Connecting to database using ADO.net

تهیه کننده: صفری

بهار ۱۳۹۷

ADO.NET چیست؟

به مجموعه کلاس هایی (class) که اشیایی (object) از آن کلاس ها برای دسترسی به داده های یک دیتابیس در .net استفاده می شود ADO.NET گفته می شود.

تکنولوژی ADO.NET برای ذخیره و بازیابی داده ها در هر نوع منبع داده ای کاربرد دارد.همچنین این تکنولوژی در هر نوع برنامه نویسی از تحت وب WEB programming تا تحت ویندوز windows programming کاربرد دارد.


Win Forms
Web Forms
ADO.NET
Data Store

نحوه استفاده از ADO.NET

کلاس های اصلی و مشترک ADO در فضای نام(name space) System.data قرار دارند.این فضای نام خود شامل چند فضای نام دیگر است که مهمترین انها عبارتند از System.Data.Sqlclient و System.Data.OleDb

- فضای نام System.Data.Sqlclient حاوی کلاس هایی است که برای دسترسی به دیتابیس هایی از نوع Sql Server مورد استفاده قرار می گیرد.
- فضای نام System.Data.OleDb نیز حاوی کلاس هایی است که برای دسترسی به دیتابیس هایی مانند ACCESS استفاده می شود.

نحوه استفاده از ADO.NET

به کمک دستور Using می توان از فضا های نام استفاده کرد.

جهت استفاده از ADO.NET در ابتدای محیط برنامه نویسی کد زیر را اضافه می کنیم

```
using System.Data;
```

جهت استفاده از دیتا بیس ساخته شده در SQL در زبان برنامه نویسی C# کد زیر را به ابتدای محیط برنامه نویسی اضافه می‌کنیم

```
using System.Data.SqlClient;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
```

کلاس های موجود در System.Data.Sqlclient

در System.Data.Sqlclient چهار کلاس اصلی زیر وجود دارد:

- SqlConnection
- SqlCommand
- SqlParameter
- SqlDataAdapter

کلاس SqlConnection

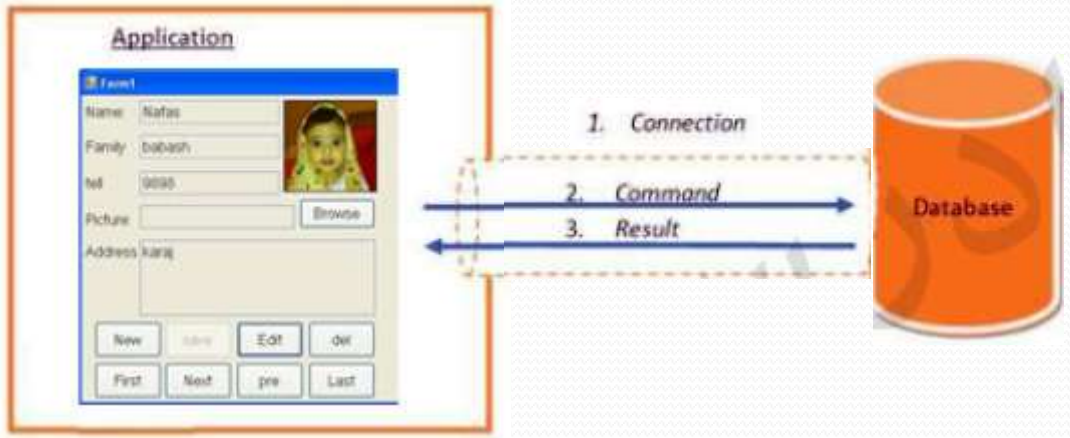
این کلاس وظیفه برقراری ارتباط بین برنامه و دیتابیس برنامه را بر عهده دارد.

برای استفاده از این کلاس کافی است یک شی یا object از این کلاس بسازیم.

مثال:

```
SqlConnection conn = new SqlConnection();
```

در اینجا شی conn از کلاس SqlConnection ساخته شده است.



اگر اتصال به دیتا بیس را با اتصال تلفنی مقایسه کنیم SqlConnection در اتصال به دیتا بیس مانند سیم کشی در اتصال تلفنی است.

همان طور که در اتصال تلفنی بعد از سیم کشی نیاز به شماره تلفن داریم در اتصال به دیتا بیس نیز نیاز به اطلاعات دیتا بیس داریم این اطلاعات را از طریق ConnectionString بدست میاوریم

کلاس SqlConnection دارای یک نوع داده به نام ConnectionString است که به **رشته اتصال** معروف است و تمام داده های لازم برای اتصال به یک دیتابیس را شامل می شود.

مثال: یک نمونه رشته اتصال یا ConnectionString

```
conn.ConnectionString = "Data Source=DatabaseServer;
Initial Catalog=Northwind; User ID=YourUserID;
Password=YourPassword";
```

نام پارامترِ رشته ی اتصال	توصیف
Data Source	سرور را تعریف می کند. می تواند دستگاه محلی، نام دامنه ی دستگاه، و یا آدرس IP باشد.
Initial Catalog(کاتالوگ اولیه)	نام پایگاه داده
Integrated Security(امنیت یکپارچه)	تنظیم برای SSPI به منظور ایجاد اتصال با ورود به ویندوزِ کاربر
User ID (نام کاربری)	نامِ پیکربندیِ کاربر در سرورِ SQL
Password(رمز عبور)	رمز ورود متناسب با SQL Server و ID کاربر

راه بدست آوردن اتوماتیک رشته اتصال

1. رفتن به پنجره server explorer در نرم افزار visual studio
2. راست کلیک و انتخاب گزینه add connection
3. وارد کردن نام سرور را در قسمت server name (اگر دیتابیس و برنامه دریک سیستم است می توان از نقطه به جای نام سرور استفاده کرد)
4. وارد کردن نام دیتابیس در قسمت enter database name
5. فشردن دکمه test connection جهت بررسی اتصال
6. فشردن دکمه advance و کپی کردن رشته اتصال از انتهای پنجره باز شده

بعد از اینکه با ایجاد connection string نحوه برقراری ارتباط با دیتابیس را مشخص کردیم، می توانیم با استفاده از متد های open و close در کلاس sqlconnection به دیتا بیس متصل شده یا اتصال خود را قطع کنیم.یک نمونه از این کار در قطعه کد زیر نشان داده شده است

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();









    conn.Close();
}
```

کلاس SqlCommand

- کلاس SqlCommand حاوی یک دستور SQL برای اعمال تغییرات بر روی داده های دریافت شده از دیتابیس می باشد.
- این دستور می تواند یک دستور select برای انتخاب داده های خاص،یک دستور insert برای درج داده های جدید، یک دستور delete برای حذف داده ها از دیتا بیس باشد.

- برای استفاده از این کلاس یک شی از این کلاس می سازیم
- SqlCommand sc = new SqlCommand();

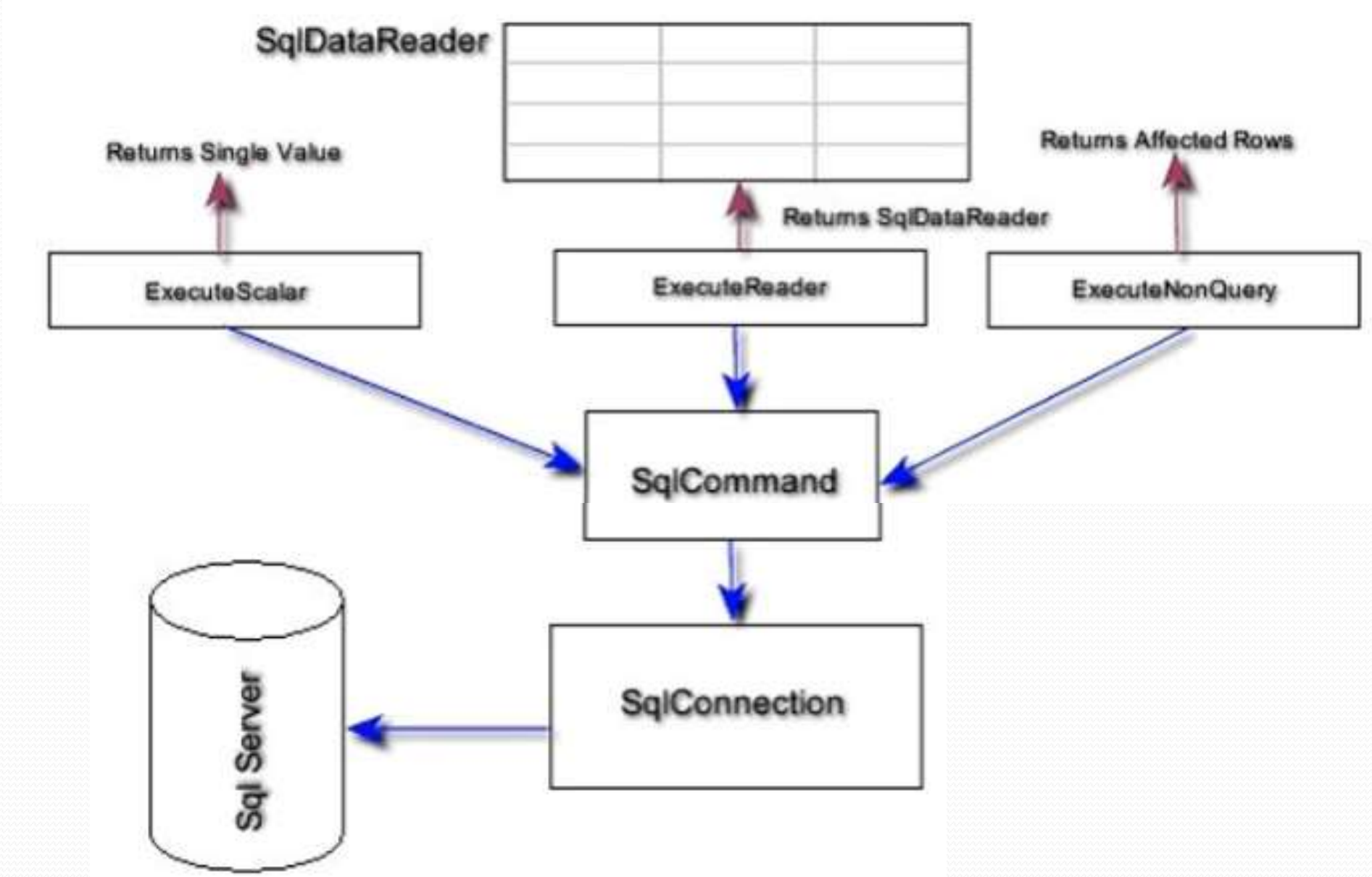
- خاصیت connection این کلاس را برابر با شی اتصال قرار می دهیم تا به کمک آن اتصال، دستور اجرا شود
- خاصیت commandtext حاوی متنی دستور SQL است که باید بر روی داده ها اجرا شود.

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "insert into friend values('ali3', 'safari')";

    conn.Close();
}
```

متدهای کلاس sqlcommand

- ExecuteNonQuery();اگر فرمان ما هیچ خروجی ای نداشت از این متد استفاده می کنیم مثلا فرمان insert و delete و update هیچ مقداری را بر نمی گرداند
- ExecuteReader(); اگر خروجی فرمان سطر یا ستونی از دیتا بیس باشد از این متد استفاده می کنیم مثلا فرمان select
- ExecuteScalar(); : اگر خروجی ما یک مقدار داده را برگرداند از این متد استفاده می شود مثلا select count()


SqlDataReader
Returns Single Value
Returns SqlDataReader
Returns Affected Rows
ExecuteScalar
ExecuteReader
ExecuteNonQuery
SqlCommand
SqlConnection
Sql Server

در اینجا چون دستورما insert است و هیچ مقداری را برنمی گرداند از ExecuteNonQuery(); استفاده کرده ایم.

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "insert into friend values('ali3', 'safari')";
    sc.ExecuteNonQuery();
    conn.Close();
}
```

دستورما delete است و هیچ مقداری را برنمی گرداند از ExecuteNonQuery(); استفاده کرده ایم.

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "delete from friend where name='ali1'";
    sc.ExecuteNonQuery();
    conn.Close();
}
```

دستورما update است و هیچ مقداری را برنمی گرداند از ExecuteNonQuery(); استفاده کرده ایم.

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "update friend set family='karimi' where name='ali3'";
    sc.ExecuteNonQuery();
    conn.Close();
}
```

- دستور select count که یک مقدار داده را بر می گرداند و از ExecuteScalar() استفاده کرده ایم

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "select count(name) from friend";
    int count=(int)sc.ExecuteScalar();
    conn.Close();
    Console.WriteLine("the nomber of name in dataset is :"+count);
    Console.ReadKey();

}
```

- دستور select که سطر یا ستونی از دیتا بیس را بر می گرداند و از ExecuteReader() استفاده کرده ایم

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "select * from friend";
    SqlDataReader dr= sc.ExecuteReader();
    while (dr.Read())
    {
        Console.WriteLine("friend is : "+dr[0]+dr[1]);
    }
    conn.Close();
    Console.ReadKey();
}
```

خاصیت parameters از کلاس sqlcommand

پارامتر ها متغیرهایی هستند که در یک دستور SQL قرار می گیرند و می تواند در زمان اجرای برنامه جای خود را به عباراتی خاص عوض کنند.

این متغییر ها با علامت @ در یک دستور مشخص می شوند.

متد AddWithValue نام یک پارامتر و متغیری که مقدار مربوط به آن را در زمان اجرای برنامه نگهداری می کند به عنوان پارامتر دریافت کرده و آن را به لیست parameters اضافه میکند.

```
private void button1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "insert into friend values(@p1,@p2)";
    sc.Parameters.AddWithValue("p1",textBox1.Text);
    sc.Parameters.AddWithValue("p2",textBox2.Text);
    sc.ExecuteNonQuery();
    conn.Close();
}
```

خاصیت parameters از کلاس sqlcommand

می توان خروجی دستور select را در یک نوع داده Datatable ذخیره کرد و سپس از طریق کنترلر dataGridView در ویندوز فرم نشان داد.

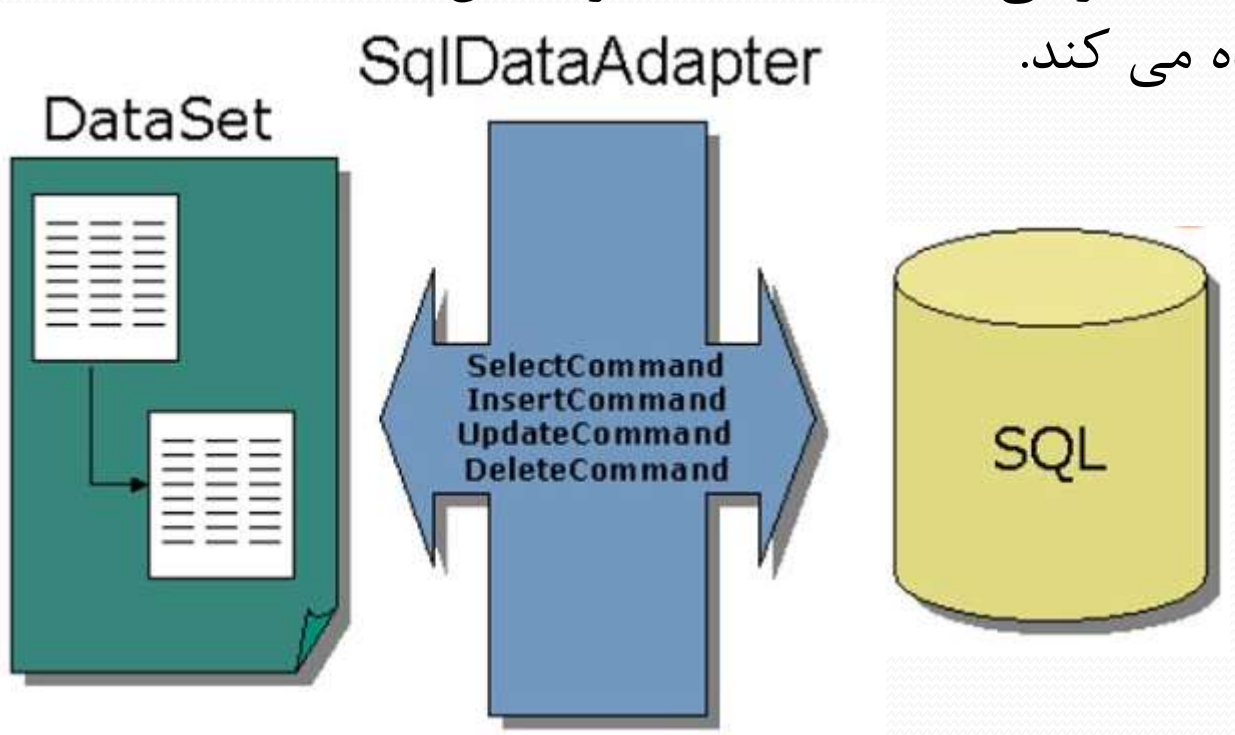
```
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    conn.Open();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "select * FROM friend ";
    SqlDataReader dr= sc.ExecuteReader();
    DataTable dt = new DataTable();
    dt.Load(dr);
    dataGridView1.DataSource = dt;
    conn.Close();
}
```

کلاس dataset

- هنگام استفاده از دستور select، بخشی از اطلاعات یک دیتابیس که درون harddisk ذخیره شده است، انتخاب شده و آن اطلاعات روی RAM قرار می گیرد. وظیفه نگهداری اطلاعات به دست آمده روی RAM برعهده کلاس dataset است.
- یک شی (object) از کلاس dataset می تواند شامل چندین جدول باشد که هر یک از انها به وسیله یک کنترل DataTable مشخص می شوند.
- DataSet ds = new DataSet();

کلاس sqlDataAdapter

- کلاس DataAdapter مانند پلی بین جداول دیتا بیس و جداول موجود در حافظه RAM که به وسیله DataSet نگهداری می شود عمل می کند.
- می توانیم بگوییم که کلاس DataAdapter برای دسترسی به بانک اطلاعات از کلاس sqlcommand و SqlConnection استفاده می کند.



کلاس sqlDataAdapter

- کلاس sqlDataAdapter دارای چهار خاصیت زیر است:
- Selectcommand : از طریق این خصوصیت DataAdapter ، دستور انتخاب (select) را بر روی دیتابیس اجرا کرده و نتایج را در کلاس هایی مانند Dataset قرار می دهند
- Insertcommand : از طریق این خصوصیت DataAdapter ، دستور اضافه کردن را بر روی دیتابیس اجرا کرده و نتایج را در کلاس هایی مانند Dataset قرار می دهند.
- Deletecommand : از طریق این خصوصیت، DataAdapter ، دستور حذف را بر روی دیتابیس اجرا کرده و نتایج را در کلاس هایی مانند Dataset قرار می دهند.
- Updatecommand : از طریق این خصوصیت DataAdapter ، دستور ویرایش داده را بر روی دیتابیس اجرا کرده و نتایج را در کلاس هایی مانند Dataset قرار می دهند

متد Fillوupdate از کلاس DataAdapter

با استفاده از متد fill کلاس DataAdapter می توان دستور SQL موجود در خاصیت Selectcommand را در دیتابیس اجرا کرده و سپس داده های برگشتی از اجرای این دستور را درون یک Dataset در حافظه قرار دهیم.

هنگامی که با استفاده از DataAdapter داده هایی را درون یک Dataset قرار می دهید ابتدا یک شی جدید از نوع DataTable ایجاد شده ، داده ها درون آن قرار داده می شوند و به DataSet اضافه می شوند.

با استفاده از متدupdate کلاس DataAdapter می توان دستور SQL موجود در هر سه خاصیت Insertcommand, Deletecommand, Updatecommand را که در Dataset تغییرات ایجاد کرده را درون دیتابیس در حافظه ذخیره نماییم.

نمونه کد استفاده از کلاس SqlDataAdapter

```
static void Main(string[] args)
{
    SqlConnection conn = new SqlConnection();
    conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated security=True";
    SqlDataAdapter da = new SqlDataAdapter();
    DataSet ds = new DataSet();
    SqlCommand sc = new SqlCommand();
    sc.Connection = conn;
    sc.CommandText = "select * from friend";
    da.SelectCommand = sc;
    da.Fill(ds, "T1");

}
```

نمونه کد استفاده از کلاس SqlDataAdapter

```
SqlDataAdapter adapter;
DataSet ds;
DataTable dt;
string connstring = "Data Source=SAFARI\\SQLEXPRESS;Initial Catalog=classa;Integrated Security=True";
private void btndataset_Click(object sender, EventArgs e)
    {
        // Using DataSet
        adapter = new SqlDataAdapter("select * from student_detail", connstring);
        ds = new DataSet();
        adapter.Fill(ds);// fill the DataSet
        dataGridView1.DataSource = ds.Tables[0];
    }
private void btnupdate_Click(object sender, EventArgs e)
    {
        SqlCommandBuilder cmb = new SqlCommandBuilder(adapter);
        adapter.Update(ds);// updating changes
    }
```

برای نمایش محتویات کلاس Dataset از کنترلdatagridveiw استفاده می شود.

```
SqlConnection conn = new SqlConnection();
conn.ConnectionString = "Data Source=.;Initial Catalog=univercity;Integrated Security=True";
SqlCommand scinsert = new SqlCommand();
scinsert.Connection = conn;
scinsert.CommandText = "insert into friend values(@p1,@p2)";
scinsert.Parameters.AddWithValue("p1",txtname.Text);
scinsert.Parameters.AddWithValue("p2", txtfname.Text);
SqlCommand scselect = new SqlCommand();
scselect.Connection = conn;
scselect.CommandText = "select * from friend";
SqlDataAdapter sd = new SqlDataAdapter();
DataSet ds = new DataSet();
sd.SelectCommand = scselect;
sd.InsertCommand = scinsert;
sd.InsertCommand.Connection.Open();
sd.InsertCommand.ExecuteNonQuery();
sd.InsertCommand.Connection.Close();
sd.Fill(ds, "t1");
dataGridView1.DataBindings.Clear();
dataGridView1.DataBindings.Add("datasource",ds,"t1");
```