

به نام خدا

آموزش کاربردی
ADO.NET
در
VB.NET و C#

حمیدرضا بیریان
انتشارات پندار پارس

سرشناسه	: بربیان، حمیدرضا، ۱۳۶۲-
عنوان و نام پدیدآور	: آموزش کاربردی ADO.NET در C# و NET.VB/حمیدرضا بربیان.
مشخصات نشر	: تهران: پندار پارس: مائلی، ۱۳۹۰.
مشخصات ظاهری	: ۳۳۶ ص:مصور، جدول.
شابک	: ۱۰۴۰۰۰ ریال: 7-60-2989-964-978
وضعیت فهرست نویسی	: فیپا
موضوع	: اکتیوایکس
موضوع	: پایگاه‌های اطلاعاتی -- طراحی
موضوع	: برنامه‌نویسی شی گرا
موضوع	: ویژوال بیسیک (زبان برنامه‌نویسی کامپیوتر)
موضوع	: سی شارپ (زبان برنامه‌نویسی کامپیوتر)
رده بندی کنگره	: QA/۷۶/۲ب ۱۳۸۹
رده بندی دیویی	: ۰۰۵/۷۵۸۵
شماره کتابشناسی ملی	: ۶۴۲۶۷۲۲

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
 تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۱۲۲۴۵۲۳۴۸
info@pendarepars.com



نام کتاب	: آموزش کاربردی ADO.NET در C# و VB.NET
ناشر	: انتشارات پندار پارس ناشر همکار: مائلی
تالیف	: حمیدرضا بربیان
چاپ اول	: بهار ۹۰
شمارگان	: ۱۰۰۰ نسخه
طرح جلد و صفحه‌آرایی	: لیلی نظری
لیتوگرافی	: ترام سنج
چاپ، صحافی	: صالحان، روشنگر

قیمت : ۱۰۴۰۰۰ تومان به همراه ۲ DVD (ویدئوی آموزشی) شابک : ۷-۶۰-۲۹۸۹-۹۶۴-۹۷۸

*هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

مقدمه

با توجه به چندین سال سابقه تدریس ADO.Net و شناخت نیاز دانشجویان و مهندسين ایرانی، بر آن شدم تا کتاب پیش روی را به رشته تحریر درآوردم تا به کمک سادگی کلام و تصاویر موجود در آن، اصول اولیه کار با بانک‌های اطلاعاتی را آموزش دهم تا به نوعی، نیازهای این طیف از کاربران مرتفع گردد.

شاید بتوان گفت برای نخستین بار در ایران به یاری انتشارات پندار پارس، اقدام به ارائه ویدئوهای آموزشی Net. به همراه کتاب نموده‌ایم. به این معنا که کاربر، افزون بر خواندن کتاب و همراه داشتن آن به عنوان یک مرجع برای مراجعات بعدی، می‌تواند از ویدئوهای آموزشی نیز بهره‌برداری نماید تا اثربخشی یادگیری، دوچندان گردد.

همچنین، به ندرت کتابی تألیفی در زمینه ADO خواهید یافت که هر دو زبان C# و VB.net را پوشش داده باشد. این کار موجب می‌گردد کاربران مبتدی که از صفر شروع به کار با ADO می‌کنند بتوانند یک زبان را به دلخواه برگزیده و یا با هر دو زبان آشنا شوند. کاربرانی که پیش‌تر با یکی از این دو زبان آشنا بوده‌اند نیز قادر به خواندن قسمت موردنظر و شناخت چگونگی کدنویسی در زبان دیگر خواهند بود. در پیوست این کتاب، تمامی Connection String های موجود برای تمامی بانک‌های اطلاعاتی گنجانده شده تا کاربران هر بانک اطلاعاتی بتوانند از آن به عنوان مرجعی در محل کار یا منزل خود استفاده نمایند.

به تجربه ثابت گردیده که مشکل عمده‌ی اغلب کاربران حین تحویل برنامه نهایی به استفاده‌کننده، ارائه نسخه Setup برنامه به همراه دیتابیس آن به شکل یک‌جاست. به همین دلیل، در یک فصل مجزا به نحوه ساخت Setup برای برنامه، به همراه بانک اطلاعاتی جهت ارائه به کاربر نهایی پرداخته‌ایم.

در سایت شخصی بنده (www.Babrian.com)، امکان دست‌یابی به ویدئوهای آموزشی بیشتر و کدهای کتاب مهیاست.

در انتها بیان این جمله کلیشه‌ای را لازم می‌بینم که هیچ اثر نگارشی، حتی پس از چندین بار بازخوانی، نمی‌تواند خالی از اشکال باشد که این امر، دامن‌گیر این کتاب نیز می‌شود. بنابراین، وجود ایرادات احتمالی نگارشی، ویراستاری، و فنی را به حساب قصور یا کم‌کاری بنده نگذارید و خواهش دارم با انتقال این نکات به بنده، مرا در ماندن در این راه پرمخاطره و نیز نگارش هرچه بهتر آثار بعدی یاری فرمایید.

بهار ۹۰

حمیدرضا ببریان

فهرست

بخش اول

فصل ۱

۱	ارتباط با بانک و نمایش اطلاعات در فرم های و ویندوز ۳
۲	پیش گفتار
۲	اجرای ویژوال استدیو
۴	ایجاد بانک
۸	ایجاد جدول
۱۳	ایجاد پروژه جدید
۱۵	آشنایی با اشیاء مورد نیاز
۱۶	آغاز کار با اشیاء
۲۰	ایجاد Connectionstring
۲۱	کار با Dataadapter
۲۲	پر کردن Dataset
۲۷	فارسی سازی ستون ها
۲۸	جستجو در بانک
۳۰	مقابله با خطا
۳۴	عدم وجود رکورد در Dataset
۳۶	نتیجه گیری

فصل ۲

۳۷	افزودن اطلاعات به بانک
۳۸	پیش گفتار
۴۰	آشنایی با Datarow و Sqlcommandbuilder
۴۳	کار با Datarow
۴۴	مقابله با خطا
۴۶	کار با Sqlcommandbuilder
۵۳	نتیجه گیری
۵۴	حذف اطلاعات از بانک
۹۸	نتیجه گیری

فصل ۳

۹۹	دستکاری داده ها
۱۰۰	پیش گفتار.....
۱۰۰	دستکاری داده ها
۱۲۶	ساخت Setup برای برنامه ها

بخش دوم

فصل ۴

۱۵۳	دستورات پیشرفته
۱۵۴	اشیاء کلاس های آنلاین
۱۵۵	کلاس های آفلاین
۱۵۶	Persist security info
۱۵۶	Integrated security
۱۵۷	ارتباط با SQL server
۱۶۱	Native SQL server
۱۶۲	ارتباط با یک named instance of SQL
۱۶۴	ارتباط با SQL Server از طریق ادرس ip
۱۶۶	ارتباط با بانک اطلاعاتی Oracle
۱۶۹	Native Oracle
۱۶۹	فایل TNSNAMES.ORA
۱۷۰	ارتباط با یک بانک اکسس
۱۷۱	ارتباط با یک بانک اکسس شامل کلمه عبور
۱۷۳	ارتباط با یک فایل اکسل
۱۷۶	ارتباط با یک فایل Text file
۱۷۹	اتصال به یک دیتا بیس دیگر با یک کانکشن ((تعویض بانک))
۱۸۰	کارکردن با اشیای آفلاین
۱۸۰	ایجاد و افزودن یک Datatable به یک Data column
۱۸۴	ایجاد یک Datatable و افزودن آن به یک Dataset
۱۸۶	انواع داده ها و نگاشت آن ها به وسیله ADO.NET
۱۹۴	افزودن یک فیلد محاسباتی به Datatable
۱۹۹	ایجاد و تعریف یک یا چند فیلد بعنوان کلید اصلی یا Primary key
۲۰۳	ایجاد یک کلید اصلی با خصوصیت افزایش خودکار
۲۰۵	ساختن یک کلید خارجی یکتا Forjine key
۲۰۸	ایجاد رابطه بین داده ها

فصل ۴

دستورات پیشرفته

۲۱۹	تبدیل و قرار دادن داده‌های درون یک Datatable داخل آرایه ای از Datarow و بالعکس
۲۲۳	ایجاد پرس و جوها.....
۲۲۶	دریافت جریانی از داده‌ها با استفاده از Datareader.....
۲۲۸	دسترسی به مقادیر داده‌های درون Datareader.....
۲۳۳	پر کردن یک Dataset با یک Datatable از روی نتایج یک جستجو.....
۲۳۶	دسترسی به مقادیر داخل یک Datatable یا یک Dataset.....
۲۴۰	بررسی اینکه آیا یک پرس و جو مقداری را بر گردانده است یا خیر.....
۲۴۳	شمردن رکوردهایی که در شرط اول خاص صدق می کند.....
۲۴۴	به دست آوردن تعداد رکورد هایی که در یک Datareader قرار گرفته اند.....
۲۴۸	اجرا و اداره یک پرس و جو که حاصل آن چندین نتیجه «result» می باشد.....
۲۵۱	توضیح در مورد مفهوم Batch command.....
۲۵۱	دسترسی به داده‌های سلسله مراتبی Hierarchcal data.....
۲۵۶	پیمایش بین رکوردها در جدول فرزند و والد درون Dataset.....
۲۵۹	فرستادن یک مقدار null بعنوان پارامتر در یک جستجو.....
۲۶۲	دسترسی به مقادیر بازگشت داده شده توسط رویه‌های ذخیره شده.....
۲۶۵	نحوه مدیریت خطاها در کار با Stored procedures.....
۲۶۹	اجرای توابع valued , scaler در SQL server.....
۲۷۱	ایجاد و اجرای پرس و جوی یک Dataset با استفاده از Linq.....
۲۷۵	بازیابی و دسترسی به داده‌های درون یک فایل متنی.....
۲۷۷	بازیابی و دسترسی به داده‌های درون excel.....

پیوست

۲۸۱ Connection strings
-----	--------------------------

فصل ۱

ارتباط با بانک
و نمایش اطلاعات
در فرم های ویندوز

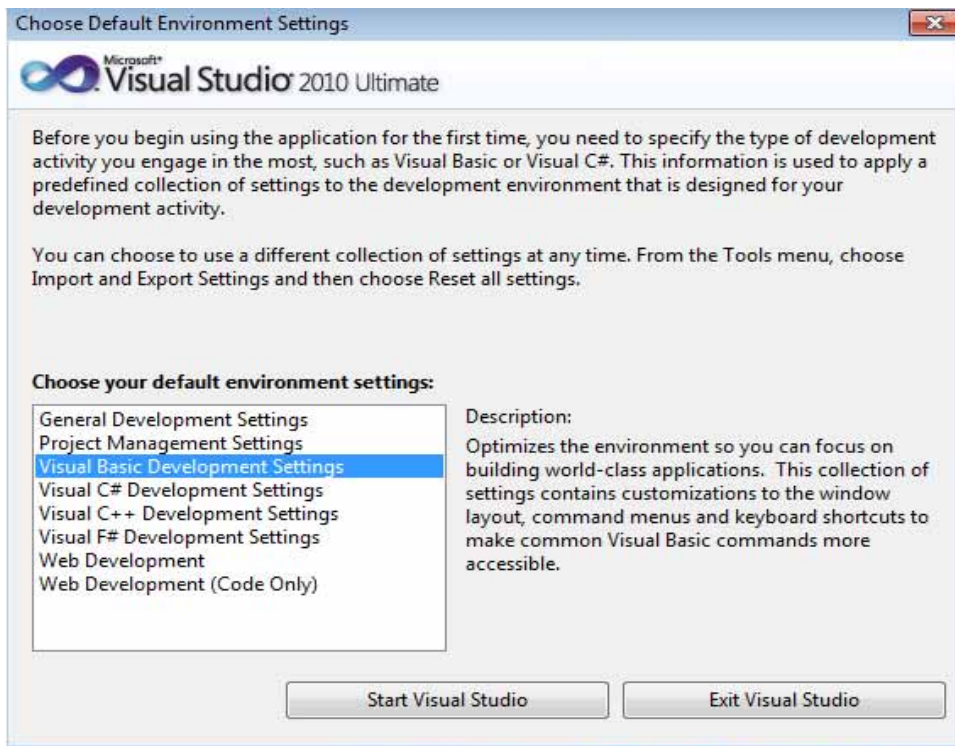
پیش‌گفتار:

گام اول در ایجاد برنامه‌های کاربردی که با بانک‌های اطلاعاتی در ارتباطند، طراحی و ساخت بانک می‌باشد. این قسمت یکی از حساس‌ترین قسمت‌های ایجاد برنامه است، به این دلیل که هرگونه اشتباه در این قسمت منجر به نوشتن کدهای بسیار زیاد، یا شکست پروژه خواهد شد. در صورتی که بعداً نیز بخواهید تغییراتی را در بانک اعمال نموده و اشتباهات قبلی را تصحیح کنید، با این مشکل مواجه خواهید شد که قسمت زیادی که از کدها و یا روال‌های ذخیره شده دیگر کار نمی‌کنند و مجبور به بازنویسی مجدد آن‌ها خواهید شد، در نتیجه اگر می‌خواهید که در این چرخه بی‌پایان گرفتار نشوید، اکیداً توصیه می‌کنم که قبل از طراحی بانک در ابتدا با مفاهیم اولیه طراحی بانک‌های اطلاعاتی، آشنا شوید و به مطالعه کتاب‌های طراحی بانک‌های اطلاعاتی بپردازید. به دلیل این که طراحی بانک‌های اطلاعاتی خارج از بحث این کتاب می‌باشد، ما فقط به تشریح مفاهیم اولیه و نحوه ایجاد بانک، به طور خلاصه می‌پردازیم.

اجرای ویژوال استدیو

قبل از هرکاری می‌بایست ویژوال استدیو ۲۰۱۰ را بر روی سیستم خود نصب کرده باشید. اگر با نحوه نصب این برنامه آشنایی دارید به انتهای کتاب و ضمیمه «نصب ویژوال استدیو» مراجعه نمایید. پس از نصب این نرم‌افزار از منوی Start اقدام به اجرای آن نمایید.

اگر برای اولین بار است که اقدام به اجرای ویژوال استدیو می‌نمایید، پنجره‌ای مطابق تصویر زیر برای شما به نمایش در خواهد آمد که از شما می‌خواهد زبان پیش فرض خود را انتخاب نمائید، با توجه به اینکه برنامه نویسی VB یا C# می‌باشید، یکی از گزینه‌ها را انتخاب نموده و سپس بر روی دکمه Start Visual Studio کلیک نمائید.

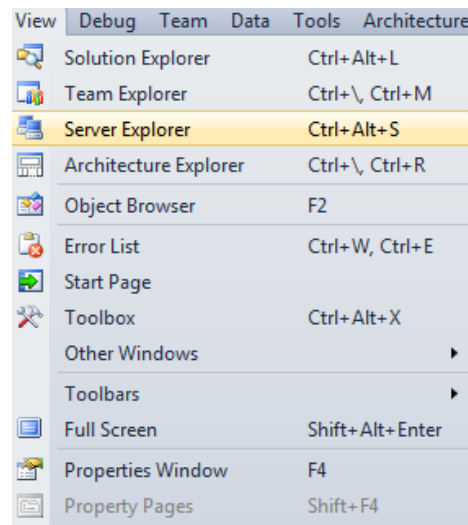


نکته

اگر در دفعات بعد بخواهید مجدداً به این صفحه دسترسی داشته و تنظیمات آن را تغییر دهید، در محیط ویژوال استدیو از منوی Tools گزینه ImpOrt And ExpOrt را انتخاب و سپس گزینه Reset All Setting را انتخاب نمائید.

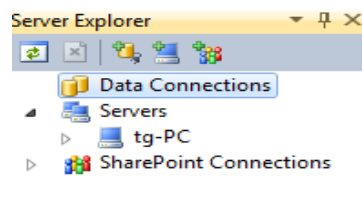
ایجاد بانک

پس از به نمایش در آمدن محیط ویژوال استدیو از منوی View مطابق تصویر زیر گزینه Server Explorer را انتخاب نمائید.

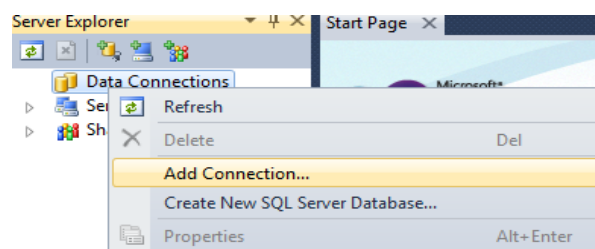


نکته

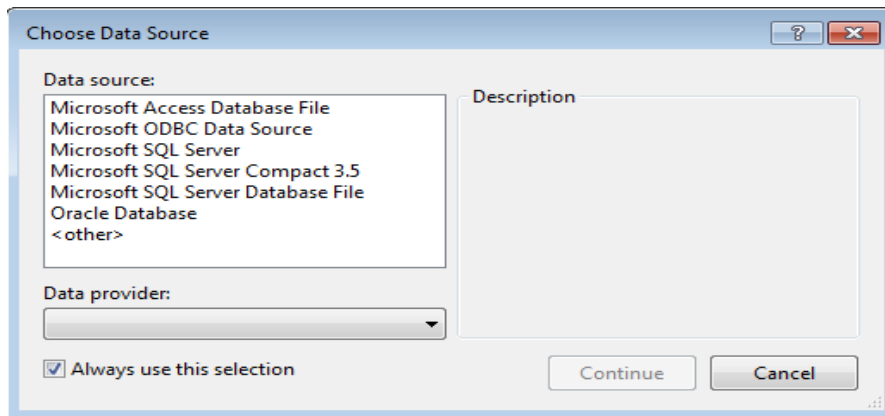
برای به نمایش در آمدن پنجره Server Explorer می توانید از کلیدهای میانبر Ctrl+Alt+S نیز استفاده نمایید. حال با صفحه ای مطابق تصویر زیر روبرو می شویم.



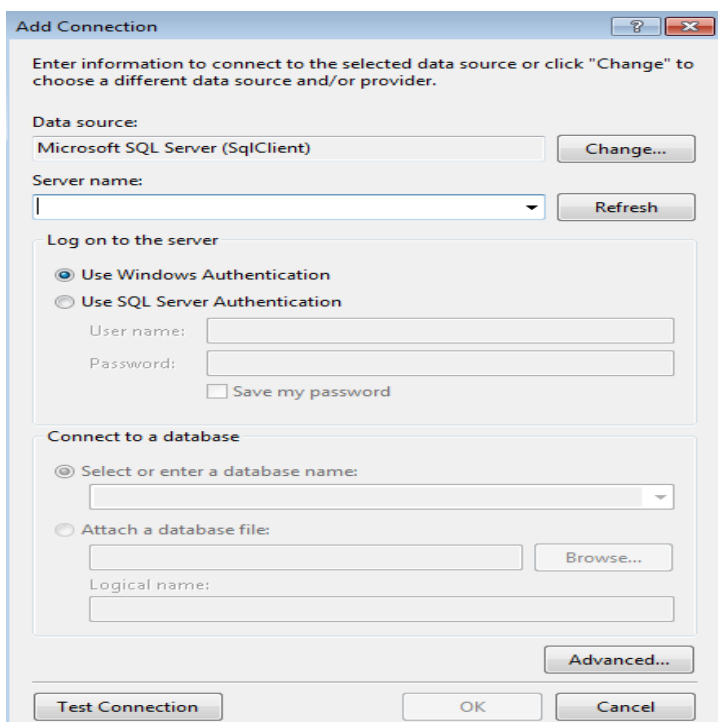
در این قسمت می توانیم یک بانک اطلاعاتی، از نوع SQL را برای خودمان ایجاد نماییم و یا به بانک های موجود که از قبل ایجاد شده اند، دسترسی داشته باشیم. در پنجره Server Explorer بر روی گزینه Data Connections راست کلیک کرده و گزینه Add Connection را انتخاب نمائید:



با این کار پنجره Choose Data Source مطابق تصویر زیر به نمایش در می آید:



در این صفحه می توانید نوع بانکی را که قصد ارتباط با آن را دارید، مشخص نمایید. مثلاً برای ارتباط با بانک های SQL که پیش تر ایجاد کرده اید می توانید گزینه های Microsoft SQL Server یا Microsoft SQL Server Data Base File را انتخاب نمایید و یا اگر قصد ارتباط با بانک Access را دارا می باشید می بایست گزینه Microsoft Access Data Base Field را انتخاب نمایید. برای مثال گزینه Microsoft SQL Server را انتخاب نموده و بر روی گزینه Continue کلیک نمایید تا وارد صفحه ای مطابق تصویر زیر شوید.



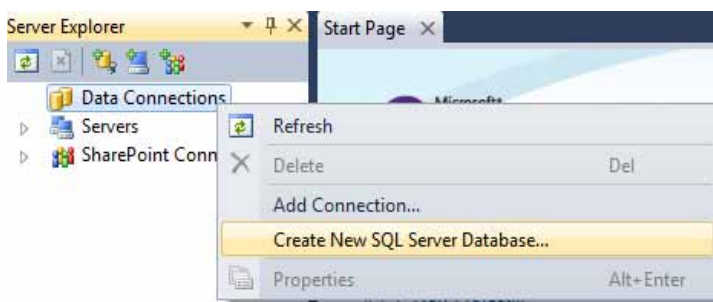
همان طوری که هنگام نصب ویژوال استدیو متوجه شدید، در زمان نصب، یک نسخه از SQL Server مایکروسافت با نام SQL Server Express بر روی سیستم شما نصب می‌گردد. این نسخه از SQL رایگان بوده و برای سازمان‌های کوچک و استفاده‌های شخصی بسیار مفید می‌باشد. در ضمن از لحاظ مصرف منابع سیستمی بسیار سبک بوده و منابع کمتری را مصرف می‌کند. بانکی را که در SQL Express R2 ایجاد می‌کنید، می‌تواند تا ظرفیت ۱۰ گیگابایت رشد داشته باشد. بنابراین توصیه می‌گردد اگر به امکاناتی بیش از این نیاز ندارید، از همین نسخه SQL استفاده نمایید.

در قسمت Server Name مطابق تصویر زیر عبارت \.SQL Express را وارد نمایید.

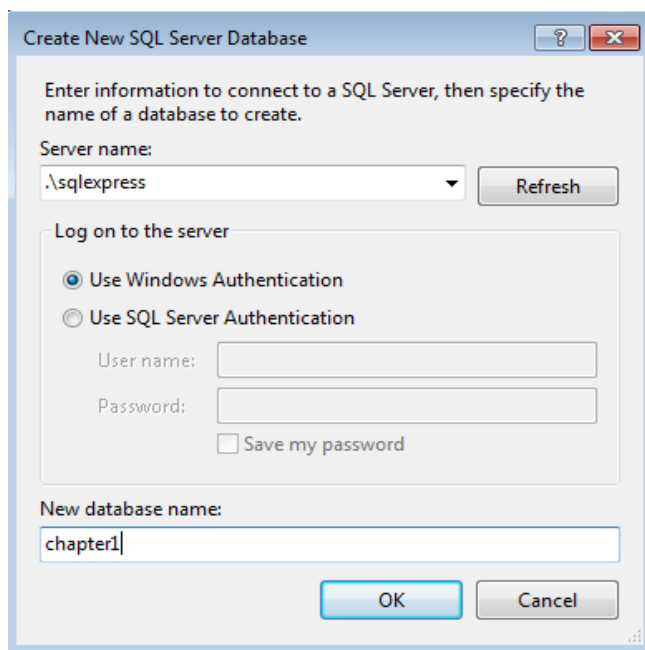
حال در قسمت Select Or Enter a Data Base Name بر روی منوی کشویی کلیک نمایید تا مطابق تصویر زیر بانک‌های موجود در این قسمت را مشاهده نمایید.

همان طوری که مشاهده می‌کنید چهار بانک Tempdb, msdb, Model, Master از قبل، در این قسمت موجود می‌باشند، این بانک‌ها به صورت پیش فرض و با نصب SQL بر روی سیستم شما قرار می‌گیرند. این بانک‌ها، از بانک‌های نوع سیستمی بوده و برای ایجاد بانک‌های جدید توسط SQL مورد استفاده قرار می‌گیرند، بنابراین اکیداً توصیه می‌گردد که تا زمانی که با نحوه کاربرد آن‌ها آشنا نشده‌اید، آن‌ها را تغییر ندهید و یا حذف ننمایید. پس از آشنایی مقدماتی با این صفحه بر روی دکمه Cancel کلیک کنید تا مجدداً به صفحه اصلی برگردید.

فرض کنید می‌خواهیم برنامه‌ای را ایجاد نماییم که نام، نام فامیل، شماره شناسنامه و آدرس کارمندان را در یک بانک ذخیره نماید. در ابتدای این کار، به ایجاد بانک می‌پردازیم. برای این کار در پنجره Server Explorer بر روی Data Connections راست کلیک کرده و گزینه Create New SQL Server Data Base را انتخاب نمایید



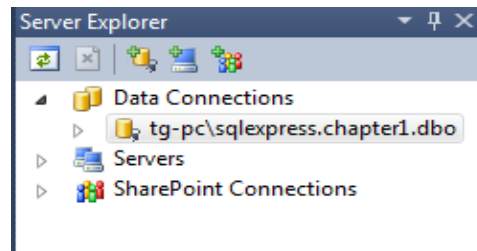
در قسمت server name عبارت \.sql express را وارد کنید. در قسمت Log on to server گزینه پیش فرض یعنی Authenticatoin Use windows را انتخاب کنید، این بدان معناست که از اعتبارسنجی خود ویندوز استفاده می کنید. در پایان نیز در قسمت New data base نام بانکی را که قصد ایجاد آن را داریم، Chapter1 وارد نمایید. هم اکنون پنجره شما می بایست مطابق تصویر زیر گردیده باشد.



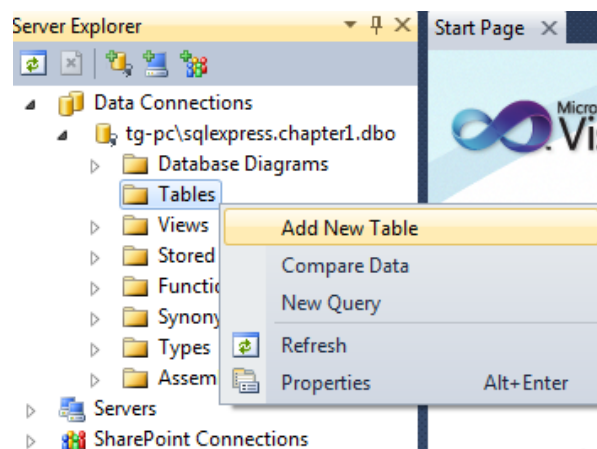
با کلیک بر روی دکمه OK یک بانک با نام Chapter1 درون SQL express ایجاد می نمایید.

ایجاد جدول

حال در پنجره Server explorer و در زیر قسمت Data connectoins بانک جدیدی را که ایجاد نموده‌اید، می‌توانید مشاهده نمایید:

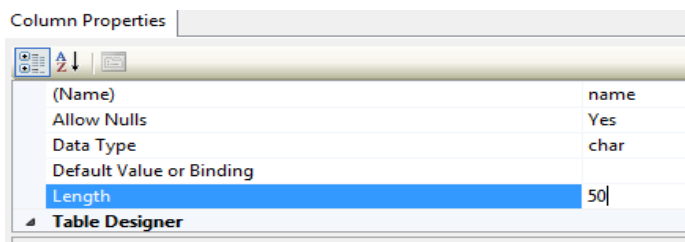


بر روی علامت مثلث کنار نام بانک، کلیک نمایید تا به سایر گزینه‌های بانک دسترسی داشته باشید. بر روی Tables راست کلیک کرده و گزینه Add New table را انتخاب نمایید:



با این کار وارد محیط ایجاد جدول خواهید شد. همان طوری که قبلاً گفتیم قصد داریم مقادیر نام و نام فامیل و شماره شناسنامه و آدرس را در این جدول ذخیره نماییم. بنابراین به چهار فیلد نیاز داریم. در پنجره‌ای که در مقابل دارید، برای ایجاد اولین فیلد در قسمت column name نام فیلد‌های مورد نظر و در قسمت Data Type نیز نوع داده‌های مورد نظر، را وارد می‌نماییم. همچنین اگر بخواهیم که کاربر بتواند درون فیلد خاصی مقداری را وارد ننموده و آن را خالی بگذارد، گزینه Allow Nulls را تیک می‌زنیم.

برای ایجاد اولین فیلد، زیر قسمت column name مقدار name را وارد کنید و Data Type آن را برابر با Char قرار دهید. حال در پایین صفحه در قسمت Column properties خاصیت Length آن را مطابق با تصویر برابر با ۵۰ قرار دهید.



برای ساخت فیلد بعدی نام فیلد را برابر با family و طول آن را نیز برابر با ۵۰ کاراکتر از نوع Char همانند فیلد قبل وارد نمایید. برای ساخت فیلد شماره شناسنامه، نام آن را برابر با sh و مقدار آن را از نوع numeric قرار دهید.

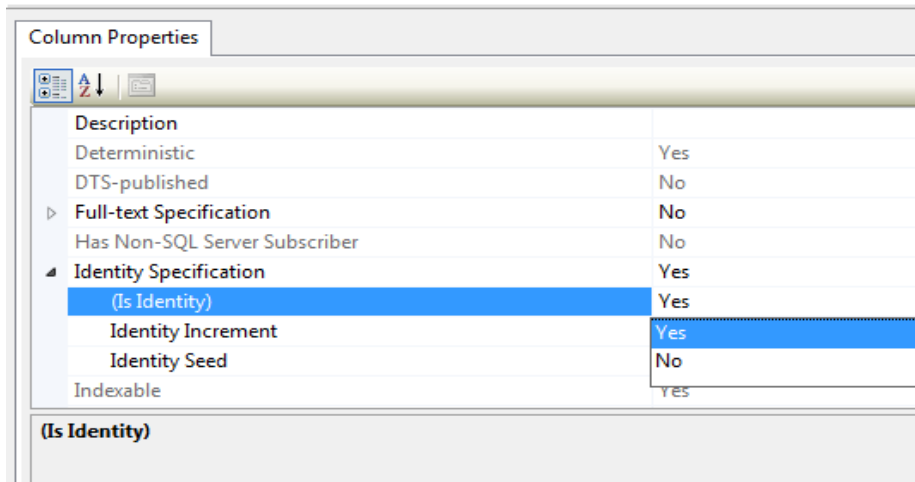
همان طوری که مشاهده می‌کنید پس از ساختن هر فیلد مقدار فیلد Allow null به طور اتوماتیک برابر با True شده و تیک دار می‌شود. از آن جایی که ممکن است طول کاراکترهای آدرس بسیار زیادتر باشد بعد از این که نام فیلد را برابر با Address قرار دادید، نوع آن را از نوع Text قرار دهید. در ظاهر طراحی بانک برای هدف ما انجام گردیده است. اما در تمامی جدول‌های ایجاد شده توسط ما می‌بایست فیلدی را به عنوان کلید اصلی یا Primary key دارا باشند. تمامی کدهایی که برای دستکاری داده‌ها می‌نویسیم، اعم از افزودن اطلاعات، حذف آن‌ها و یا به روز رسانی، نیاز به این فیلد دارند.

کاربرد اصلی کلید اصلی در بانک، ایجاد رابطه بین جداول می‌باشد. فیلدی را می‌توان به عنوان کلید اصلی برگزید که اولاً مقادیر آن در تمام جدول یکتا بوده و تکراری نباشد. مثلاً مقدار نام یا فامیل نمی‌تواند کلید اصلی باشد زیرا این امکان هست که نام یا فامیل دو نفر متشابه باشد. سایر فیلدها نیز همین شکل را دارند. دومین شرط نیز برای کلید این می‌باشد که فیلد مورد نظر هیچ گاه خالی نباشد.

برای مثال در جدولی، اگر فیلدی با نام شماره پلاک ماشین وجود داشته باشد، این فیلد نمی‌تواند کلید اصلی یا primary key گردد. زیرا ممکن است شخصی اصلاً ماشین نداشته باشد و در نتیجه ممکن است در این فیلد مقداری قرار نگیرد. حال به جدول خودمان برمی‌گردیم. همان طوری که متوجه شده‌اید هیچکدام از فیلدها دارای شرط یک نمی‌باشند.

مثلاً اگر در سازمان شما دو برادر از یک آدرس مشغول به کار باشند، این فیلد نیز خاصیت کلید بودن را از دست می‌دهد. در چنین مواردی خودمان یک فیلد جدید به جداولمان اضافه کرده و آن را کلید اصلی می‌نماییم. برای این کار، زیر فیلدهایی که تاکنون ایجاد نموده‌اید، در قسمت column name مقدار ID را وارد نموده و نوع آن را برابر با numeric قرار دهید.

حال در پایین صفحه و در قسمت column properties خصوصیت Identity specification را پیدا کنید و بر روی مثلث کنار آن کلیک نمایید و گزینه Is Identity آن را برابر با «yes» قرار دهید.



با این کار، این فیلد به طور خودکار مقدار می‌گیرد. یعنی با افزودن یک رکورد به جدول عدد ۱ درون این فیلد قرار می‌گیرد و با افزودن رکوردهای دیگر به ترتیب اعداد ۲ و ۳ و... درون این فیلد قرار داده می‌شود. بدیهی است که در این صورت دیگر هیچ وقت مقدار تکراری درون فیلد قرار نخواهد گرفت.

نکته

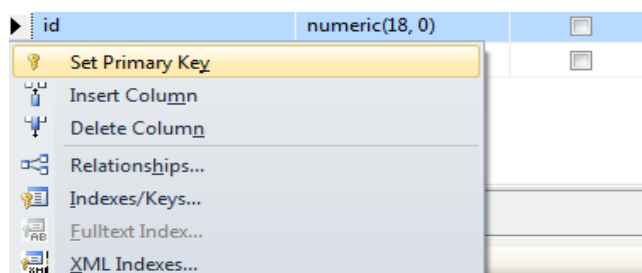
اگر بخواهیم مقدار این فیلد به جای یک واحد، یک واحد، به مقدار دیگری افزایش پیدا کند، در قسمت Identity Increment عدد مورد نظر را وارد نمایید. مثلاً اگر در این قسمت عدد ۳ را وارد نماییم مقدار فیلد ۳ واحد، ۳ واحد افزایش می‌یابد.

نکته

مقدار اولیه فیلد را می‌توان در قسمت Identity seed مشخص نمود. مثلاً اگر این مقدار را برابر با ۷ قرار دهیم. مقداردهی آن از ۷ شروع شده و به آن یک واحد، یک واحد افزوده می‌گردد.

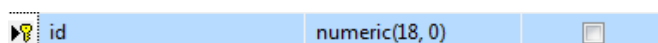
حال مجدداً بر روی فیلد ID که قبلاً آن را ایجاد نموده بودیم برگشته و در انتها الیه سمت چپ آن، بر روی آن راست کلیک نموده و گزینه Set primary key را انتخاب نمایید.

با این کار، این فیلد با عنوان کلید اصلی یا primary key در جدول انتخاب می‌گردد.



نکته

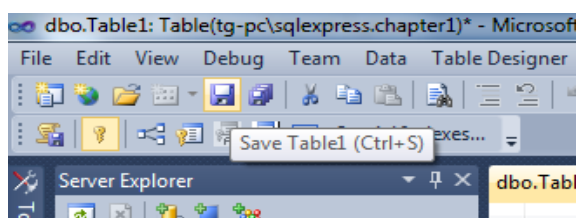
همان طوری که مشاهده می‌نمایید، پس از انتخاب فیلد ID به عنوان کلید ستون Allow nulls خودبه خود بدون تیک می‌شود و در کنار فیلد ID تصویر یک کلید قرار می‌گیرد.



در انتها می‌بایست محیط طراحی جدول شما به صورت زیر درآمده باشد:

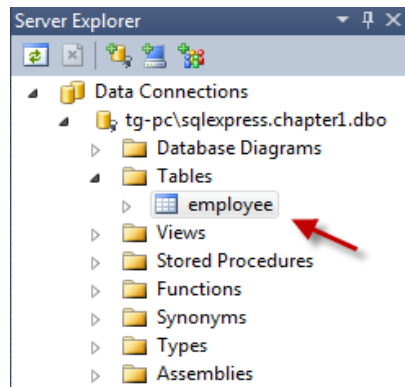
Column Name	Data Type	Allow Nulls
name	char(50)	<input checked="" type="checkbox"/>
family	char(50)	<input checked="" type="checkbox"/>
sh	numeric(18, 0)	<input checked="" type="checkbox"/>
address	text	<input checked="" type="checkbox"/>
id	numeric(18, 0)	<input type="checkbox"/>

از Tool bar بالای صفحه بر روی گزینه Save table کلیک نمایید

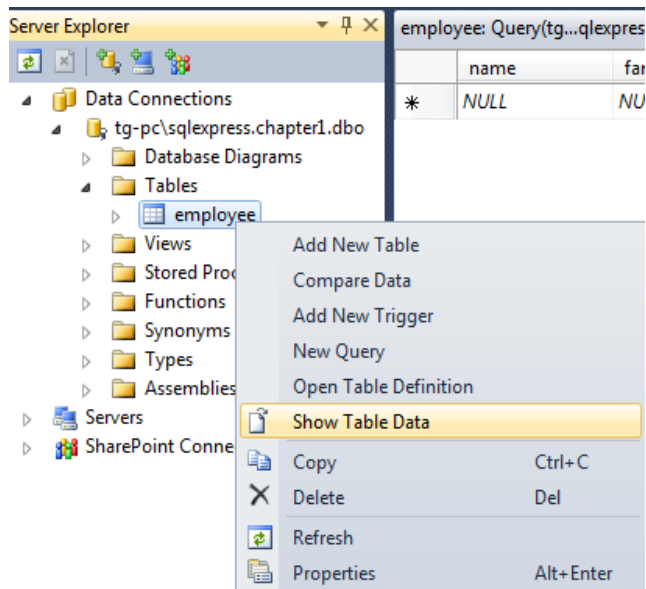


در صفحه ای که به نمایش در می‌آید نام جدول را برابر با employee قرار دهید و بر روی دکمه ok کلیک نمایید.

با این کار جدول employee می بایست مطابق تصویر زیر در Server explorer به نمایش در آید.



حال می خواهیم برنامه ای بنویسیم که داده های درون این جدول را برای ما به نمایش در آورد بنابراین قبل از هرکاری، چند رکورد به صورت فرضی وارد بانک مان می نماییم. برای این کار بر روی جدول employee راست کلیک کرده و گزینه show table data را مطابق تصویر زیر انتخاب نمایید.



در صفحه ای که به نمایش درمی آید می توانید اطلاعاتی را به صورت فرضی وارد نمایید. شما نیز می توانید مطابق تصویر اطلاعات را وارد جدول employee نمایید.

employee: Query(tg...qlxpress.chapter1) X Start Page					
	name	family	sh	address	id
	hamid reza	babrian	5	tehran	1
	farhad	rezaee	12365	shiraz	2
	leila	nazari	87456	tehran	3
	soheila	karimi	123	tabriz	4
	ali	majd	789	hamedan	5
**	NULL	NULL	NULL	NULL	NULL

نکته

همان طوری که در هنگام ورود اطلاعات به جدول موردنظر، متوجه شدید، پس از این که اطلاعات را درون فیلدی وارد نمودیم و قصد رفتن به فیلد بعدی را داریم درون فیلد قبلی علامت تعجب قرمز رنگی مطابق تصویر زیر به نمایش درمی آید، که بدان معناست که هنوز اطلاعات فیلد قبلی درون بانک ذخیره نگردیده است.

employee: Query(tg...qlxpress.chapter1) X Start Page					
	name	family	sh	address	id
	hamid reza	babrian	5	tehran	1
	farhad	rezaee	12365	shiraz	2
	leila	nazari	87456	tehran	3
	soheila	karimi	87456	tabriz	4
	ali	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL

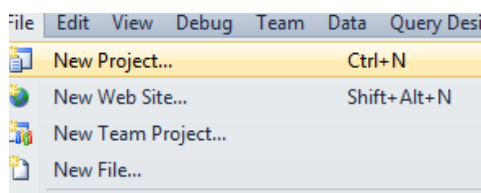
This Cell has changed.
The change has not been committed to the database.
The original Data is NULL.

نکته

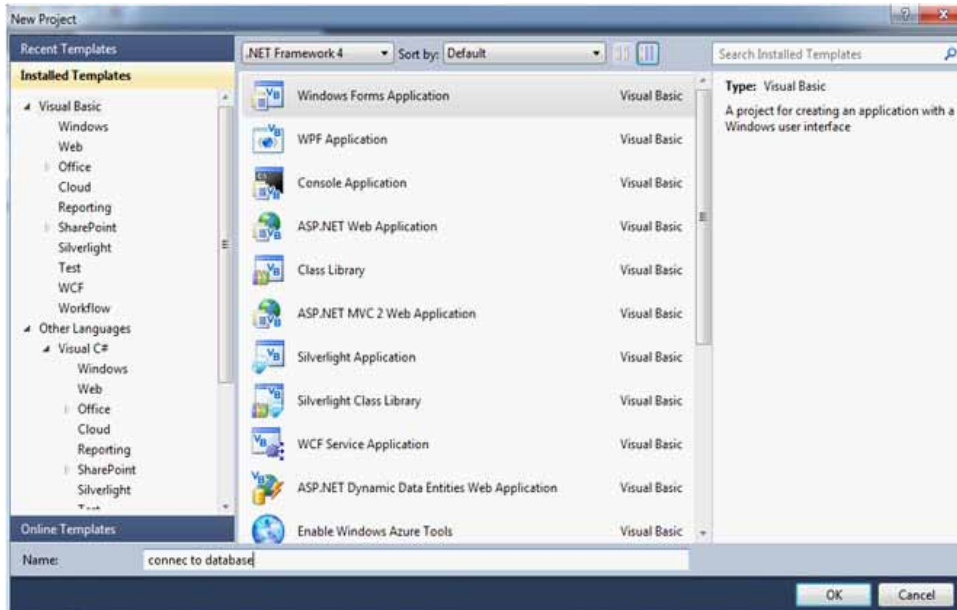
همان طوری که متوجه شده اید مقدار فیلد ID به طور اتوماتیک درون آن قرار گرفته . کافی است اطلاعات درون یک رکورد را وارد نموده و سراغ رکورد بعدی بروید. با این کار مقدار فیلد ID خودبه خود درون آن قرار می گیرد.

ایجاد پروژه جدید

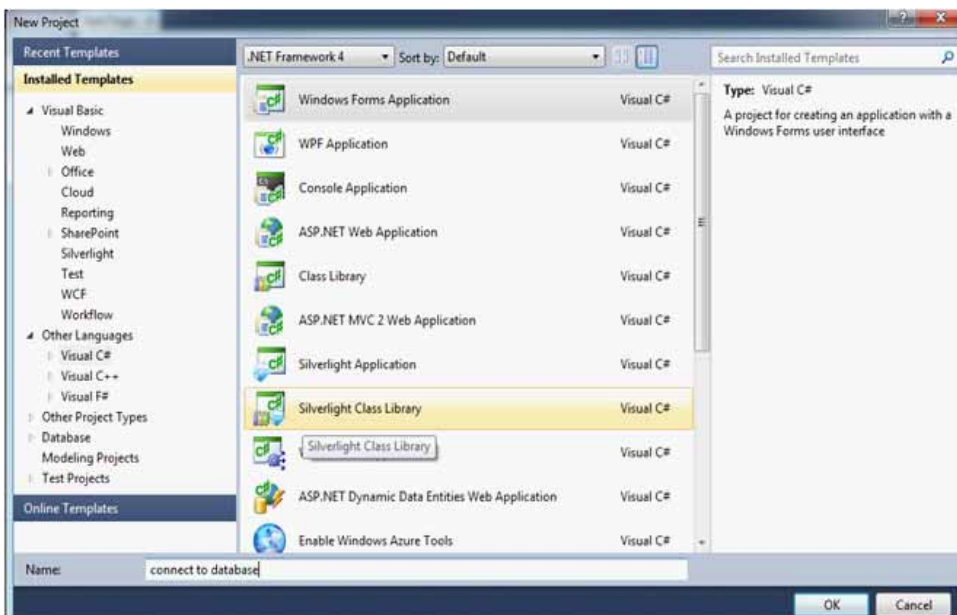
هم اکنون زمان آن فرارسیده تا در عمل از داده های درون بانک مان استفاده نماییم. از منوی فایل گزینه New project را انتخاب نمایید.



در پنجره New project اگر می‌خواهید از زبان Visual Basic استفاده نمایید، بر روی Visual Basic کلیک کرده و در پنل وسطی گزینه windows Form application را انتخاب نمایید. در قسمت name نیز عبارت connect to data base را وارد نمایید.



اگر قصد استفاده از زبان C# را به عنوان زبان برنامه‌نویسی خود دارید بر روی Visual C# کلیک نموده و سپس در پانل وسطی گزینه windows forms application را انتخاب کنید در قسمت Name نیز عبارت connect to data base را وارد کرده و سپس بر روی دکمه ok کلیک نمایید.



آشنایی با اشیاء مورد نیاز

قبل از این که به ادامه کار بپردازیم لازم است تا کمی درباره سه شیء مهم و پرکاربرد در ADO.NET توضیحاتی را ارائه نماییم. قبل از هر چیزی برنامه ما می بایست بانک اطلاعاتی موردنظر را پیدا کرده و به آن وصل شود. این وظیفه در ADO.NET بر عهده Connection می باشد. حال اگر از بانک SQL استفاده می کنیم این وظیفه به عهده SQL Connection بوده و اگر از بانک Access استفاده می نماییم این وظیفه به عهده OLEDB connection می باشد. پس وظیفه Connection SQL و یا OLEDB connection برقراری ارتباط بین برنامه کاربردی ما و بانک اطلاعاتی موجود می باشد. هرکدام از دو شیء بالا به وسیله connection string، بانک اطلاعاتی را پیدا کرده و با آن ارتباط برقرار می نمایند.

فرض کنید می خواهید به یک فروشگاه بروید و از آن جا چیزی را برای خود بخرید. و یا به درب خانه دوستی رفته و از او کتابی بگیرید. در هر دو مثال اولین قدم این است که آدرس فروشگاه و یا منزل دوست خود را بدانید. Connection string را می توانید به نوعی آدرس بانک، همراه با اطلاعات اضافی دیگری فرض نمایید.

شیء بعدی Data adapter می باشد که اگر با بانک SQL در ارتباط باشیم، SQL data adapter نامیده شده و در سایر موارد OLEDB data adapter نامیده می شود. این شیء، پل ارتباطی برنامه ما و جداول بانک می باشد. فرض کنید از طریق connection به یک برنامه وصل شده ایم که دارای چندین جدول می باشد.

به کدام یک از جدول ها نیاز داریم؟ از هر جدول به چه فیلدهایی نیاز داریم؟ اینجاست که Data adapter به کمک ما می آید و درخواست ما را اجرا می نماید. پس در Data adapter این که کدام جدول و کدام فیلد را، با چه شروطی لازم داریم، تعیین می نماییم. سومین شیء Dataset می باشد. Dataset را فضایی از RAM کامپیوتر خودتان در نظر بگیرید که درخواست ها در آن قرار می گیرد.

فرض کنید می خواهید کلیه نام های درون جدول employee را به نمایش درآورید. ابتدا SQL connection ارتباط فرم مان را با بانک Chapter1 برقرار می نماید، سپس SQL data adapter به جدول employee وصل شده و کلیه نام ها را برای ما برداشته و به درون Dataset می ریزد. حال دیگر تمامی اعمال ما بر روی Dataset انجام شده و همچنین بر روی فرم های برنامه هایمان نیز داده های درون Dataset را مشاهده می کنیم.

پس از توضیحات فوق به برنامه ایجاد شده توسط خودمان برمی گردیم. در محیط ویژال استدیو بر روی فرم ظاهر شده دو بار کلیک کنید تا وارد فرم کد برنامه شوید.

برای این که بتوانیم از سه شیء که قبل تر معرفی کردیم استفاده نماییم می بایست فایل های اسمبلی system. Data. SQL client و system. Data. SQL connection را به عنوان Refrence به برنامه اضافه می نماییم.

برای آشنایی بیشتر با فایل های اسمبلی و چگونگی کارکرد و همچنین ایجاد و استفاده از آن ها می توانید به کتاب برنامه نویسی شیء گرا در ویژوال استدیو تألیف اینجانب، انتشارات پندار پارس مراجعه نمایید. به طور خلاصه می توان اینطور بیان کرد که برای استفاده از سه شیء اصلی SQL connection و SQL data adapter و Dataset و می بایست از دو فایل اسمبلی dll system data و system. Data. SQL connection استفاده نمایید.

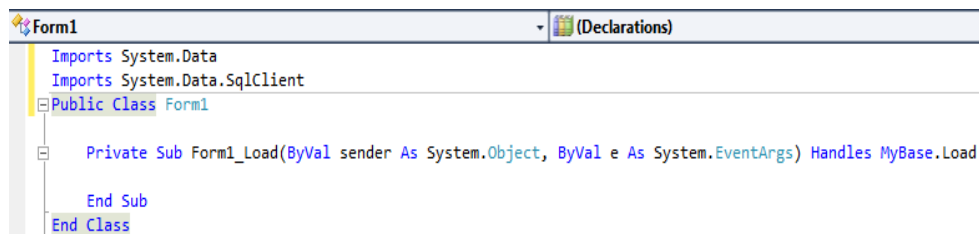
آغاز کار با اشیاء

بعد از این که بر روی فرم ایجاد شده دو بار کلیک کردید تا وارد فرم کد شوید، در ویژوال بیسیک بالای عبارت Public class Form1 یعنی در بالاترین نقطه صفحه، عبارت زیر را تایپ کنید:

```
Imports system . data
```

```
Imports system . data . SQL client
```

هم اکنون صفحه کد شما می بایست شبیه تصویر زیر گردیده باشد.



```
Form1 (Declarations)
Imports System.Data
Imports System.Data.SqlClient
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub
End Class
```

در C# به جای عبارت Imports از using استفاده می نماییم. بعد از وارد شدن به صفحه کد، متوجه می شوید که از قبل، عبارت using system. Data در فرم قرار دارد، بنابراین ما فقط عبارت

```
Using system. Data. SQLclient;
```

را زیر سایر ارجاع ها تایپ می کنیم.

حاصل کار می بایست شبیه تصویر زیر گردیده باشد.

```
WindowsFormsApplication1.Form1
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

هم اکنون زمان آن رسیده است که سه متغیری که قبلاً معرفی نمودیم را در برنامه خودمان تعریف نماییم. از آن جایی که می خواهیم این متغیرها در تمامی روال های فرم، قابل استفاده باشند، آن ها را دقیقاً زیر عبارت `Public class Form1` قرار می دهیم. علاوه بر سه متغیر فوق دو رشته نیز برای قرار دادن `Connection string` و دستور `SQL` در آن ها تعریف می نماییم.
در ویژوال بیسیک قرار می دهیم:

```
Form1 (Declarations)
Imports System.Data
Imports System.Data.SqlClient

Public Class Form1
    Dim con As SqlConnection
    Dim da As SqlDataAdapter
    Dim ds As New DataSet
    Dim s1, s2 As String
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    End Sub
End Class
```