑	删除
0102001	网络操作系统	4	编辑	删除
0102002	JAVA程序设计	6	编辑	删除
0102003	微机原理	4	编辑	删除
0102004	SQL Server数据库	5.5	编辑	删除
1 2 3				

查询全部课程任课情况　添加课程　课程分配

图 8.2.4　成绩信息管理页面

“查询方式”下拉列表控件包含“按课号”、“按课名”、“按学号”等 4 类查询条件，文本框控件中录入查询内容，按钮控件的 Click()事件完成组合条件查询。用户可以通过 DataGrid 控件的“编辑”列对查询的成绩进行修改。

在成绩统计中，“统计范围”下拉列表控件包含“班级”、“个人”等查询条件，录入成绩具体范围、课程编号、统计内容后，通过 Button 控件的 Click()事件完成组合条件的查询，并且在该事件中完成的统计数据将显示于 Label 控件 Lbl-average、high-Lbl-all、Lbl-a、Lbl-unpass 中，分别表示成绩平均分、最高分、所有学生人数、优秀学生人数和不及格学生人数。匹配过程用到了 SQL Server 2008 数据库中的 AVG()、MAK()、COUNT()等统计函数。

成绩信息管理页面的后台支持类（grade-manage.aspx.cs）的统计内容主要相关代码如下。

“查询”按钮单击事件的代码如下。

```
    protected void Button5-Click（object sender,EvenArgs e）
```

```
        {
        SplConnection con = new SqlConnection
（System.Configuration.ConfigurationManager.AppSettings["dsn"].ToString()）;
        con.Open();
        If（this.DropDownlistl.Selecteditem.Value.Equals（"0"））//判断查询条件
        {
        SqCommand com = new SqlCommd（"select c.cno，c.grade，s.sno，s.sname，sclassno
        From choic c lef outer join student s on s. sno=c.sno where
c.cno+this.TextBox5.
        Text+"",con;
        SqlDataAdapter sda = new AqlDataAdapter();
        Sda.SelectCommand = com;
        DataSet ds = new SqlDataSet();
        sda.Fill（ds，"tl"）;
        this.score dgl.DataSource = ds.Tables["tl"].DefaultView;
        this.score_dgl.DataBind( )
        {;
        else if （this.DropDownListl.SelectedItem.Value.Equals（"1"）
        }{
        SqlCommand com = new SqlCommand （"select c. con，c.grade，s.sname，s.classno
        From choice c left outer join student s on s.sno=c.sno where c.sno=""+TetBox5.
        Text+"",con）;
        SqlDataAdapter sda = new SqlDataAdapter();
        Sda.Fill（ds,"tl"）;
        this.score_dgl.DataSource=ds.Tables["tl"].DefaultView;
        this.score_dgl.Data.Bind();
        }
        Con.Close();
        }
```

“统计”按钮单击事件的代码如下。

```
        Protected void Button6_Click（object sender,EventArgs e ）
        {
        SqlConnection con = new SqlConnection（System.
Configuration.ConfigurationManaer.
        AppSettings["dsn"].ToString()）;
        Con.Open();
        String t4 = this.TextBox4.Text.Trim();
        String t6 = this.TextBox6.Text.Trim();
        if （ this. DropDownList2.SelectedItem.Value.Equals（"0"） &&this.
DropDownlist3.
        SelectedItem.Value.Equals（"0"））//判断统计条件
        {
        SqlCommand com=new SqlCommand("select max(grade max(grade) from choice where
sno=+t4
        +"", con）;
        Int n= Convert.ToInt32（com.ExecuteScalar()）;
```

```
        this.Label7.Text=this.TextBox4. Text+"的"+this. DropDownList3.
SelectedItem. Text+"为"+n+"分";
        }
        If （this. DropDownLise2. SelectedItem. Value. Equals （"o"） &.& this.
DropDownList3.SelectedItem. Value. Equals（"1"）//查询个人信息时不可以输入课程
        {
        SqlCommand com = new SqlCommand（"select avg （grade） from choice where sno='"+
t4+"'", com）;
        int n = Convert. ToInt32（com.ExecuteScalar()）; //返回记录数
        this. Label7. Text = this. TextBox4. Text + "的"+ this.
DropDownList3.SelectedItem. Text+"分"+n"分"
        }
        if （this. DropDownListe2. SelectedItem. Value. Equals（"1"） &.& this.
DropDownList3. SelectedItem. Value. Equals（"0"））
        {
        SqlCommand com =new SqlCommand（"select max（grade） from choice c left outer
join student s on c. sno=s. sno where s. classno='"+t4+"'and c. cno='"+t6+"'",
con）; int n = Convert. ToInt32（com.ExecuteScalar()）;
        this. Label7. Text = this. TextBox4. Text +”的”+this. DropDownList3.
SelectedItem.
        Text + "为" + n+ "分";
        SqlCommand coml = new SqICommand（"select count（*）from choice c left outer
join student s on c. sno=s. sno where s. classno='"+ t4 + "' and c. cno='" +t6+"'
and grade<'60'", con）;
        N= Convert. ToInt32（com2. ExecuteScalar()）;
        }
        if （ this. DropDownList2. SelectedItem. Value. Equals（"1"） &.& this.
DropDownList3. SelectedItem. Value. Equals（"1"））
        {
        SqICommand com = new SqICommand （"select avg（grade） from choice c left outer
join student s on c. sno=s. sno where s. classno='" +t4+"'", com）;
        Int n = Convert. ToInt32（com. ExecuteScalar()）:
        this. Label7. Text = this. TextBox4. Text +"的" + this. DropDownList3.
SelectedItem. Text + “为” + n + "分";
        SqlCommand coml = new SqlCommand （" select count（*） from choice c left outer
join student s on c. sno=s. sno where s. classon='" +t4 + "'and c. cno='" +t6 +"'and
grde >='85'", com）;
        n=Convert.Tolnt32（coml.ExecuteScalar()）;
        this.Label9.Text=n+"人";
        sqlCommand com2 = new sqlCommand（"sqlCt count（*）from chice c left outer
join
        student s on c . sno where s.classno="+t4+"'+t6+"'and grade
        ‹'60', con）;
        n= covert. Tolont32（com2.ExecuteScuteScalar()）;
        this . Labell . Text = n+人;
        con.Close();
```

第 5 步：管理员操作模块中的学生选课管理主页面代码编写。

在学生选课管理页面中，“课程名称”下拉列表的数据在页面初始化事件 Page_Load ()中绑定。绑定内容为数据库中的所有课程名，当选择某一个课程中的所有课程名，或者选择某一门课程时，“教师姓名”下拉列表显示相应的任课教师，此时单击“选课学生总数”按钮，可以统计选课的总人数，若总人数超出预订人数，则管理员有权删除选课时间靠后的学生，通过 DateGrid 控件的“删除”列即可直接完成该操作。

后台支持类（student course . aspx .cs）的主要相关代码如下。

```
       页面载入时对下拉列表和数据网格控件进行绑定。
       Protcctde void Page  load （object sender, EventAregs e ）
       if （! Ispostback）
       Sqlconnection con = new sqlconnection （SqlConnection （System .
ConfigurationManager.Appsetting 〔"dsn"）. Tostring()〕;
       Con . open (); sqlcommand coml = new sqlconmmand （"select c . cname , c  .
cname, c, cno from taching t left outer join  course c on t .cno = c.cno ", cno）;
       sqlDataReader dr = coml. ExecuteReader();
       this . DropDownListl . DataTextfied="cname"
       this . Dropdownlistl.databind()valuefreld .="con"
       this.  Dropdownlistl.datasource=dr
       this.  Dropdownlistl.databind();
       dr. close ();
       sqldataadapter sda = new sqldataadapter();
       sdda.Fill （ds,"t1"）
       this slcourse dgl. Datasource
       sqlcommand com 2 = new sqlcommand ();
       this . DropDownList2 .data =textfeid ="tname"
       this . DropDownList2 dataTexField +"tname"
       this . DropDownList2 datasourec=drl;
       this . DropDownList2 databind()
       con.close();
```

“选课学生总数”按钮单击事件的代码如下。

```
       Protected void Button5 click （object sender， EventArgs e ）
       Sqlconnention con =new sqlconnecting（"select count（*）from choice where
cno='"+this.
       DropDownListl. Selectedvalue+'''，con）
       Int n = cover .Toint 32 （com. Executescalar()）;
       This . Lable7. Text = "所选课程的总人数为"+n"
```

通过对上述几个模块的实现，我们可以试着设计班级管理子模块和教师管理子模块。要实现这些功能，则要理清各数据表之间的关系，再通过 SQL 语句来返回正确的结果，最后利用 ASP.NET 中的控件来显示。其他管理页面采用 DataGrid 控件的 DataSet 数据集返回所有学生信息，分页显示，并可以对学生信息进行添加、修改、查找或删除操作。

此页面中，“查询学生”按钮的 Click 事件把 Panel 的 Visible 属性重设为 true，用来显示输入查询条件的表格。根据提示，用户输入查询条件，“确定”按钮的 Click 事件通过生成 SQL 语句来实现查询功能，查询的结果最终显示在 DataGrid 控件 Dgd_student 中，在该控件中设置

“编辑”和“删除”列，提供数据的修改、删除操作。在“显示所有信息”控件的 Click()事件中，完成 DataGrid 控件 Dgd_student 的数据绑定操作，使其显示所有学生信息。同时，容纳查询条件的 Panel 控件的 Visible 属性设为 false，因为此时系统不接受直接的查询条件，只有当触发“查询学生”按钮 Click 事件后，才能重新显示查询条件。

学生选课管理页面的后台支持类（student.aspx.cs）主要代码如下。

页面加载在事件中进行数据绑定。

```
        Protected void Page_Load (object sender, EventArgs e)
        {
        If (! IsPostBack)
        {
        SqlConnection con=new SqlConnection (System.Configurgtion.
ConfigurationManager.AppSettings["dsn"].ToString())
        Con.Open()
        218页
        mand com=new SqlCommand ("select s.sno, s.sname, s, ssex, s.sbirhday, s.sscore,
c.classnama from student s left outer join class c on s.classno", con) ;

        SqlDataAdapter sda = new SqlDateAdapter();
        Sda.SelectCommand = com;
        DataSet ds = new DataSet();
        Sda.Fill (ds , " t1) ;
        This.stu_dg1.DataKeyField = "sno";// 要设置此语句，才可以查找控件
        this.stu_dg1.DataSource = ds.Tables ["t 1"].DefaultView;
        this.stu_dg1.DataBind();
        con.Close()
        this.panel1.Visble = false;//要存放在PSOTBACK中，表示第一次执行有效
        this.panel1.Visble = false;//
        } //DataGrid中的数据要用样式表固定
        }
```

“添加新生”按钮单击事件的代码如下。

```
        Protected void Buttonl_Click (object sender,EventArgs e)
        {
        this.panel2.Visible = false;
        this.pane.l1Visble = true;
        }
```

“编辑”记录事件，需要重新绑定。

```
        Protected void stu_dg1_EditCommand (objest source, DataGridCommand EventArgs
e)
        {
        this.stu_dg1.EdidtIitemlndex = e.ltem.ltemlndex;
        SqlConnectioncon=newSqlConnection
 (System.Configuration.Configurgtion.ConfigurationManager.App
Setting["dsn"].ToString())
        Com.Open();
        SqlCommand com = new SqlCommand ("select s.sno ,s.sname ,s. ssex,
s.sbirthday ,s. sscore ,
```