ADO.NETیا ActiveX Data Object مجموعه کامپوننت هایی است که برای دسترسی به داده های بانک اطلاعاتی در .NET. استفاده می شود . گر چه در این سایت مقالات و کتابها و سورسهایی در رابطه با استفاده از ADO.NET در سی شارپ وجود دارد ولی آوردن این مبحث مهم در سایت خالی بود که در این پست قرار میدهم .

برای مشاهده این مبحث به ادامه مطلب بروید.

تعریف : به مجموعه کامپوننت هایی که برای دسترسی به داده های بانک اطلاعاتی در .NET.
استفاده می شود Ado.Net گفته می شود.
می توانیم بگوییم که دو روش برای اتصال به بانک وجود دارد اتصال متصل(online)و غیر متصل.

کار با بانک اطلاعاتی بصورت متصل:
هنگام استفاده از اشیا و متد های مربوط به این نوع اتصال ارتباط بین فرم و بانک در تمام مدت باید برقرار باشد برای همین موضوع این نوع کار با بانک متصل می نامند. معمولا این نوع کار با بانک از سرعت بسیار بالاتری نسبت به روش دوم که غیر متصل نام دارد برخوردار است.

کلاسهای ارتباط با بانکهای اطلاعاتی:
توجه - در اینجا در مثالها از بانک اطلاعاتی SqlServer استفاده میشود.
برای استفاده از این کلاس ها باید فضای نام زیر را با استفاده از using به برنامه اضافه کرد.

System.Data.SqlClient ;

1- کلاس SqlConnection :

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

-هنگامی که می خواهید یک نمونه از ان کلاس را ایجاد کنید باید پارامتری را به نام Connection String به آن ارسال کنید.

Connection Stringرشته ای است که شامل تمام داده های مورد نیاز برای برقراری اتصال به یک بانک اطلاعاتی می شود.

ویژوال استودیو با استفاده از ویزارد AddConnection و اطلاعاتی که کاربر وارد می کند چنین رشته ای را ایجاد کرده و در اختیار SqlConnection قرار می دهد.

اغلب بهتر است که متن لازم برای ConnectionString را خودمان بنویسیم که به صورت زیر است:

"Data Source = local;Initial Catalog = university;Integrated Security = True"

در کد بالا local :نام سروری است که بانک بر روی آن قرار دارد که در اینجا چون سرور خود کامپیوتر ماست مقدار آن را local قرار داده ایم که می توانیم به جای آن از "." (نقطه) هم استفاده کنیم.

Universityنام بانکی است که قرار است ما با اطلاعات آن کار کنیم.

-متن ConnectionString به صورت پارامتر به شیئ جدید SqlConnection فرستاده می شود به صورت زیر:

```
SqlConnection Con = New SqlConnection( "Data Source = local;Initial Catalog = university;Integrated Security = True");
```

در کد بالا Con : یک نمونه جدید از نوع SqlConnection است که برای استفاده از آن آن را ساخته ایم.

متصل و قطع کردن اتصال به یک بانک اطلاعاتی:

با استفاده از متدهای Open و Close در کلاس SqlConnection به بانک متصل شده و یا اتصال خود را قطع کنیم.

Con.Open(); و Con.Close();

2- کلاس SqlCommand :

این کلاس حاوی یک دستور Sql برای اجرا بر روی داده های دریافت شده از بانک اطلاعاتی است این دستور می تواند یک دستور SELECT برای انتخاب داده هایی خاص ، یک دستور INSERT برای درج داده های جدید در بانک اطلاعاتی ، یک دستور DELETEبرای حذف داده ها از بانک اطلاعات و یا حتی فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی می باشد.

ایجاد آن به صورت زیر می باشد:

SqlCommand Cmd = New SqlCommand();

نکته – در برنامه های بانک اطلاعاتی معمولا از اشیای ایجاد شده از کلاس SqlCommand به تنهایی استفاده نمی کنند بلکه آنها را همراه با DataSet ها و DataAdapter ها به کار می برند.

همچنین اشیای SqlCommand می توانند به همراه اشیای ایجاد شده از کلاس DataReader مورد استفاده قرار گیرند.

خاصیت connection کلاس: SqlCommand

قبل از اینکه بتوانیم از یک شئ از کلاس SqlCommand استفاده کنیم باید بعضی از خاصیت های آن را تنظیم کنیم اولین خاصیتی که باید تنظیم شود خاصیت Conectionاست.

این خاصیت می تواند یک مقدار از نوع SqlConnection را دریافت کند :

```
Cmd .Connection = Con ;
```

توضیح کد بالا :

Cmdهمان شیئی است که از کلاسSqlCommand قبلا ساخته ایم.

Conشیئی است که از نوع SqlConnection قبلا ساخته ایم .

خاصیت CommandText کلاس: SqlCommand

خاصیتبعدی که باید از کلاس SqlCommand تنظیم شود خاصیت CommandText است.

این خاصیت متنی را دریافت می کند که می تواند حاوی یک دستور Sql و یا فراخوانی یک پروسیجر ذخیره شده در بانک اطلاعاتی باشد که باید روی داده ها اجرا شود.

متد: ExecuteNonQuery

این متد دستورات را بر روی بانک اطلاعات اجرا می کند.

برای استفاده از این متد باید ابتدا اتصال خود را به بانک اطلاعاتی برقرار کنید سپس با فراخوانی این متد دستور موجود در شئ Command را اجرا کنید.

کلاس : SqlDataReader

از طریق این کلاس می توانیم اطلاعات را از بانک دریافت کنیم . که به صورت زیر تعریف می شود:

```
SqlDataReader dr;
```

متد : ExecuteReader

برای دریافت اطلاعات از بانک از متد ExecuteReader شی تقاضا استفاده می کنیم ، یک شی از نوع SqlDataReaderتعریف کرده و مقدار ExecuteReader را برابر آن قرار می دهیم

```
SqlDataReader dr = Cmd.ExecuteReader();
```

سپس از طریق متد Read شی DataReader اطلاعات را در یک حلقه واکشی می کنیم .

ارائه مراحل کلی یک ارتباط و مثال

برای تقاضا از یک جدول در بانک اطلاعاتی باید مراحل زیر طی شود:

-1اضافه نمودن فضا نام های مورد نیاز : برای ارتباط با بانک های اطلاعاتی به فضا نام System.Data و همچنین فضا نام System.Data.SqlClient برای کار با دیتابس SQLServer و فضا نام System.Data.Oledbبرای کار با بانک Access نیازمندیم .

-2تعیین رشته اتصال : (ConnectionString) رشته اتصال رشته شامل تنظیماتی جهت اتصال به بانک اطلاعاتی می باشد ، این رشته برای هر بانک متفاوت خواهد بود

-3تعیین شی اتصال : (Connection Object) کلاسی است برای برقراری ارتباط با بانک اطلاعاتی است ، این شی که از کلاس DbConnectionارث بری می کند اعمالی مانند باز و بسته کردن اتصال با بانک را از طریق رشته اتصال انجام می دهد .

-4تعیین رشته تقاضا : (Command Text) همان دستورات SQL است که جهت یک تقاضا ارائه می گردد ، این تقضا جهت دریافت اطلاعات (Select) درج اطلاعات (Insert) ، ویرایش اطلاعات (Update)و یا حذف اطلاعات (Delete)یا... صورت می گیرد .

-5تعیین شی تقاضا : (Command Object) کلاسی است جهت ارسال و دریافت تقاضا از طریق شی اتصال به بانک اطلاعاتی

-6باز کردن اتصال

-7اجرای درخواست

-8دریافت اطلاعات (در صورتی که تقاضا Select باشد (

-9بستن اتصال

» -درج ، حذف و بروز رسانی » برای درج ، حذف و بروز رسانی به صورت متصل و مستقیم از متد

ExecuteNoneQueryمربوط به شی Command استفاده می شود . این شی درخواست مربوط به insert,delete,updateیا هر درخواست دیگری را بدون هیچ درخواستی انجام میدهد و خروجی آن تعداد سطر های تحت تاثیر درخواست می باشد .

```
string connectionString = "Data Source=(local);Initial Catalog=univercity;Integrated Security=true;";
SqlConnection Con = new SqlConnection(connectionString);
string commandText = "insert into student(name,family) values('ali','arefi')";
SqlCommand Cmd = new SqlCommand(CommandText, Con);
Con.Open();
Cmd.ExecuteNonQuery();
Con.Close();
```

معرفی خاصیت ها و دو کد نمونه Placeholderها :متغیرهایی هستند که در یک دستور Sql قرار می گیرندو می توانند در زمان اجرای برنامه جای خود را با عبارتی خاص عوض کنند این متغیرها با علامت @ در یک دستور مشخص می شوند . و هنگامی که از آنها در یک دستور Sql استفاده کنیم قبل از اجرای دستور باید تمامی آنها را با مقادیر مناسب تعویض کنیم . که این کار به صورت اتوماتیک توسط برنامه در زمان اجرای دستور انجام می شود.اما باید پارامترهایی را ایجاد کرده و آن را در لیست parametersدر شئ ایجاد شده از کلاس SqlCommand قرار دهیم تا برنامه بداند هنگام اجرای دستور هر placeholder را باید با مقدار چه متغیری در برنامه عوض کند . نکته – هیچ ضرورتی ندارد که نام یک placeholder همنام فیلدی باشد که قرار است مقدار placeholder در آن قرار بگیرد .خاصیت parametersکلاس : SqlCommandبرای دسترسی به لیست پارامترهایی که در یک شئ از کلاس SqlCommandوجود دارد می توانیم از خاصیت parameters در این کلاس استفاده کنیم . این خاصیت حاوی لیستی از placeholder ها به همراه متغیرهای وابسته به آنها است بنابراین در کد قبل از اجرای دستور ، باید به وسیله ی این لیست مشخص کنیم که هر placeholder با مقدار چه متغیری باید تعویض شود .مثالی از درج رکورد در بانک اطلاعات : در این کد می خواهیم در جدول student از بانک

university، یک رکورد اضافه کنیم این جدول شامل سه فیلد می باشد که اطلاعات آن توسط کاربر در TextBoxها وارد می شود و برنامه با گرفتن این اطلاعات آنها را در جدول بانک ذخیره می کند .

```
Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security = True";
SqlConnection Con = New SqlConnection(Strcon);

SqlCommand Cmd = New SqlCommand();

Cmd.Connection = Con;

Cmd.CommandText= " insert into student id=@id,lname=@lname,fname=@fname";
Cmd.Parameters.AddWithValue("@id",TextBox1.Text);

Cmd.Parameters.AddWithValue("@lname",TextBox2.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox3.Text);

Con.Open();

Cmd.ExecuteNonQuery();

Con.Close();
```

توضیح کد بالا : در خط ۱ یک متغیر از نوع String تعریف کردیم و راطلاعات اتصال به بانک را در آن قرار دادیم . (connectinString)خط ۲ : شیئی از نوع SqlConnection ساختیم و رشته StrConرا به عنوان پارامتر به آن فرستادیم .خط ۳ : شیئی از نوع SqlCommand ساختیم .خط ۴ : خاصیت Connection کلاس SqlCommand را برابر شیئ ساخته شده از کلاس SqlConnection قرار دادیم .خط ۵ : دستور اجرایی Sql را به شیئ SqlCommandنسبت دادیم (با استفاده از خاصیت.(CommandText خط ۶ ، ۷ ، ۸ : با استفاده از خاصیت Parameters کلاس SqlCommand به placeholder ها مقدار دادیم .خط ۹ : اتصال به بانک را برقرار می کنیم .خط ۱۰ : متد اجرایی ExecuetNonQueryرا اجرا می کنیم .خط ۱۱ : اتصال برنامه با بانک را قطع می کنیم .مثالی از اصلاح (update) اطلاعات یک رکورد :حالا می

خواهیم اطلاعات یک رکورد از جدول student را اصلاح کنیم و تغییرات را ثبت نماییم . برای این منظور جدول مورد نظر دارای یک کلید است که می توان با استفاده از آن به تمامی اطلاعات رکورد موردنظر دسترسی پیدا کرد . در این جدول فیلد) id شماره دانشجویی) کلید جدول است .

```
Strcon= "Data Source = (local);Initial Catalog = university;Integrated Security = True";

SqlConnection Con = New SqlConnection(Strcon);
SqlCommand Cmd = New SqlCommand();
Cmd.Connection = Con;

Cmd.CommandText= "update student lname=@lname,fname=@fname where id=@id";

Cmd.Parameters.AddWithValue("@id",TextBox1.Text);

Cmd.Parameters.AddWithValue("@lname",TextBox2.Text);

Cmd.Parameters.AddWithValue("@fname",TextBox3.Text);

Con.Open();

Cmd.ExecuteNonQuery();
Con.Close();
```

در توضیح کد بالا باید این رو بگم که تمام مراحل آن مانند کد insert می باشد به جز دستور Sql که در خط ۵ آمده است و در اینجا update می باشد .

کلاس SqlDataAdapter

کلاس: SqlDataAdapter این کلاس در برنامه های بانک اطلاعاتی ، همانند پلی بین جداول اطلاعاتی و نیز داده های موجود در حافظه که به وسیله ی DataSet نگهداری می شوند ، عمل می کنند . و برای استفاده از آن در برنامه باید یک شیئ از نوع آن ساخته شود .

```
SqlDataAdapter da = New SqlDataAdapter ();
```

این کلاس برای دسترسی به بانک اطلاعاتی از شیئ ایجاد شده از کلاس SqlCommand ای که به آن نسبت داده می شود استفاده می کند .و برای دسترسی به بانک اطلاعات از کلاس SqlCommandو SqlConnectionاستفاده می کند .

```
da . SelectCommand = New SqlCommand();
```

خاصیت : SelectCommand

کلاس SqlDataAdapterدارای خاصیتی این خاصیت است .خاصیت SelectCommand حاوی شیئی از نوع SqlCommand است که از دستور موجود در آن شیئ برای دریافت داده های موردنیاز در برنامه از بانک اطلاعاتی به کار می رود یعنی SqlDataAdapter ، دستوری را که در خاصیت SqlCommand نگهداری می شود را روی بانک اطلاعاتی اجرا کرده و نتایج آن را در کلاس هایی مانند DataSet و یا DataTableقرار می دهد تا در برنامه مورد استفاده قرار گیرند .علاوه بر این ، کلاس SqlDataAdapter دارای خاصیت هایی به نام DeleteCommand ، InsertCommandو UpdatCommand است که به هر یک شیئی از نوع SqlCommand را قبول می کنند و DataAdapter از دستور ذخیره شده در هر یک از آنها به ترتیب بای حذف ، درج و ویرایش داده ها در بانک اطلاعاتی استفاده می کند .

*هنگامی که بخواهید با استفاده از کلاس DataAdapterاطلاعات مورد نیاز خود را از یک بانک اطلاعاتی دریافت کنید ابتدا باید خاصیت SelectCommand را در DataAdapter تنظیم کنید

*خاصیت SelectCommand شیئی از نوع SqlCommandدریافت کرده که این شیئ مشخص می کند داده ها چگونه باید از بانک اطلاعات انتخاب شده و نیز چه داده هایی باید انتخاب شوند .

*اشیاء از نوع SqlCommand نیز دارای خاصیتهایی هستند که قبل از استفاده باید آنها را تنظیم کرد این خاصیت ها عبارتند از :

Connection : -یک شیئ از کلاس SqlConnection در این قسمت قرار گرفته و نحوه ی اتصال به بانک اطلاعاتی را مشخص می کند .

da.SelectCommand.Connection = Con;

CommandText : دستور Sql -و یا پروسیجر ذخیره شده در بانک اطلاعاتی که باید توسط این شیئ اجرا شود ، در این قسمت ذخیره می شود .

da.SelectCommand.CommandText = "select fields from table ";

توضیح کد بالا :در اینجا نوع دستور selectمی باشد و منظور از fields ، نام فیلدهایی است که می خواهیم اطلاعات آنها را استخراج کنیم اگر بخواهیم همه ی آنها را استخراج کنیم از * استفاده می کنیم و منظور از table نیز نام جدولی است که می خواهیم اطلاعات را از آن استخراج کنیم . نمونه کد :برای مثال می خواهیم اطلاعات فیلدهای نام ، نام خانوادگی و شماره دانشجویی از جدول دانشجو را استخراج کنیم و برای کارمورد نظر استفاده کنیم (پس ما در اینجا فقط اطلاعات را استخراج می کنیم .)

SqlConnection Con = New SqlConnection(connectionstring۱ - ;(

SqlDataAdapter da = New SqlDataAdapter۲ - ;()

da.SelectCommand.Connection = Con۳ - ;

da.SelectCommand.CommandText" = select fname,lname,id from student ۴ - ;"

دسترسی اطلاعات و اتصال داده ها

دسترسی به اطلاعات :

در ویژوال C# برای دسترسی به اطلاعات و نمایش آنها سه کامپوننت مهم و اصلی وجود دارند که عبارتند از :

Binding Source، BindingNavigator ، Table Adapter ، DataSet .

* کامپوننتهای BindingNavigator، Binding Source و DataSetرا می توانید در قسمت Data جعبه ابزار ببینید .

* کامپوننت TableAdapter نیز بر اساس مسیری که برای دسترسی به اطلاعات درون بانک اطلاعاتی

و نمایش آنها طی می کنیم به صورت اتوماتیک ایجاد می شود . در ادامه توضیح مختصری در مورد کامپوننتهای مطرح شده با هم مرور می کنیم .

کامپوننت DataSet :

مانند یک موتور اطلاعاتی کوچک عمل می کند با استفاده از DataSet ابتدا به بانک وصل می شویم اطلاعات مورد نیاز را در حافظه DataSet قرار می دهیم سپس ارتباط با بانک را قطع می کنیم از این پس هر تغییراتی که خواستیم می توانیم بر روی اطلاعات درون DataSet اعمال کنیم سپس در آخر تمام تغییرات را بر روی بانک اطلاعاتی اعمال کنیم .

- با استفاده از این کامپوننت اطلاعات درون جداولی نگهداری می شوند و با استفاده از کامپوننت DataView به چندین روش پرس و جوهایی را روی داده ها انجام داد .

کامپوننت DataGridView :

این کنترل برای نمایش داده های موجود در یک بانک اطلاعاتی در فرم برنامه به کار می رود .برای کار با آن کافی است آن را به منبع داده های خود ، مثلا یکی از جدولهای موجود در بانک اطلاعاتی متصل کرده و سپس این کنترل را تنظیم کنیم تا دادهای جدول مورد نظر همانند یک جدول نمایش دهد (ستونهای این جدول نام فیلدها و ردیفهای آن اطلاعات مربوط به فیلدها که هر کدام در یک رکورد نگهداری می شوند) .

- علاوه بر این به وسیله این کنترل می توانید عنوان ستونهای داده ها و یا نوع نمایش آنها را نیز بدلخواه تعیین کنیم

کامپوننت BindingSource :

این کنترل همانند پلی برای ایجاد ارتباط بین داده های موجود در منبع داده ای شما (مانند DataSet) و کنترل هایی که برای نمایش داده ها مورد استفاده قرار می گیرند(مانند TextBox) به کار می رود . بنابراین هنگامی که بخواهید به وسیله ی کنترل هایی و یا به هر دلیل دیگری بخواهید به آنها د منبع اطلاعاتی دسترسی داشته باشید ، این ارتباط باید از طریق این کامپوننت صورت بگیرد .

کامپوننت BindingNavigator :

این کنترل یک رابط گرافیکی استاندارد برای حرکت بین رکوردهای موجود در یک بانک اطلاعاتی ایجاد

می کند .

همچنین مانند کنترل DataGridView می تواند به کنترل BindingSource متصل شده و از طریق آن به داده های موجود در برنامه دسترسی داشته باشد .

کامپوننت TableAdapter :

این کامپوننت در جعبه ابزار وجود ندارد بلکه با توجه به روشی که کامپوننت های داده ای دیگر را در برنامه قرار داده و آنها را تنظیم می کنید و به صورت اتوماتیک ایجاد می شود

- این کامپوننت حاوی پرس و جوهایی برای انتخاب داده های موجود در بانک اطلاعاتی و نیز اطلاعاتی در مورد نحوه اتصال برنامه به بانک است .

- همچنین حاوی متدهایی است که به وسیله آنها می توان داده ها را از جداول بانک اطلاعاتی بدست آورد و در کامپوننت هایی مانند DataSet قرار داد و سپس در برنامه از آن داده ها استفاده کرد .

- این کامپوننت این قابلیت را دارد که بر اساس دستور Select ای که برای انتخاب داده ها از بانک اطلاعاتی برای آن وارد می کنید دستورات Insert ، Update و نیز Delete مناسب برای تغییر داده های انتخاب شده در بانک اطلاعاتی ایجاد کند .

* اتصال داده ها :

اتصال داده یعنی اینکه داده های را که به وسیله ی کامپوننت BindingSource به آنها دسترسی دارید را به یک کنترل خاص نسبت دهید (مثلا به یک DataGridView یا TextBox ، ...) .

به عبارت دیگر یک کنترل را بتوانید به نحوی تنظیم کنید که داده های مورد نیاز خود را به وسیله کامپوننت های دسترسی داده ها در برنامه دریافت کند و سپس آنها را به صورت اتوماتیک به کاربر نمایش دهد .

- در C# بعضی از کنترل ها وجود دارند که مخصوص این کار طراحی شده اند مانند کنترل DataGridView و یا TextBox .