

Web Based Programming

**Sanabad Institute
Of Higher Education**

**Provided by:
Farzad Zandi**



سرویس دهنده داده (DATA SERVER):

در صورت نیاز به بانک اطلاعاتی در برنامه ها نیاز به یک نرم افزار DBMS یا بخشی از آن است که امکان مدیریت بانک اطلاعاتی و دسترسی برنامه ها به آن را فراهم می کند. این سرویس دهنده درخواست ها را به زبانی مثل SQL گرفته و پاسخ آنرا به برنامه بر می گرداند.

این نرم افزار می تواند Access, Oracle, MySQL, SQL Server و ... باشد.



همیشه سرویس گیرنده، شروع کننده ارتباط است، وقتی درخواست (Request) به وب سرور رسید، نوع فایل درخواستی بررسی می شود:

- اگر مربوط به یک فایل HTML یا فایل های دیگری مثل txt و ... بود اگر فایل وجود داشته باشد، محتوای آن به درخواست کننده ارسال می شود.
- اگر مربوط به اسکریپت برنامه نویسی بود از نرم افزار مخصوص آن برای اجرا کمک گرفته می شود.
- اگر دستورات برنامه برای دسترسی به بانک اطلاعاتی بود از نرم افزار فراهم کننده داده ها استفاده می شود .



تعاریف اولیه (انواع اشیاء

۱- اشیایی که برای نگهداری و مدیریت داده استفاده می شوند:

Dataset, DataTable, ...

۲- اشیایی که برای اتصال با منبع داده به کار می روند:

**DataSource, Connection, Command,
DataReader**



تعاریف اولیه (فضای نام Data

کلاسهای اصلی در فضای نام System.Data قرار دارند.
این فضای نام (Name Space) خود شامل چند فضای نام دیگر است :

○ System.Data.SqlClient: شامل کلاس هایی است که برای کار با بانک های اطلاعاتی که به وسیله SQL Server ایجاد شده به کار می روند .

○ System.Data.OleDb: حاوی کلاس هایی است که برای کار با بانک های اطلاعاتی Ole (مثل Access) به کار می رود.



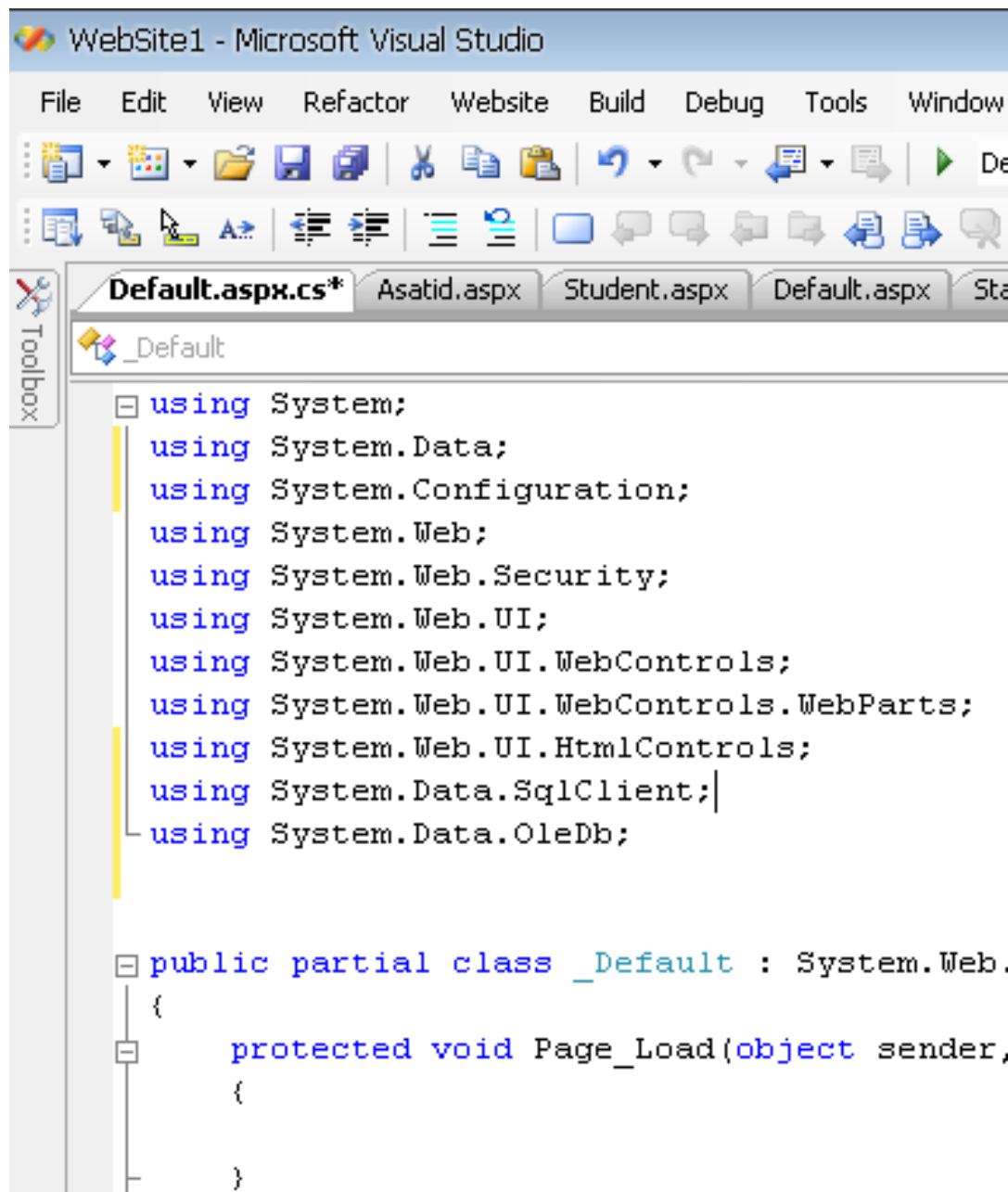
تعاریف اولیه (فضای نام Data

○ هر دو گروه بالا مثل هم عمل می کنند.

○ کلاس هایی که برای `sql server` طراحی شده اند از کارایی بهتر برخوردار بوده و فقط با `sql server` مایکروسافت کار می کنند .

○ اشیاء `oledb` با بانک اطلاعاتی دیگر مانند اکسس و اوراکل و ... کار می کنند. همچنین می توان برای کار با `sql server` نیز از `oledb` استفاده کرد.





```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.Data.OleDb;

public partial class _Default : System.Web.
{
    protected void Page_Load(object sender,
    {
    }
```

تعاریف اولیه (فضای نام Data

○ فضاهای نام باید با دستور **using** در برنامه وارد شود.



تعاریف اولیه (فضای نام Data

- OleDb ,SqlClient را سرویس دهنده اطلاعاتی گویند.
- کلاس های موجود در این سرویس دهنده ها شبیه یکدیگر هستند.
- کلاس های این سرویس دهنده ها زیاد بوده که در ادامه برخی از کلاس های مهم Sql Client را بررسی می شود.



SqlClient

عملیات های اصلی بانک اطلاعاتی :

1. نمایش اطلاعات از بانک اطلاعاتی
2. اضافه کردن رکورد جدید در جداول بانک اطلاعاتی
3. بروز رسانی اطلاعات یک رکورد در جداول بانک اطلاعاتی
4. حذف یک رکورد از جداول بانک اطلاعاتی



روش های نمایش اطلاعات از بانک اطلاعاتی:

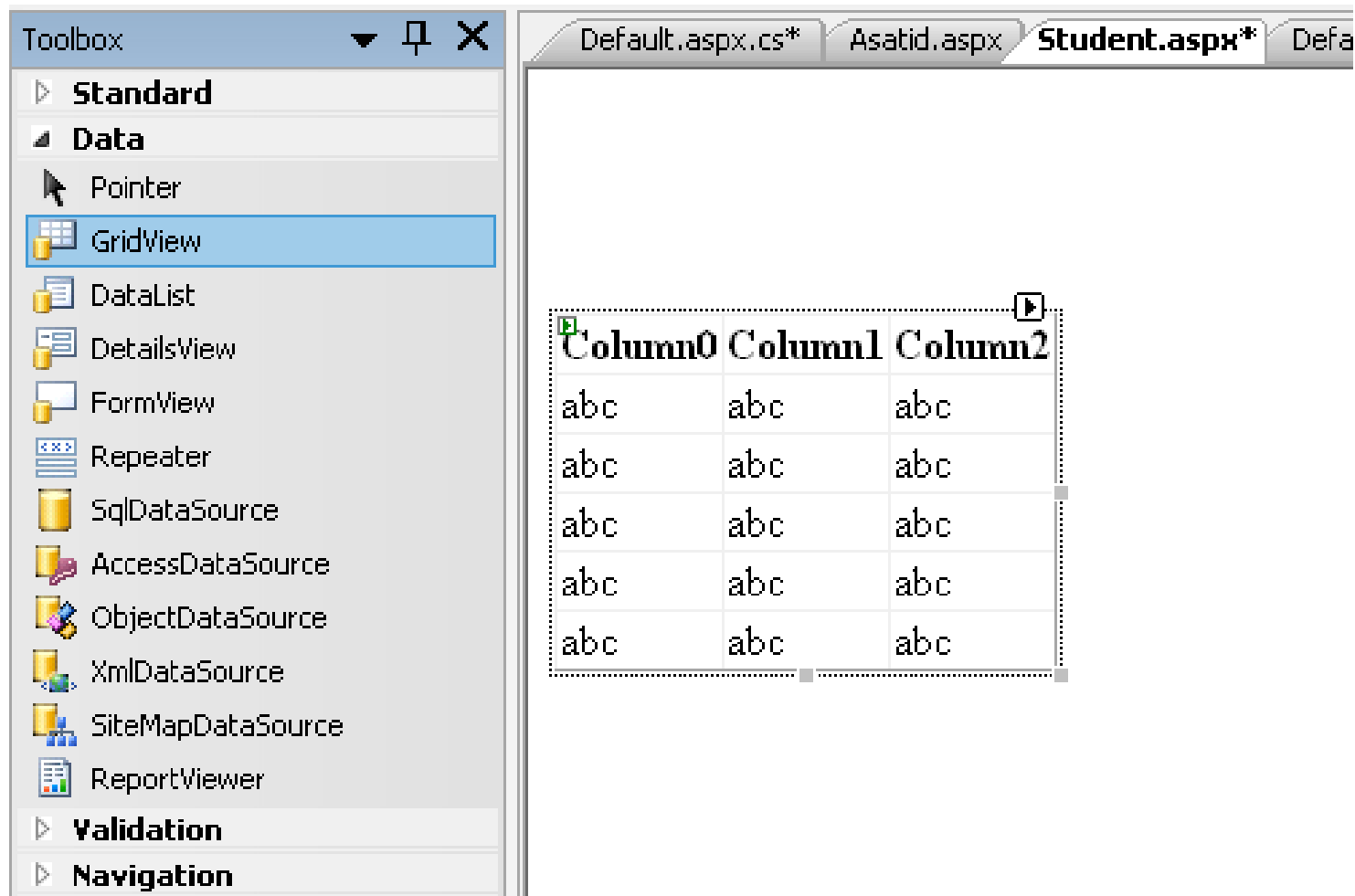
۱- بصورت **Visual** و با استفاده از کنترل های **Toolbox**

۲- با استفاده از کد نویسی



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله اول) کلیک بر روی کنترل **GridView** در سربرگ **Data**



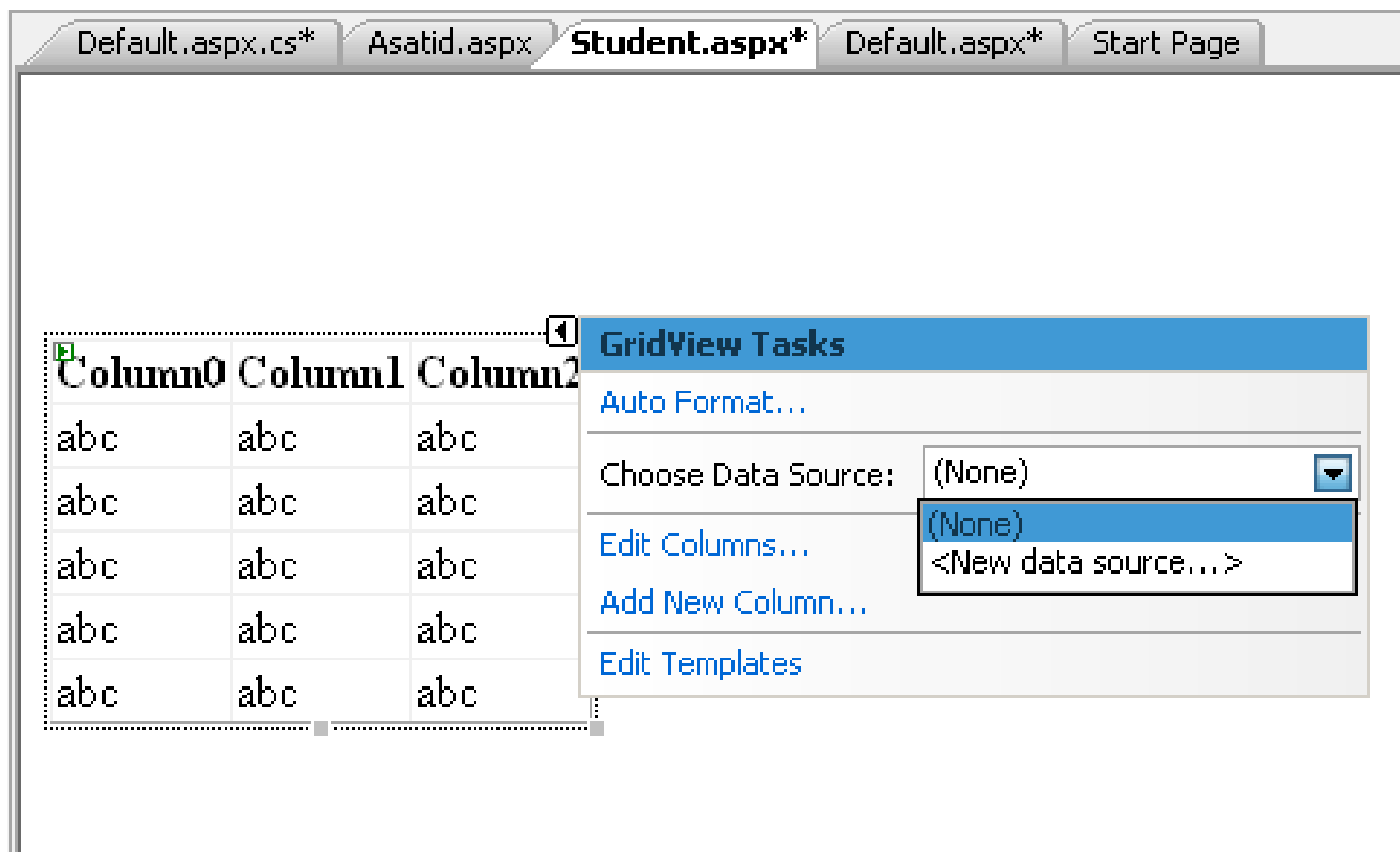
The screenshot shows the Visual Studio IDE. On the left, the **Toolbox** is open to the **Data** category, where **GridView** is selected. On the right, a web browser window displays the rendered output of the GridView control, which is a table with 5 rows and 3 columns. The columns are labeled **Column0**, **Column1**, and **Column2**, and each cell contains the text **abc**.

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله دوم) کلیک بر روی فلش GridView (بالا سمت راست آن) و سپس انتخاب Data Source (مخزن داده) برای آن



Default.aspx.cs* Asatid.aspx Student.aspx* Default.aspx* Start Page

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

GridView Tasks

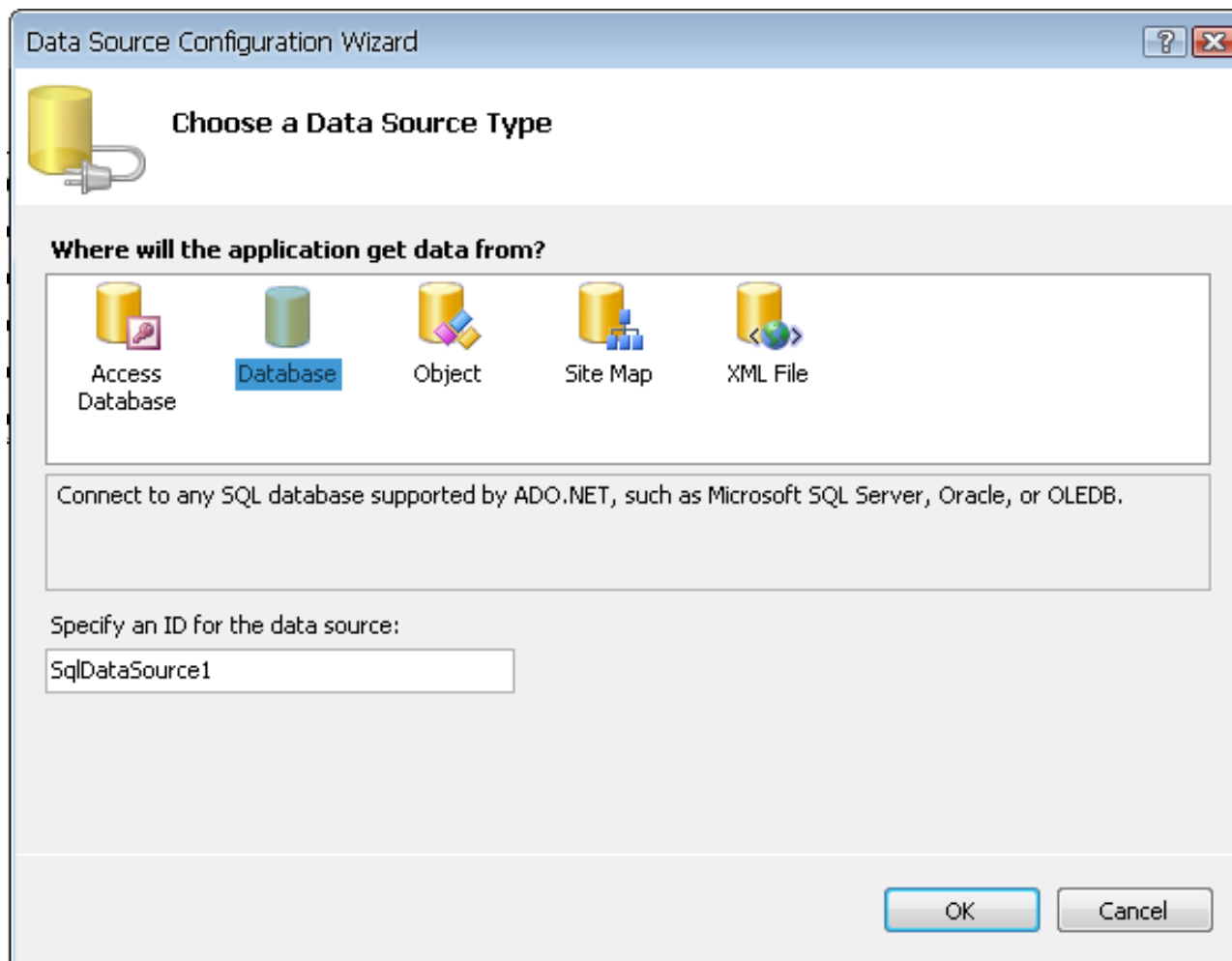
- Auto Format...
- Choose Data Source: (None)
- Edit Columns...
- Add New Column...
- Edit Templates

(None)
<New data source...>



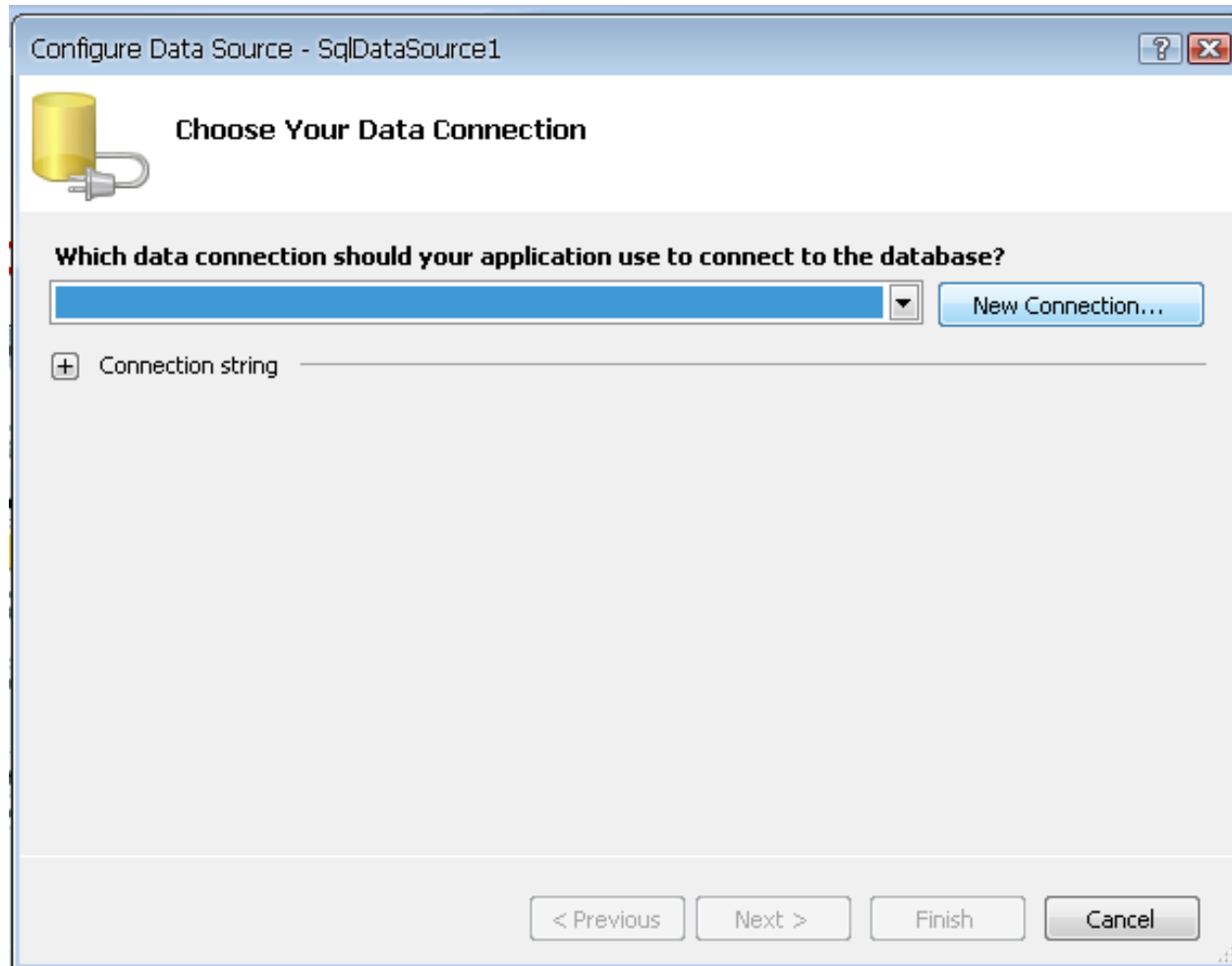
نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله سوم) انتخاب گزینه Database



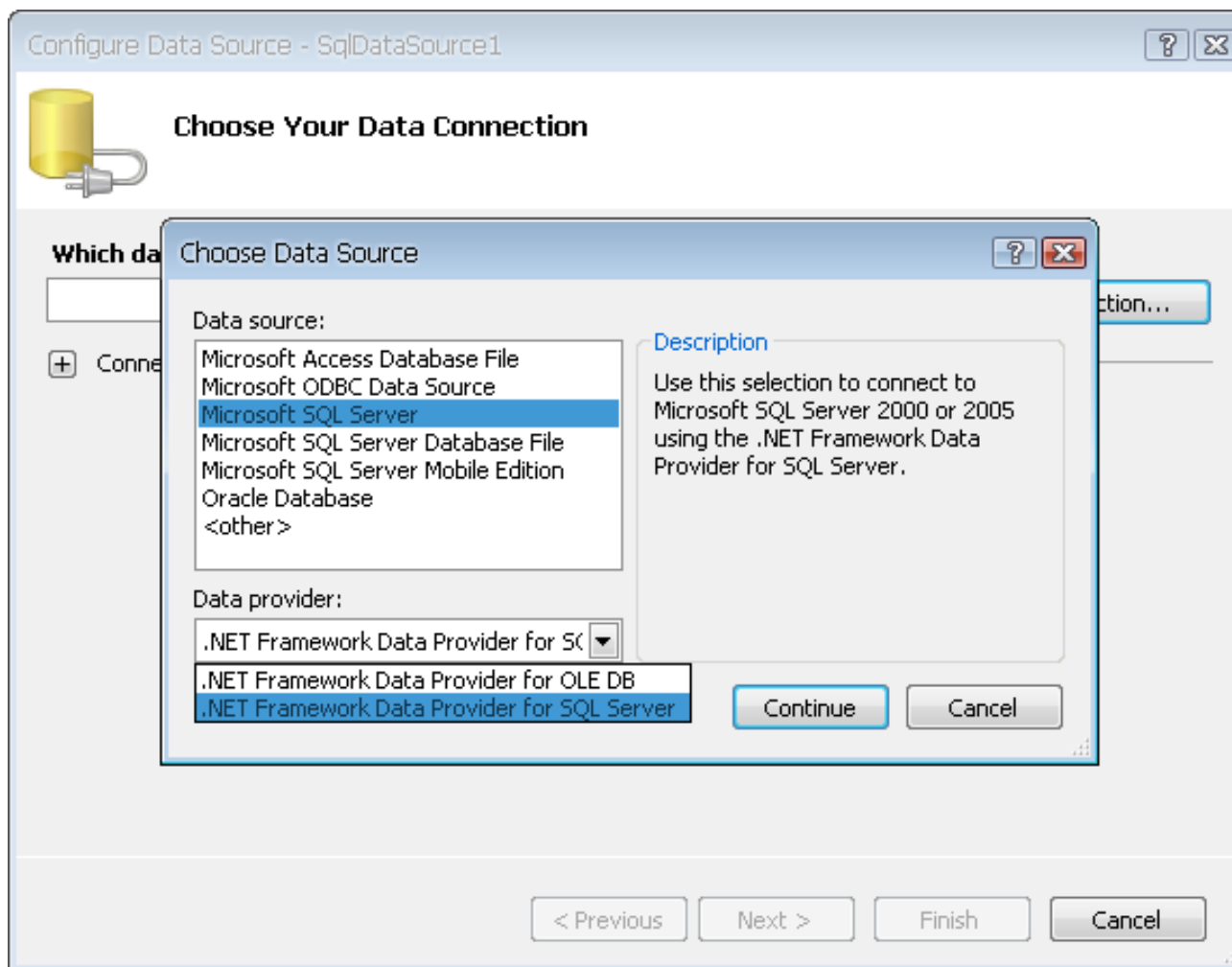
نمایش اطلاعات از بانک اطلاعاتی : روش **Visual**

مرحله چهارم) کلیک بر روی دکمه **New Connection**



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله پنجم) انتخاب نوع بانک اطلاعاتی



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
AGHIGHI\SQLEXPRESS Refresh

Log on to the server

Use Windows Authentication
 Use SQL Server Authentication

User name: sa
Password:
 Save my password

Connect to a database

Select or enter a database name:
|
GROCERY
master
model
msdb
tempdb
test
Uni

Advanced...

Test Connection OK Cancel

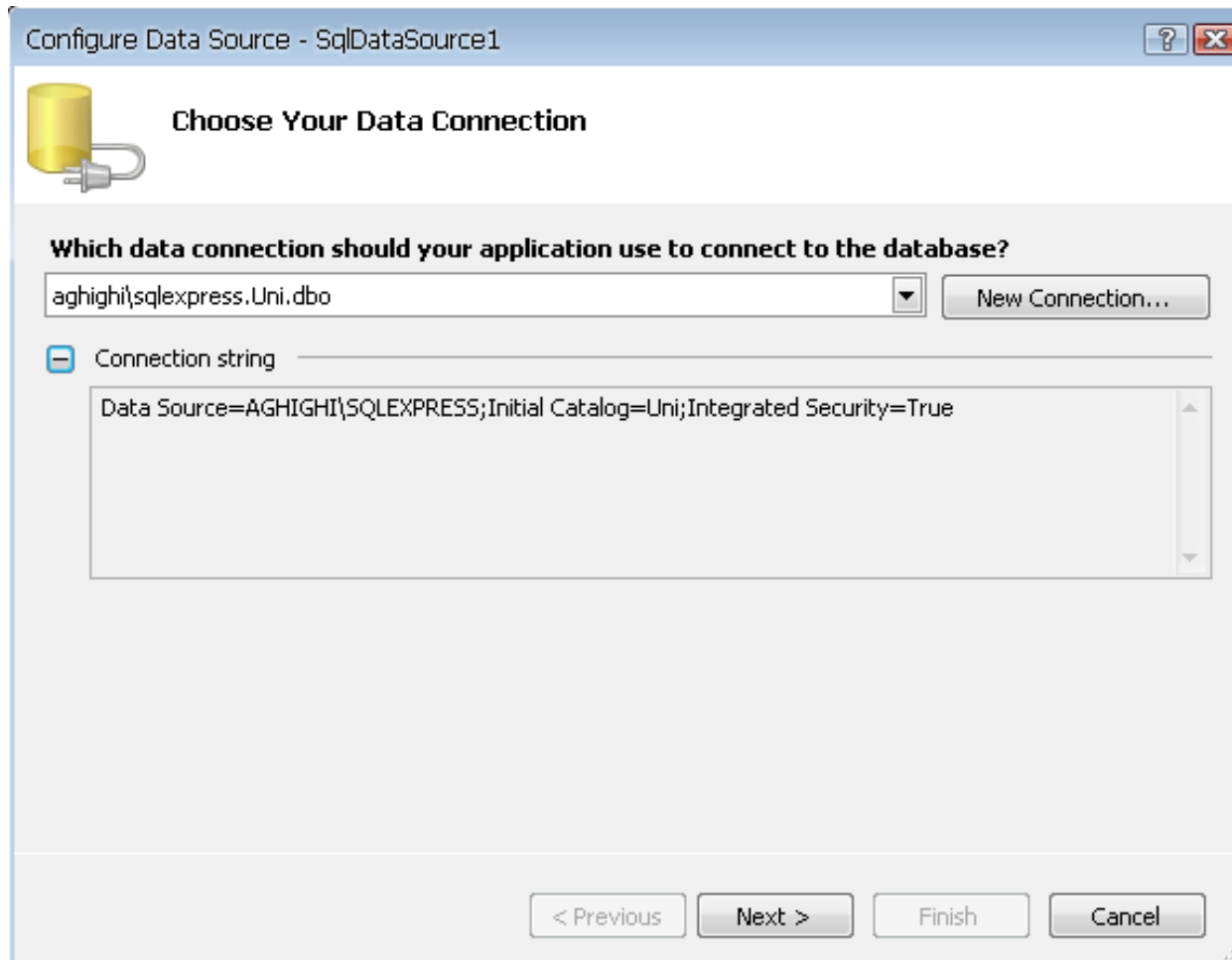
نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله ششم)
انتخاب نام بانک اطلاعاتی



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله هفتم) تشکیل رشته اتصال (Connection String)



Configure Data Source - SqlDataSource1

Choose Your Data Connection

Which data connection should your application use to connect to the database?

aghighi\sqlexpress.Uni.dbo

Connection string


Data Source=AGHIGHI\SQLEXPRESS;Initial Catalog=Uni;Integrated Security=True



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله هشتم) ذخیره رشته اتصال (Connection String)

Configure Data Source - SqlDataSource1



Save the Connection String to the Application Configuration File

Storing connection strings in the application configuration file simplifies maintenance and deployment. To save the connection string in application configuration file, enter a name in the text box and then click Next. If you choose not to do this, the connection string is saved in the page as a property of the data source control.

Do you want to save the connection in the application configuration file?

Yes, save this connection as:

UniConnectionString


< Previous Next > Finish Cancel



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله نهم) تنظیم دستور select بصورت بصری

Configure Data Source - SqlDataSource1



Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure

Specify columns from a table or view

Name:
student

Columns:

<input checked="" type="checkbox"/> *	<input type="checkbox"/> phone
<input type="checkbox"/> student_number	
<input type="checkbox"/> first_name	
<input type="checkbox"/> last_name	
<input type="checkbox"/> age	
<input type="checkbox"/> average	

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
SELECT * FROM [student]

< Previous Next > Finish Cancel



نمایش اطلاعات از بانک اطلاعاتی: روش Visual

مرحله دهم) پیچیده تر نمودن دستور select با استفاده از دکمه where

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column:
average

Operator:
>

Source:
None

SQL Expression:
[average] > @average

WHERE clause:

SQL Expression	Value

Parameter properties

Value:
17


Value:
17

Add

Remove

OK

Cancel



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله (۱) پیچیده تر نمودن دستور با استفاده از **Order By** و **Advanced**

Configure Data Source - SqlDataSource1

Configure the Select Statement

How would you like to retrieve data from your database?

Specify a custom SQL statement or stored procedure

Specify columns from a table or view

Name: student

Columns:

- *
- student_number
- first_name
- last_name
- age
- average

Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

```
SELECT * FROM [student] WHERE ([average] > @average)
```

< Previous Next > Finish Cancel



نمایش اطلاعات از بانک اطلاعاتی : روش Visual

مرحله ۱۲) تست دستور و کلیک بر روی Finish

Configure Data Source - SqlDataSource2

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

student_number	first_name	last_name	age	average	phone
88322111	Ali	Rezaee	21	17.5	021654987
88145256	Muhmmad	Irani	22	19	051123654
87216549	Reza	Tehrani	20	20	021564871

Test Query

SELECT statement:

```
SELECT * FROM [student] WHERE ([age] > @age)
```

< Previous Next > Finish Cancel



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس های مهم SqlClient

SqlConnection ○

SqlCommand ○

SqlDataAdapter ○

SqlDataReader ○

تذکر :

❖ برای استفاده از این کلاسها باید فضای نام System.Data.SqlClient را به برنامه اضافه کنیم.

❖ کلاس های پایه دیگر در ADO.net مانند : **DataSet**, **DataView** در فضای نام System.Data قرار دارند.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

مراحل نمایش اطلاعات از SQL Server

۱- ایجاد SqlConnection و تعیین نوع بانک اطلاعاتی و مسیر آن

۲- باز کردن Connection

۳- انجام دستورات بر روی بانک اطلاعاتی با استفاده از

SqlDataAdapter

۴- بستن Connection



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlConnection** (

- وظیفه ارتباط بین بانک اطلاعاتی و برنامه را برعهده دارد.
- هنگام ایجاد این شی از این کلاس باید پارامتری را به نام رشته اتصال (Connection string) تعریف کرده و به آن نسبت دهیم.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlConnection** (تعریف رشته اتصال

○ رشته اتصال یک متغیر رشته ای است که در آن پارامترهای لازم برای ایجاد یک اتصال به بانک اطلاعاتی تنظیم می شود.

```
// SQL 2005 connection string
```

```
string conStr = “server = .\sqlexpress; Initial  
catalog=Uni; integrated security = true ;”
```

```
// SQL 2000 connection string
```

```
string conStr = “server=(local); uid=; pwd=;  
database=Uni”
```



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlConnection** (تعریف رشته اتصال

```
string conStr = "server = .\\sqlexpress; Initial catalog=Uni;  
integrated security = true ;"
```

- بجای کلمه server می توان از کلمه data source استفاده نمود.
- نقطه نشان دهنده سرور محلی می باشد که می توان بجای نقطه از کلمه (local) نیز استفاده نمود.
- نام بعد از \\ نشان دهنده نمونه sqlserver می باشد که در محیط SQL تعریف می شود.
- بخش initial Catalog=Uni نام پایگاه داده sqlserver را که می خواهید باز کنید مشخص می کند.
- بخش integrated security = true نشان دهنده این است که از احراز هویت ویندوز استفاده شده است که می توان مقدار spsi را به جای true استفاده کرد.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlConnection** (تعریف رشته اتصال

```
string conStr = "server=(local); uid=; pwd=;  
database=Uni"
```

○ بجای کلمه initial Catalog می توان از کلمه database نیز استفاده نمود.

○ بجای کلمه uid می توان از کلمه user id نیز استفاده نمود.

○ بجای کلمه pwd می توان از کلمه password نیز استفاده نمود.

○ اگر احراز هویت از نوع ویندوزی باشد نیازی به نوشتن یا مقداردهی uid و pwd نمی باشد.

○ اگر احراز هویت از نوع SQL باشد می بایست uid و pwd مقداردهی شوند که مقدار پیش فرض uid مساوی sa و مقدار پیش فرض pwd خالی می باشد.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlConnection** (ایجاد **connection**

```
// create connection string
```

```
string conStr =
```

```
“server=(local); uid=; pwd=; database=Uni”
```

```
// create connection
```

```
SqlConnection sc =
```

```
new SqlConnection(conStr);
```



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlDataAdapter** (

○ برای اجرای دستور موردنظر روی بانک اطلاعاتی و استخراج نتایج آن بکار می رود.

○ خروجی آن می تواند درون یک شی از کلاس **Dataset** قرار گیرد ، بدین ترتیب **SqlDataAdapter** پلی بین جداول بانک اطلاعاتی و داده های موجود در **Dataset** می باشد



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

Adapter کلاس **SqlDataAdapter** (تعریف یک شی

```
SqlDataAdapter adapter = new  
SqlDataAdapter(string commandtext,  
sqlconnection)
```

(مثال)

```
// create SqlDataAdapter  
SqlDataAdapter adapter = new  
SqlDataAdapter ("select * from student", sc);
```

sc نام شی **connection** می باشد.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **SqlDataAdapter** (متد **Fill()**)

این کلاس دارای متد های گوناگونی می باشد که یکی از آنها متد **Fill()** بوده که دستور sql موجود را اجرا کرده و داده های برگشتی از جدول بانک اطلاعاتی را به **dataset** منتقل می کند.

SqlDataAdapter.Fill (ds, tableName);

- **ds** نام شی تعریف شده از کلاس **DataSet** است.
- **tableName** نام جدولی است که در **DataSet** ایجاد می کنیم تا داده های استخراج شده از بانک اطلاعاتی درون آن قرار گیرد. این فیلد از نوع رشته ای می باشد.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس DataSet (

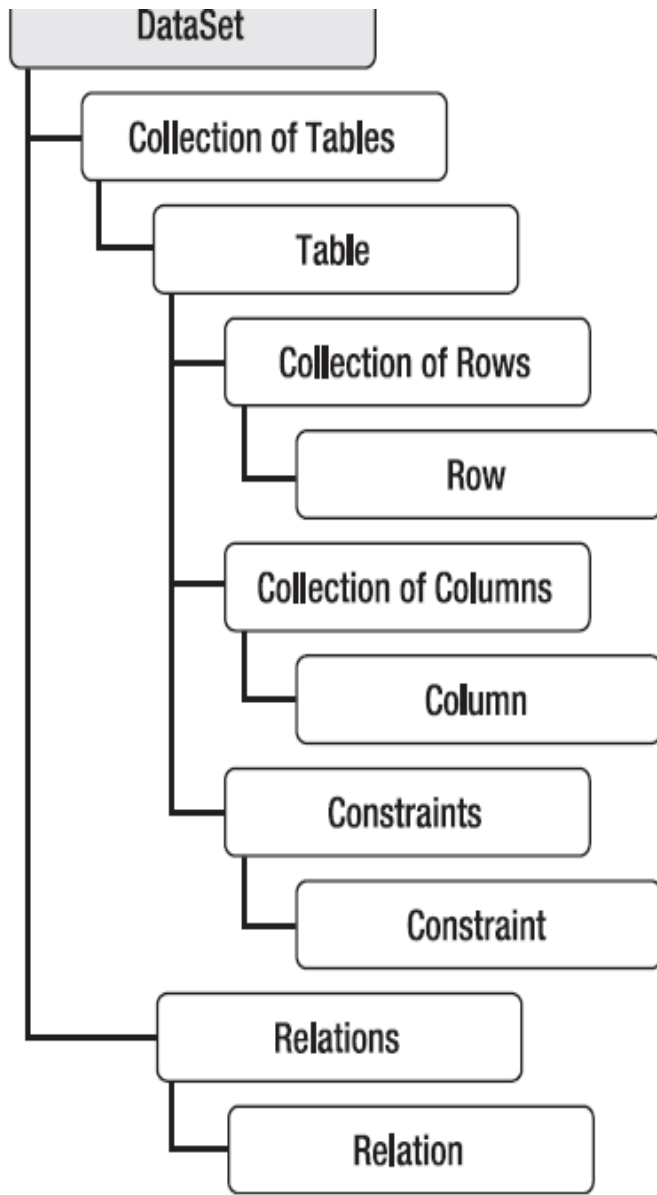
- کار یک بافر را انجام داده و اطلاعات حاصل از تقاضای بانک اطلاعاتی در آن ذخیره می شود.
- بعد از اینکه اطلاعات بانک اطلاعاتی را در آن ذخیره کردیم ، بدون اتصال با بانک اطلاعاتی قابل استفاده است.
- شامل عناصر :

DataView ✓

DataTable ✓ (شامل DataRow , DataColumn)

DataRelation ✓





نمایش اطلاعات از بانک اطلاعاتی :
روش کدنویسی

کلاس DataSet (

Figure 13-1. Dataset architecture



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس DataSet (

- DataSet شامل چند DataTable است که خود DataTable نماینده یک جدول بانک اطلاعاتی است و می تواند نتیجه یک پرس و جو (query) از بانک اطلاعاتی را نگهداری نماید.
- DataView می تواند روش دیگری جهت مشاهده محتویات یک DataTable باشد.
- می توان محتویات یک DataView را جستجو ، اصلاح و ... کرد.



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

کلاس **DataSet** (ایجاد یک شی **dataSet**

```
Dataset ds = new Dataset();
```



نمایش اطلاعات از بانک اطلاعاتی : روش کدنویسی

```
public partial class Student : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string conStr =
            "Data Source=.\SQLEXPRESS;Initial Catalog=Uni;Integrated Security=True";
        SqlConnection sc = new SqlConnection(conStr);
        SqlDataAdapter adapter =
            new SqlDataAdapter("select * from student where average>17", sc);
        DataSet ds = new DataSet();
        sc.Open();
        adapter.Fill(ds, "clever");
        sc.Close();
        GridView1.DataSource = ds.Tables["clever"];
        GridView1.DataBind();
    }
}
```

نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی : با استفاده از کلاس **SqlCommand**

■ مراحل اجرای یک دستور در بانک اطلاعاتی

- ۱- ایجاد **SqlConnection** و تعیین نوع بانک اطلاعاتی و مسیر آن
- ۲- ایجاد و تنظیم یک شی **SqlCommand** (مثل مشخص نمودن اتصال و رشته دستور موردنظر)

۲- باز کردن **Connection**

- ۳- انجام دستورات بر روی بانک اطلاعاتی با فراخوانی **یکی از متدهای** شی **SqlCommand**

۴- بستن **Connection**



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (

○ برای تنظیم یک دستور جهت اجرا روی بانک اطلاعاتی که هیچ رکوردی را بر نمی گرداند از **SqlCommand** استفاده می شود.

○ نحوه تعریف یک شی از این کلاس بصورت زیر است

```
SqlCommand command = new  
SqlCommand();
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (خاصیت **Connection**

برای مشخص کردن اتصال موردنظر به بانک اطلاعاتی است.

```
command.Connection = sc;
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (خاصیت **CommandType**

برای مشخص کردن نوع دستور است :

۱- متنی

```
command.CommandType =  
CommandType.Text;
```

۲- روال ذخیره شده

```
command.CommandType =  
CommandType.StoredProcedure;
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (خاصیت **CommandText**

■ برای تعریف دستور موردنظر جهت اجرا روی بانک اطلاعاتی است. در اینجا هر دستور بانک اطلاعاتی را می توان استفاده نمود.
(select,insert,delete,update)

■ این فیلد از نوع رشته ای بوده و می بایست کوتیشن های موردنیاز در دستور SQL در آن مورد توجه قرار بگیرد.
مثال) یک دستور insert را می توانیم به آن اختصاص دهیم.

command.CommandText =

```
"INSERT INTO student (st_number,  
first_name,last_name,age,average,phone) VALUES  
(`88111111` , `Ali` , `Irani` , 21 , 18 , `09120000000`)"
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

ExecuteNonQuery (کلاس **SqlCommand**) متد

✓ این متد ، دستور موردنظر را روی بانک اطلاعاتی اجرا نموده و خروجی خاصی ندارد.

✓ بنابراین نمی توان برای اجرای دستور **select** از آن استفاده نمود ولی برای اجرای دستورات **Insert, delete , update** مناسب است.

```
sc.Open();
```

```
command.ExecuteNonQuery();
```

```
sc.Close();
```



درج یک رکورد جدید در بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteNonQuery**)

```
string conStr = "server=(local); uid=; pwd=; database=Uni" ;  
SqlConnection sc = new SqlConnection(conStr);  
SqlCommand command = new SqlCommand();  
command.Connection = sc;  
command.CommandText =  
    "insert into student (st_number, first_name, last_name")  
    values ('88111111' , 'ali' , 'irani' );  
sc.Open();  
command.ExecuteNonQuery();  
sc.Close();
```



حذف یک رکورد از بانک اطلاعاتی :

ExecuteNonQuery (متد کلاس SqlCommand)

```
string conStr = "server=(local); uid=; pwd=; database=Uni" ;  
SqlConnection sc = new SqlConnection(conStr);  
SqlCommand command = new SqlCommand();  
command.Connection = sc;  
command.CommandText =  
"delete from student where average < 14";  
sc.Open();  
command.ExecuteNonQuery();  
sc.Close();
```



بروزرسانی یک رکورد در بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteNonQuery**)

```
string conStr = "server=(local); uid=; pwd=; database=Uni" ;  
SqlConnection sc = new SqlConnection(conStr);  
SqlCommand command = new SqlCommand();  
command.Connection = sc;  
command.CommandText = "update student set  
average=19.25 where first_name='ali' and  
last_name='irani'";  
sc.Open();  
command.ExecuteNonQuery();  
sc.Close();
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteScalar**)

✓ این متد ، دستور موردنظر را روی بانک اطلاعاتی اجرا نموده و می تواند خروجی داشته باشد.

✓ بنابراین برای اجرای دستور **select** و نیز دستورات **Insert, delete , update** مناسب است.

مثال (اجرای یک دستور **select**)

```
command.CommandText =
```

```
"select max(average) from Student";
```

```
sc.Open();
```

```
int max_avg = (int)command.ExecuteScalar();
```

```
sc.Close();
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteScalar**)

```
string conStr = "server=(local); uid=; pwd=; database=Uni" ;
```

```
SqlConnection sc = new SqlConnection(conStr);
```

```
SqlCommand command = new SqlCommand();
```

```
command.Connection = sc;
```

```
command.CommandText =
```

```
    "select max(average) from student";
```

```
sc.Open();
```

```
int x = (int)command. ExecuteScalar();
```

```
sc.Close();
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteReader**

✓ این متد ، دستور موردنظر را روی بانک اطلاعاتی اجرا نموده و می تواند خروجی داشته باشد.

✓ بنابراین برای اجرای دستور **select** و نیز دستورات **Insert, delete , update** مناسب است.

✓ برای استفاده از این متد نیاز به یک شی از کلاس **SqlDataReader** داریم.

✓ مزیت این روش سرعت بالای آن است.



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteReader**)

✓ این متد ، دستور موردنظر را روی بانک اطلاعاتی اجرا نموده و می تواند خروجی داشته باشد.

✓ بنابراین برای اجرای دستور **select** و نیز دستورات **insert, delete, update** مناسب است.

✓ خروجی این متد در یک شی از کلاس **SqlDataReader** قرار می گیرد ، پس برای استفاده از این متد نیاز به یک شی از این کلاس داریم.

✓ مزیت این روش سرعت بالای آن است.



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteReader**

```
SqlDataReader dr;  
SqlCommand cmd =  
    new SqlCommand("select last_name from student",  
        sc);  
sc.Open();  
dr = cmd.ExecuteReader();  
while ( dr.Read() )  
    DropDownList1.Items.Add(dr["last_name"].ToString(  
        ));  
dr.Close();  
sc.Close();
```



نمایش ، درج ، حذف و بروزرسانی رکوردهای بانک اطلاعاتی :

کلاس **SqlCommand** (متد **ExecuteReader**

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlDataReader dr;
    SqlCommand cmd =
        new SqlCommand("select last_name from student", sc);
    sc.Open();
    dr = cmd.ExecuteReader();
    while (dr.Read())
        DropDownList1.Items.Add(dr["last_name"].ToString());
    dr.Close();
    sc.Close();
}
```

