第9章　ADO.NET数据库访问技术

9.1 数据库概述

9.1.1 关系数据库的基本结构

1. 表
2. 记录
3. 字段
4. 关系
5. 索引
6. 视图
7. 存储过程

Access 数据库文件Stud.mdb：

表9.1 学生情况表student

学号	姓名	性别	民族	班号
1	王华	女	汉族	07001
2	孙丽	女	满族	07002
3	李兵	男	汉族	07001
6	张军	男	汉族	07001
8	马棋	男	回族	07002

表9.2 学生成绩表score

学号	课程名	分数
1	C语言	80
1	数据结构	83
2	C语言	70
2	数据结构	52
3	C语言	76
3	数据结构	70
6	C语言	90
6	数据结构	92
8	C语言	88
8	数据结构	79

9.1.2 结构化查询语言（SQL）

1. SQL语言的组成

SQL语言包含查询、操纵、定义和控制等几个部分。它们都是通过命令动词分开的，各种语句类型对应的命令动词如下：

- 数据查询的命令动词为SELECT。
- 数据定义的命令动词为CREATE、DROP。
- 数据操纵的命令动词为INSERT、UPDATE、DELETE。
- 数据控制的命令动词为GRANT、REVOKE。

2. 数据定义语言

（1）CREATE语句

CREATE语句用于建立数据表，其基本格式如下：

CREATE TABLE 表名

(列名1数据类型1 [NOT NULL]

[,列名2数据类型2 [NOT NULL]]…)

（2）DROP语句

DROP语句用于删除数据表，其基本格式如下：

DROP TABLE 表名

3. 数据操纵语言

（1）INSERT语句

INSERT语句用于在一个表中添加新记录，然后给新记录的字段赋值。其基本格式如下：

INSERT INTO 表名[(列名1[,列名2, …])]

VALUES(表达式1[,表达式2, …])

（2）UPDATE语句

UPDATE语句用于新的值更新表中的记录。其基本格式如下：

UPDATE 表名

SET 列名1 = 表达式1

[,SET 列名2 = 表达式2]…

WHERE 条件表达式

（3）DELETE语句

DELETE语句用于删除记录，其基本格式如下：

DELETE FROM 表名

[WHERE 条件表达式]

4. 数据查询语句

SQL的数据查询语句是使用很频繁的语句。SELECT的基本格式如下：

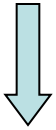
SELECT 字段表
FORM 表名
WHERE 查询条件
GROUP BY 分组字段
HAVING 分组条件
ORDER BY 字段[ASC|DESC]

各子句的功能如下：

- ◆SELECT：指定要查询的内容。
- ◆FORM：指定从其中选定记录的表名。
- ◆WHERE：指定所选记录必须满足的条件。
- ◆GROUP BY：把选定的记录分成特定的组。
- ◆HAVING：说明每个组需要满足的条件。
- ◆ORDER BY：按特定的次序将记录排序。

查询所有学生的学号、姓名、课程名和分数，要求按学号排序：

```
SELECT student.学号,student.姓名,score.课程名,score.分数
FROM student,score
WHERE student.学号=score.学号
ORDER BY student.学号
```



查询1 ： 选择查询

学号	姓名	课程名	分数
1	王华	数据结构	83
1	王华	C语言	80
2	孙丽	数据结构	52
2	孙丽	C语言	70
3	李兵	数据结构	70
3	李兵	C语言	76
6	张军	数据结构	92
6	张军	C语言	90
8	马棋	数据结构	79
8	马棋	C语言	88

记录: 1 共有记录数: 10

查询每个班每门课程的平均分：

```
SELECT student.班号,score.课程名,AVG(score.分数) AS '平均分'
    FROM student,score
WHERE student.学号=score.学号
GROUP BY student.班号,score.课程名
```

查询1 ： 选择查询

	班号	课程名	'平均分'
▶	07001	C语言	82
	07001	数据结构	81.66666666667
	07002	C语言	79
	07002	数据结构	65.5

记录： 1 共有记录

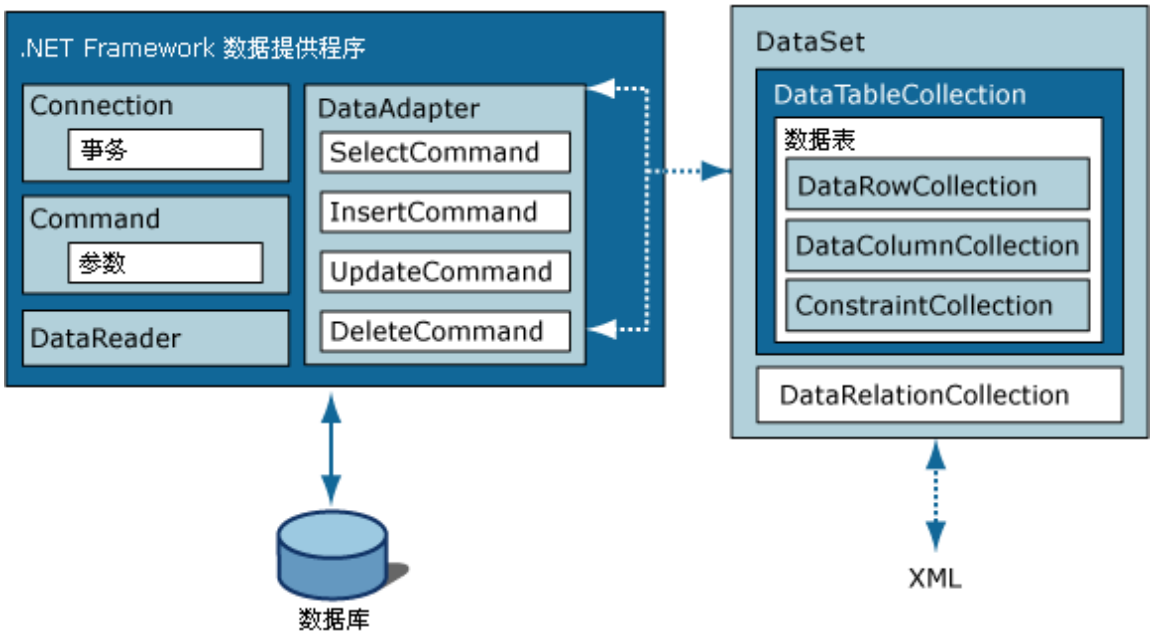
9.2 ADO.NET模型

9.2.1 ADO.NET简介

ADO.NET是在.NET Framework上访问数据库的一组类库，它利用.NET Data Provider（数据提供程序）以进行数据库的连接与访问。

通过ADO .NET，数据库程序设计人员能够很轻易地使用各种对象来访问符合自己需求的数据库内容。

9.2.2 ADO.NET体系结构



1. .NET Data Provider

.NET Data Provider是指访问数据源的一组类库，主要是为了统一对于各类型数据源的访问方式而设计的一套高效能的类数据库。

下表给出了.NET Data Provider中包含的4个对象。

对象名称	功能说明
Connection	提供和数据源的连接功能。
Command	提供运行访问数据库命令，传送数据或修改数据的功能，例如运行SQL命令和存储过程等。
DataAdapter	是DataSet对象和数据源间的桥梁。DataAdapter使用4个Command对象来运行查询、新建、修改、删除的SQL命令，把数据加载到DataSet，或者把DataSet内的数据送回数据源。
DataReader	通过Command对象运行SQL查询命令取得数据流，以便进行高速、只读的数据浏览。

在.NET Framework中常用的有如下4组数据提供程序：

（1）SQL.NET Data Provider

（2）OLEDB.NET Data Provider

（3）ODBC.NET Data Provider

（4）ORACLE.NET Data Provider

2. DataSet

DataSet（数据集）是ADO .NET离线数据访问模型中的核心对象，主要使用时机是在内存中暂存并处理各种从数据源中所取回的数据。

DataSet其实就是一个存放在内存中的数据暂存区，这些数据必须通过DataAdapter对象与数据库进行数据交换。在DataSet内部允许同时存放一个或多个不同的数据表（DataTable）对象。

这些数据表是由数据列和数据域所组成的，并包含有主索引键、外部索引键、数据表间的关系（Relation）信息以及数据格式的条件限制（Constraint）。

9.2.3 ADO.NET数据库的访问流程

ADO.NET数据库访问的一般流程如下：

（1）建立Connection对象，创建一个数据库连接。

（2）在建立连接的基础上可以使用Command对象对数据库发送查询、新增、修改和删除等命令。

（3）创建DataAdapter对象，从数据库中取得数据。

（4）创建DataSet对象，将DataAdapter对象填充到DataSet对象（数据集）中。

（5）如果需要，可以重复操作，一个DataSet对象可以容纳多个数据集合。

（6）关闭数据库。

（7）在DataSet上进行所需要的操作。数据集的数据要输出到窗体中或者网页上面，需要设定数据显示控件的数据源为数据集。

9.3 ADO.NET的数据访问对象

9.3.1 OleDbConnection对象

在数据访问中首先必须是建立数据库的物理连接。

.NET Data Provider使用OleDbConnection类的对象标识与一个数据库的物理连接。

1. OleDbConnection类

OleDbConnection类的属性	说　　明
ConnectionString	获取或设置用于打开数据库的字符串。
ConnectionTimeout	获取在尝试建立连接时终止尝试并生成错误之前所等待的时间。
Database	获取当前数据库或连接打开后要使用的数据库的名称。
DataSource	获取数据源的服务器名或文件名。
Provider	获取在连接字符串的“Provider=”子句中指定的OLEDB提供程序的名称。
State	获取连接的当前状态。其取值及其说明如表13.7所示。

OleDbConnection类的方法	说　　明
Open	使用 ConnectionString 所指定的属性设置打开数据库连接。
Close	关闭与数据库的连接。这是关闭任何打开连接的首选方法。
CreateCommand	创建并返回一个与 OleDbConnection 关联的 OleDbCommand 对象。
ChangeDatabase	为打开的OleDbConnection 更改当前数据库。

2. 建立连接字符串ConnectionString

建立连接字符串的方式是：先创建一个OleDbConnection对象，将其ConnectionString属性设置为如下值：

Provider=Microsoft.Jet.OLEDB.4.0;DataSource=Access数据库;

UserId=用户名;Password=密码;

其中Provider和Data Source是必选项，如果Access数据库没有密码，后两者都可以省略。由Access数据库是基于文件的数据库，因此在实际项目中应该将Data Source属性值转化为服务器的绝对路径。

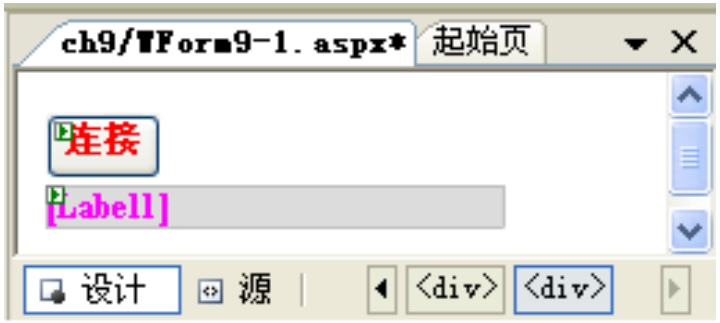
最后用Open方法打开连接。

【例9.10】 设计一个说明直接建立连接字符串的连接过程的网页WForm9-1.aspx。

其设计步骤如下：

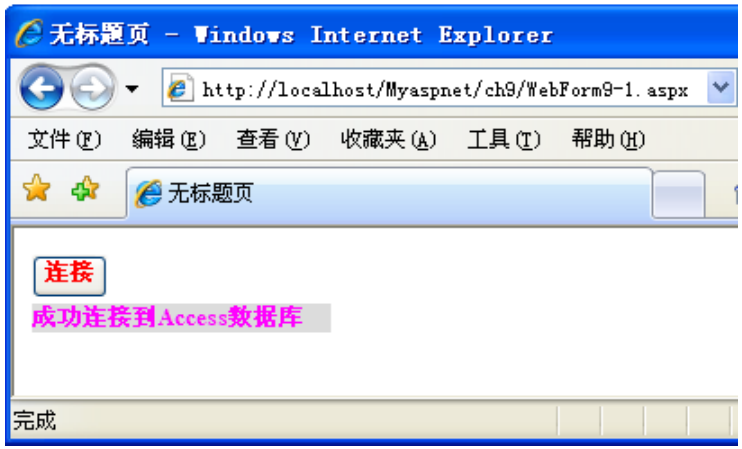
（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-1的空网页。

（2）其设计界面如下图所示，其中包含一个Button控件Button1和一个标签Label1，将该网页的StyleSheetTheme属性设置为Blue。



在该网页上设计如下事件过程：

```
Protected Sub Button1_Click(ByVal sender As Object,
        ByVal e As System.EventArgs) Handles Button1.Click
    Dim mystr As String
    Dim myconn As New OleDbConnection()
    mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
      "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
    myconn.ConnectionString = mystr
    myconn.Open()
    If myconn.State = Data.ConnectionState.Open Then
      Label1.Text = "成功连接到Access数据库"
    Else
      Label1.Text = "不能连接到Access数据库"
    End If
    myconn.Close()
End Sub
```



3. 将连接字符串存放在Web.config文件中

可以在Web.config文件中保存用于连接数据库的连接字符串，再通过对Web.config文件加密，从而达到保护连接字符串的目的。例如，在<configuration>节中插入以下代码：

```
<connectionStrings>
    <add name="myconnstring"
        connectionString="Provider=Microsoft.Jet.OLEDB.4.0;
        Data Source=|DataDirectory|Stud.mdb"
        providerName="System.Data.OleDb" />
</connectionStrings>
```

这样，以下代码自动获取Web.config文件中的连接字符串myconnstring：

```
Dim mystr As String =
    ConfigurationManager.ConnectionStrings("myconnstring").ToString()
Dim myconn As New OleDbConnection()
myconn.ConnectionString = mystr
myconn.Open()
```

也可以在Web.config文件的<configuration>节中插入以下代码：

```
<appSettings>
        <add key = "myconnstring"
          value="Provider = Microsoft.Jet.OLEDB.4.0;
          Data Source = |DataDirectory|Stud.mdb" />
</appSettings>
```

这样，以下代码自动获取Web.config文件中的连接字符串myconnstring：

```
Dim mystr As String =
        ConfigurationManager.AppSettings("myconnstring")
Dim myconn As New OleDbConnection()
myconn.ConnectionString = mystr
myconn.Open()
```

9.3.2 OleDbCommand对象

建立数据连接之后，就可以执行数据访问操作和数据操纵操作了。一般对数据库的操作被概括为CRUD—Create、Read、Update和Delete。在ADO.NET中定义OleDbCommand类去执行这些操作。

1. OleDbCommand类的属性和方法

OleDbCommand类的属性	说　　明
CommandText	获取或设置要对数据源执行的 T-SQL 语句或存储过程。
CommandTimeout	获取或设置在终止执行命令的尝试并生成错误之前的等待时间。
CommandType	获取或设置一个值，该值指示如何解释CommandText属性。其取值如表13.10所示。
Connection	数据命令对象所使用的连接对象
Parameters	参数集合（OleDbParameterCollection）

OleDbCommand类的方法	说　　明
CreateParameter	创建OleDbParameter对象的新实例。
ExecuteNonQuery	针对Connection 执行SQL语句并返回受影响的行数。
ExecuteReader	将CommandText发送到Connection并生成一个OleDbDataReader。
ExecuteScalar	执行查询，并返回查询所返回的结果集中第一行的第一列。忽略其他列或行。

2. 创建OleDbCommand对象

OleDbCommand类的主要构造函数如下：

OleDbCommand();

OleDbCommand(cmdText);

OleDbCommand(cmdText,connection);

其中，cmdText参数指定查询的文本。connection参数是一个OleDbConnection，它表示到Access数据库的连接。

例如，以下语句创建一个OleDbCommand对象mycmd：

```
Dim myconn As New OleDbConnection()
Dim mystr As String = "Provider = Microsoft.Jet.OLEDB.4.0;" &
    "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
myconn.ConnectionString = mystr
myconn.Open()
Dim mycmd As New OleDbCommand("SELECT * FROM student",myconn)
```

3. 通过OleDbCommand对象返回单个值

在OleDbCommand的方法中，ExecuteScalar方法执行返回单个值的SQL命令。

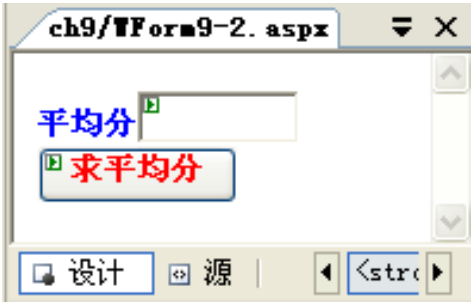
例如，如果想获取Student数据库中学生的总人数，则可以使用这个方法执行SQL查询SELECT Count(*) FROM student。

【例9.11】 设计一个通过OleDbCommand对象求score表中的平均分的网页WForm9-2.aspx。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-2的空网页。

（2）其设计界面如下图所示，其中包含一个HTML标签、一个文本框TextBox1和一个Button控件Button1，将该网页的StyleSheetTheme属性设置为Blue。



在该网页上设计如下事件过程：

```
Protected Sub Button1_Click(ByVal sender As Object,
            ByVal e As System.EventArgs) Handles Button1.Click
    Dim mystr As String, mysql As String
    Dim myconn As New OleDbConnection()
    Dim mycmd As New OleDbCommand()
    mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
       "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
    myconn.ConnectionString = mystr
    myconn.Open()
    mysql = "SELECT AVG(分数) FROM score"
    mycmd.CommandText = mysql
    mycmd.Connection = myconn
    TextBox1.Text = mycmd.ExecuteScalar().ToString()
    myconn.Close()
End Sub
```


无标题页 - Windo...
http://localhost/
文件(F) 编辑(E) 查看(V) 收藏夹
无标题页
平均分 78
求平均分
本地 Intranet
100%

网页运行界面

4. 通过OleDbCommand对象执行修改操作

在OleDbCommand的方法中，ExecuteNonQuery方法执行不返回结果的SQL命令。

该方法主要用来更新数据，通常使用它来执行UPDATE、INSERT和DELETE语句。

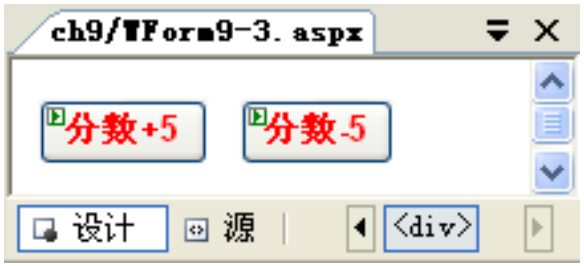
该方法不返回行，对于UPDATE、INSERT和DELETE语句，返回值为该命令所影响的行数，对于所有其他类型的语句，返回值为-1。

【例9.12】 设计一个通过OleDbCommand对象将score表中所有分数增5分和减5分的网页WForm9-3。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-3的空网页。

（2）其设计界面如下图所示，其中包含两个Button控件Button1和Button2，将该网页的StyleSheetTheme属性设置为Blue。



网页设计界面

在该网页上设计如下事件过程：

```
Imports System.Data.OleDb                    '引用命名空间
Partial Class ch9_WForm9_3
    Inherits System.Web.UI.Page
    Private mycmd As New OleDbCommand()                    '私有字段
    Private myconn As New OleDbConnection()                '私有字段
    Protected Sub Page_Load(ByVal sender As Object,
                            ByVal e As System.EventArgs) Handles Me.Load
        Dim mystr As String
        mystr = "Provider = Microsoft.Jet.OLEDB.4.0; " & _
            "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
        myconn.ConnectionString = mystr
        myconn.Open()
    End Sub
```

```
    Protected Sub Page_Unload(ByVal sender As Object,
                              ByVal e As System.EventArgs) Handles Me.Unload
        myconn.Close()  '关闭本网页时关闭连接
    End Sub
    Protected Sub Button1_Click(ByVal sender As Object,
                              ByVal e As System.EventArgs) Handles Button1.Click
        Dim mysql As String
        mysql = "UPDATE score SET 分数=分数+5"
        mycmd.CommandText = mysql
        mycmd.Connection = myconn
        mycmd.ExecuteNonQuery()
    End Sub
    Protected Sub Button2_Click(ByVal sender As Object,
                              ByVal e As System.EventArgs) Handles Button2.Click
        Dim mysql As String
        mysql = "UPDATE score SET 分数=分数-5"
        mycmd.CommandText = mysql
        mycmd.Connection = myconn
        mycmd.ExecuteNonQuery()
    End Sub
End Class
```

5. 在OleDbCommand对象的命令中指定参数

OLEDB.NET Data Provider支持执行命令中包含参数的情况，也就是说，可以使用包含参数的数据命令或存储过程执行数据筛选操作和数据更新等操作，其主要流程如下：

（1）创建Connection对象，并设置相应的属性值。

（2）打开Connection对象。

（3）创建Command对象并设置相应的属性值，其中SQL语句含有占位符。

（4）创建参数对象，将建立好的参数对象添加到Command对象的Parameters集合中。

（5）为参数对象赋值。

（6）执行数据命令。

（7）关闭相关对象。

例如，下面的更新语句：

UPDATE course SET cName = @Name WHERE cID = @ID

其中course是一个课程表，有cID（课程号）和cName（课程名）两个列。该命令是将指定cID的课程记录的cName替换成指定的值。其中@ID和@Name均为参数，在执行该语句之前需要为参数赋值。

可以使用以下命令向Parameters参数集合中添加参数值：

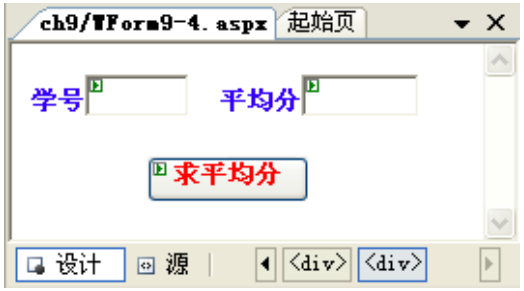
```
mycmd.Parameters.Add("@Name",OleDbType.VarChar,10).Value = Name1;
mycmd.Parameters.Add("@ID", OleDbType.VarChar,5).Value = ID1;
```

【例9.13】 设计一个通过OleDbCommand对象求出指定学号学生的平均分的网页WForm9-4。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-4的空网页。

（2）其设计界面如下图所示，其中包含两个HTML标签、两个文本框（TextBox1和TextBox2）和一个Button控件Button1，将该网页的StyleSheetTheme属性设置为Blue。



网页设计界面

在该网页上设计如下事件过程：

```
Protected Sub Button1_Click(ByVal sender As Object,
            ByVal e As System.EventArgs) Handles Button1.Click
    Dim mystr As String, mysql As String
    Dim myconn As New OleDbConnection()
    Dim mycmd As New OleDbCommand()
    mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
       "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
    myconn.ConnectionString = mystr
    myconn.Open()
    mysql = "SELECT AVG(分数) FROM score WHERE 学号=@no"
    mycmd.CommandText = mysql
    mycmd.Connection = myconn
    mycmd.Parameters.Add("@no", OleDbType.VarChar, 5).Value = _
                        TextBox1.Text  '设置参数值
    TextBox2.Text = mycmd.ExecuteScalar().ToString()
    myconn.Close()
End Sub
```


无标题页 - Windows...
http://localhost/
文件(F) 编辑(E) 查看(V) 收藏夹(A)
无标题页
学号 8
平均分 83.5
求平均分
本地 Intranet
100%

网页运行界面

9.3.3 DataReader对象

当执行返回结果集的命令时，需要一个方法从结果集中提取数据。处理结果集的方法有两个：

（1）使用DataReader对象（数据阅读器）；

（2）同时使用DataAdapter对象（数据适配器）和ADO.NET DataSet。

1. DataReader类的属性和方法

属性	说明
FieldCount	获取当前行中的列数
IsClosed	获取一个布尔值，指出DataReader对象是否关闭
RecordsAffected	获取执行SQL语句时修改的行数

方法	说明
Read	将DataReader对象前进到下一行并读取，返回布尔值指示是否有多行
Close	关闭DataReader对象
IsDBNull	返回布尔值，表示列是否包含NULL值
NextResult	将DataReader对象移到下一个结果集，返回布尔值指示该结果集是否有多行
GetBoolean	返回指定列的值，类型为布尔值
GetString	返回指定列的值，类型为字符串
GetByte	返回指定列的值，类型为字节
GetInt32	返回指定列的值，类型为整型值
GetDouble	返回指定列的值，类型为双精度值
GetDataTime	返回指定列的值，类型为日期时间值
GetOrdinal	返回指定列的序号或数字位置（首列序号为0）
GetBoolean	返回指定列的值，类型为对象

2. 创建DataReader对象

在ADO.NET中从来不会显式的使用DataReader对象的构造函数创建的DataReader对象。事实上，DataReader类没有提供公有的构造函数。人们通常调用Command类的ExecuteReader方法，这个方法将返回一个DataReader对象。

例如，以下代码创建一个OleDbDataReader对象myreader：

```
Dim cmd As New OleDbCommand(CommandText, ConnectionObject)
Dim myreader As OleDbDataReader = cmd.ExecuteReader()
```

注意：OleDbDataReader对象不能使用New来创建。

3. 遍历OleDbDataReader对象的记录

当ExecuteReader方法返回DataReader对象时，当前光标的位置是第一条记录的前面。必须调用OleDbDataReader对象的Read方法把光标移动到第一条记录，然后，第一条记录将变成当前记录。

如果OleDbDataReader对象中包含的记录不止一条，Read方法就返回一个Boolean值true。想要移动到下一条记录，需要再次调用Read方法。重复上述过程，直到最后一条记录，此时Read方法将返回false。

经常使用While循环来遍历记录：

```
While myreader.Read()
    '读取数据
End While
```

只要Read方法返回的值为true，就可以访问当前记录中包含的字段。

4．访问字段中的值

（1）Item属性

每一个DataReader对象都定义了一个Item属性，此属性返回一个代码字段属性的对象。Item属性是DataReader对象的索引。需要注意的是Item属性总是基于0开始编号的：

```
myreader(FieldName)
myreader(FieldIndex)
```

（2）Get方法

每一个DataReader对象都定义了一组Get方法，那些方法将返回适当类型的值。例如，GetInt32方法把返回的字段值作为32位整数，每一个Get方法都将接受字段的索引。例如，在上面的例子中，使用以下的代码可以检索ID字段和cName字段的值：

```
myreader.GetInt32(0)
myreader.GetString(1)
```

【例9.14】 设计一个通过OleDbDataReader对象在一个列表框中输出所有学生记录的网页WForm9-5。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-5的空网页。

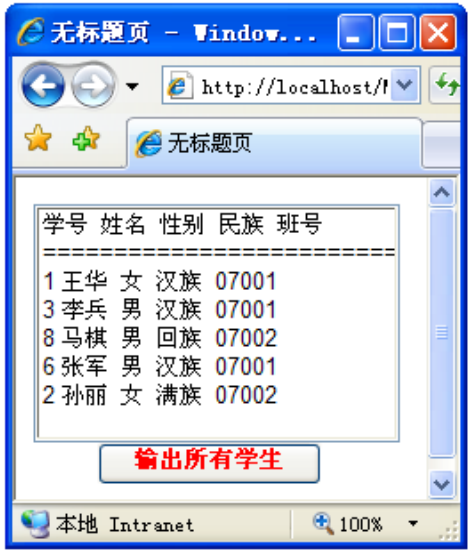
（2）其设计界面如下图所示，其中包含一个列表框ListBox1（Rows属性设为8）和一个Button控件Button1，将该网页的StyleSheetTheme属性设置为Blue。

网页设计界面

在该网页上设计如下事件过程：

```
Protected Sub Button1_Click(ByVal sender As Object,
        ByVal e As System.EventArgs) Handles Button1.Click
    Dim mystr As String, mysql As String
    Dim myconn As New OleDbConnection()
    Dim mycmd As New OleDbCommand()
    mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
      "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
    myconn.ConnectionString = mystr
    myconn.Open()
    mysql = "SELECT * FROM student"
    mycmd.CommandText = mysql
    mycmd.Connection = myconn
    Dim myreader As OleDbDataReader = mycmd.ExecuteReader()
    ListBox1.Items.Add("学号 姓名 性别 民族 班号")
    ListBox1.Items.Add("==============================")
```

```
'循环读取信息
    While myreader.Read()
        ListBox1.Items.Add(String.Format("{0} {1} {2} {3} {4}", _
        myreader(0).ToString(), myreader(1).ToString(), _
        myreader(2).ToString(), myreader(3).ToString(), _
        myreader(4).ToString()))
    End While
    myconn.Close()
    myreader.Close()
End Sub
```


无标题页 - Window...
http://localhost/
无标题页
学号 姓名 性别 民族 班号
==========
1 王华 女 汉族 07001
3 李兵 男 汉族 07001
8 马棋 男 回族 07002
6 张军 男 汉族 07001
2 孙丽 女 满族 07002
输出所有学生
本地 Intranet
100%

网页运行界面

9.3.4 OleDbDataAdapter对象

OleDbDataAdapter对象（数据适配器）可以执行SQL命令以及调用存储过程、传递参数，最重要的是取得数据结果集，在数据库和DataSet对象之间来回传输数据。

1. OleDbDataAdapter类的属性和方法

属性	说明
SelectCommand	获取或设置SQL语句用于选择数据源中的记录。该值为OleDbCommand对象
InsertCommand	获取或设置SQL语句用于将新记录插入到数据源中。该值为OleDbCommand对象
UpdateCommand	获取或设置SQL语句用于更新数据源中的记录。该值为OleDbCommand对象
DeleteCommand	获取或设置SQL语句用于从数据集中删除记录。该值为OleDbCommand对象
AcceptChangesDuringFill	获取或设置一个值，该值指示在任何Fill操作过程中时，是否接受对行所做的修改
AcceptChangesDuringUpdate	获取或设置在Update期间是否调用AcceptChanges
FillLoadOption	获取或设置LoadOption，后者确定适配器如何从DbDataReader中填充DataTable
MissingMappingAction	确定传入数据没有匹配的表或列时需要执行的操作
MissingSchemaAction	确定现有DataSet架构与传入数据不匹配时需要执行的操作
TableMappings	获取一个集合，它提供源表和DataTable之间的主映射

方法	说明
Fill	用来自动执行OleDbDataAdapter对象的SelectCommand属性中相对应的SQL语句，以检索数据库中的数据，然后更新数据集中的DataTable对象，如果DataTable对象不存在，则创建它
FillSchema	将DataTable添加到DataSet中，并配置架构以匹配数据源中的架构
GetFillParameters	获取当执行SQL SELECT语句时由用户设置的参数
Update	用来自动执行UpdateCommand、InsertCommand或DeleteCommand属性相对应的SQL语句，以使数据集中的数据来更新数据库。

2. 创建OleDbDataAdapter对象

OleDbDataAdapter类有以下构造函数：

```
Public Sub New()
Public Sub New(selectCommandText As String)
Public Sub New(selectCommandText As String,
        selectConnection As OleDbConnection)
Public Sub New (selectCommandText As String,
        selectConnectionString As String)
```

例如：

```
Dim mystr As String,mysql As String
Dim myconn As New OleDbConnection()
mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
"Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
myconn.ConnectionString = mystr
myconn.Open()
mysql = "SELECT * FROM student"
Dim myadapter As New OleDbDataAdapter(mysql,myconn)
myconn.Close()
```

3. 使用Fill方法

Fill方法用于向DataSet对象填充从数据源中读取的数据。调用Fill方法的语法格式有多种，常见的格式如下：

OleDbDataAdapter对象名.Fill(DataSet对象名，"数据表名");

其中第一个参数是数据集对象名，表示要填充的数据集对象；第二个参数是一个字符串，表示在本地缓冲区中建立的临时表的名称。

例如，以下语句用course表数据填充数据集mydataset1：

```
OleDbDataAdapter1.Fill(mydataset1,"course");
```

4. 使用Update方法

Update方法用于将数据集DataSet对象中的数据按InsertCommand属性、DeleteCommand属性和UpdateCommand属性所指定的要求更新数据源，即调用3个属性中所定义的SQL语句来更新数据源。

Update方法常见的调用格式如下。

OleDbDataAdapter对象名.Update(DataSet对象名,[数据表名]);

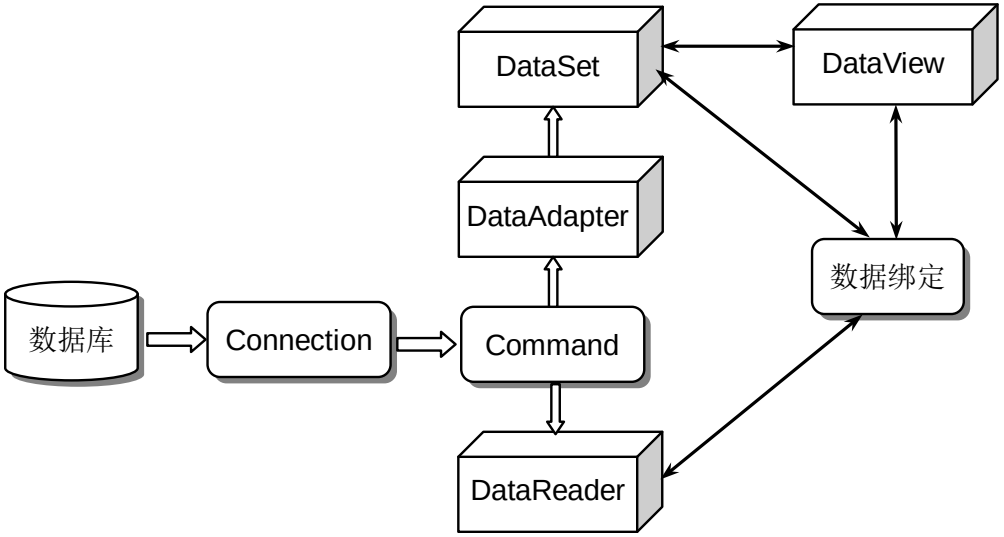
其中第一个参数是数据集对象名，表示要将哪个数据集对象中的数据更新到数据源中；第二个参数是一个字符串，表示临时表的名称。

9.4 DataSet对象

DataSet是ADO.NET数据库访问组件的核心，主要是用来支持ADO.NET的不连贯连接及数据分布。

它的数据驻留内存，可以保证和数据源无关的一致性的关系模型，并用于多个异种数据源的数据操作。

9.4.1 DataSet对象概述



创建DataSet对象有多种方法，既可以使用设计工具，也可以使用程序代码来创建DataSet对象。使用程序代码创建DataSet对象的语法格式如下：

Dim 对象名 As New DataSet()

或

Dim 对象名 As New DataSet(dataSetName)

其中，dataSetName为一个字符串，用于指出DataSet的名称。

9.4.2 Dataset对象的属性和方法

属性	说明
CaseSensitive	获取或设置一个值，该值指示DataTable对象中的字符串比较是否区分大小写
DataSetName	获取或设置当前DataSet的名称
Relations	获取用于将表链接起来并允许从父表浏览到子表的关系的集合
Tables	获取包含在DataSet中的表的集合

DataSet对象的方法	说明
AcceptChanges	提交自加载此DataSet或上次调用AcceptChanges以来对其进行的所有更改
Clear	通过移除所有表中的所有行来清除任何数据的DataSet
CreateDataReader	为每个DataTable返回带有一个结果集的DataTableReader，顺序与Tables集合中表的显示顺序相同
GetChanges	获取DataSet的副本，该副本包含自上次加载以来或自调用AcceptChanges以来对该数据集进行的所有更改
HasChanges	获取一个值，该值指示DataSet是否有更改，包括新增行、已删除的行或已修改的行
Merge	将指定的DataSet、DataTable或DataRow对象的数组合并到当前的DataSet或DataTable中
Reset	将DataSet重置为初始状态

9.4.3 Tables集合和DataTable对象

DataSet对象的Tables属性由表组成，每个表是一个DataTable对象。实际上，每一个DataTable对象代表了数据库中的一个表，每个DataTable数据表都由相应的行和列组成。

可以通过索引来引用Tables集合中的一个表，例如，Tables(i)表示第i个表，其索引值从0开始编号。

1. Tables集合的属性和方法

Tables集合的属性	说明
Count	Tables集合中表个数
Item	检索Tables集合中指定索引处的表

Tables集合的方法	说明
Add	向Tables集合中添加一个表
AddRange	向Tables集合中添加一个表的数组
Clear	移除Tables集合中的所有表
Contains	确定指定表是否在Tables集合中
Equqls	判断是否等于当前对象
GetType	获取当前实例的Type
Insert	将一个表插入到Tables集合中指定的索引处
IndexOf	检索指定的表在Tables集合中的索引
Remove	从Tables集合中移除指定的表
RemoveAt	移除Tables集合中指定索引处的表

2. DataTable对象

一个DataTable对象包含一个Columns属性即列集合和一个Rows属性即行集合。

属性	说明
CaseSensitive	指示表中的字符串比较是否区分大小写
ChildRelations	获取此DataTable的子关系的集合
Columns	获取属于该表的列的集合
Constraints	获取由该表维护的约束的集合
DataSet	获取此表所属的DataSet
DefaultView	返回可用于排序、筛选和搜索DataTable的DataView
ExtendedProperties	获取自定义用户信息的集合
ParentRelations	获取该DataTable的父关系的集合
PrimaryKey	获取或设置充当数据表主键的列的数组
Rows	获取属于该表的行的集合
TableName	获取或设置DataTable的名称

方法	说明
AcceptChanges	提交自上次调用AcceptChanges以来对该表进行的所有更改
Clear	清除所有数据的DataTable
Compute	计算用来传递筛选条件的当前行上的给定表达式
CreateDataReader	返回与此DataTable中的数据相对应的DataTableReader
ImportRow	将DataRow复制到DataTable中，保留任何属性设置以及初始值和当前值
Merge	将指定的DataTable与当前的DataTable合并
NewRow	创建与该表具有相同架构的新DataRow
Select	获取DataRow对象的数组

3. 建立包含在数据集中的表

建立包含在数据集中的表的方法主要有以下两种。

（1）利用数据适配器的Fill方法自动建立DataSet中的DataTable对象

先通过OleDbDataAdapter对象从数据源中提取记录数据，然后调用其Fill方法，将所提取的记录存入DataSet中对应的表内，如果DataSet中不存在对应的表，Fill方法会先建立表再将记录填入其中。例如，以下语句向DataSet对象myds中添加一个表course及其包含的数据记录：

```
Dim myds As New DataSet()
Dim myda As New OleDbDataAdapter("SELECT * From course",myconn)
 myda.Fill(myds, "course")
```

（2）将建立的DataTable对象添加到DataSet中

先建立DataTable对象，然后调用DataSet的表集合属性Tables的Add方法，将DataTable对象添加到DataSet对象中。

例如，以下语句向DataSet对象myds中添加一个表，并返回表的名称course：

```
Dim myds As New DataSet()
Dim mydt As New DataTable("course")
myds.Tables.Add(mydt)
textBox1.Text = myds.Tables("course").TableName
   '文本框中显示“course”
```

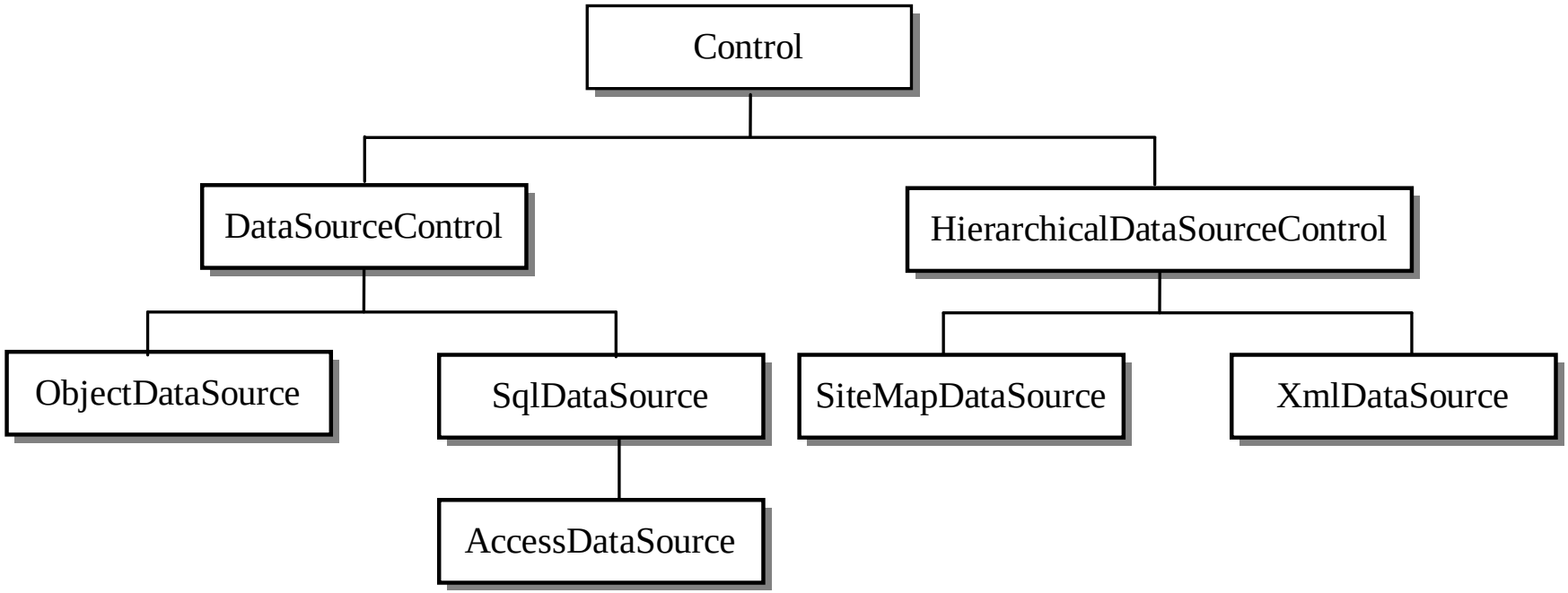
以下内容自学：

- ✓ **Columns集合和DataColumn对象**
- ✓ **Rows集合和DataRow对象**
- ✓ **Relations集合和DataRelation对象**

9.5 数据源控件

数据源控件允许用户使用不同类型的数据源，如数据库、XML文件或中间层业务对象。

数据源控件连接到数据源，从中检索数据，并使得其他控件可以绑定到数据源而无需代码。数据源控件还支持修改数据。


Control
DataSourceControl
HierarchicalDataSourceControl
ObjectDataSource
SqlDataSource
SiteMapDataSource
XmlDataSource
AccessDataSource

9.5.1 SqlDataSource控件

SqlDataSource控件对应的类为SqlDataSource，它表示到Web应用程序中数据库的直接连接。数据绑定控件（如GridView、DetailsView和FormView控件）可以使用SqlDataSource控件自动检索和修改数据。可以将用来选择、插入、更新和删除数据的命令指定为SqlDataSource控件的一部分，并让该控件自动执行这些操作。用户无需编写代码（例如，使用System.Data命名空间中的类的ADO.NET代码）来创建连接并指定用于查询和更新数据库的命令。

可以使用SqlDataSource控件连接到SQL Server、Access和ODBC数据库。SqlDataSource类派生出AccessDataSource类，后面重点介绍AccessDataSource类的使用方法。

9.5.2 AccessDataSource控件

AccessDataSource控件简化与Microsoft Access数据库文件（.mdb文件）建立连接。

1. AccessDataSource控件的属性、方法和事件

属性	说明
ConnectionString	获取用来连接到Access数据库的连接字符串
DataFile	获取或设置Access.mdb文件的位置
DataSourceMode	获取或设置AccessDataSource控件获取数据所用的数据检索模式
DeleteCommand	获取或设置AccessDataSource控件从基础数据库删除数据所用的SQL字符串
DeleteCommandType	获取或设置一个值，该值指示DeleteCommand属性中的文本是SQL语句还是存储过程的名称
DeleteParameters	从与AccessDataSource控件相关联的AccessDataSourceView对象获取包含DeleteCommand属性所使用的参数的参数集合。
EnableCaching	获取或设置一个值，该值指示AccessDataSource控件是否启用数据缓存
FilterExpression	获取或设置调用Select方法时应用的筛选表达式
FilterParameters	获取与FilterExpression字符串中的任何参数占位符关联的参数的集合

InsertCommand	**获取或设置AccessDataSource控件将数据插入基础数据库所用的SQL字符串**
InsertCommandType	**获取或设置一个值，该值指示InsertCommand属性中的文本是SQL语句还是存储过程的名称**
InsertParameters	**从与AccessDataSource控件相关联的AccessDataSourceView对象获取包含InsertCommand属性所使用的参数的参数集合**
ProviderName	**获取用于连接到Access数据库的AccessDataSource控件的 .NET数据提供程序的名称**
SelectCommand	**获取或设置AccessDataSource控件从基础数据库检索数据所用的SQL字符串**
SelectCommandType	**获取或设置一个值，该值指示SelectCommand属性中的文本是SQL查询还是存储过程的名称**
SelectParameters	**从与AccessDataSource控件相关联的AccessDataSourceView对象获取包含SelectCommand属性所使用的参数的参数集合**
SortParameterName	**获取或设置存储过程参数的名称，在使用存储过程执行数据检索时，该存储过程参数用于对检索到的数据进行排序**
UpdateCommand	**获取或设置AccessDataSource控件更新基础数据库中的数据所用的SQL字符串**
UpdateCommandType	**获取或设置一个值，该值指示UpdateCommand属性中的文本是SQL语句还是存储过程的名称**
UpdateParameters	**从与AccessDataSource控件相关联的AccessDataSourceView控件获取包含UpdateCommand属性所使用的参数的参数集合**

方法	说明
DataBind	将数据源绑定到被调用的服务器控件及其所有子控件
Delete	使用DeleteCommand SQL字符串和DeleteParameters集合中的所有参数执行删除操作
Select	使用SelectCommand SQL字符串以及SelectParameters集合中的所有参数从基础数据库中检索数据
Update	使用UpdateCommand SQL字符串和UpdateParameters集合中的所有参数执行更新操作

2. AccessDataSource控件的功能

功能	要求
排序	将DataSourceMode属性设置为DataSet值
筛选	将FilterExpression属性设置为在调用Select方法时用于筛选数据的筛选表达式
分页	AccessDataSource不支持直接在Access数据库上进行分页操作。如果将DataSourceMode属性设置为DataSet值，则数据绑定控件（如GridView）可以在AccessDataSource返回的项上进行分页
更新	将UpdateCommand属性设置为用来更新数据的SQL语句。此语句通常是参数化的
删除	将DeleteCommand属性设置为用来删除数据的SQL语句。此语句通常是参数化的
插入	将InsertCommand属性设置为用来插入数据的SQL语句。此语句通常是参数化的
缓存	将DataSourceMode属性设置为DataSet值，EnableCaching属性设置为true，并根据希望缓存数据所具有的缓存行为设置CacheDuration和CacheExpirationPolicy属性

3. 使用AccessDataSource控件连接到Access数据库

可以将AccessDataSource控件连接到Access数据库，然后使用某些控件（例如GridView）来显示或编辑数据。其操作步骤如下：

（1）新建用来连接到Access数据库的网页WForm9-7.aspx。

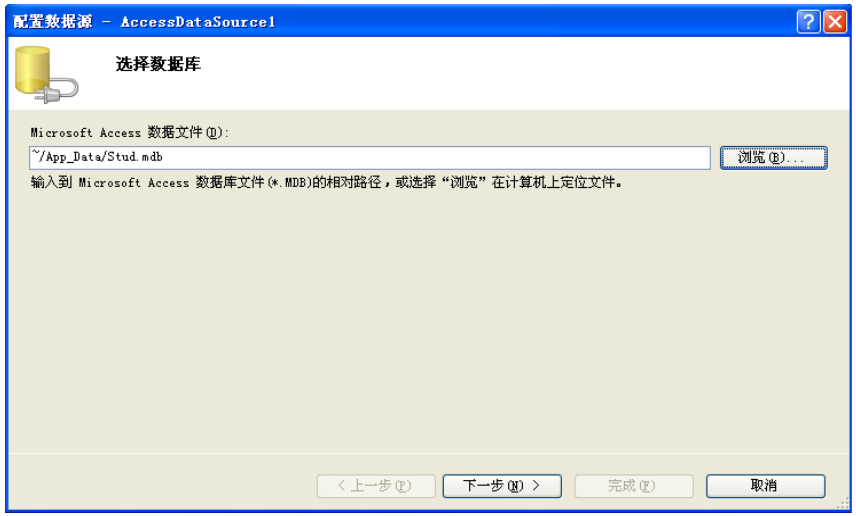
（2）切换到“设计”视图。

（3）从工具箱的“数据”选项卡中将AccessDataSource控件拖动到页面上。

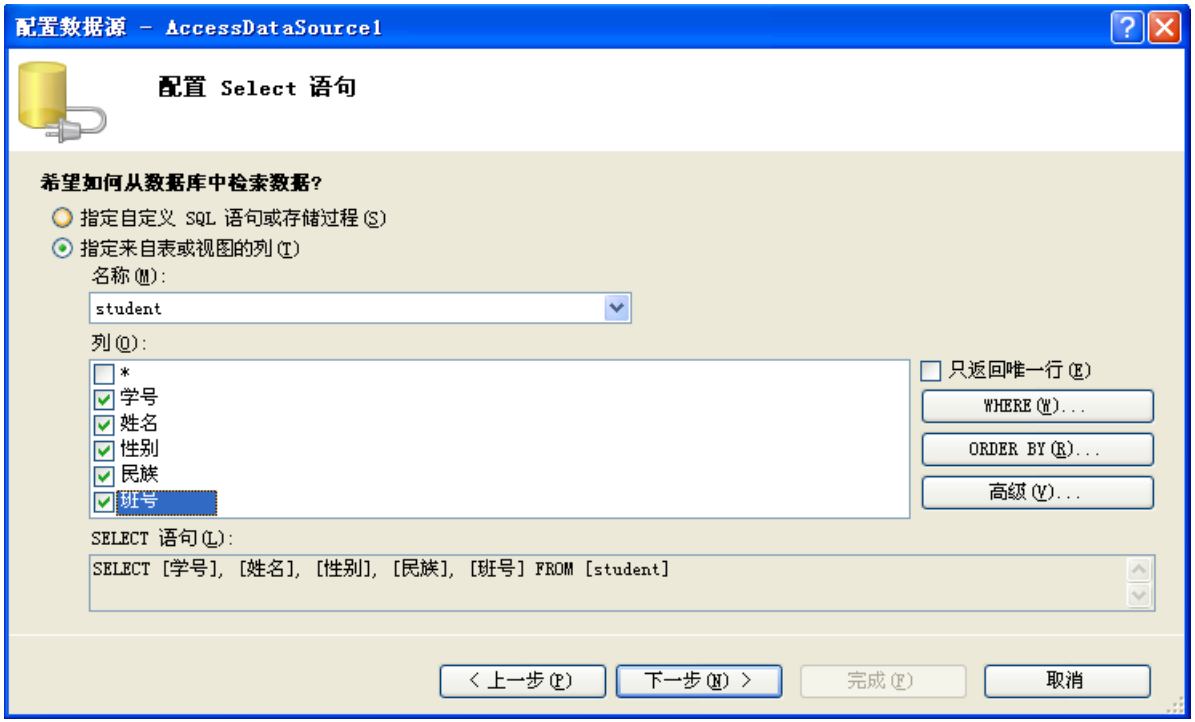
（4）如果智能标记面板没有显示，单击该控件右上方的智能标记。

（5）在“AcessDataSource任务”列表中，单击“配置数据源”将显示“配置数据源”向导。

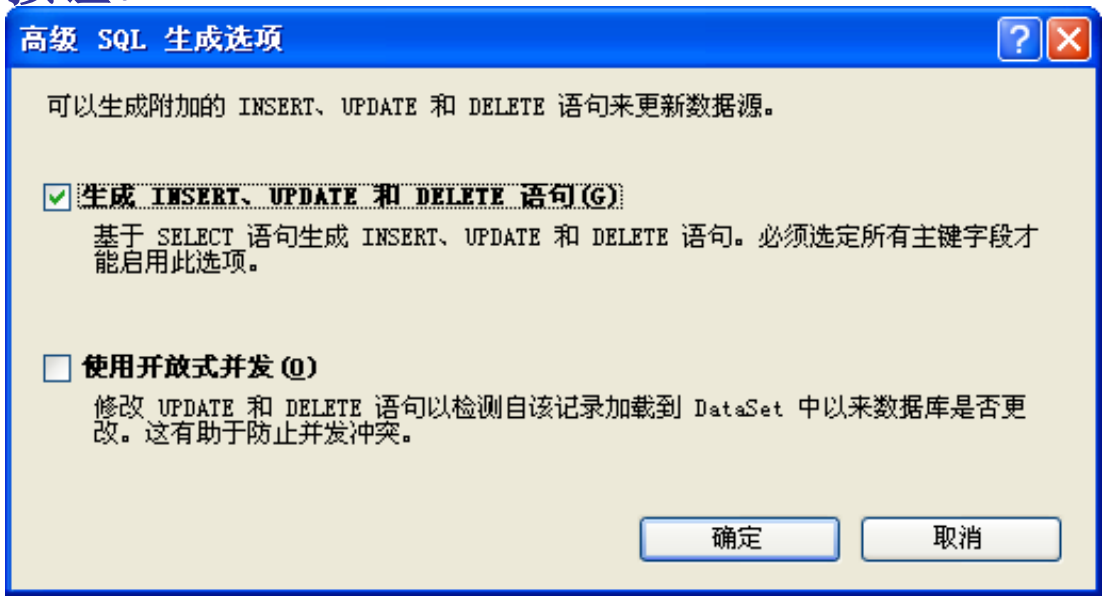
（6）在“选择数据库”对话框中，键入或选择Access数据库的路径，该数据库的扩展名为.mdb。如下图所示选择数据库为Stud.mdb。单击“下一步”按钮。



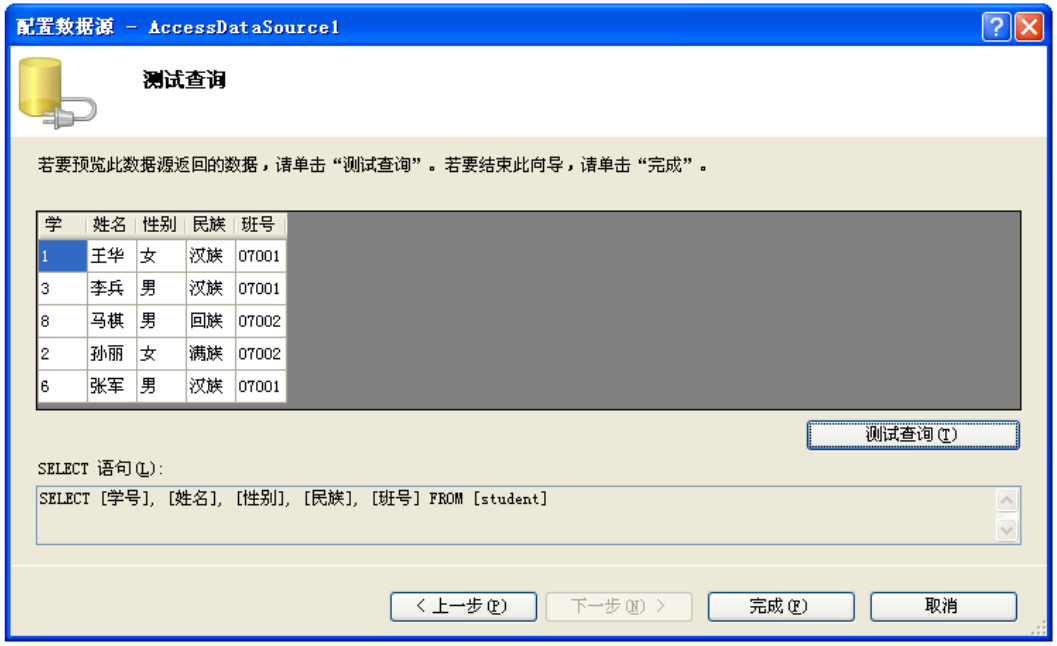
（7）出现“配置Select语句”对话框，选择student表，勾选所有列名，如下图所示。



（8）单击“高级”按钮，出现“高级SQL生成选项”对话框，勾选“生成INSERT、UPDATE和DELETE语句”选项，如下图所示，单击“确定”按钮。



（9）单击“下一步”按钮，在出现的对话框中单击“测试查询”按钮，出现如下图所示的对话框，表示查找成功。单击“完成”按钮。



这样就在网页中建立好了AccessDataSource控件，其SelectQuery属性自动设置为：

SELECT [学号], [姓名], [性别], [民族], [班号] FROM [student]

InsertQuery属性自动设置为：

INSERT INTO [student] ([学号], [姓名], [性别], [民族], [班号]) VALUES (?, ?, ?, ?, ?)

UpadteQuery属性自动设置为：

UPDATE [student] SET [姓名] = ?, [性别] = ?, [民族] = ?, [班号] = ? WHERE [学号] = ?

DeleteQuery属性自动设置为：

DELETE FROM [student] WHERE [学号] = ?

9.5.3 ObjectDataSource控件

自学，并了解与AccessDataSource控件的不同点。

9.6 数据绑定控件

数据绑定就是把数据连接到网页的过程。在数据绑定后，可以通过网页界面来操作数据库中的数据。

9.6.1数据绑定概述

数据绑定控件将数据以标记的形式呈现给请求数据的浏览器。

数据绑定控件可以绑定到数据源控件，并自动在页请求生命周期的适当时间获取数据。数据绑定控件可以利用数据源控件提供的功能，包括排序、分页、缓存、筛选、更新、删除和插入。数据绑定控件通过其DataSourceID属性连接到数据源控件。

9.6.2 列表控件

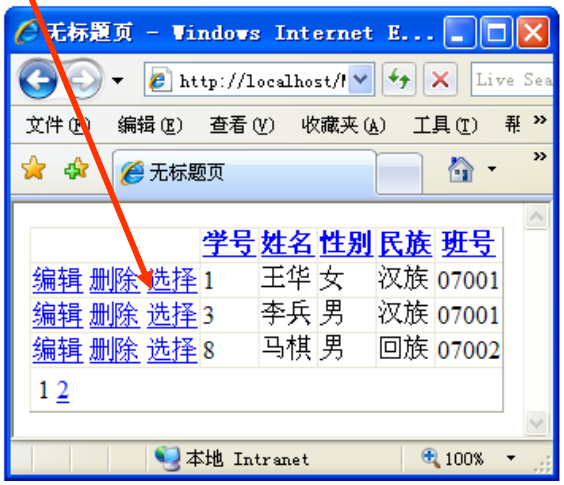
列表控件在实现数据绑定时主要指定控件的DataSource和DataTextField属性，然后调用DataBind方法。

例如，以下语句建立了一个供用户选择民族的DropDownList控件DropDownList1：

```
Dim mystr As String =
        ConfigurationManager.AppSettings("myconnstring")
Dim myconn As New OleDbConnection()
myconn.ConnectionString = mystr
myconn.Open()
Dim myds As New DataSet()
Dim myda As New OleDbDataAdapter("SELECT distinct 民族
        FROM student", myconn)
myda.Fill(myds, "student")
DropDownList1.DataSource = myds.Tables("student")
DropDownList1.DataTextField = "民族"
DropDownList1.DataBind()
myconn.Close()
```

9.6.2 GridView控件

GridView控件在表中显示数据源的值，其中每列表示一个字段，每行表示一条记录。它允许用户选择和编辑这些项以及对它们进行排序等。



1. GridView控件的属性

（1）PagerSettings属性

使用PagerSettings属性控制GridView控件中页导航行的设置。它是一个PagerSettings对象的引用。

例如：

```
<PagerSettings
    Mode = "NextPreviousFirstLast"
    FirstPageText = "第一页"
    LastPageText = "末页">
</PagerSettings>
```

其中，PagerSettings属性的Mode有4种取值：

- ■NextPrevious：上一页按钮和下一页按钮。
- ■NextPreviousFirstLast：上一页按钮、下一页按钮、第一页按钮和最后一页按钮。
- ■Numeric：可直接访问页面的带编号的链接按钮。
- ■NumericFirstLast：带编号的链接按钮、第一个链接按钮和最后一个链接按钮。

在Mode属性设置为NextPrevious、NextPreviousFirstLast或NumericFirstLast值时，可以通过设置PagerSettings属性的以下属性来自定义非数字按钮的文字：

- FirstPageText：第一页按钮的文字。
- PreviousPageText：上一页按钮的文字。
- NextPageText：下一页按钮的文字。
- LastPageText：最后一页按钮的文字。

也可以通过设置PagerSettings属性的以下属性为非数字按钮显示图像：

- FirstPageImageUrl：为第一页按钮显示的图像的URL。
- PreviousPageImageUrl：为上一页按钮显示的图像的URL。
- NextPageImageUrl：为下一页按钮显示的图像的URL。
- LastPageImageUrl：为最后一页按钮显示的图像的URL。

（2）PageSize、PageCount和PageIndex属性

PageSize属性获取或设置一个页面中显示的记录个数。PageCount属性获取或设置总的页数，PageIndex获取或设置当前的页号。正确地使用这些属性可以实现分页功能。

（3）AutoGenerateColumns属性

该属性获取或设置网页运行时是否基于关联的数据源自动生成列。其默认值为True，也就是说，一旦指定了GridView控件的数据源，便自动生成相应的列。

在有些情况下，不希望自动生成列，而是通过“GridView任务”列表的“编辑列”命令来设置相关的列，此时要将AutoGenerateColumns属性设为False。

（4）DataKeyNames和DataKeys属性

DataKeyNames属性是一个字符串数组，指定表示数据源主键的字段。当设置了DataKeyNames属性时，GridView控件自动为该控件中的每一行创建一个DataKey对象。

DataKey对象包含在DataKeyNames属性中的指定的字段的值。

DataKey对象随后被添加到控件的DataKeys集合中。使用DataKeys属性检索GridView控件中特定数据行的DataKey对象。这提供了一种访问每个行的主键的便捷方法。

例如：

```
GridView1.DataKeyNames = New string() {"学号"}
 …
TextBox1.Text = GridView1.DataKeys(0).Value.ToString()
```

（5）Rows属性

获取表示GridView控件中数据行的GridViewRow对象的集合。

2. 基本数据操作

GridView控件提供了很多内置功能，这些功能使得用户可以对控件中的项进行排序、更新、删除、选择和分页。当GridView控件绑定到某个数据源控件时，GridView控件可利用该数据源控件的功能并提供自动排序、更新和删除功能。

注意，GridView控件可为其他类型的数据源提供对排序、更新和删除的支持；但必须提供一个适当的事件处理程序，其中包含对这些操作的实现。

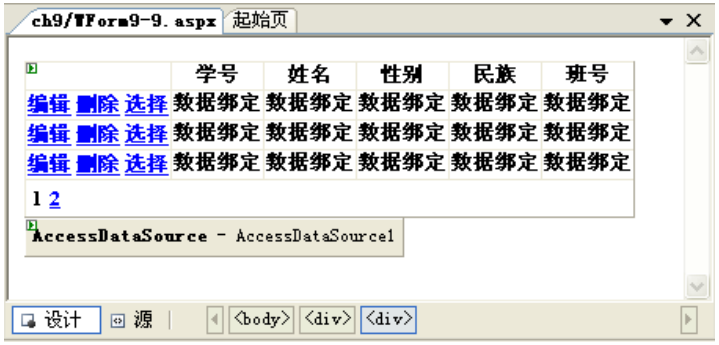
排序允许用户通过单击某个特定列的标题来根据该列排序GridView控件中的项。若要启用排序，需将AllowSorting属性设置为true。

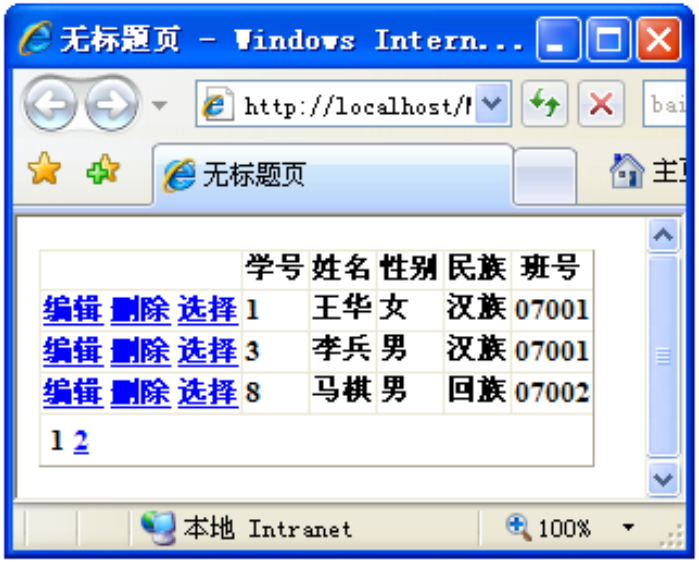
当单击ButtonField或TemplateField列字段中命令名分别为“Edit”、“Delete”和“Select”的按钮时，自动更新、删除和选择功能启用。如果AutoGenerateEditButton、AutoGenerateDeleteButton或AutoGenerateSelectButton属性分别设置为true时，GridView控件可自动添加带有“编辑”、“删除”或“选择”按钮的CommandField列字段。

注意，GridView控件不直接支持将记录插入数据源。但是，通过将GridView控件与DetailsView或FormView控件结合使用则可以插入记录。

GridView控件可自动将数据源中的所有记录分成多页，而不是同时显示这些记录。若要启用分页，需将AllowPaging属性设置为true。

应用例子：


ch9/WForm9-9.aspx
起始页
学号
姓名
性别
民族
班号
编辑 删除 选择 数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
编辑 删除 选择 数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
编辑 删除 选择 数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
1 2
AccessDataSource - AccessDataSource1
设计
源
<body>


无标题页 - Windows Intern...
http://localhost/
无标题页
学号 姓名 性别 民族 班号
编辑 删除 选择 1 王华 女 汉族 07001
编辑 删除 选择 3 李兵 男 汉族 07001
编辑 删除 选择 8 马棋 男 回族 07002
1 2
本地 Intranet
100%

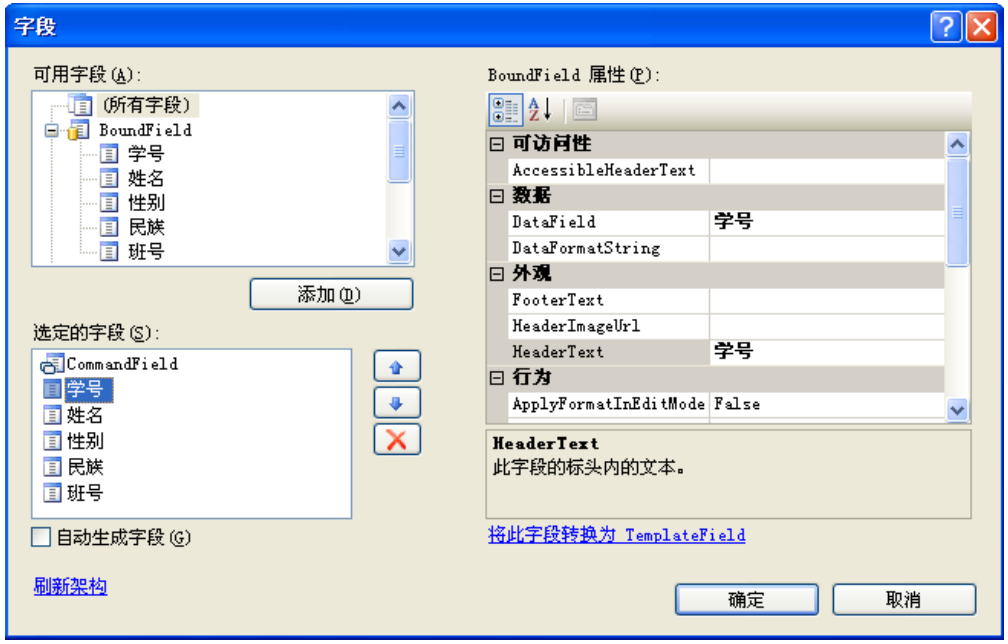
4. 列对象处理

（1）列字段

GridView控件中的每一列由一个DataControlField对象表示。

在默认情况下，AutoGenerateColumns属性被设置为true，为数据源中的每一个字段创建一个AutoGeneratedField对象。然后每个字段作为GridView控件中的列呈现，其顺序同于每一字段在数据源中出现的顺序。

选择“GridView列表”中的“编辑列”命令，打开“字段”对话框，如下图所示，通过该对话框可进行列字段的添加、删除，或对列字段的属性进行设置等，如利用HeaderText属性设置列字段呈现的文本。



（2）使用列模板

添加列模板有两种方式：

（1）通过“字段”对话框中“可用字段”列表中选取TemplateField选项，单击“添加”按钮，就在“选定的字段”列表中创建了一个模板列。

（2）通过把一个“选定的字段”列表中现有的列转换为模板列，通过单击“将些字段转换为TemplateField”超链接完成。

在完成模板的转换后，选择“GridView任务”中的“编辑模板”命令，就可以编辑该模板。TemplateField类包含如下表所示的多个模板。

模板类型	说明
HeaderTemplate	在列头部显示一个单元格
FootTemplate	在列尾部显示一个单元格
ItemTemplate	显示模式下的每行单元格
AlternatiingItemTemplate	显示模式下的隔行单元格
EditItemTemplate	编辑模式下的单元格
EmptyDataTemplate	当没有相应数据时，呈现给用户的外观表示
PagerTemplate	分页模式下呈现的外观

【例9.17】 设计一个通过GridView控件显示score表记录的网页WForm9-10，要求显示分数对应的等级。

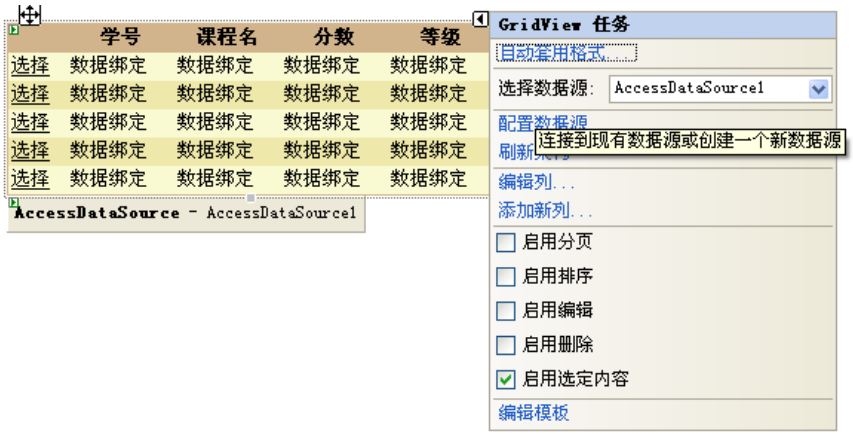
其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-10的空网页。

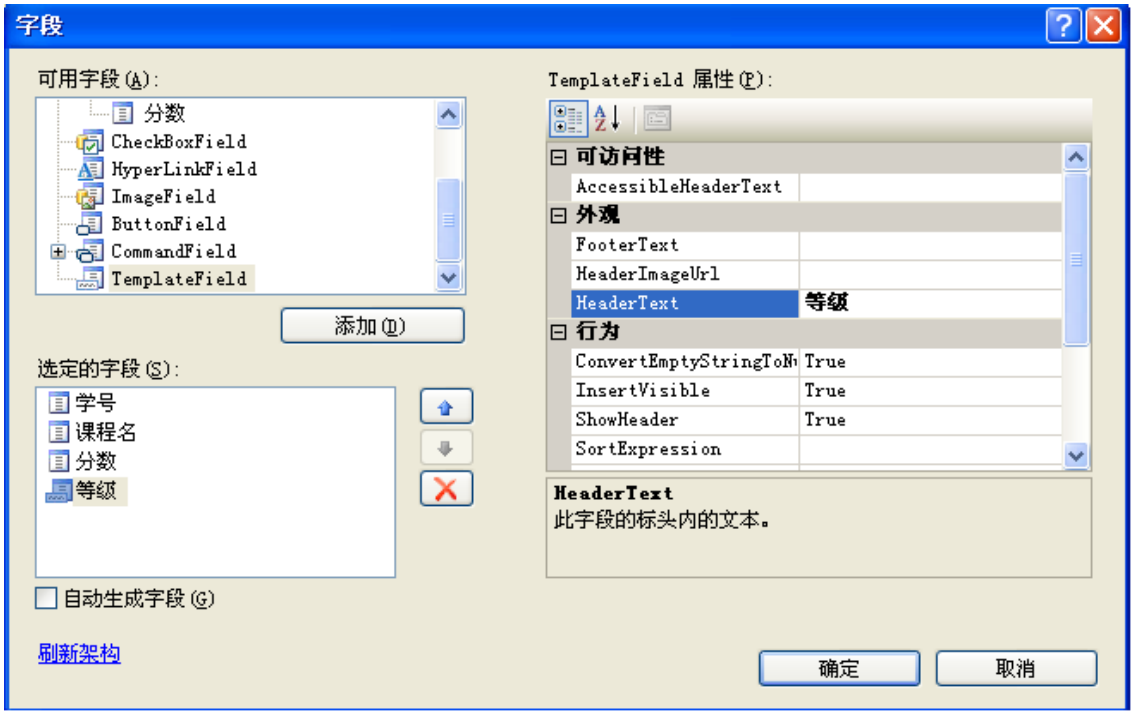
（2）向其中拖放一个GridView控件GridView1。

（3）在“GridView任务”列表中选择“自动套用格式”命令设置GridView1控件的外观，再选择“新建数据源”命令，在出现的“选择数据源类型”对话框，选中“Access数据库”，保持默认的名称为AccessDataSource1，单击“确定”按钮。

（4）按照前面建立AccessDataSource控件数据源的方式指定数据源为Stud.mdb数据库的score表。此时的“GridView任务”列表如图9.39所示，勾选中相关选项。



（5）从“GridView任务”列表中选择“编辑列”命令，出现“字段”对话框，从“可用字段”列表中选中“TemplateField”，单击“添加”按钮将其加入到“选定的字段”列表中，选中它，在TemplateField属性中将HeaderText属性改为“等级”，如下图所示。

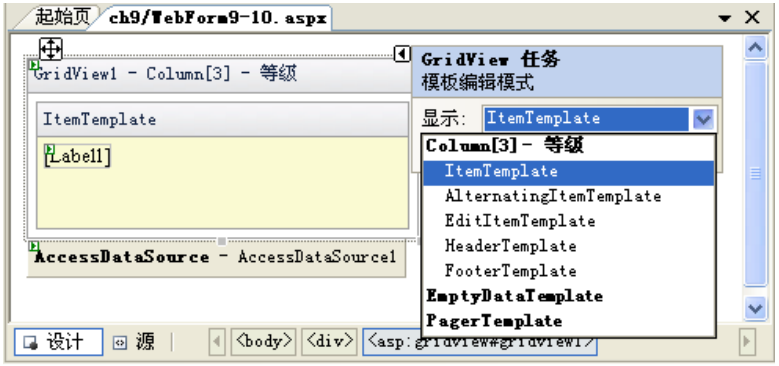


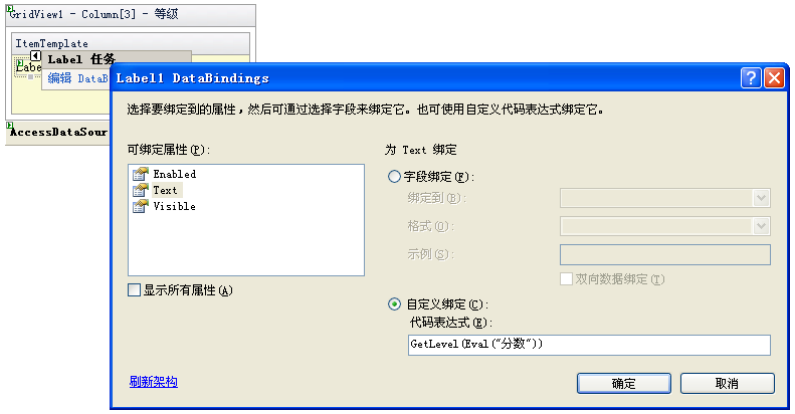
（6）在“GridView任务”列表中选择“编辑模板”命令，出现如左图所示的“模板编辑模式”列表，从中选择ItemTemplate。

在ItemTemplate区中拖放一个Label控件，从“Label任务”列表中选择“编辑DataBindings”命令，打开“Label1 DataBindings”对话框，选中“自定义绑定”选项，在代码表达式文本框中输入：

GetLevel(Eval(“分数”))

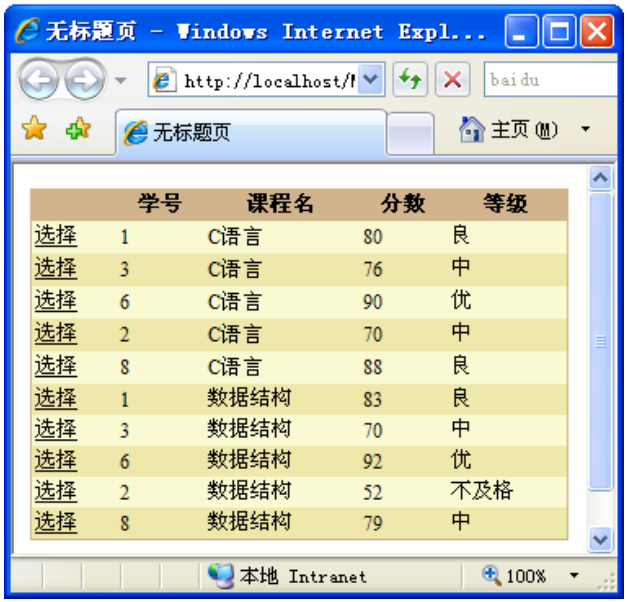
如右图所示。单击“确定”按钮。返回后再单击“GridView任务”列表中的“结束编辑模板”命令。





（7）在本网页上设计如下事件过程：

```
Public Function GetLevel(ByVal s As Object) As String
    Dim fs As Integer = CInt(s.ToString())
    If fs >= 90 Then
        Return ("优")
    ElseIf fs >= 80 Then
        Return ("良")
    ElseIf (fs >= 70) Then
        Return ("中")
    ElseIf fs >= 60 Then
        Return ("及格")
    Else
        Return ("不及格")
    End If
End Function
```



	学号	课程名	分数	等级
选择	1	C语言	80	良
选择	3	C语言	76	中
选择	6	C语言	90	优
选择	2	C语言	70	中
选择	8	C语言	88	良
选择	1	数据结构	83	良
选择	3	数据结构	70	中
选择	6	数据结构	92	优
选择	2	数据结构	52	不及格
选择	8	数据结构	79	中

运行界面

5. DataBinder类

该类提供对应用程序快速开发设计器的支持以生成和分析数据绑定表达式语法。在网页数据绑定语法中可以使用此类的静态方法Eval在运行时计算数据绑定表达式。

Eval方法的基本语法格式如下：

```
Public Shared Function Eval(container As Object,expression As String)
    As Object
```

其中，参数container指出进行计算的对象引用。expression指出从container到要放置在绑定控件属性中的公共属性值的导航路径。其返回值为Object，它是数据绑定表达式的计算结果。

必须将<%#和%>标记放在数据绑定表达式的两端；这些标记也用于标准ASP.NET数据绑定。当数据绑定到模板列表中的控件时，此方法尤其有用。

对于所有的列表Web控件，如DataGrid、DataList或Repeater，container参数值均应为“Container.DataItem”。如果要对页进行绑定，则container参数值应为“Page”。

例如，使用Eval方法以绑定到Price字段，其使用代码如下：

```
<%# DataBinder.Eval(Container.DataItem, "Price") %>
```

6. GridViewRow类

GridView控件中每个单独行都是一个GridViewRow对象。

GridView控件将其所有数据行都存储在Rows集合中。若要确定Rows集合中GridViewRow对象的索引，需使用RowIndex属性。

通过使用Cells属性，可以访问GridViewRow对象的单独单元格。如果某个单元格包含其他控件，则通过使用单元格的Controls集合，可以从单元格检索控件。如果控件指定了ID，还可以使用单元格的FindControl方法来查找该控件。

若要从BoundField字段列或自动生成的字段列检索字段值，需使用单元格的Text属性。若要从将字段值绑定到控件的其他字段列类型检索字段值，先从相应的单元格检索控件，然后访问该控件的相应属性。

下面的示例演示如何使用GridViewRow对象，从GridView控件中的单元格检索字段值，然后在该页上显示该值：

```
Dim selectRow As GridViewRow = GridView1.SelectedRow
Dim txt As String = selectRow.Cells(1).Text
```

【例9.18】 设计一个通过GridView控件修改student表记录班号的网页WForm9-11。

其设计步骤如下：

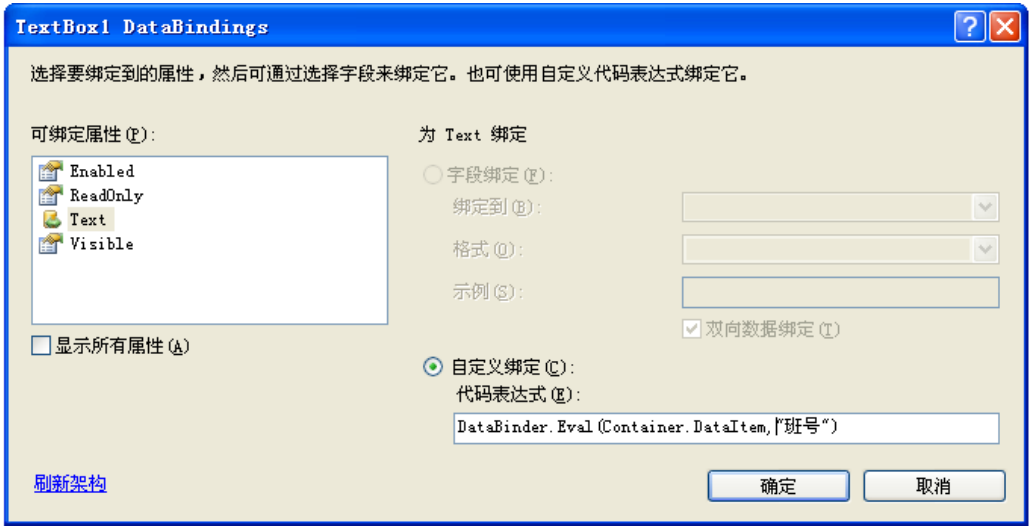
（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-11的空网页。

（2）向其中拖放一个GridView控件GridView1。在其下方拖放一个Button控件Button1。将本网页的StyleSheetTheme属性指定为Blue。

（3）在“GridView任务”列表中选择“自动套用格式”命令设置GridView1控件的外观，再选择“新建数据源”命令，在出现的“选择数据源类型”对话框，选中“Access数据库”，保持默认的名称为AccessDataSource1，单击“确定”按钮。

（4）在“GridView任务”列表中选择“编辑列”命令，打开“字段”对话框，从“选定的字段”列表中选中“班号”，单击“将此字段转换为TemplateField”超链接，单击“确定”按钮返回。

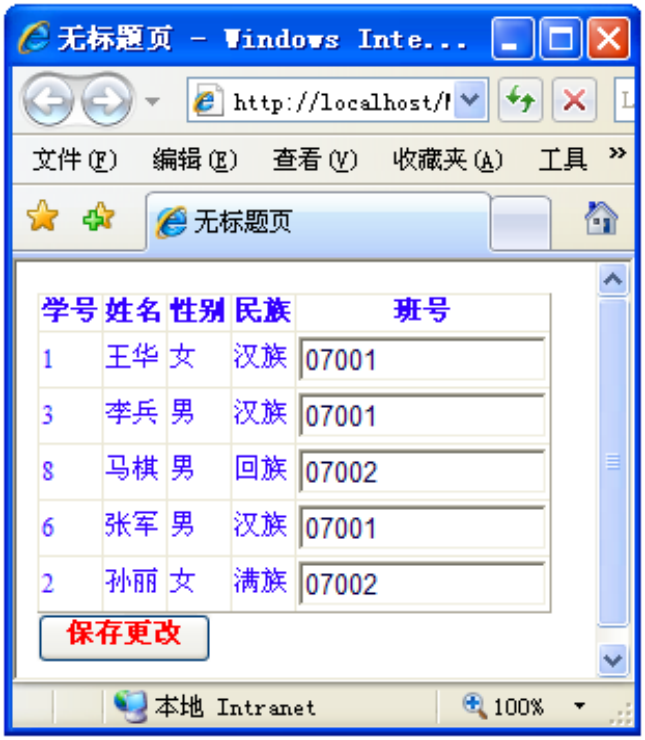
（5）在“GridView任务”列表中选择“编辑模板”命令，指定ItemTemplate模板，向其中拖放一个TextBox控件TextBox1。从“TextBox任务”列表中选择“编辑DataBindings”命令，打开“TextBox1 DataBindings”对话框，选中“自定义绑定”选项，在代码表达式文本框中输入DataBinder.Eval(Container.DataItem,“班号”)，如下图所示。单击“确定”按钮。返回后再单击“GridView任务”列表中的“结束编辑模板”命令。



（6）在本网页中设计如下事件过程：

```
Protected Sub Button1_Click(ByVal sender As Object,
      ByVal e As System.EventArgs) Handles Button1.Click
    Dim no As String
    Dim txtbh As TextBox
    Dim i As Integer
    For i = 0 To GridView1.Rows.Count - 1
        txtbh = GridView1.Rows(i).FindControl("TextBox1")
        no = GridView1.Rows(i).Cells(0).Text
        Update(no, txtbh.Text)
    Next
End Sub
```

```
Public Sub Update(ByVal no As String, ByVal nbh As String)
    Dim mycmd As New OleDbCommand()
    Dim myconn As New OleDbConnection()
    Dim mystr As String, mysql As String
    mystr = "Provider = Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source =" & Server.MapPath("~\\App_data\\Stud.mdb")
    myconn.ConnectionString = mystr
    myconn.Open()
    mysql = "UPDATE student SET 班号='" & nbh & _
      "' WHERE 学号 = '" & no & "'"
    mycmd.CommandText = mysql
    mycmd.Connection = myconn
    mycmd.ExecuteNonQuery()
    myconn.Close()
End Sub
```


无标题页 - Windows Inte...
http://localhost/
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具
无标题页
学号 姓名 性别 民族 班号
1 王华 女 汉族 07001
3 李兵 男 汉族 07001
8 马棋 男 回族 07002
6 张军 男 汉族 07001
2 孙丽 女 满族 07002
保存更改
本地 Intranet
100%

7. DataView类

在GridView控件实现数据绑定的第2种方法中经常用DataView对象。DataView对象能够创建DataTable中所存储数据的不同视图，用于对DataSet中的数据进行排序、过滤和查询等操作。

DataView对象类似于数据库中的视图功能，提供DataTable列（Column）排序、过滤记录（Row）及记录的搜索，它的一个常见用法是为控件提供数据绑定。

DataView对象的构造函数如下：

```
Public Sub New()
Public Sub New(table As DataTable)
Public Sub New(table As DataTable, RowFilter As String, Sort As String,RowState As DataViewRowState)
```

DataTable对象提供DefaultView属性返回默认的DataView对象。例如：

```
Dim mydv As New DataView()
mydv = myds.Tables("student").DefaultView
```

属性	说明
AllowDelete	设置或获取一个值，该值指示是否允许删除
AllowEdit	获取或设置一个值，该值指示是否允许编辑
Allownew	获取或设置一个值，该值指示是否可以使用Addnew方法添加新行
ApplyDefaultSort	获取或设置一个值，该值指示是否使用默认排序
Count	在应用RowFilter和RowStateFilter之后，获取DataView中记录的数量
Item	从指定的表获取一行数据
RowFilter	获取或设置用于筛选在DataView中查看哪些行的表达式
RowStateFilter	获取或设置用于DataView中的行状态筛选器
Sort	获取或设置 DataView 的一个或多个排序列以及排序顺序
Table	获取或设置源DataTable。

方法	说明
Addnew	将新行添加到DataView中
Delete	删除指定索引位置的行
Find	按指定的排序关键字值在 DataView 中查找行
FindRows	返回DataRowView对象的数组，这些对象的列与指定的排序关键字值匹配
ToTable	根据现有DataView中的行，创建并返回一个新的DataTable

【例9.19】 设计一个通过GridView控件查找和排序student表中记录的网页WForm9-12。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-12的空网页。

（2）其设计界面如下图所示。将该网页的StyleSheetTheme属性设置为Blue。

WForm9-12网页设计界面

在该网页上设计如下事件过程：

```
Imports System.Data
Imports System.Data.OleDb
Partial Class ch9_WForm9_12
    Inherits System.Web.UI.Page
    Protected Sub Page_Load(ByVal sender As Object,
                            ByVal e As System.EventArgs) Handles Me.Load
        If Not Page.IsPostBack Then
            Dim mystr As String = _
              ConfigurationManager.AppSettings("myconnstring")
            Dim myconn As New OleDbConnection()
            myconn.ConnectionString = mystr
            myconn.Open()
            '以下设置DropDownList1的绑定数据
            Dim myds1 As New DataSet()
            Dim myda1 As New OleDbDataAdapter("SELECT distinct 民族 " & _
              "FROM student", myconn)
            myda1.Fill(myds1, "student")
            DropDownList1.DataSource = myds1.Tables("student")
            DropDownList1.DataTextField = "民族"
            DropDownList1.DataBind()
```

```
'以下设置DropDownList2的绑定数据
        Dim myds2 As New DataSet()
        Dim myda2 As New OleDbDataAdapter("SELECT distinct 班号 " & _
          "FROM student", myconn)
        myda2.Fill(myds2, "student")
        DropDownList2.DataSource = myds2.Tables("student")
        DropDownList2.DataTextField = "班号"
        DropDownList2.DataBind()
        '以下设置DropDownList3的数据
        DropDownList3.Items.Add("学号")
        DropDownList3.Items.Add("姓名")
        DropDownList3.Items.Add("性别")
        DropDownList3.Items.Add("民族")
        DropDownList3.Items.Add("班号")
        myconn.Close()
        TextBox1.Text = ""
        TextBox2.Text = ""
        RadioButton1.Checked = False
        RadioButton2.Checked = False
        RadioButton3.Checked = False
        RadioButton4.Checked = False
    End If
End Sub
```

```
Protected Sub Button1_Click(ByVal sender As Object,
                            ByVal e As System.EventArgs) Handles Button1.Click
    Dim condstr As String = ""
    '以下根据用户输入求得条件表达式condstr
    If TextBox1.Text <> "" Then
       condstr = "学号 Like '" & TextBox1.Text & "%'"
    End If
    If TextBox2.Text <> "" Then
       If condstr <> "" Then
          condstr = condstr & " AND 姓名 Like '" & TextBox2.Text & "%'"
       Else
          condstr = "姓名 Like '" & TextBox2.Text & "%'"
       End If
    End If
```

```
If RadioButton1.Checked = True Then
        If condstr <> "" Then
            condstr = condstr & " AND 性别 = '男'"
        Else
            condstr = "性别 = '男'"
        End If
    ElseIf RadioButton2.Checked = True Then
        If condstr <> "" Then
            condstr = condstr & "AND 性别 = '女'"
        Else
            condstr = "性别 = '女'"
        End If
    End If
```

```
    If condstr <> "" Then
        condstr = condstr & " AND 民族 = '" & _
          DropDownList1.SelectedValue + "'"
    Else
        condstr = "民族 = '" & DropDownList1.SelectedValue & "'"
    End If
    condstr = condstr & " AND 班号 = '" & _
      DropDownList2.SelectedValue & "'"
    Dim orderstr As String = ""
    '以下根据用户输入求得排序条件表达式orderstr
    If RadioButton3.Checked Then
        orderstr = DropDownList3.SelectedValue & " ASC"
    Else
        orderstr = DropDownList3.SelectedValue & " DESC"
    End If
    Server.Transfer("WForm9-12-1.aspx?" & "condstr=" & condstr & _
      "&orderstr=" & orderstr)
End Sub
```

```
Protected Sub Button2_Click(ByVal sender As Object,
                            ByVal e As System.EventArgs) Handles Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        RadioButton1.Checked = False
        RadioButton2.Checked = False
        RadioButton3.Checked = False
        RadioButton4.Checked = False
        DropDownList1.SelectedIndex = -1
        DropDownList2.SelectedIndex = -1
    End Sub
End Class
```

WForm9-12-1网页，其设计界面如下图所示，其中有一个HTML标签和一个GridView控件GridView1。GridView1控件的PageSize属性设为2，AllowPaging属性设为true，选择自动套用格式为“沙滩和天空”

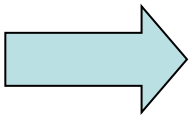
WForm9-12-1网页设计界面

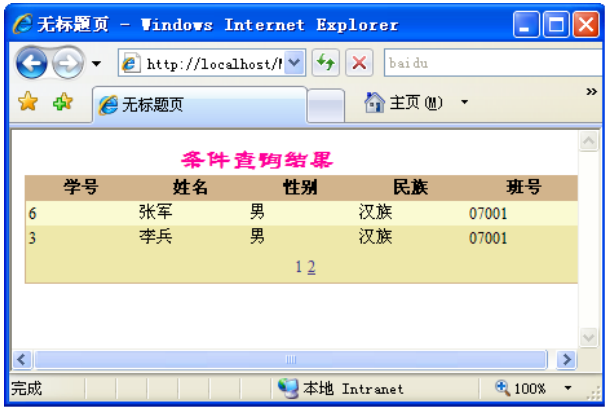
在WForm9-12-1网页上设计如下事件过程：

```
Imports System.Data
Imports System.Data.OleDb
Partial Class ch9_WForm9_12_1
    Inherits System.Web.UI.Page
    Private mydv As New DataView()
    Protected Sub Page_Load(ByVal sender As Object,
            ByVal e As System.EventArgs) Handles Me.Load
        Dim condstr As String = Request.QueryString("condstr")
        Dim orderstr As String = Request.QueryString("orderstr")
        Dim mystr As String = _
          ConfigurationManager.AppSettings("myconnstring")
        Dim myconn As New OleDbConnection()
        myconn.ConnectionString = mystr
        myconn.Open()
        Dim myds As New DataSet()
        Dim myda = New OleDbDataAdapter("SELECT" + "  * FROM student",
myconn)
        myda.Fill(myds, "student")
```

```
        mydv = myds.Tables("student").DefaultView
         '获得DataView对象mydv
        mydv.RowFilter = condstr     '过滤DataView中的记录
        mydv.Sort = orderstr         '对DataView中记录排序
        GridView1.DataSource = mydv
        GridView1.DataBind()
    End Sub
    Protected Sub GridView1_PageIndexChanging(ByVal sender As Object,
      ByVal e As System.Web.UI.WebControls.GridViewPageEventArgs)
      Handles GridView1.PageIndexChanging
        GridView1.PageIndex = e.NewPageIndex
        GridView1.DataSource = mydv
        GridView1.DataBind()
    End Sub
End Class
```


无标题页 - Windows Internet Explorer
http://localhost/
baidu
无标题页
主页(M)
查询条件设置
学号
姓名
性别
男
女
民族
汉族
班号
07001
排序条件设置
学号
升序
降序
确定
重置
WForm9-12
本地 Intranet
100%




无标题页 - Windows Internet Explorer
http://localhost/
baidu
无标题页
主页(M)
条件查询结果
学号 姓名 性别 民族 班号
6 张军 男 汉族 07001
3 李兵 男 汉族 07001
1 2
完成
本地 Intranet
100%

9.6.3 DetailsView控件

DetailsView控件显示来自数据源的单条记录的值，其中每个数据行表示该记录的一个字段。

它可与GridView控件结合使用，以用于主/详细信息方案。DetailsView控件支持的功能有：绑定到数据源控件（如AccessDataSource），内置插入功能，内置更新和删除功能，内置分页功能，以编程方式访问DetailsView对象模型以动态设置属性、处理事件等。

属性	说明
AllowPaging	获取或设置一个值，该值指示是否启用分页功能
AutoGenerateDeleteButton	获取或设置一个值，该值指示用来删除当前记录的内置控件是否在DetailsView控件中显示
AutoGenerateEditButton	获取或设置一个值，该值指示用来编辑当前记录的内置控件是否在DetailsView控件中显示
AutoGenerateInsertButton	获取或设置一个值，该值指示用来插入新记录的内置控件是否在DetailsView控件中显示
AutoGenerateRows	获取或设置一个值，该值指示对应于数据源中每个字段的行字段是否自动生成并在DetailsView控件中显示
DataItem	获取绑定到DetailsView控件的数据项
DataKey	获取一个DataKey对象，该对象表示所显示的记录的主键
DataMember	当数据源包含多个不同的数据项列表时，获取或设置数据绑定控件绑定到的数据列表的名称
DataSource	获取或设置对象，数据绑定控件从该对象中检索其数据项列表
GridLines	获取或设置DetailsView控件的网格线样式
PageCount	获取数据源中的记录数
PageIndex	获取或设置所显示的记录的索引
SelectedValue	获取DetailsView控件中的当前记录的数据键值

方法	说明
ChangeMode	将DetailsView控件切换为指定模式newMode。newMode的取值为DetailsViewMode.Edit（DetailsView控件处于编辑模式，这样用户就可以更新记录的值）、DetailsViewMode.Insert（DetailsView控件处于插入模式，这样用户就可以向数据源中添加新记录）或DetailsView.ReadOnly（DetailsView控件处于只读模式，这是通常的显示模式）
DataBind	将来自数据源的数据绑定到控件
DeleteItem	从数据源中删除当前记录
InsertItem	将当前记录插入到数据源中
UpdateItem	更新数据源中的当前记录

【例9.20】 设计一个通过DetailsView控件操作student表记录的网页WForm9-13。

其设计步骤如下：

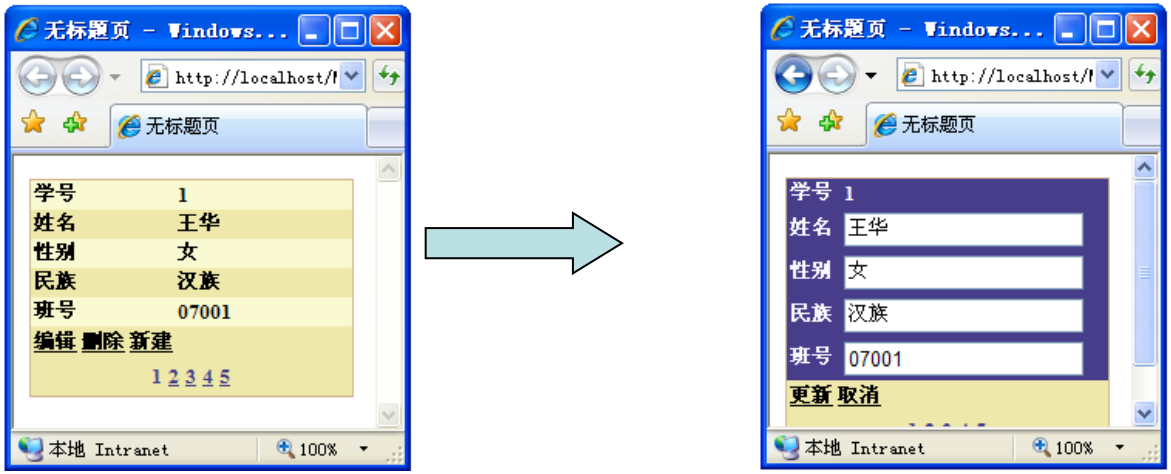
（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-13的空网页。

（2）向其中拖放一个DetailsView控件DetailsView1。

（3）在“DetailsView任务”列表中选择“新建数据源”命令，如下图所示。

（4）建立数据源AccessDataSource1的方法与前例相同，单击“完成”按钮。

（5）出现新的“GridView任务”列表如下图所示，勾选所有启动选项。选择自动套用格式为“沙滩和天空”


无标题页 - Windows...
http://localhost/l
无标题页
学号 1
姓名 王华
性别 女
民族 汉族
班号 07001
编辑 删除 新建
1 2 3 4 5
本地 Intranet
100%
无标题页 - Windows...
http://localhost/l
无标题页
学号 1
姓名 王华
性别 女
民族 汉族
班号 07001
更新 取消
本地 Intranet
100%

网页运行界面

【例9.21】设计一个通过GridView控件和DetailsView控件操作student表记录的网页WForm9-14。

其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-14的空网页。

（2）向其中拖放一个GridView控件GridView1。采用上例的方法设置其数据源控件为AccessDataSource1。通过“GridView任务”列表勾选“启动分页”、“启动排序”、“启动删除”和“启动选定内容”。指定自动套用格式为“穆哈咖啡”。设置PageSize属性为3。通过“字段”对话框将“删除”超链接移到最后并将前景颜色设置为Red（将ItemStyle→Font→ForeColor属性设置为Red）。

（3）向其中拖放一个DetailsView控件DetailsView1。指定其DataSourceID为AccessDataSource1，通过“DetailsView任务”列表勾选“启动插入”和“启动编辑”。指定自动套用格式为“沙滩和天空”。

（4）再添加两个起提示作用的HTML标签，网页设计界面如下图所示。

网页设计界面

在该网页上设计如下事件过程：

```
Protected Sub GridView1_SelectedIndexChanged(ByVal sender As Object,
      ByVal e As System.EventArgs) Handles GridView1.SelectedIndexChanged
    DetailsView1.ChangeMode(DetailsViewMode.ReadOnly)
    DetailsView1.PageIndex = GridView1.PageIndex*GridView1.PageSize _
      + GridView1.SelectedIndex
End Sub
Protected Sub GridView1_PageIndexChanged(ByVal sender As Object,
      ByVal e As System.EventArgs) Handles GridView1.PageIndexChanged
    DetailsView1.ChangeMode(DetailsViewMode.ReadOnly)
    DetailsView1.PageIndex = GridView1.PageIndex * GridView1.PageSize
End Sub
```


无标题页 - Windows Internet Explorer
http://localhost/
baidu
无标题页
主页(M)
学生记录操作
学号 姓名 性别 民族 班号
选择 6 张军 男 汉族 07001 删除
选择 2 孙丽 女 清族 07002 删除
1 2
单个学生记录编辑
学号 2
姓名 孙丽
性别 女
民族 清族
班号 07002
编辑 新建
完成
本地 Intranet
100%

网页运行界面

9.6.4 FormView控件

FormView控件与DetailsView控件在功能上有很多相似之处，也是用来显示数据源中的一条记录，分页显示下一条记录，支持数据的添加、删除、修改、分页等功能。

FormView控件与DetailsView控件之间的不同之处在DetailsView控件使用表格布局，在此布局中，记录的每个字段都各自显示一行，而FormView控件不指定用于显示距离的预定义布局，用户必须使用模板指定用于显示的布局。

9.6.5 DataList控件

1. 使用DataList控件

DataList控件默认情况下以表格形式显示数据。

其优点是用户可以通过模板定义为数据创建任意格式的布局，不仅可以横排数据，也可以竖排数据，还支持选择和编辑等。

DataList控件支持的模板

模板名称	说明
AlternatingItemTemplate	如果已定义，则为DataList中的交替项提供内容和布局。如果未定义，则使用ItemTemplate
EditItemTemplate	如果已定义，则为DataList中当前编辑的项提供内容和布局。如果未定义，则使用ItemTemplate
FooterTemplate	如果已定义，则为DataList的脚注部分提供内容和布局。如果未定义，将不显示脚注部分
HeaderTemplate	如果已定义，则为DataList的页眉节提供内容和布局。如果未定义，将不显示页眉节
ItemTemplate	为DataList中的项提供内容和布局所要求的模板
SelectedItemTemplate	如果已定义，则为DataList中当前选定项提供内容和布局。如果未定义，则使用ItemTemplate
SeparatorTemplate	如果已定义，则为DataList中各项之间的分隔符提供内容和布局。如果未定义，将不显示分隔符

【例9.22】 设计一个通过DataList控件显示student表记录的网页WForm9-15。

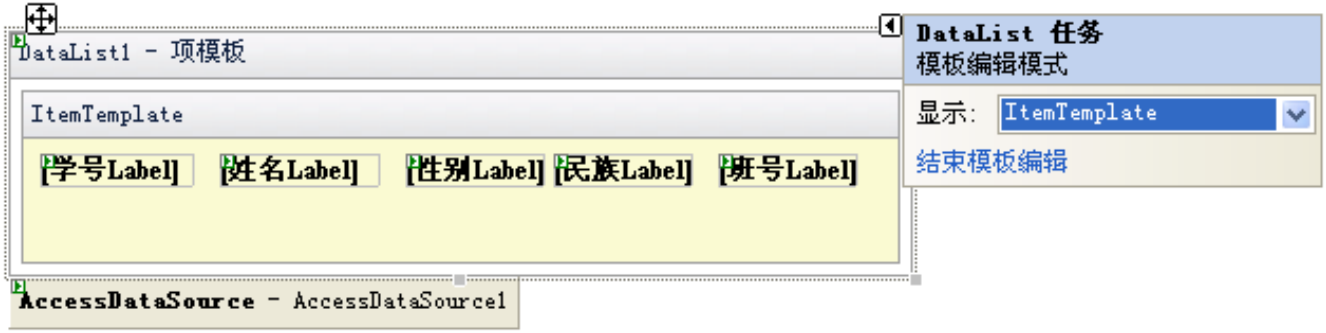
其设计步骤如下：

（1）在Myaspnet网站的ch9文件夹中添加一个名称为WForm9-15的空网页。

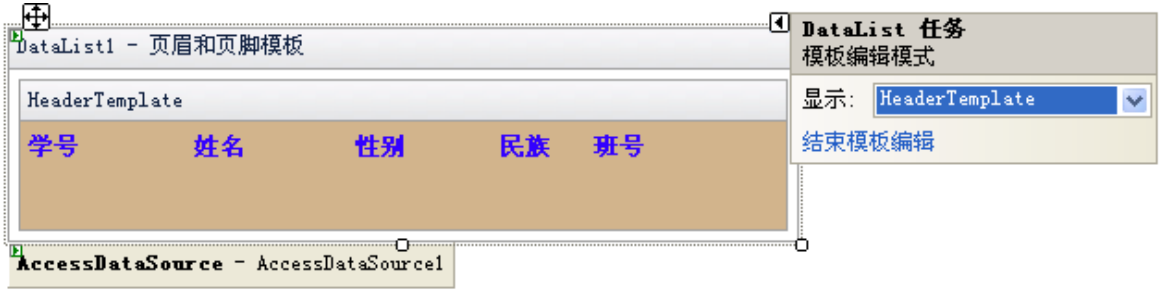
（2）向其中拖放一个DataList控件DataList1。

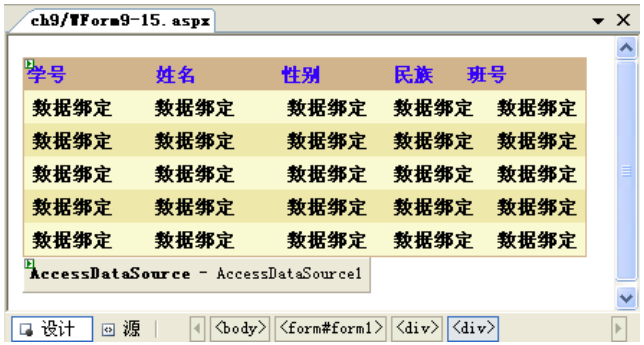
（3）通过“DataList任务”列表选择“新建数据源”命令新建数据源控件AccessDataSource1，用于访问student表。

（4）选择“DataList任务”列表中的“编辑模板”命令，定义ItemTemplate模板如下：



定义HeaderTemplate模板如下：




ch9/WForm9-15.aspx
学号 姓名 性别 民族 班号
数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
数据绑定 数据绑定 数据绑定 数据绑定 数据绑定
AccessDataSource - AccessDataSource1
设计
源
<body>
<form#form1>

设计界面


无标题页 - Windows Internet Explorer
http://localhost/
Live Search
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T) 帮助(H)
无标题页
学号 姓名 性别 民族 班号
1 王华 女 汉族 07001
3 李兵 男 汉族 07001
8 马棋 男 回族 07002
6 张军 男 汉族 07001
2 孙丽 女 清族 07002
本地 Intranet
100%

运行界面