



دانشگاه علمی کاربردی

واحد آموزش و پژوهش صنایع ایران

عنوان مستند :

# آموزش Microsoft Visual Studio

## VC#

استاد :

مهندس امیر عباس کرمی

توسط دانشجو::

فخرالسادات میرزمانی

رشته : مهندسی فناوری اطلاعات (IT)

سال تحصیلی : ۹۰



به نام دوست

که هر آنچه داریم از اوست

---

تقدیم به

دوستان همیشه دوست

---



---

از زحمات و راهنمایی های صمیمانه استاد عزیزم  
مهندس امیر عباس کرمی که همواره بی دریغ و  
دلسوزانه فقط به ما آموخت بسیار سپاسگزاریم و  
بی نهایت تشکر و قدردانی می کنم .

---



صفحه	عنوان
۱۳	چکیده
<b>فصل اول : مقدمه</b>	
۱۶	۱-۱- مقدمه
۱۷	۱-۲- سیستم های مورد نیاز جهت اجرای نرم افزار
<b>فصل دوم : مروری بر نرم افزار VC#</b>	
۲۰	۲-۱- معرفی نرم افزار VC#
۲۰	۲-۲- نحوه اجرای برنامه
۲۱	۲-۳- قسمت ها، اجزاء و زبان ها
۲۴	۲-۴- معرفی محیط
۲۴	۲-۴-۱- Server Explorer
۲۵	۲-۴-۲- Toolbars
۲۸	۲-۴-۳- Solution Explorer
۳۳	۲-۴-۴- Properties
۳۴	۲-۵- یک تمرین ساده
<b>فصل سوم : اجرای پروژه</b>	
۳۸	۳-۱- طریقه اجرای پروژه
۳۹	۳-۲- معرفی عناصر
۳۹	۳-۲-۱- Form
۳۹	۳-۲-۲- Name
۴۴	۳-۲-۳- BackColor
۴۴	۳-۲-۴- BackgroundImage
۴۶	۳-۲-۵- BackgroundImageLayout
۴۷	۳-۲-۶- Cursor



۴۸ ..... Enable -۳-۲-۷

### فصل چهارم: بررسی Menu

۵۴ ..... Menu ویژگیهای -۴-۱

۵۴ ..... Font -۴-۱-۱

۵۵ ..... ForeColor -۴-۱-۲

۵۵ ..... Image -۴-۱-۳

۵۶ ..... FormBorderStyle -۴-۱-۴

۵۷ ..... Anchor -۴-۱-۵

۵۸ ..... Icon -۴-۱-۶

۵۸ ..... Languge -۴-۱-۷

۵۸ ..... Maximize Box -۴-۱-۸

۵۸ ..... Minimize Box -۴-۱-۹

۵۸ ..... RightToLeft -۴-۱-۱۰

۵۹ ..... RightToLeft -۴-۱-۱۰-۱

۵۹ ..... RightToLeftLayout -۴-۱-۱۱

۵۹ ..... Text -۴-۱-۱۲

۵۹ ..... WindowsState -۴-۱-۱۳

۶۰ ..... TextBox -۴-۱-۱۴

۶۰ ..... Name -۴-۱-۱۵

۶۰ ..... DataBindings -۴-۱-۱۶

۶۴ ..... Windows تمرین نوشتن برنامه ماشین حساب

### فصل پنجم: استفاده از عکس

۷۶ ..... Load عکس -۵-۱

۷۶ ..... PictureBox -۵-۱-۱

۷۸ ..... Name -۵-۱-۲

۷۸ ..... Image -۵-۱-۳

۷۹	.....	SizeMode -۵-۱-۴
۸۰	.....	FileName -۵-۱-۵
۸۱	.....	Filter -۵-۱-۶
۸۱	.....	Title -۵-۱-۷
۸۲	.....	۵-۲- یک تمرین ساده

### فصل ششم : بانک اطلاعاتی

۸۶	.....	۶-۱- آشنایی با بانک اطلاعاتی
۹۱	.....	۶-۲- اتصال به بانک اطلاعاتی
۹۱	.....	DataGridview -۶-۲-۱
۹۳	.....	Web Service -۶-۲-۲
۹۳	.....	Object -۶-۲-۳
۹۴	.....	۶-۳- Wizard اتصال به بانک اطلاعاتی
۹۶	.....	۶-۴- معرفی امکانات
۹۶	.....	Name -۶-۴-۱
۹۶	.....	Header Text -۶-۴-۲
۹۷	.....	MaxInputLength -۶-۴-۳
۹۷	.....	MinimumWith -۶-۴-۴
۹۸	.....	ReadOnly -۶-۴-۵
۹۸	.....	SortMode -۶-۴-۶
۹۹	.....	ToolTipText -۶-۴-۷
۹۹	.....	ColumnType -۶-۴-۸
۱۰۱	.....	۶-۵- جستجو

### فصل هفتم : دسترسی به اطلاعات

۱۰۶	.....	۷-۱- Component های دسترسی به اطلاعات
۱۰۶	.....	Dataset -۷-۱-۱
۱۰۸	.....	BindingNavigator -۷-۱-۲

۱۰۹	BindingSource -۷-۱-۳
۱۰۹	TableAdapter -۷-۱-۴

### فصل هشتم : کلاس نویسی

۱۱۴	ADO.NET -۸-۱
۱۱۴	کلاس نویسی -۸-۲
۱۱۵	کلاس های موجود در SQL Client -۸-۳
۱۱۶	متصل شدن و قطع کردن اتصال به یک بانک اطلاعاتی -۸-۴
۱۱۶	خاصیت Parameter -۸-۵
۱۱۶	خاصیت CommandText -۸-۶
۱۱۷	کد مربوط به دستور Delete و Insert -۸-۷

### فصل نهم : مزایا و معایب محیط VC#

۱۲۲	مزایا -۹-۱
۱۲۲	معایب -۹-۲

### فصل دهم : خلاصه ، نتیجه گیری ، پیشنهادات آتی

۱۲۴	خلاصه -۱۰-۱
۱۲۵	نتیجه گیری -۱۰-۲
۱۲۵	پیشنهادات آتی -۱۰-۳
۱۲۷	پیوست ۱- فصل دوم
۱۲۷	پیوست ۲- فصل دوم
۲۰۳	پیوست ۳- فصل دوم
۲۲۱	پیوست ۴- فصل دهم
۲۴۵	پیوست ۵- پیوست ۴
۲۵۱	<b>ABSTRACT</b>

## چکیده:

در شرایط دنیای امروز، انجام امور بر پایه های علمی و اصولی و با استفاده از دانش پیشرفته روز، یک الزام است. نرم افزار Visual Studio، یک مجموعه از برنامه‌هایی است که ارتباط بسیار نزدیک با هم دارند و Microsoft آن را به توسعه دهندگان و برنامه نویسان برنامه‌های کاربردی اهدا نمود تا آنها را وادار نماید در محیطی توسعه یافته بر روی پلت فرم‌های ویندوز و دات نت به ساخت برنامه‌های خود بپردازند.

Visual Studio می‌تواند برای نوشتن برنامه‌های کنسولی، ویندوزی، سرویس‌های ویندوز، برنامه‌های کاربردی موبایل، برنامه‌های کاربردی ASP.NET و سرویس‌های وب ASP.NET بنا به انتخاب شما همراه با زبان‌هایی مانند C++, C#, VB.NET, J# استفاده شود. اینکه با Visual Studio واقعا "چه کارهایی می‌توان انجام داد؟ و چه استفاده‌هایی می‌توان از آن کرد و چگونگی انجام آنها مواردی است که مختصرا" به آن خواهیم پرداخت.

لازم به ذکر است که قابلیت‌های منحصر به فرد و خاص این نرم افزار جذابیت‌های آن را نسبت به سایر نرم افزارهای مشابه بیشتر و قابل توجه تر نموده است.

اهمیت آموزش این نرم افزار از آن جهت است که به راحتی می‌توان از آن برای برنامه نویسی‌های خاص و سفارشی با در نظر گرفتن خواسته‌های مشتری استفاده نمود.

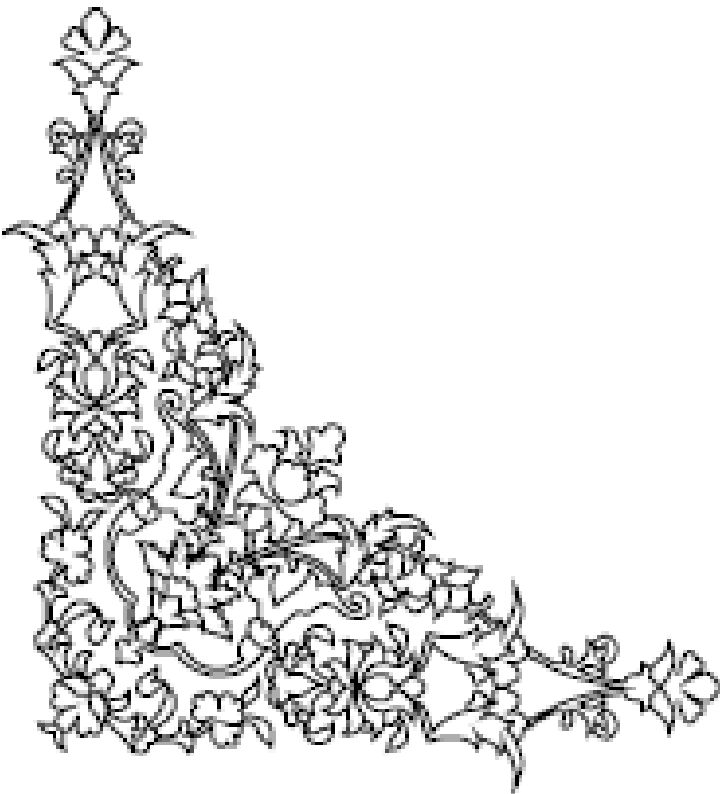
آموزش گام به گام ارائه شده می‌تواند کار در محیط Visual Studio را بصورت کارگاهی و آموزشی برای کسانی که اشتیاق برای فراگیری و هم چنین برنامه نویسی با آن را دارند فراهم سازد.





# فصل اول

## مقدمه





ویژوال استودیو نام مجموعه برنامه‌نویسی شرکت مایکروسافت است که دارای چند زبان برنامه‌نویسی است. این مجموعه ویژوال سی و ویژوال بیسیک و ویژوال فاکس‌پرو و چند ابزار دیگر را درون خود جای داده‌است. شرکت مایکروسافت در نوزدهم نوامبر ۲۰۰۷ ( بیست و هشتم آبان ماه هشتاد و شش ) ، رسماً نسخه ASP. NET ۳.۵ و ویژوال استودیو ۲۰۰۸ را عرضه کرد .

ویژوال استودیو ۲۰۰۸ ، ابزاری قدرتمند برای پیاده سازی برنامه های ASP. NET است . برخلاف نسخه قبل که صرفاً "مختص یک نسخه خاص frame work بود ( به عنوان نمونه در ویژوال استودیو ۲۰۰۳ ، نسخه ۱.۱ ASP.NET و در ویژوال استودیو ۲۰۰۵ نسخه ۲.۰ ASP. NET ) ، از ویژوال استودیو ۲۰۰۸ می توان به همراه چندین نسخه frame work استفاده کرد . شما می توانید از طریق یک لیست drop down ، نوع frame work خود جهت پیاده سازی برنامه ها ( به عنوان نمونه ASP. NET ۲.۰ ، ASP. NET ۳.۰ و یا ASP. NET ۳.۵ ) را انتخاب نمائید .

در ویژوال استودیو ۲۰۰۸ ، در محیط طراحی نیز تغییراتی در جهت قدرتمند تر شدن آن ایجاد شده است . افزودن امکاناتی نظیر هوشمندی در تایپ و اشکال زدائی کدهای جاوا اسکریپت و قابلیت مشاهده و حتی توقف در هسته کد frame work در حین اشکال زدائی نمونه هائی در این زمینه می باشند .

## ۱-۲- سیستم مورد نیاز جهت اجرای نرم افزار:



### ۱. Supported Architectures:

# x۸۶

# x۶۴ (WOW)

### ۲. Supported Operating Systems:

# Windows XP Service Pack ۳ or above

# Windows Server ۲۰۰۳ Service Pack ۱ or above

# Windows Server ۲۰۰۳ R۲ or above

# Windows Vista

# Windows Server ۲۰۰۸

### ۳. Hardware Requirements:

# Minimum: ۱.۶ GHz CPU, ۳۸۴ MB RAM, ۱۰۲۴x۷۶۸ display, ۵۴۰۰ RPM hard disk

# Recommended: ۲.۲ GHz or higher CPU, ۱۰۲۴ MB or more RAM, ۱۲۸۰x۱۰۲۴ display, ۷۲۰۰ RPM or higher hard disk

# On Windows Vista: ۲.۴ GHz CPU, ۷۶۸ MB RAM







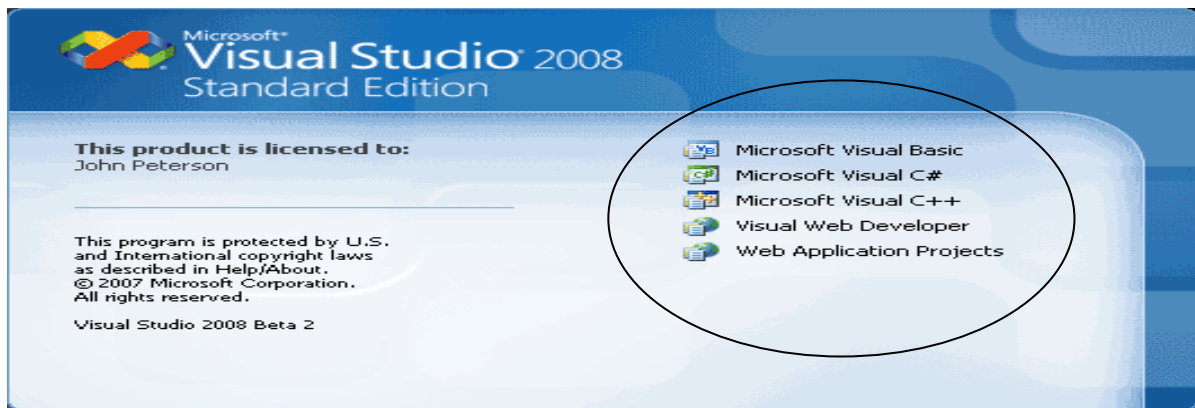
**فصل دوم**

# **مروری بر نرم افزار**

**VC#**



## ۲-۱- معرفی نرم افزار:



این نرم افزار محصول شرکت Microsoft بوده و شامل زبانهای برنامه نویسی :

Visual C# ,Visual Basic ,Visual J# ,Visual F#,Visual C++ ,Asp.net(VB,C#,g#),SQL Server, Visual Studio Solution& . . .

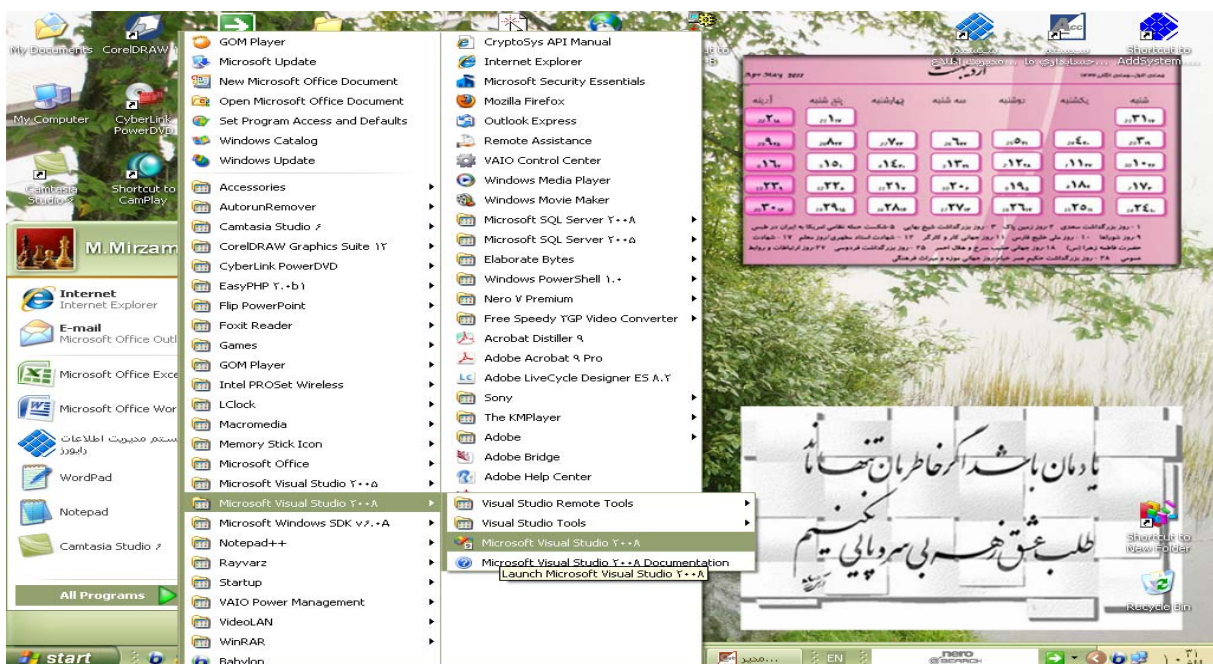
که نسخه ۲۰۰۳ و بالاتر از آن برنامه نویسی به سبک معماری لایه ای را نیز حمایت می کند .

قبل از این در C++ (پیوست ۱) هم با این معماری آشنا شدیم .

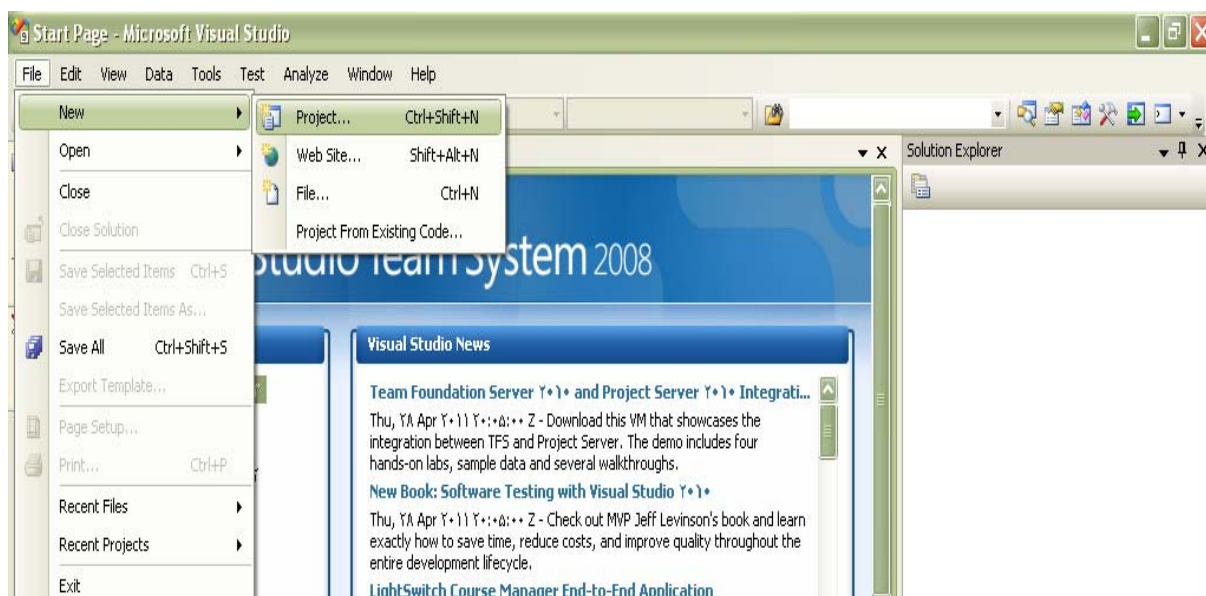
## ۲-۲- نحوه اجرای برنامه :

نحوه اجرای برنامه به این شرح است و با طی این مسیر ، نرم افزار اجرا می شود :

Start\All Program\Microsoft Visual Studio



File\New\Project



یک پروژه جدید ایجاد می کنیم .

### ۳-۲- قسمت ها ، اجزاء و زبان ها :

پس از اجرای برنامه ، پنجره آن ظاهر می شود که شامل قسمت های زیر است :

#### ۱- Project types

در این قسمت زبان برنامه نویسی موردنظر را که می تواند هر یک از

Visual C#

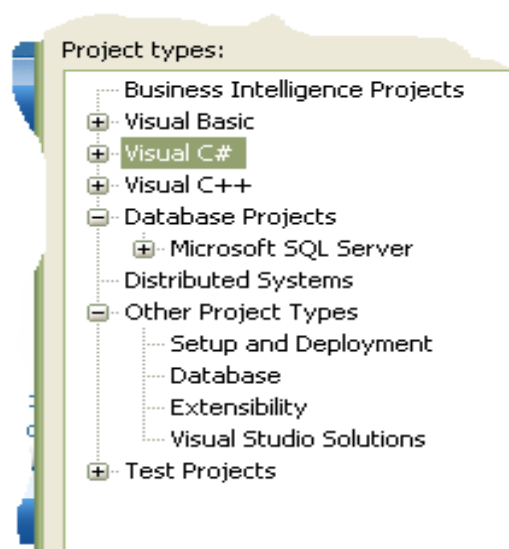
Visual Basic

Visual J#

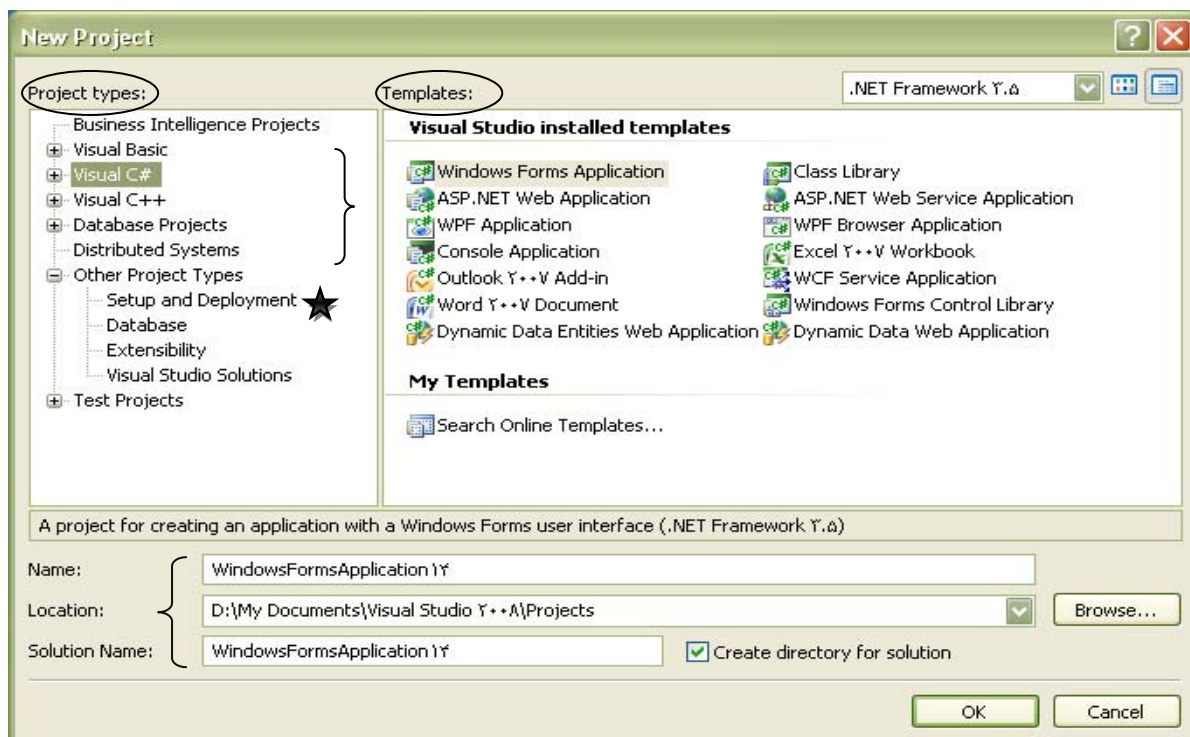
SQL Server

Visual Studio Solution

& . . .



باشد را انتخاب می کنیم .



★ در قسمت **Other Project** آیتم های **Database** (ایجاد پایگاه داده) و یا **Setup And Deployment**

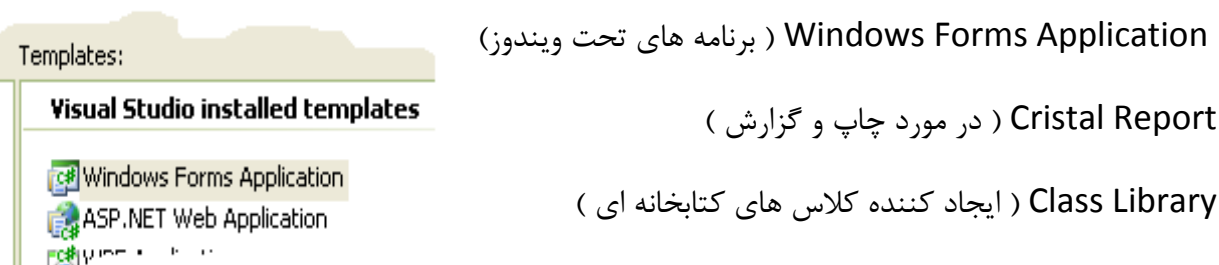
( ساختن فایل **Setup\*** ای از پروژه ) را می توانیم انتخاب کنیم .

\*یعنی می خواهیم برنامه امان را به صورت **Package** درست کنیم، که آن را هر کجا بگذاریم **Setup** شود .

ما در این قسمت گزینه **Visual Studio Solution** را انتخاب می کنیم ، این زبان برنامه نویسی با

ساختار معماری لایه ای را حمایت می کند . (که پروژه ها(لایه ها) می تواند به آن اضافه گردند)

قسمت بعدی پنجره ظاهر شده **Templates** می باشد ، در این قسمت نوع پروژه که می تواند :

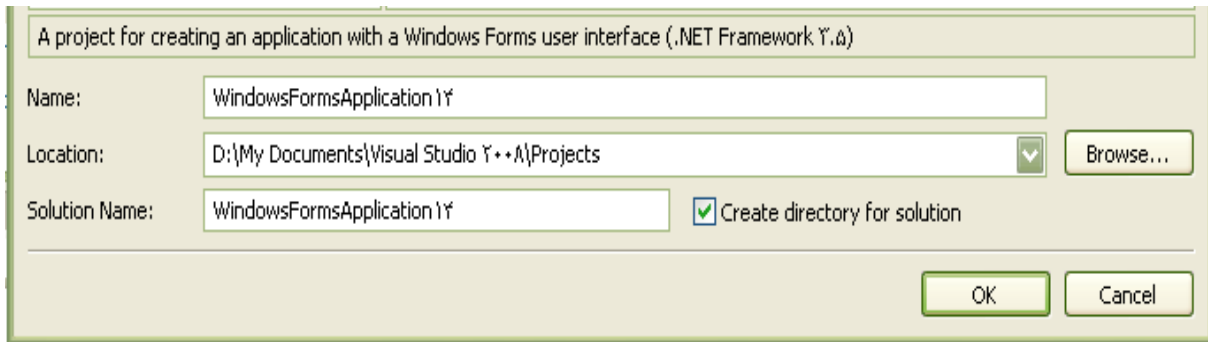


و ... باشد را می توانیم انتخاب کنیم ،

که ما معمولاً "نوع **Windows Forms Application** ( برنامه های تحت ویندوز) را انتخاب می کنیم .

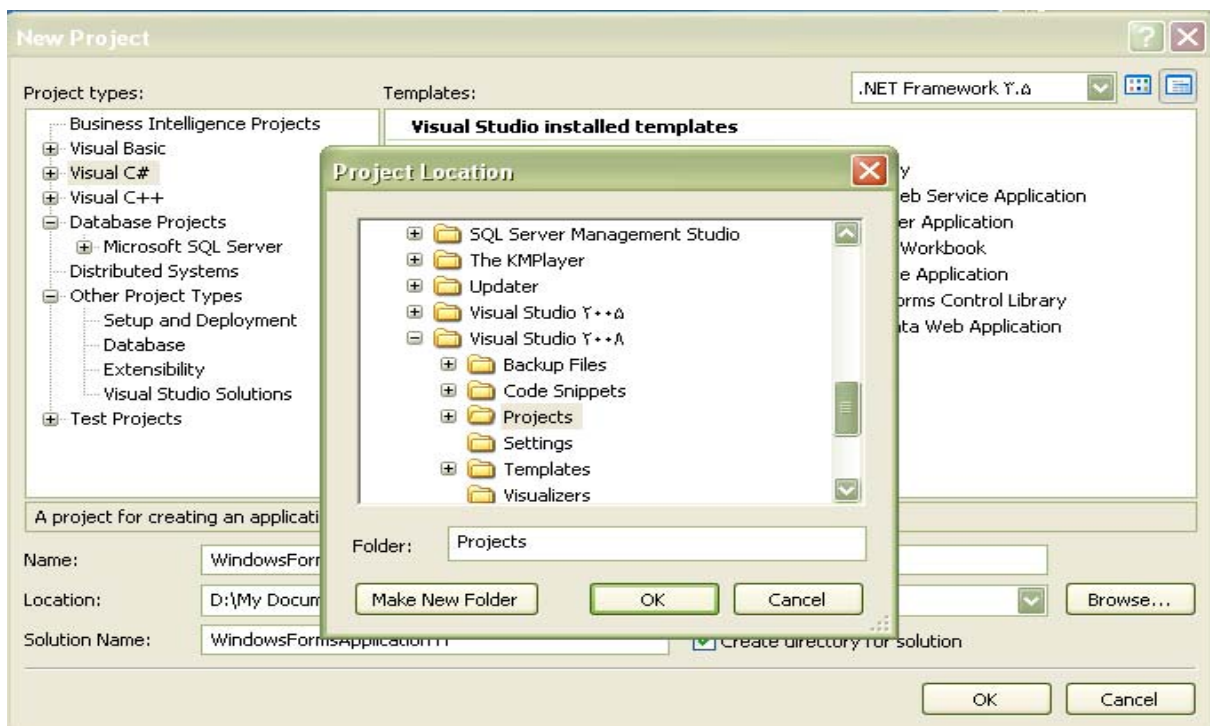
\*برای انجام کارهای وب سایتی ، قسمت **ASP.Net** را انتخاب می کنیم .

پنجره **New Project** دارای قسمت های دیگری هم می باشد که به شرح زیر است :



**Name** : در این قسمت نام پروژه را تعیین می کنیم .

**Location** : در این قسمت مکان ذخیره سازی را خواهیم داشت .



**Create directory for solution** : در این قسمت **Solution** لازم برای **Support** معماری لایه ای

ساخته می شود .

بنابراین برای اینکه بتوانیم معماری لایه ای را نیز حمایت کنیم ، در **BOX** مربوط به قسمت ، یا گزینه

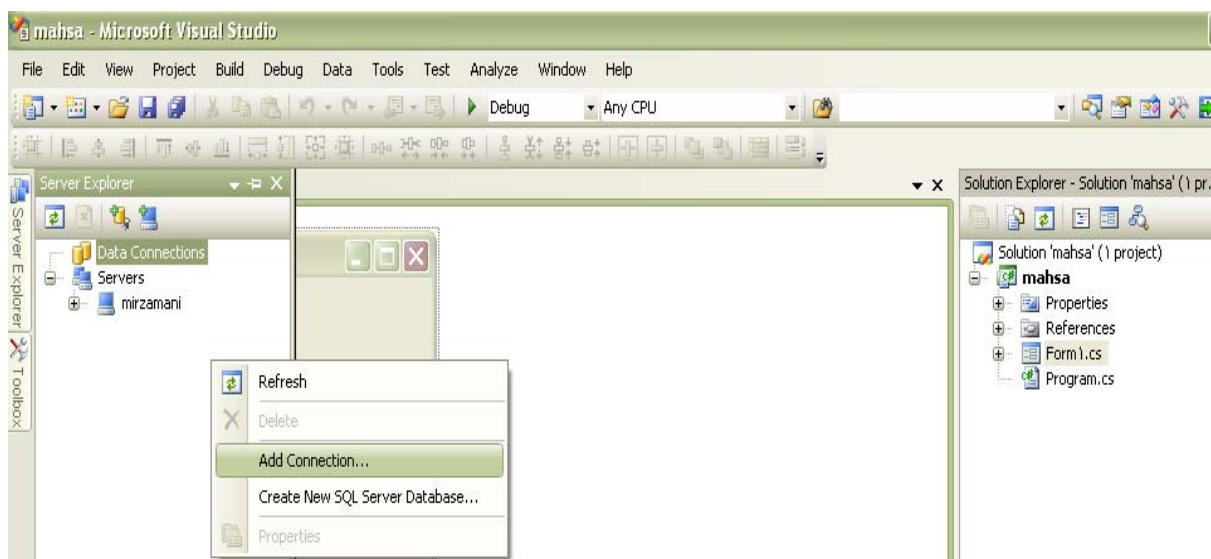
**Create directory for solution** تیک را می زنیم و در قسمت **Solution Name** یک نام هم برای

**Solution** امن تعیین می نمائیم . ( به این ترتیب یک **Solution** داریم که می توانیم پروژه ها و لایه ها

را به آن **Add** کنیم و چند لایه به آن اضافه کنیم )

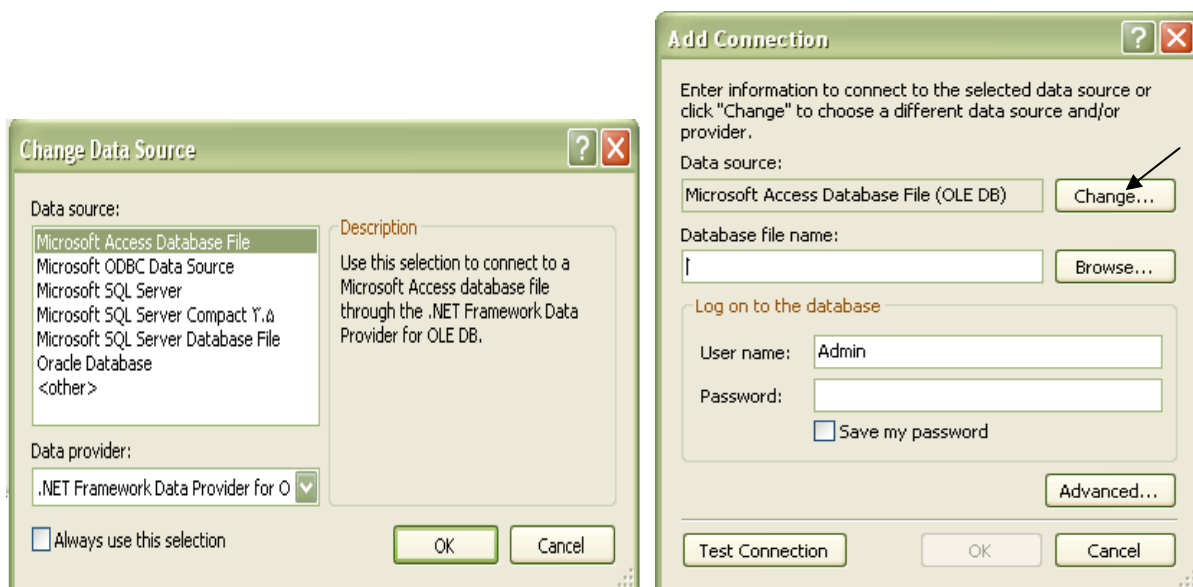
و در نهایت هم **OK** می کنیم .

## ۴-۲- معرفی محیط :

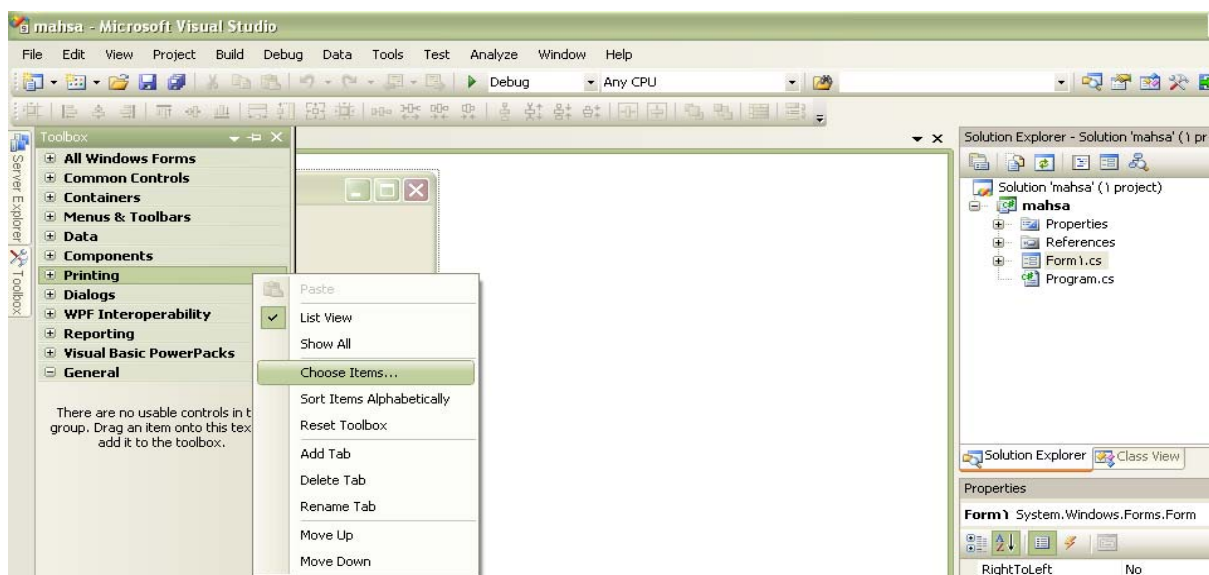


### ۱-۴-۲- Server Explorer :

این بخش ، جهت اتصال به انواع منابع داده و بانکهای اطلاعاتی به کار می رود که به طور پیش فرض SQLServer ( پیوست ۲ ) به کار می رود ،(طریقه فعال کردن از طریق menu view می باشد). برای این کار روی قسمت Data Connection رایت کلیک می کنیم ، و سپس با کلیک برروی Add Connection ارتباط مورد نظرمان را انتخاب می کنیم که مثلاً" می خواهیم به چه چیزی وصل شویم .

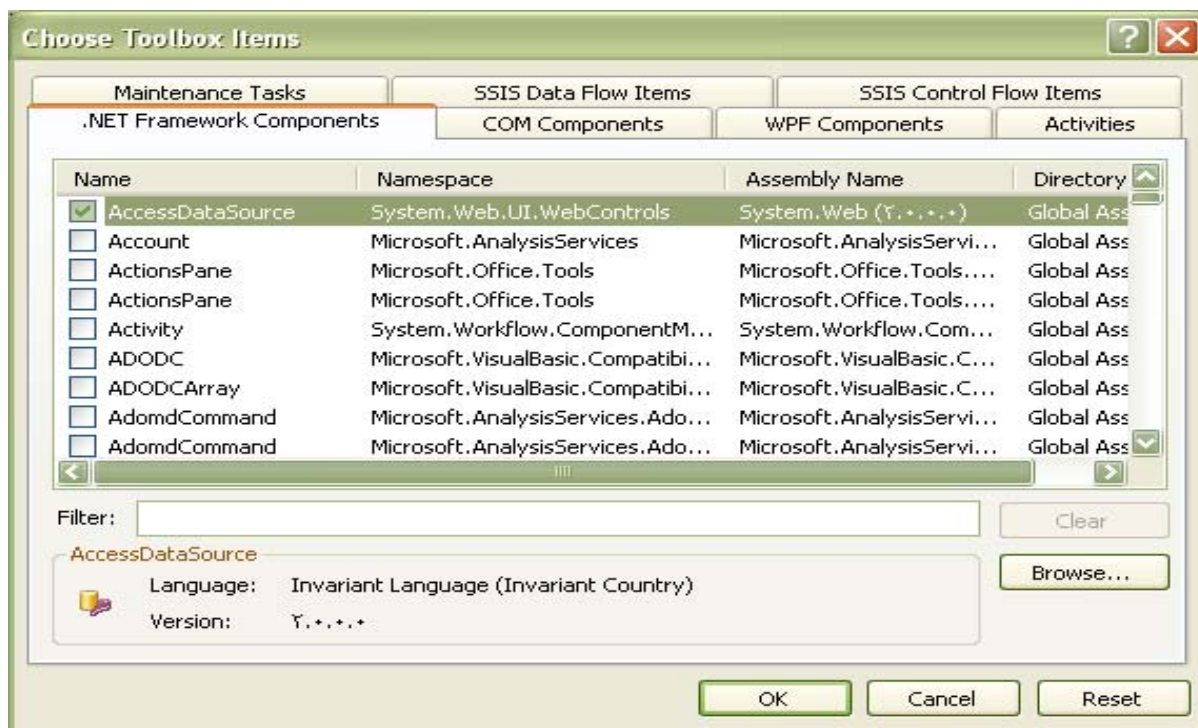


## ۲-۴-۲ : Toolbars



در این پنجره ، بیشتر Class ها و Component های موجود و قابل دسترس هستند و معروفترین آنها لیست شده اند ، برای دسترسی به بقیه کلاس ها و حتی اضافه کردن Component جدید می توانیم از دانلود توسط اینترنت و یا CD و ... استفاده کنیم . ( مثل استفاده از قفل های سخت افزاری )

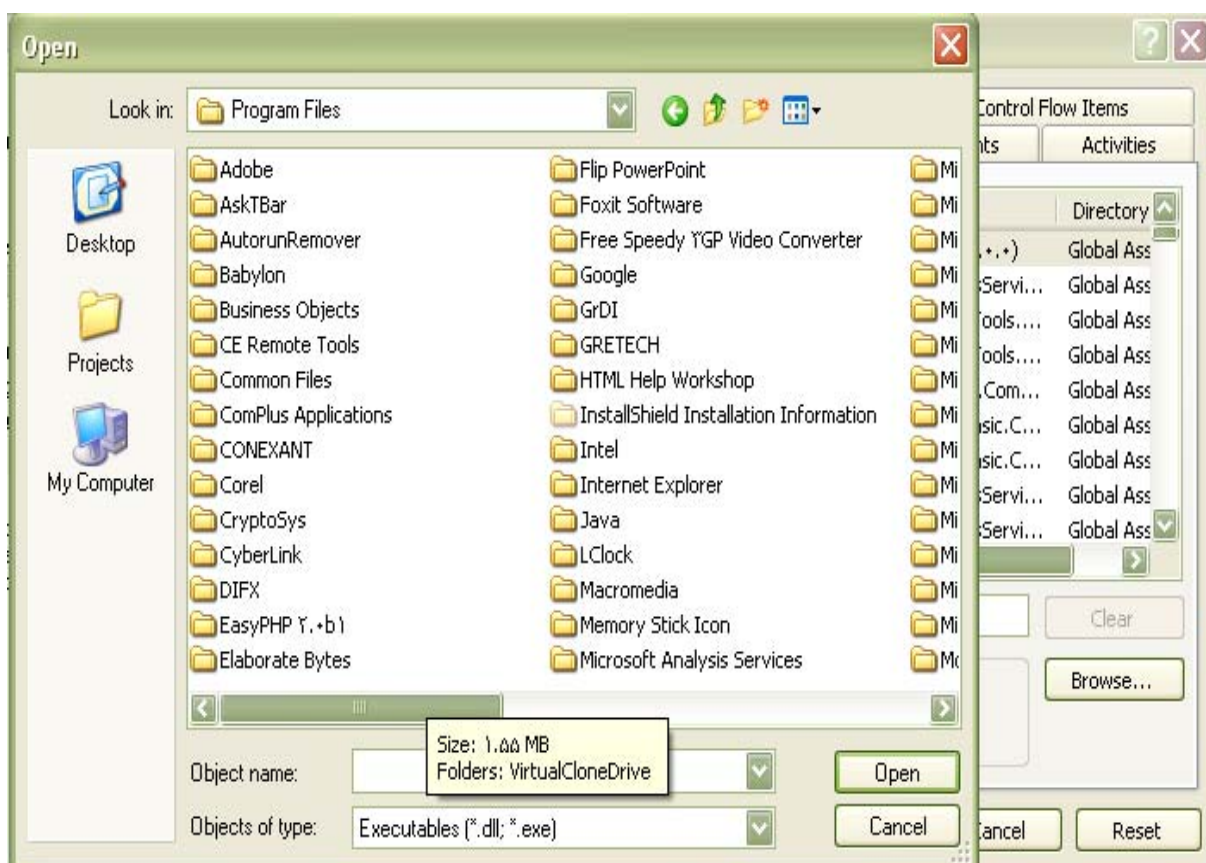
برای این کار کافیست که روی Toolbars رایت کلیک کنیم و فرمان Choose Item را انتخاب کنیم ، تا پنجره آن ظاهر شود ،



تیک کنار Component مورد نظر را می زنیم ، و ok می کنیم ، به لیست Toolbars اضافه می شود.



ویا با کلیک بر روی دکمه Browse می توانیم به دلخواه Component های مورد نظرمان را انتخاب کنیم .

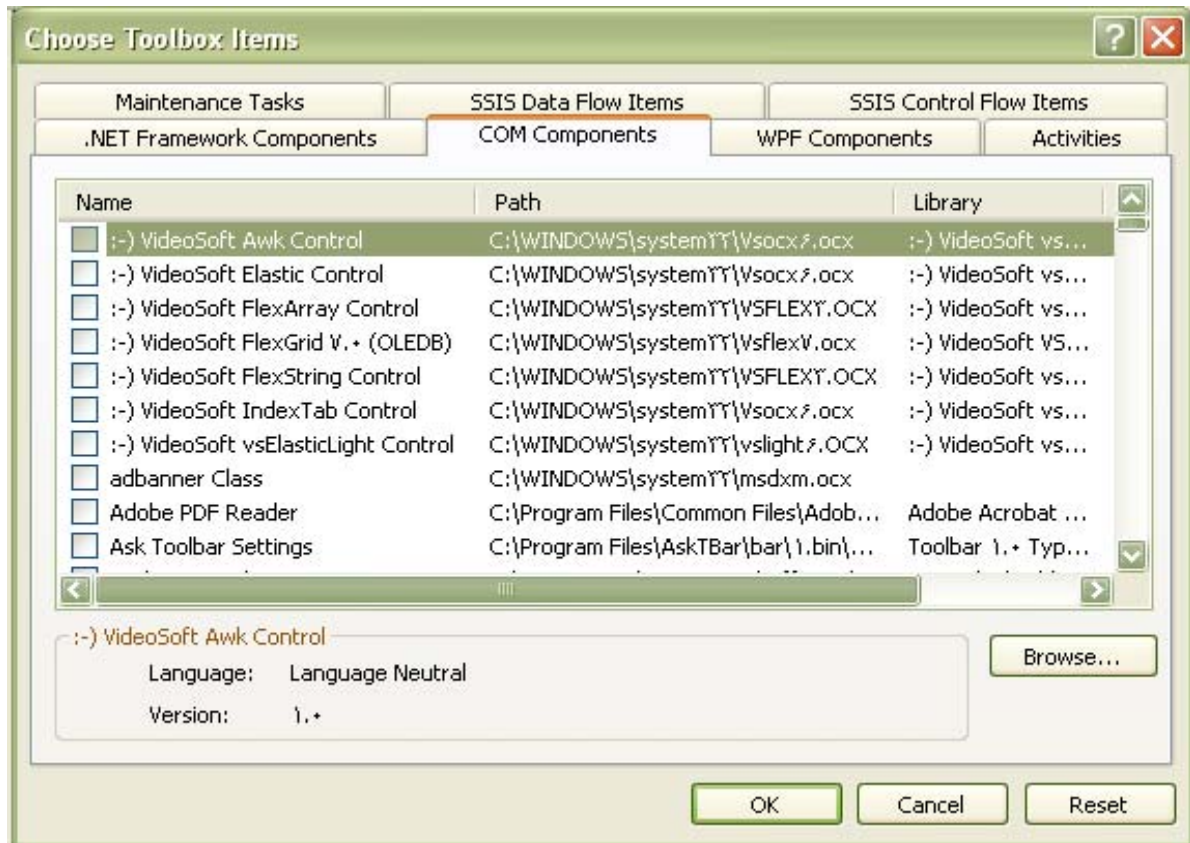


\*نکته : در زمان انتخاب هر یک از Component ها می بایست ، توجه داشته باشیم که آن Component دارای Expired Date نباشد ، چراکه در این صورت پس از مدتی ، با به پایان رسیدن زمان مصرف آن Component خاص ، برنامه ما دچار مشکل می گردد .

( مثل Component های مربوط به دکمه ها و ..... که ممکن است پس از مدتی Expired Date گردد و تمام برنامه ما دچار مشکل گردد.)

مثلاً" برای پخش و یا اجرای فیلم در این قسمت می توانیم برنامه Media Player را Add کنیم که این برنامه را با استفاده از امکانات Windows برای ما فعال می کند .

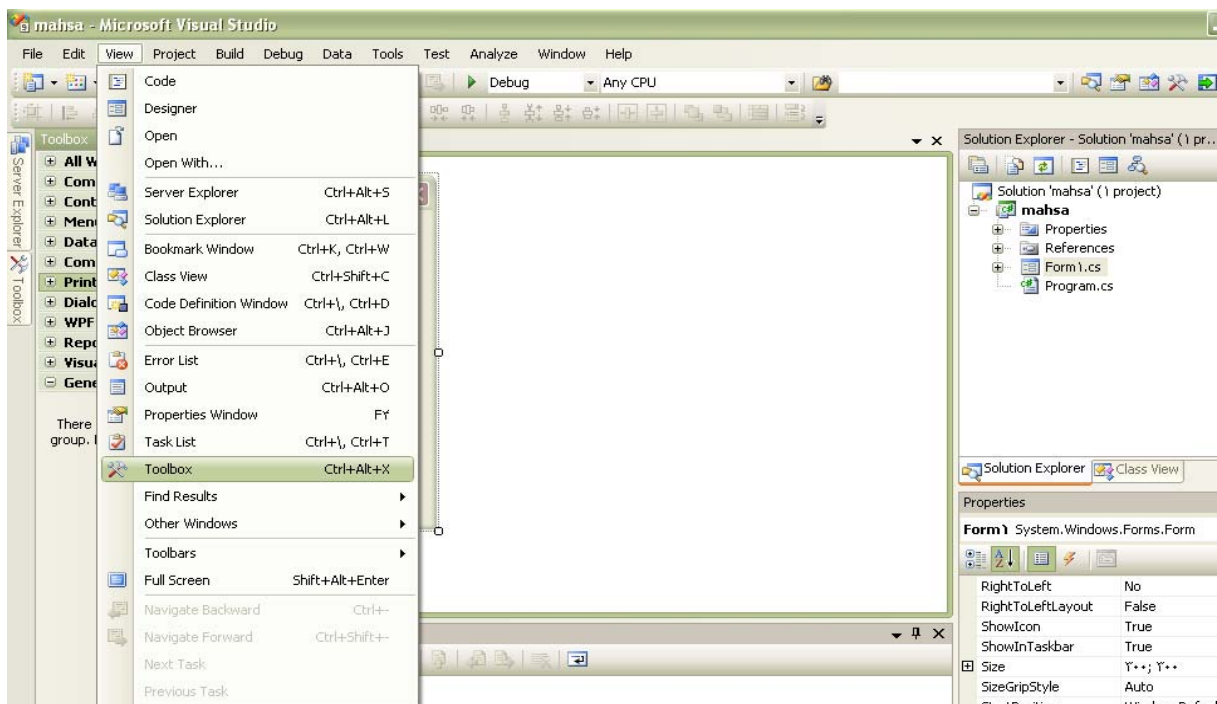
Component های دلخواه را در قسمت COM Component می توانیم انتخاب کنیم ،



که پس از انتخاب مثلاً "Media Player" و انتخاب کردن فایل دلخواه با Debug کردن آن را اجرا

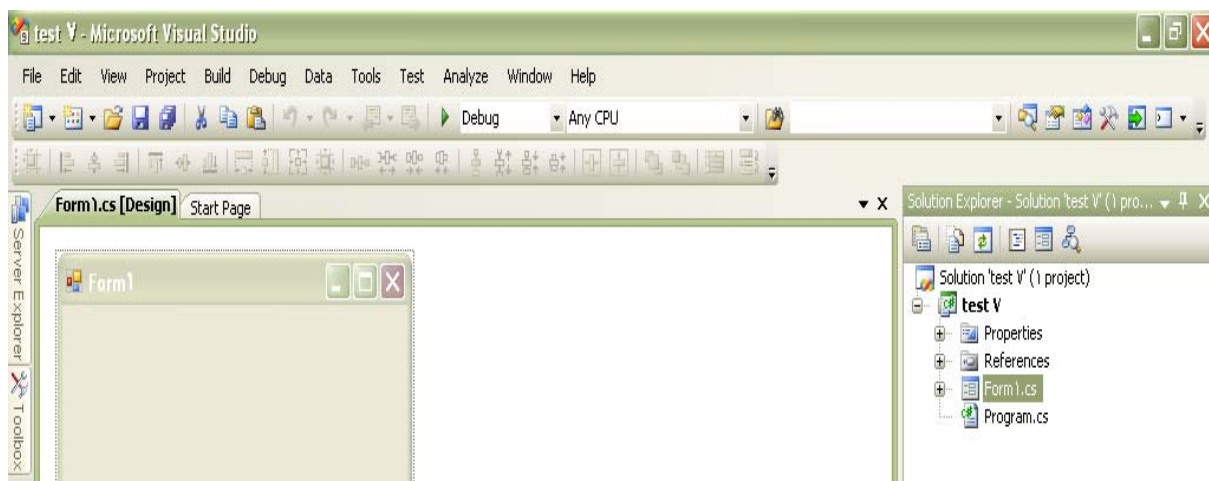
می کنیم و مشاهده می کنیم که Component انتخاب شده ، فعالیت مورد نظر را انجام می دهد .

\*اگر Toolbox فعال نبود، از طریق قسمت View\Toolbox هم می توانیم به آن دسترسی داشته باشیم .



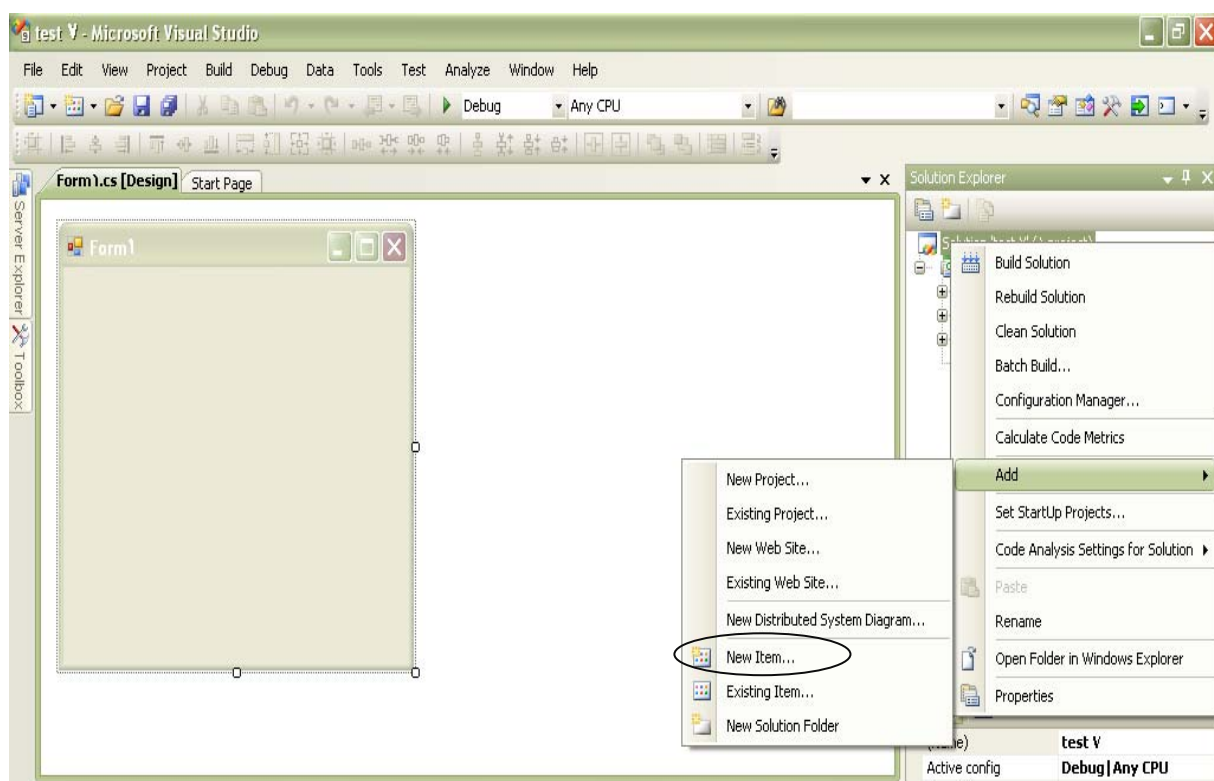
## ۳-۴-۲ - Solution Explorer

Solution Explorer، خود Solution، لایه ها (پروژه ها) و متعلقات هر لایه، شامل Form ها و همچنین Web Form ها Class ها و..... در این پنجره قابل دسترس می باشند، در صورتیکه ما در محیط Web فعالیت کنیم Web Form ها را مشاهده خواهیم کرد و فایل های کد نویسی و....



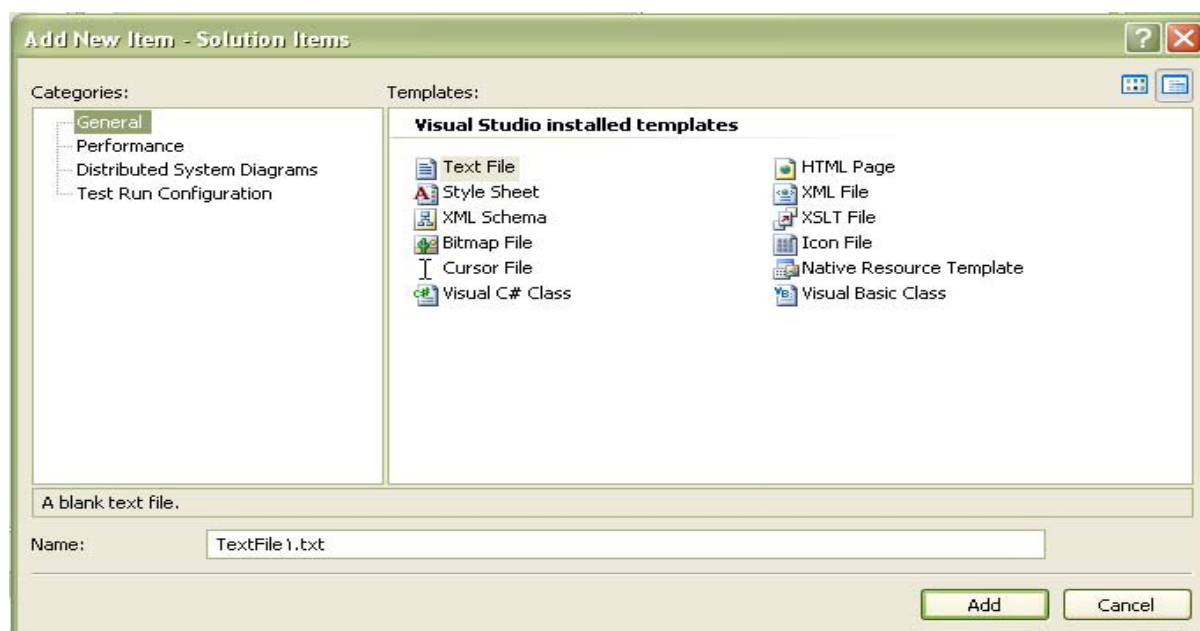
هم در این پنجره قابل دسترس می باشند.

همچنین در این پنجره می توانیم با رایت کلیک هر چه را که می خواهیم Add کنیم، کافایت که روی نام Solution رایت کلیک کرده و مورد دلخواه را انتخاب کنیم، برای آن یک نام تعیین کرده،



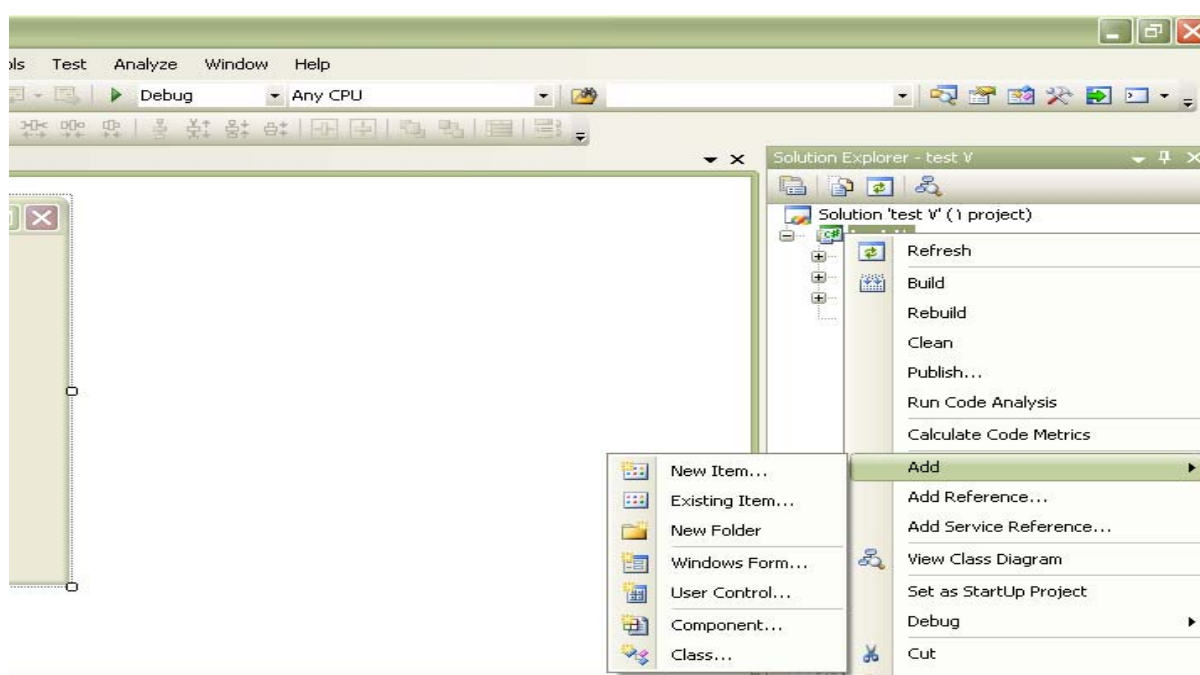
آنچه که Add می کنیم ، می تواند یک پروژه ، یک Item ، یک Class ، یک سرویس ویندوزی ، یک پایگاه داده و .... باشد .

\*اگر بخواهیم آیتم جدیدی را به Solution اصلی امان Add کنیم ، روی Solution اصلی رایت کلیک می کنیم و پس از انتخاب Add ،

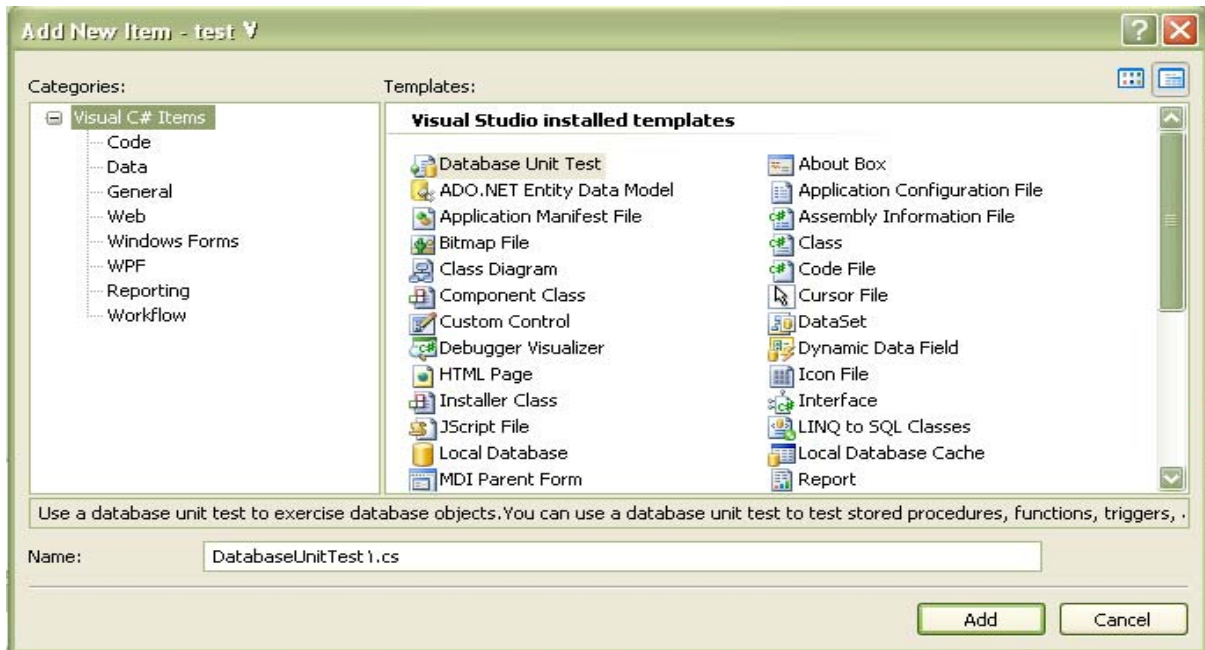


آن آیتم جدیدی که به پروژه ما اضافه می شود ،

و یا اگر بخواهیم آیتم جدیدی را به پروژه امان Add کنیم ، روی نام پروژه رایت کلیک می کنیم و پس از انتخاب Add ،



آن آیت می تواند :



- یک کلاس باشد که در بحث شیء‌گرایی اضافه می شود

- یک فرم دیگر باشد

- یک سرویس ویندوزی باشد

- یک پایگاه داده از نوع Database ای باشد و.....

مثلاً" اگر یک Class جدید Add کنیم ، یک کلاس در بحث شیء‌گرایی اضافه می شود که ما می توانیم با

استفاده از آن کار حرفه ای Database ای انجام دهیم .

مثلاً" ما می توانیم خودمان کلاسهای مربوط به :

Insert -

Delete - و.....را بنویسیم .

در قسمت اضافه کردن یک فرم می توانیم فرم دیگری را به پروژه امان اضافه کنیم و اگر کار چاپی داشته

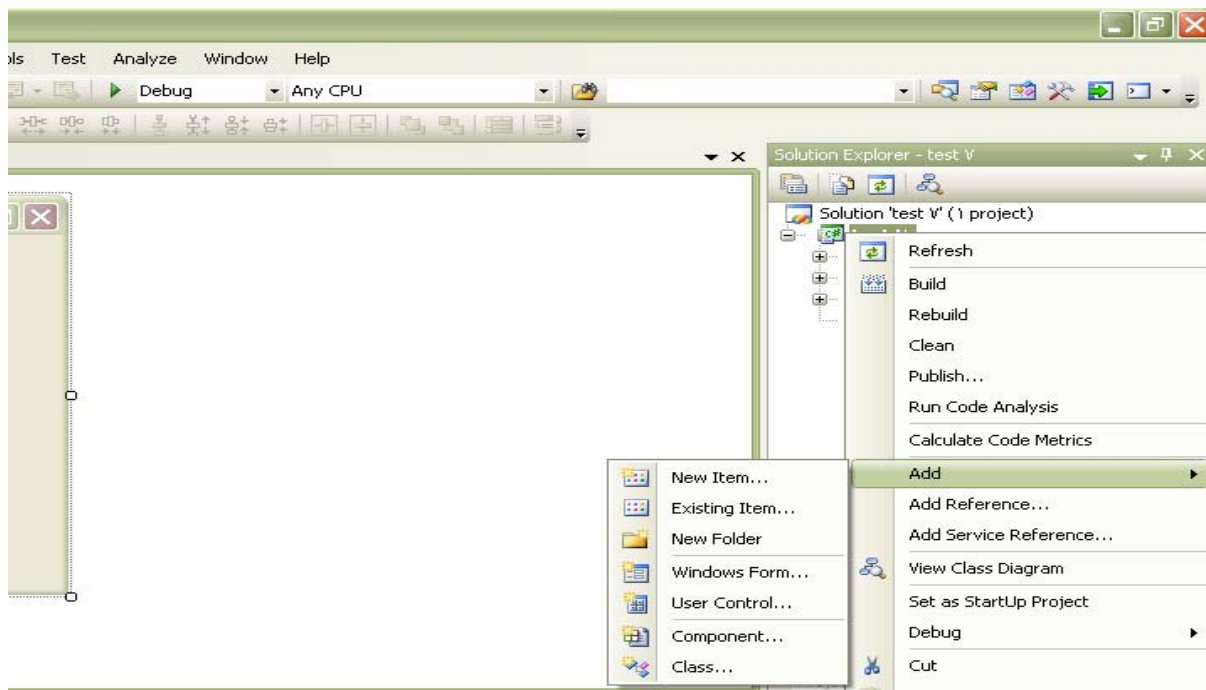
باشیم می توانیم با استفاده از Crystal Report به یک Wizard دسترسی پیدا کنیم که می پرسد که ما

می خواهیم به کدام قسمت از Database وصل شویم و چه کاری انجام دهیم و چه قسمتی را می خواهیم

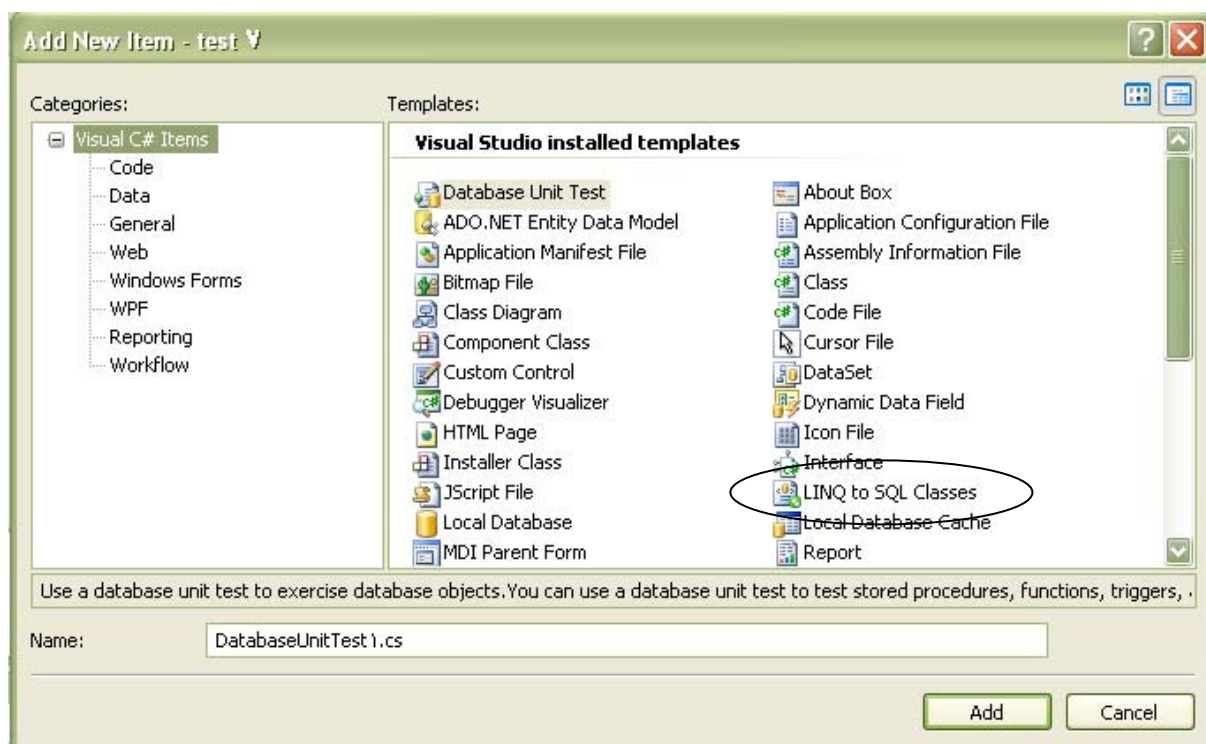
چاپ کنیم . ( که در پروژه ما آن قسمت خاص صفحه چاپ خواهد شد )

و بسیاری از مباحث دیگر که در ادامه به آنها خواهیم پرداخت .

\*اگر بخواهیم کلاس ها را به صورت اتوماتیک خودش ایجاد کند ، روی نام پروژه امان رایت کلیک می کنیم  
و پس از انتخاب Add ،

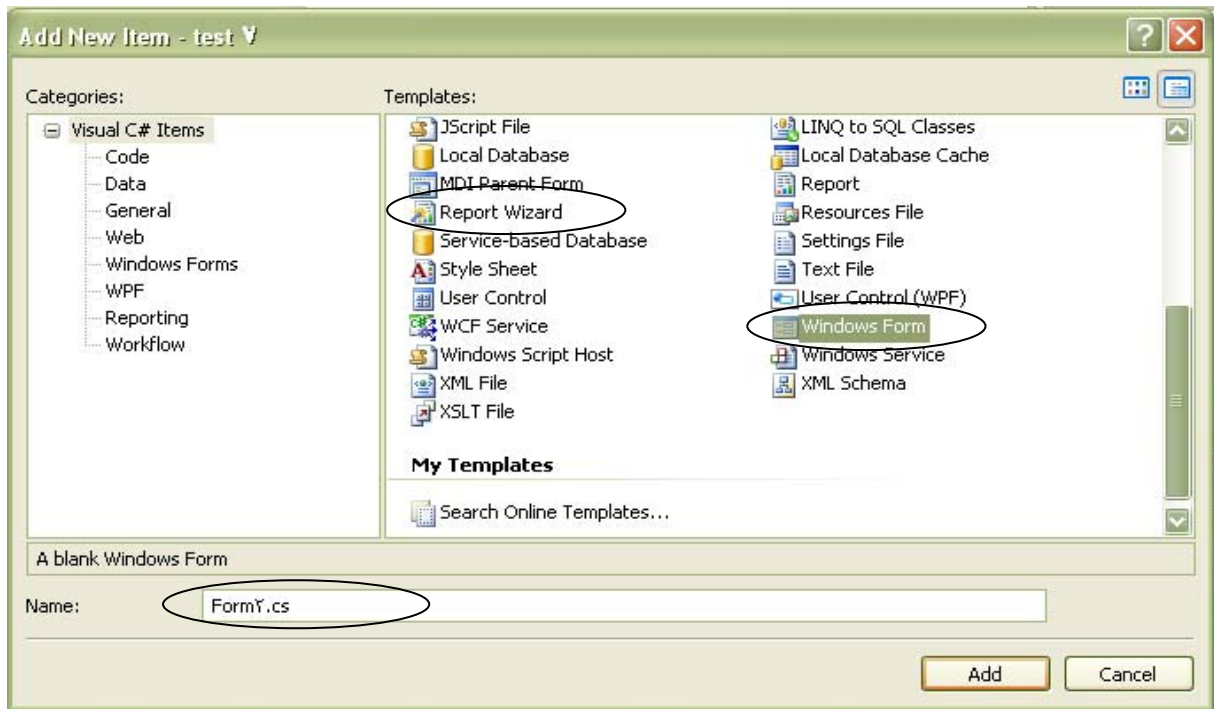


با انتخاب New Item از پنجره جدید باز شده آئتم LINQ to SQL Classes استفاده می کنیم ،



که از Version ۲۰۰۸ این امکان در این نرم افزار آورده شده است و قبل از آن ، مثلاً" در نسخه ۲۰۰۵ این امکان وجود نداشت .

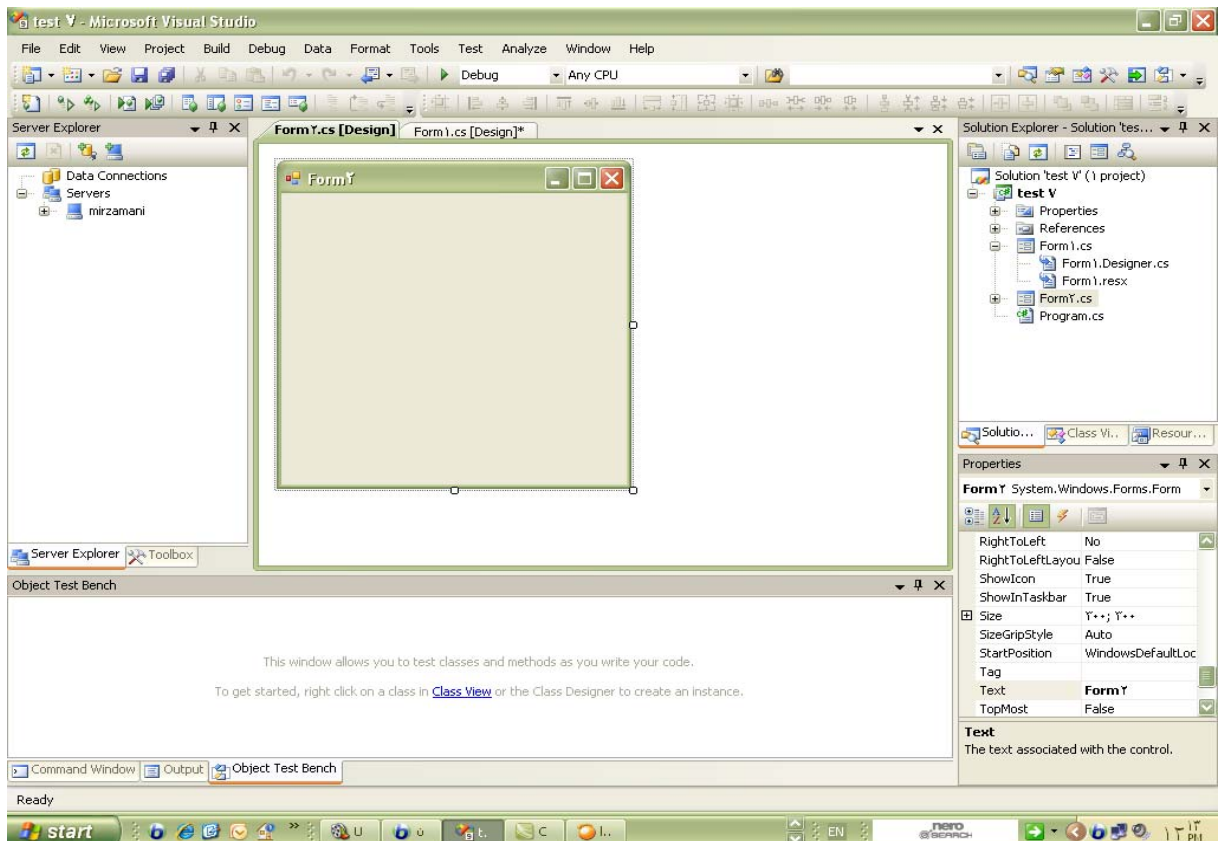
بنابراین اگر بخواهیم یک فرم داشته باشیم از Windows Form استفاده می کنیم ،



و همینطور از قسمت Report Wizard برای کارهای چاپ استفاده می کنیم .

با انتخاب Windows Form و ثبت یک نام ، یک فرم مثلاً "Form۲" به پروژه ما اضافه و در لیست ما

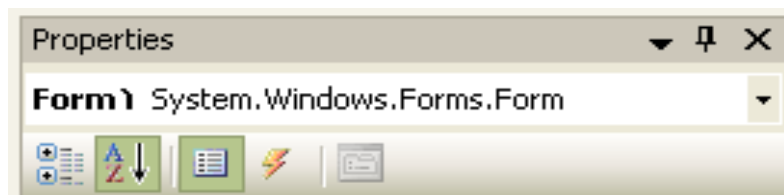
نمایش داده می شود .




همانگونه که مشاهده می کنیم ، این محیط در مقایسه با Access (پیوست ۳) بسیار سخت تر است ، ولی امکانات و ویژگیهای بسیاری دارد که در اختیار ما قرار می دهد و در تهیه پروژه دلخواه دست ما را باز تر و امکانات ما را بیشتر می کند .


#### ۴-۲-۴ Properties :


پنجره بعدی ، پنجره Properties می باشد ،

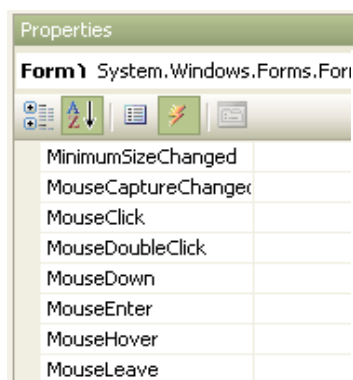


که صفات و عملیات شیء جاری انتخابی در این پنجره قابل دسترس می باشد .  
مثلاً" با انتخاب هر یک از اشیاء قسمت های مختلف زیر را خواهیم داشت :

۱-  Categorized که گروه بندی را نشان می دهد .

۲-  Alphabetical که مرتب کننده به ترتیب حروف الفبا است .

۳-  Events که بیانگر رخداد ها و اتفاق هایی است که پیش می آید ، مثلاً" :



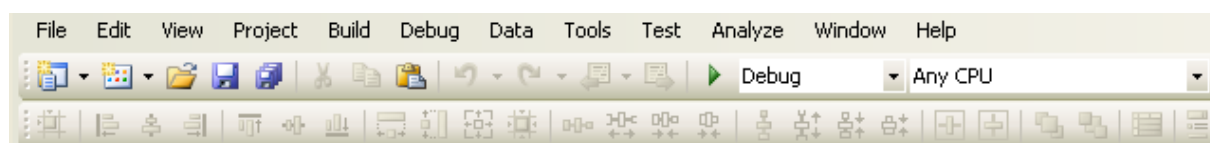
وقتی که با Mouse روی آن شیء قرار می گیریم ، چه اتفاقی رخ دهد ،

یا وقتی که با Mouse روی آن شیء کلیک می کنیم ، چه اتفاقی رخ دهد ،

یا وقتی که با Mouse روی آن شیء عبور می کنیم ، چه اتفاقی رخ دهد ،

و ...

طریقه اجرای پروژه به این ترتیب است که روی علامت Play کلیک کرده ، از کلید میانبر F5 استفاده کرده و یا در منوی Debug فرمان Start Debugging را اجرا می کنیم .

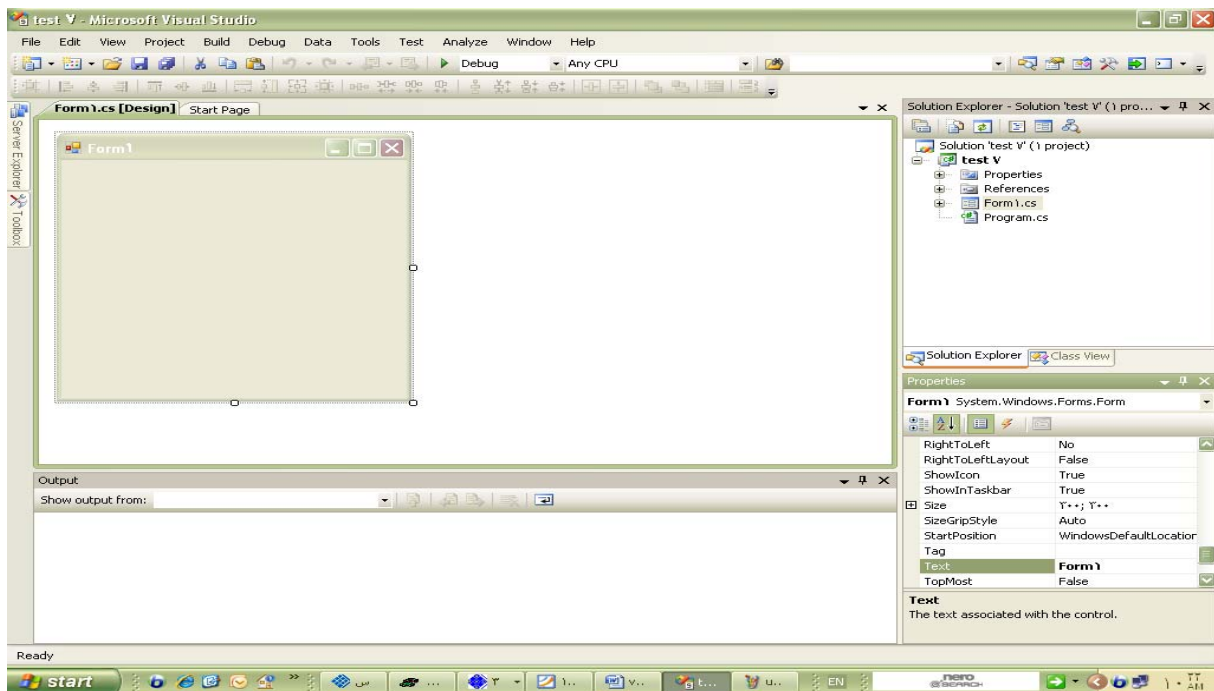


طریقه خروج از پروژه در منوی Debug فرمان Stop Debugging و یا کلید میانبر Shift + F5 می باشد.



## ۵-۲- یک تمرین ساده :

به عنوان تمرین یک پروژه باز می کنیم ، در قسمت New با انتخاب Project نوع پروژه امن را در قسمت زبان برنامه نویسی Visual c# نوع Windows Forms Application را انتخاب و تیک مربوط به Create directory for solution را هم می زنیم و نام solution را هم تعیین می کنیم ، سپس با انتخاب یک نام و محل ذخیره سازی و کلیک بر روی OK پروژه ما ایجاد می شود .  
به این ترتیب فرم برنامه ظاهر می گردد .

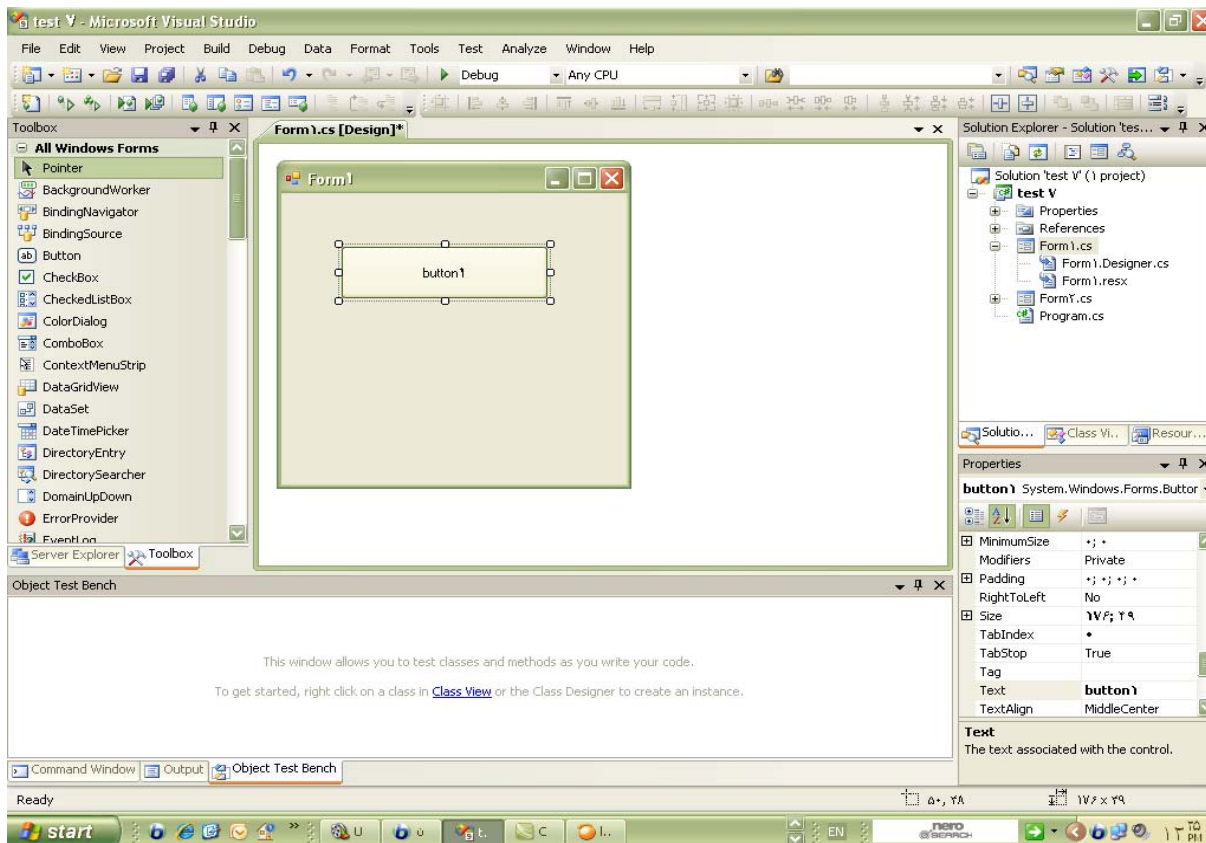


قبلاً" با قسمت های مختلف محیط آشنا شدیم ، در قسمت Solution Explorer خود Solution و لایه های مختلف آن را مشاهده می کنیم ، اگر بخواهیم لایه جدیدی به پروژه اضافه کنیم ، روی Solution رایت کلیک می کنیم و با انتخاب Add می توانیم آنچه را که می خواهیم انتخاب و اضافه کنیم و اگر بخواهیم به پروژه امن چیزی اضافه کنیم روی پروژه رایت کلیک می کنیم و آنچه را که می خواهیم Add می کنیم .

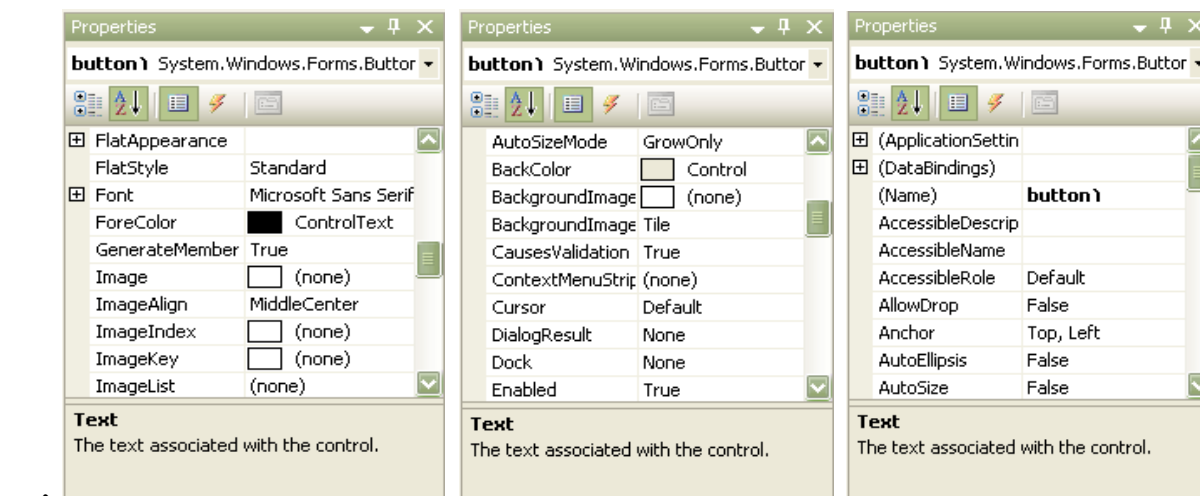
مثلاً" یک فرم اضافه می کنیم ، پروژه ما به طور پیش فرض دارای ۱ Form است و با Add کردن مثلاً" یک فرم ، به صورت Windows Form یک ۲ Form نیز به پروژه ما اضافه می شود .

\*توجه داریم که پروژه ما می تواند چندین Form دلخواه داشته باشد .

در پنجره **Properties** هم همانطوری که گفته شد ، صفات و رفتار های مربوط به هر یک از شیء ها را خواهیم داشت . مثلاً " وقتی بر روی یک **Button** قرار می گیریم ، رفتارهای مربوط به آن **Button** را خواهیم داشت و می توانیم برای هر یک از آنها دستورات و **Action** های لازم را بنویسیم .



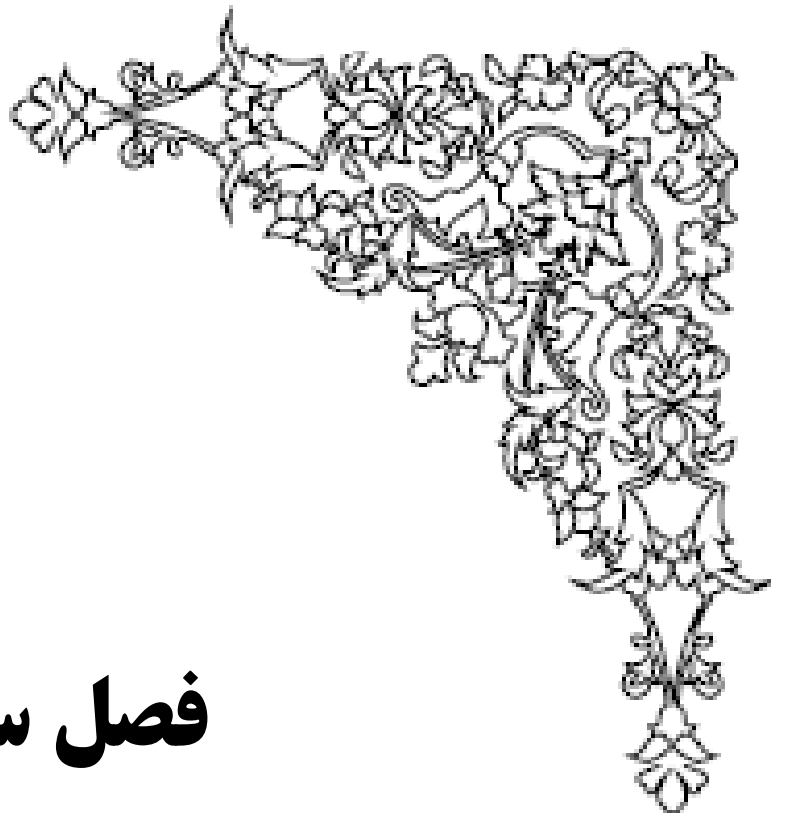
همینطور صفات مربوط به هر عنصر را در قسمت **Properties** مربوط به آن شیء به راحتی می توانیم مشاهده ،



....و

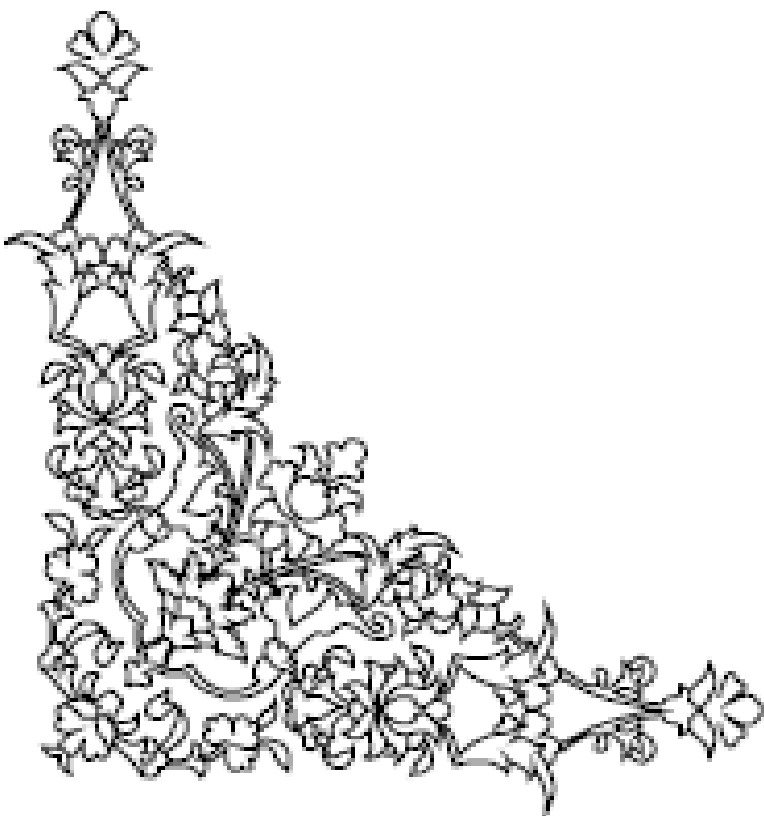
مقدار دهی و یا تغییر دهیم .





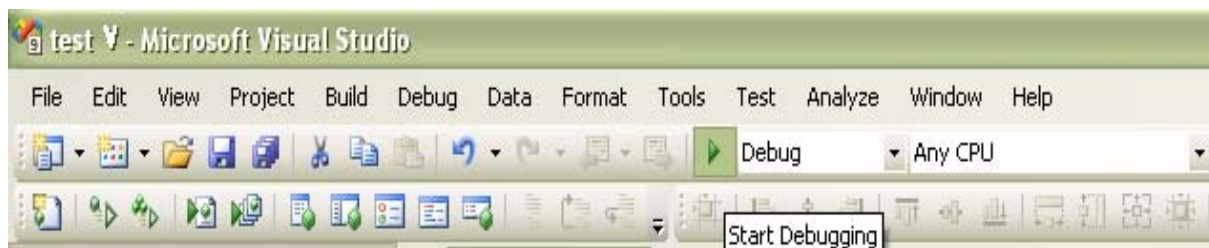
**فصل سوم**

# **اجرای پروژه**



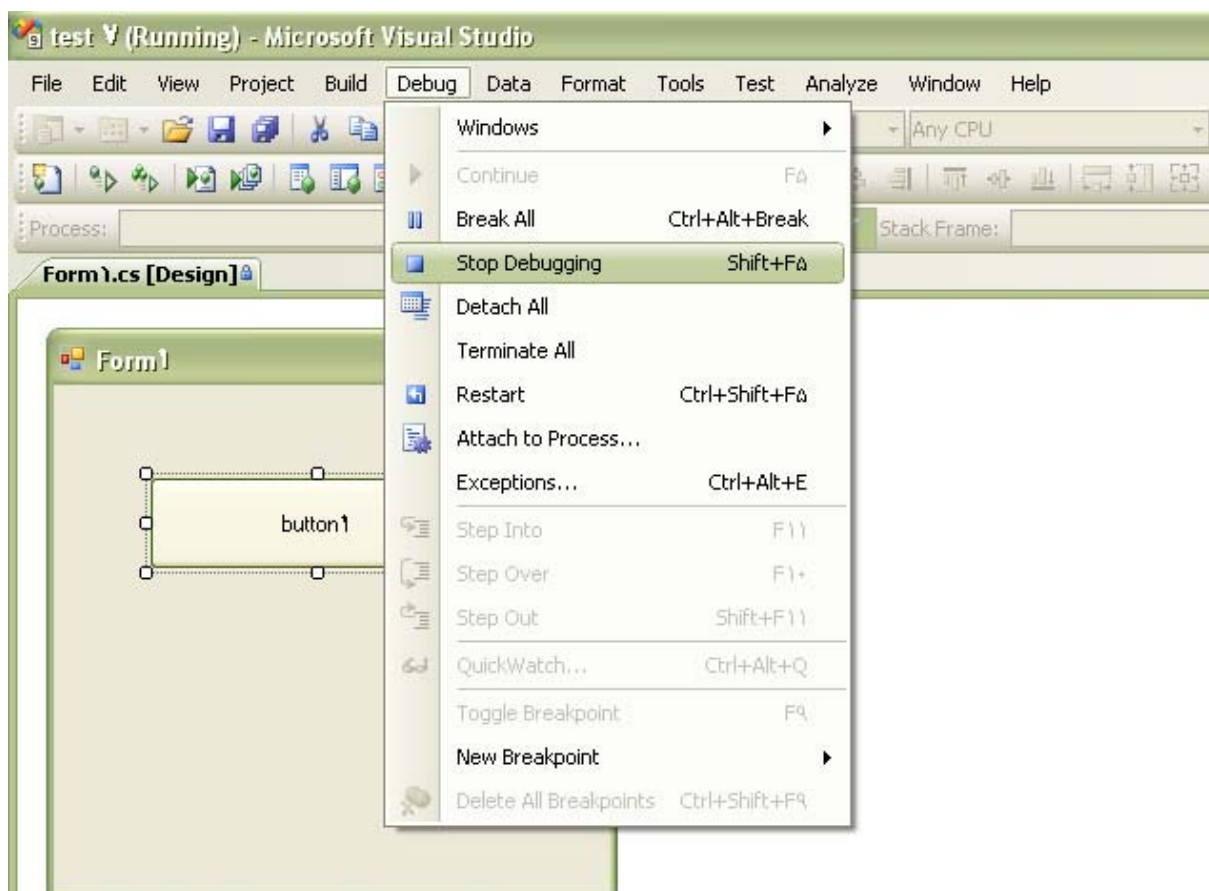
### ۳-۱ - طریقه اجرای پروژه :

همانطوریکه قبلاً گفته شد برای اجرای پروژه در منوی Debug فرمان Start Debugging ،



را انتخاب می کنیم و یا روی آیکن Debug روی علامت Play کلیک می کنیم و یا اینکه از کلید میانبر F5 استفاده می کنیم .

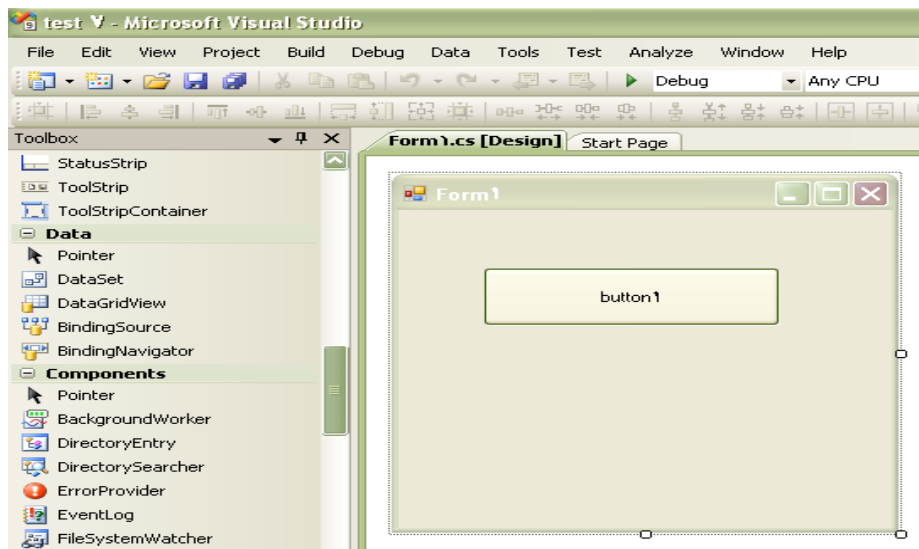
پس از هریک از این کارها ، پروژه ما در مد اجرا قرار می گیرد ، طریقه خروج از حالت یا مد اجرا به این ترتیب است که از همان منوی Debug فرمان Stop Debugging را انتخاب



و یا با استفاده از کلید های Shift +Debug به همراه هم اجرای پروژه متوقف می گردد.

## ۳-۲- معرفی عناصر:

حال می خواهیم به معرفی عناصر بپردازیم ، اولین عنصری که معرفی کنیم عنصر Form است .



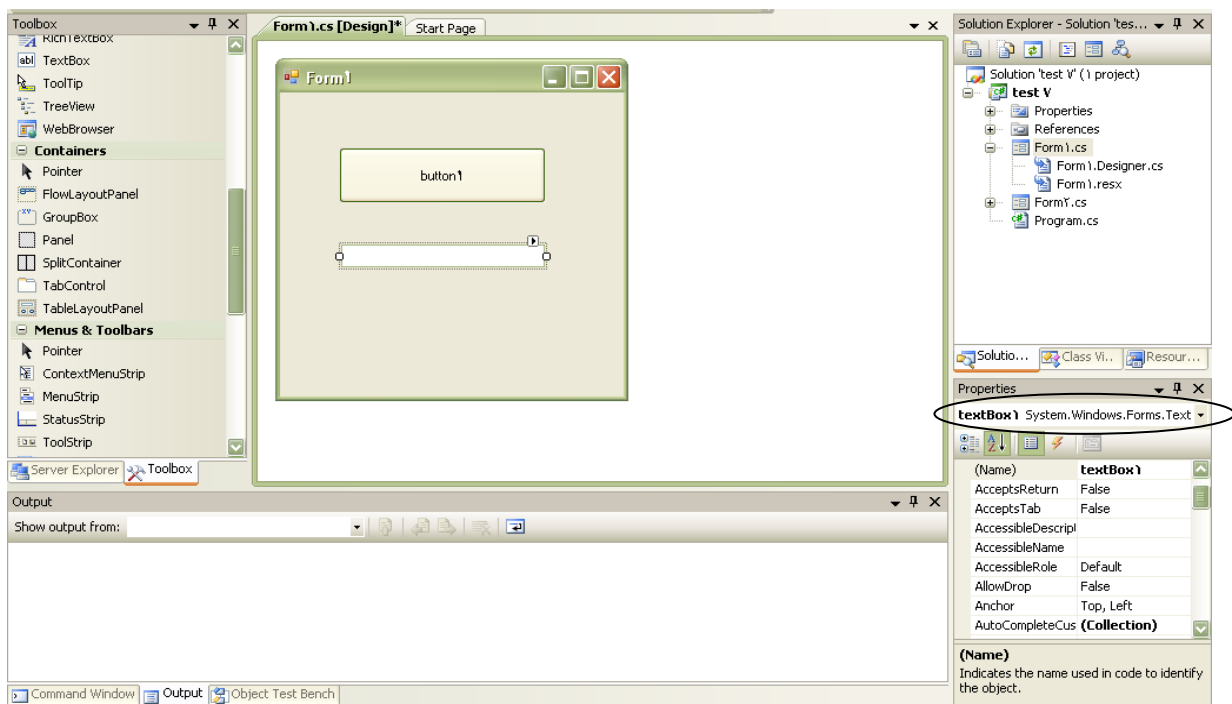
۳-۲-۱- **Form** : اصلی ترین عنصر در محیط Visual Studio می باشد که در برگزیده عناصر دیگر

بوده و صفاتی را به آنها ارث می دهد،

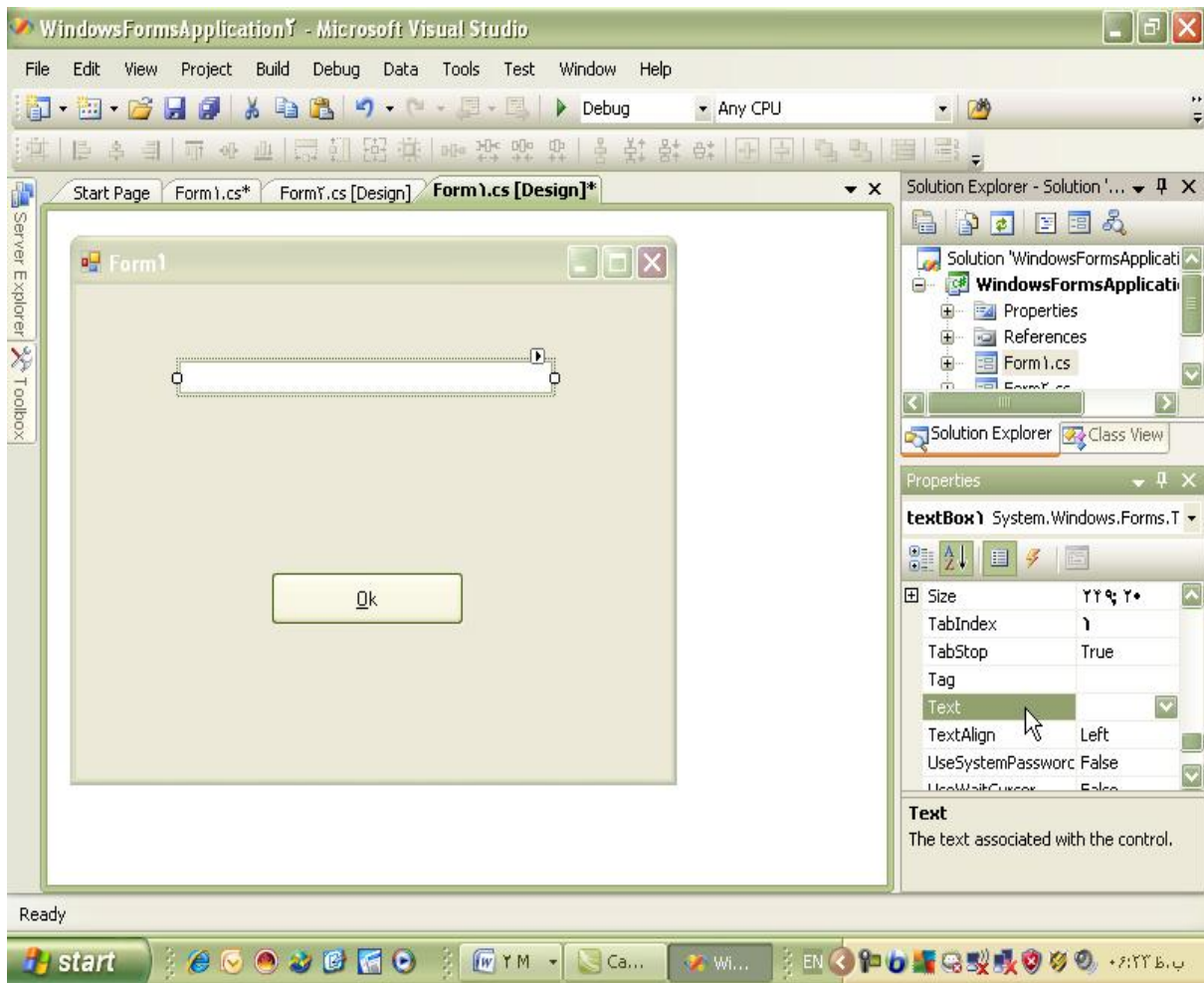
مهم ترین آن صفات عبارتند از :

۳-۲-۲- **Name** : نام هر عنصر اولین و مهم ترین صفت آن می باشد که توسط این صفت می توان به

بقیه صفات عنصر دسترسی داشت .



صفتی که به محتویات عنصر دسترسی داریم صفت Text عنصر است :



نکته : توجه داریم که برخی از این توضیحات کلی بوده و صرفاً "مربوط به محیط C# نمی باشد.

در ASP.net ، Visual C# ، Visual J#

صفت مورد نظر . نام عنصر

در Visual C++

صفت مورد نظر -> نام عنصر

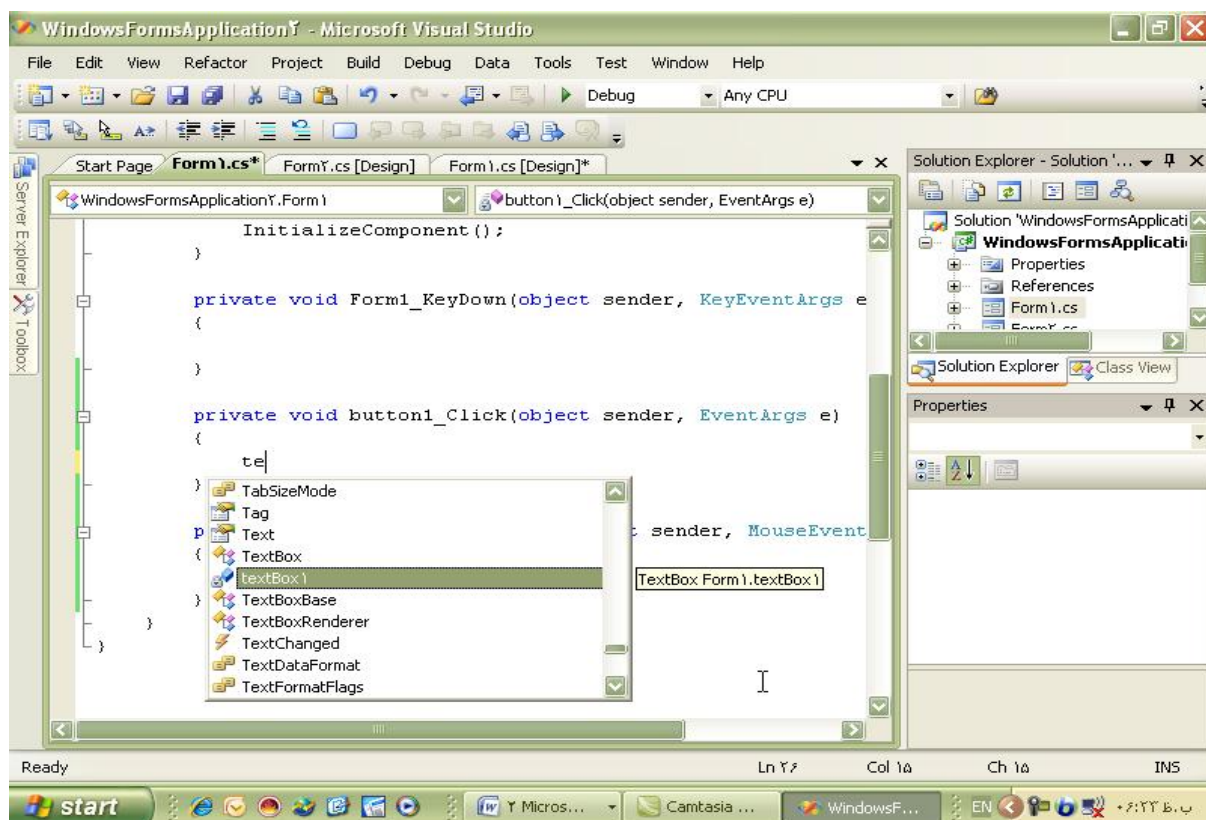
به صفات عنصر دسترسی پیدا می کنیم .

مثلاً" وقتی textbox ایجاد می کنیم ، نام آن عنصر را که مثلاً "textbox است خواهیم داشت ، صفتی

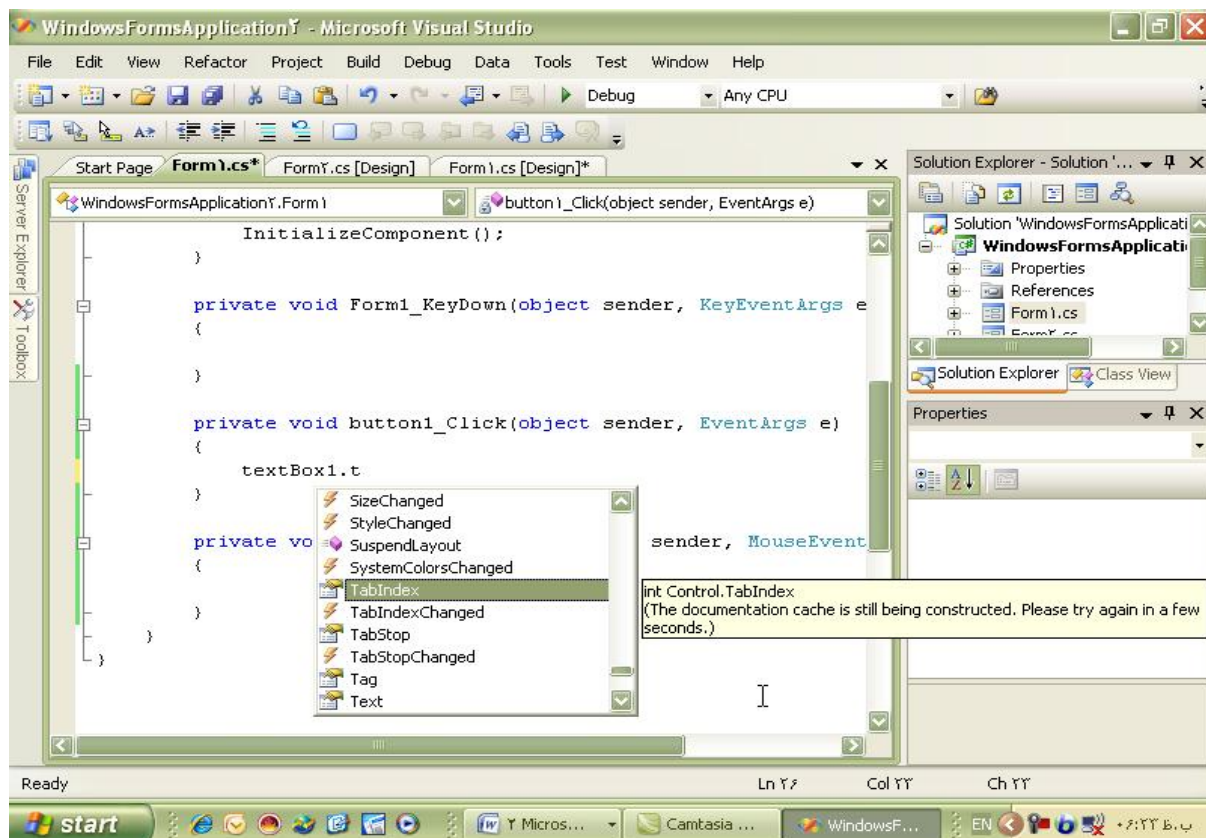
که توسط آن به محتویات این عنصر دسترسی پیدا می کنیم ، صفت Text آن است ، یعنی مثلاً

Text textbox را که داشته باشیم ، صفات آن عنصر را خواهیم داشت .

\*دوبار کلیک بر روی عنصر به محیط دستورات وارد می شویم و CTRL+ Space bar به دستورات ،

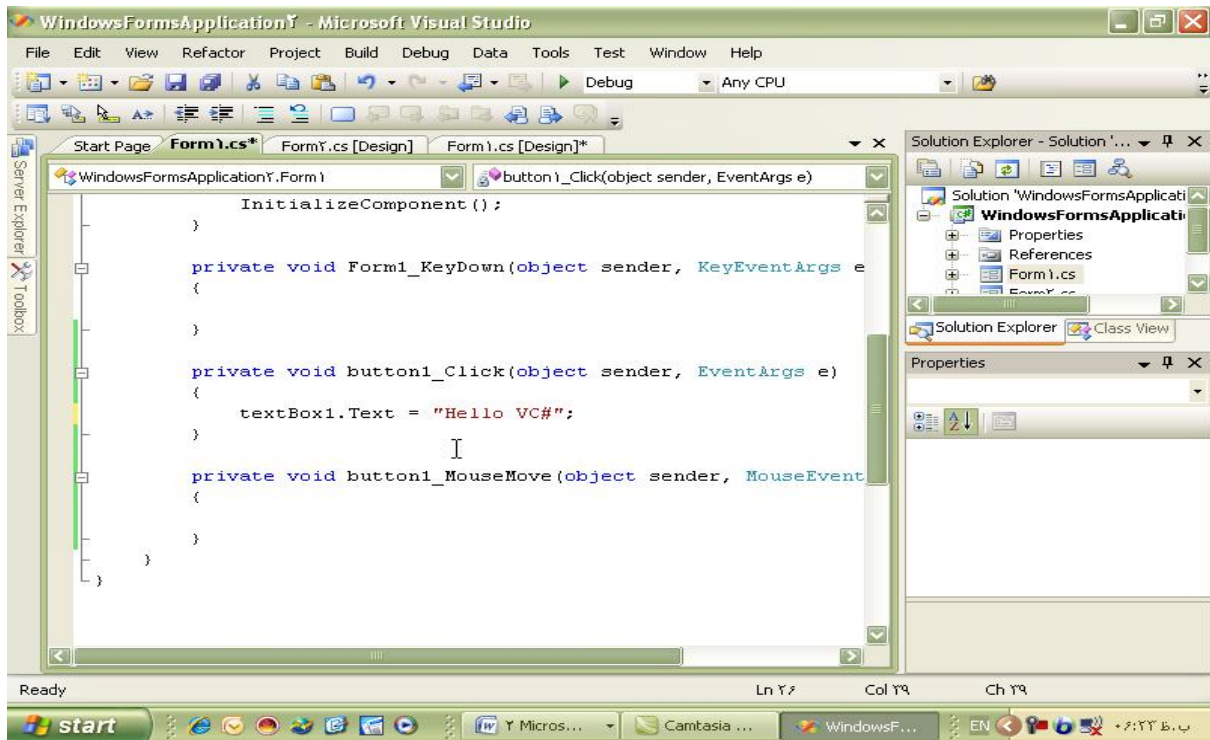


وبا استفاده از ". " به صفات عنصر دسترسی پیدا می کنیم .





مثلاً با اجرای دستور :

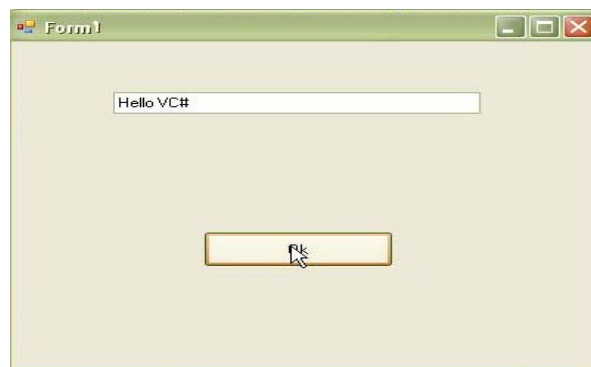


textBox).Text="Hello Visual C#";

متن دلخواه را که می خواهیم در textBox نمایش دهد را اعلام می کنیم .

برای کلیک کردن یک دکمه ترسیم می کنیم و با کلیک کردن بر روی دکمه مورد نظر ، متن دلخواه ، مثلاً

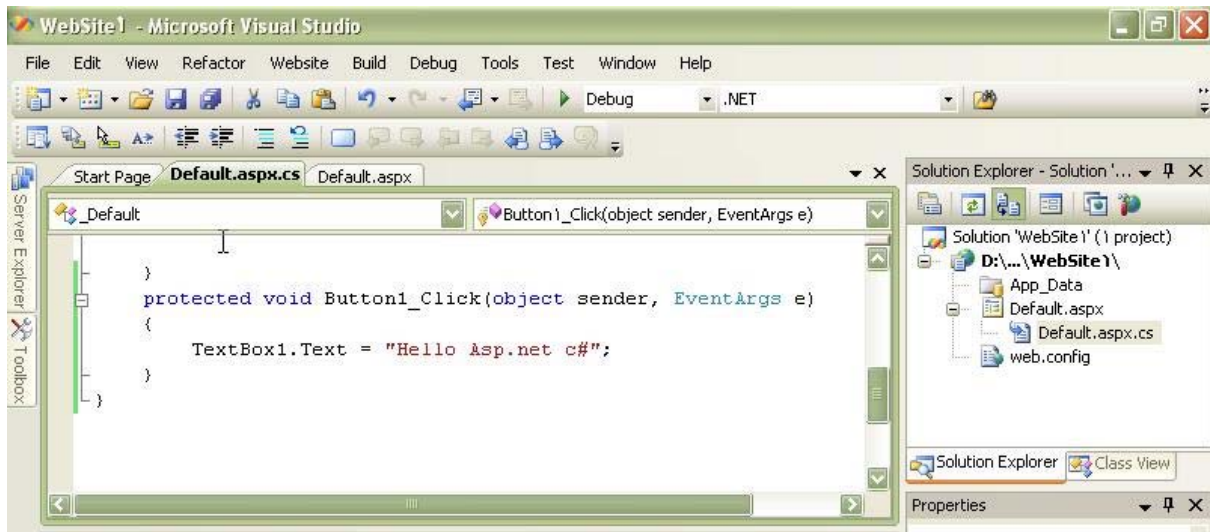
Hello Visual C# در درون Textbox نوشته می شود.



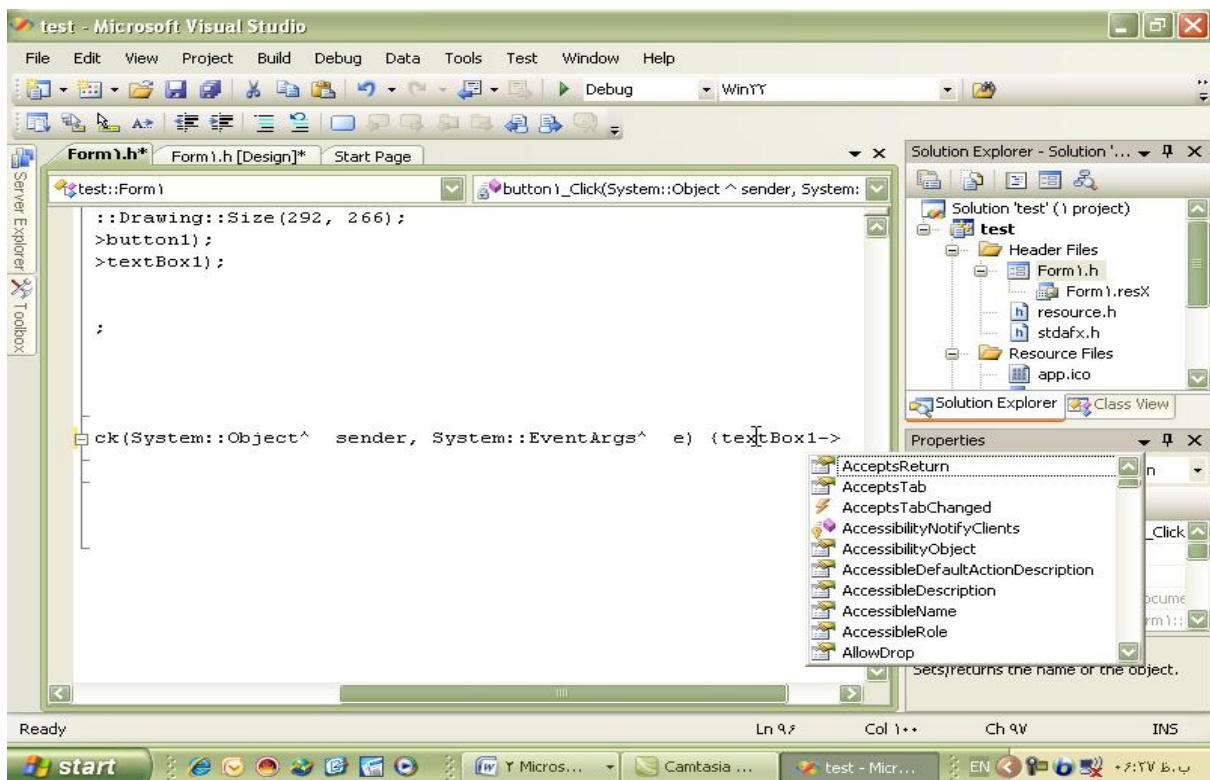
این کار در محیط Asp.net هم قابل اجرا می باشد.



در محیط Asp.net هم با استفاده از ". " به صفات عنصر دسترسی پیدا می کنیم .



ولی در Visual C++ اگر ما از نقطه استفاده کنیم ، به هیچ چیزی دسترسی نخواهیم داشت ، ولی با استفاده از -> کلیه صفات را مشاهده می کنیم ،

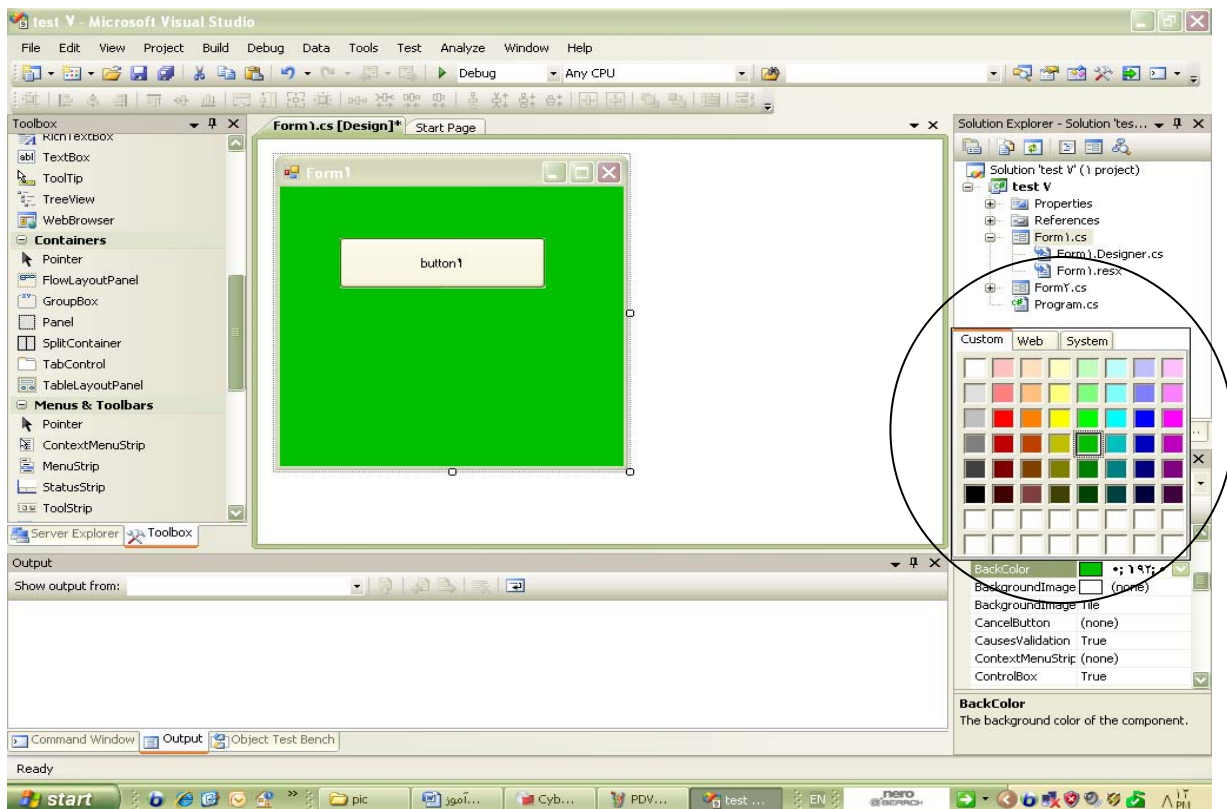


و با انتخاب صفت Text و نوشتن مابقی دستورات کار را مانند قبل دنبال می کنیم .

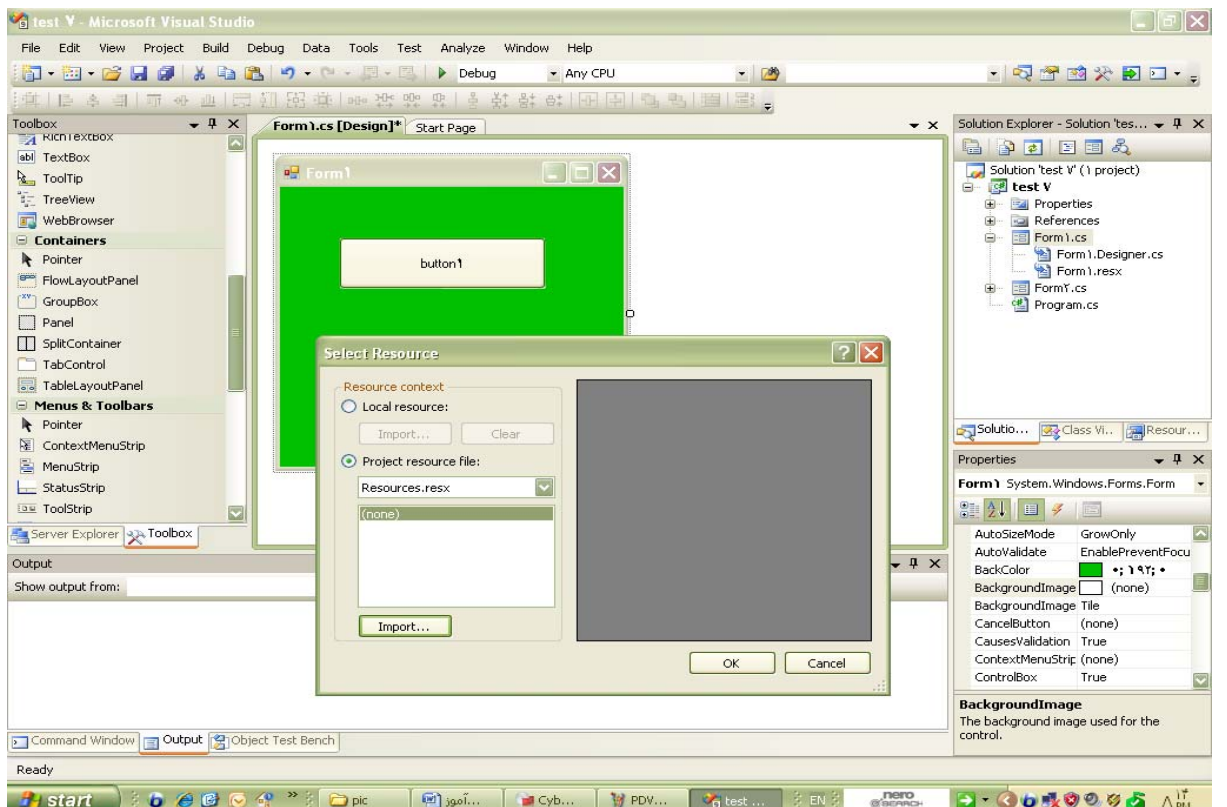
نکته : توجه داریم که ما کد ها را به زبان C# می نویسیم و خودش به ASP.Net هم Convert می کند .

پس صفت نام صفت بسیار مهمی است که توسط آن می توانیم به صفات دیگر عنصر با استفاده از ". " دسترسی داشته باشیم .

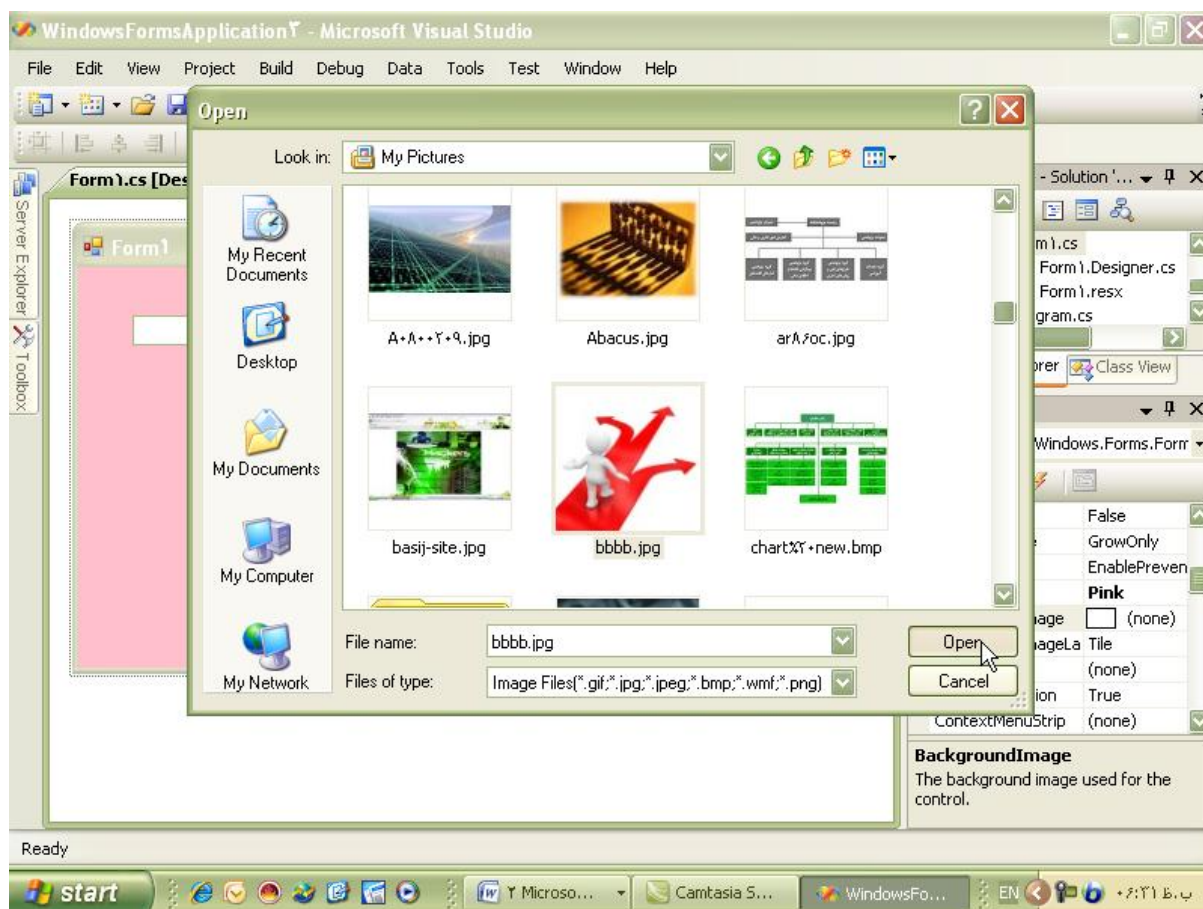
۳-۲-۳ - BackColor : با استفاده از این صفت رنگ زمینه را تغییر می دهیم .



۳-۲-۴ - BackgroundImage : توسط این صفت می توانیم عکس زمینه را انتخاب و Set کنیم .



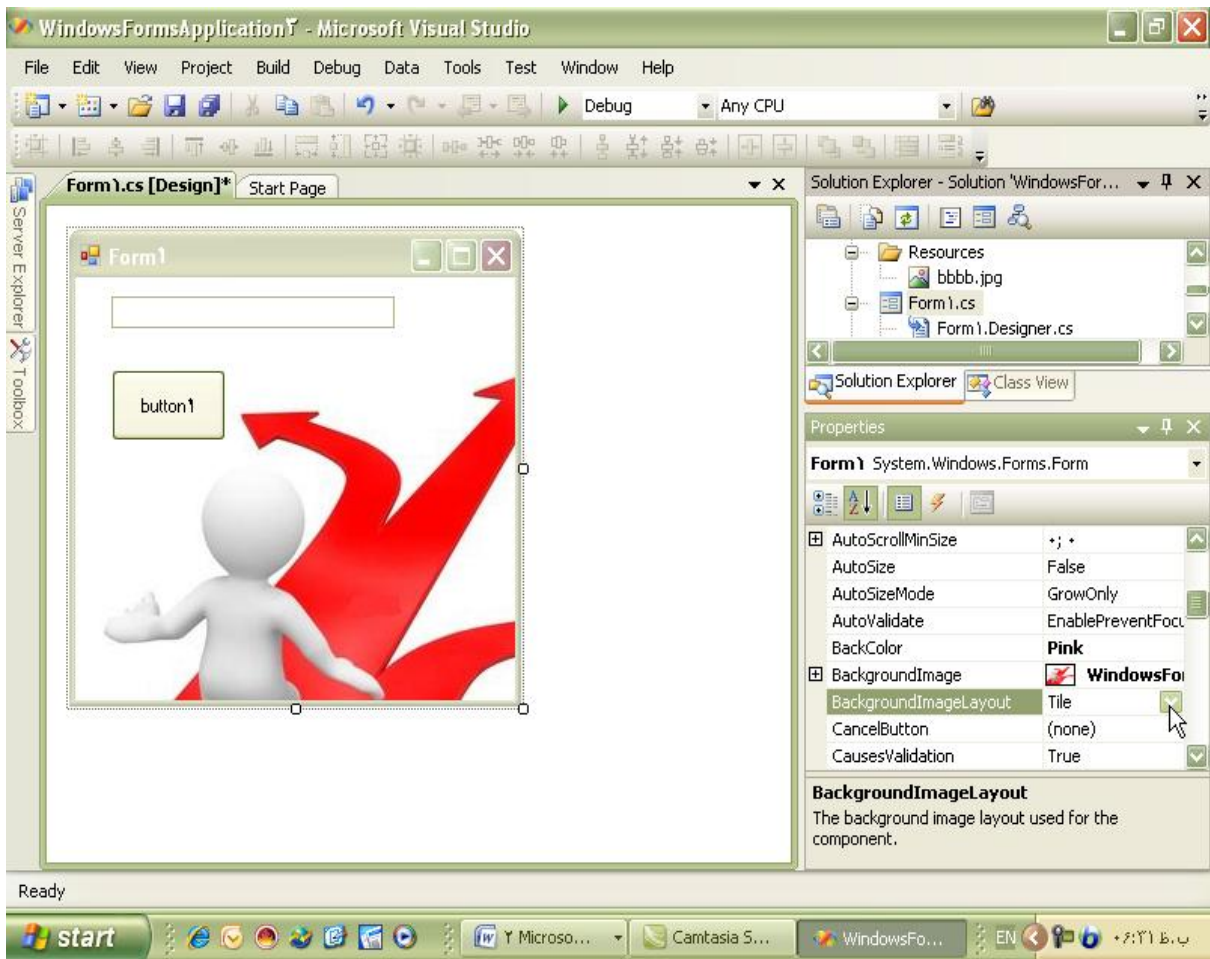
برای استفاده از این صفت دکمه مقابل **BackgroundImage** را می‌زنیم و در صفحه جدید دکمه **Import** را زده و از آدرس مورد نظر **Image** دلخواه را انتخاب و **Open** می‌کنیم



و پس از کلیک بر روی **Ok**،



تصویر مورد نظر در متن ما قرار می گیرد .



۵-۲-۳ - **BackgroundImageLayout** : با این صفت می توانیم چیدمان عکس زمینه را تعیین کنیم،

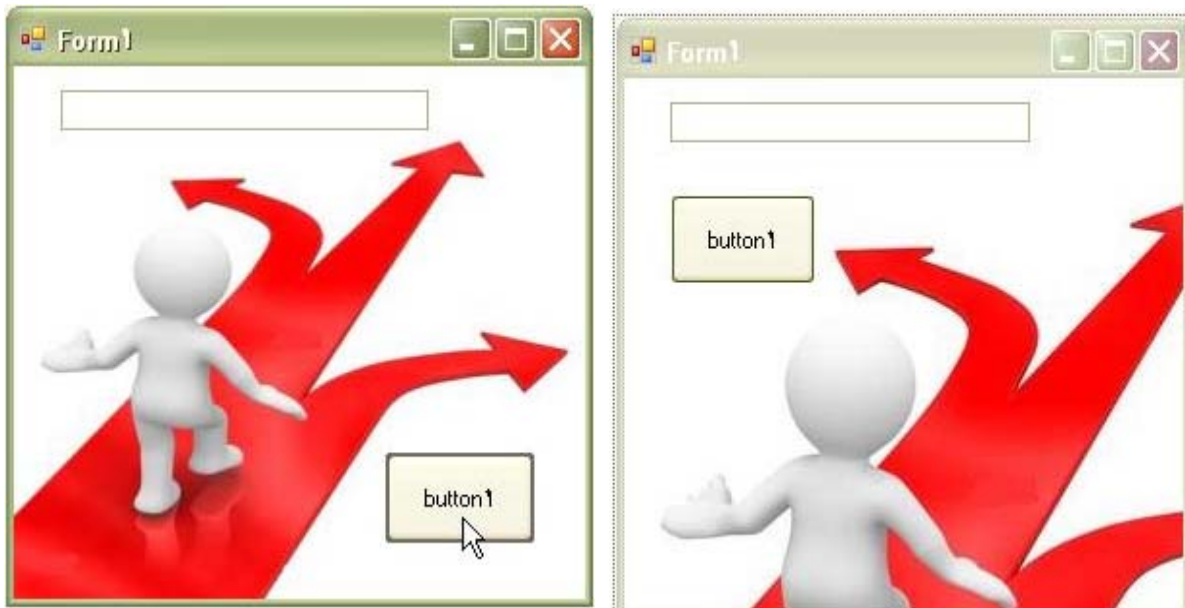


با باز کردن دکمه کشویی کنار آن به حالت‌های مختلفی دسترسی پیدا کرده و می توانیم حالت دلخواه را انتخاب کنیم .

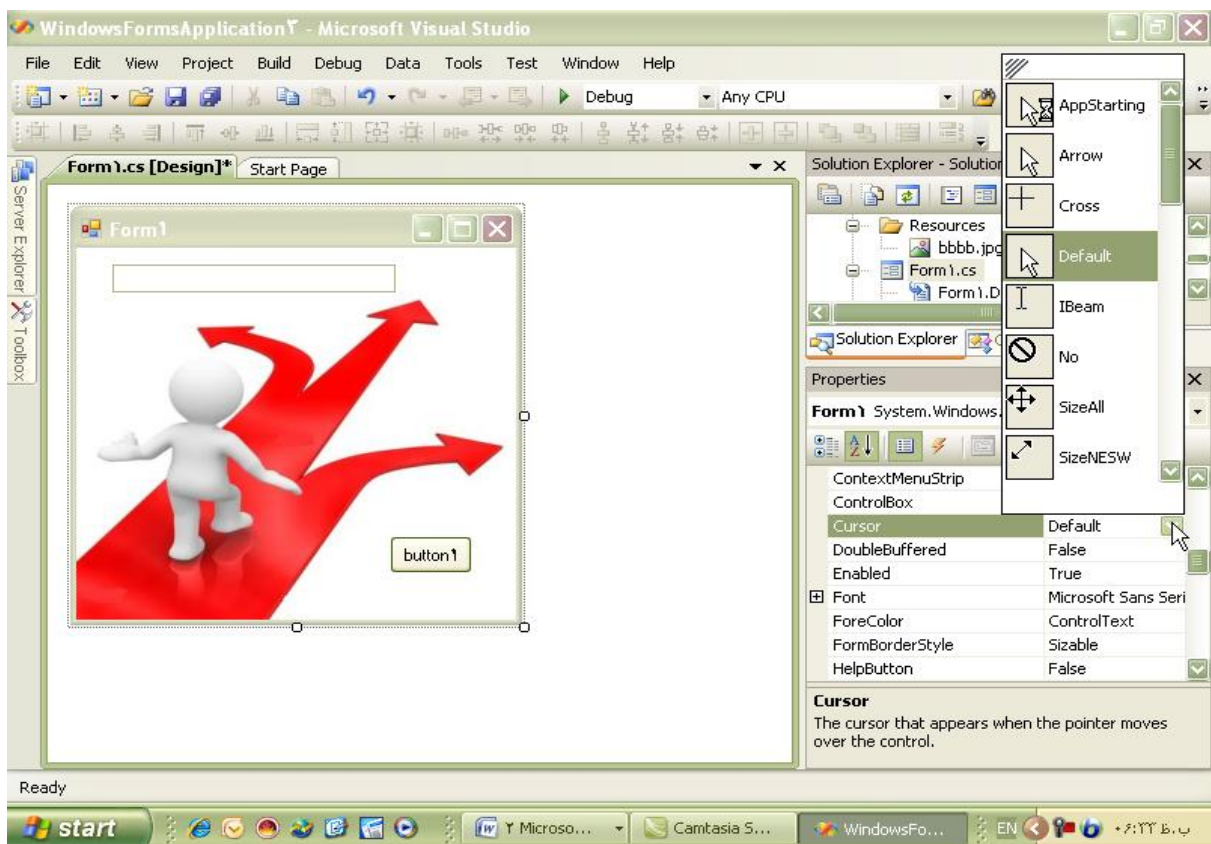
مثلاً" در حالت Stretch عکس زمینه خودش را به اندازه عنصر در می آورد.

Stretch

Tile



۶-۲-۳- Cursor : شکل مکان نمای ماوس را در زمان اجرا تعیین می کند ، مثلاً" زمانی که ماوس روی عنصر قرار می گیرد.



\*برای هریک از عناصر می توانیم حالت ماوس را جداگانه و به دلخواه تعیین کنیم .

۷-۲-۳ - Enable : این صفت دارای دو حالت True و False است :



} Enable در حالت True یعنی عنصر فعال و قابل دسترس است .  
} Enable در حالت False یعنی عنصر غیر فعال و غیر قابل دسترس است .

مثلاً" اگر صفت Enable یک Button را در حالت False قرار داده باشیم ، اگر برای آن Button دستوراتی را نوشته باشیم ، با قرار گرفتن بر روی آن دیگر آن دستورات اجرا نمی شوند و دکمه ما به رنگ خاکستری و غیر قابل دسترس خواهد بود ، چراکه Enable آن غیر فعال است .

این موضوع در مورد منو ها کاربرد بیشتری دارد، مثلاً" با استفاده از امکان `Menus in toolbar` یک منو می سازیم و با انتخاب گزینه `Menu Strip` یک منو باز می کنیم ، نام منو را انتخاب می کنیم و زیر مجموعه های آن را هم تعریف می کنیم توجه داریم که :

\* با تایپ یک خط تیره در هر یک از `Box` های زیر مجموعه های منو می توانیم یک خط کلی داشته باشیم .

\* برای هر یک از زیر مجموعه ها هم می توانیم یک زیر مجموعه دیگر تعریف کنیم .

\* در هر یک از `Box` های منو می توانیم متن را به زبان دلخواه داشته باشیم .

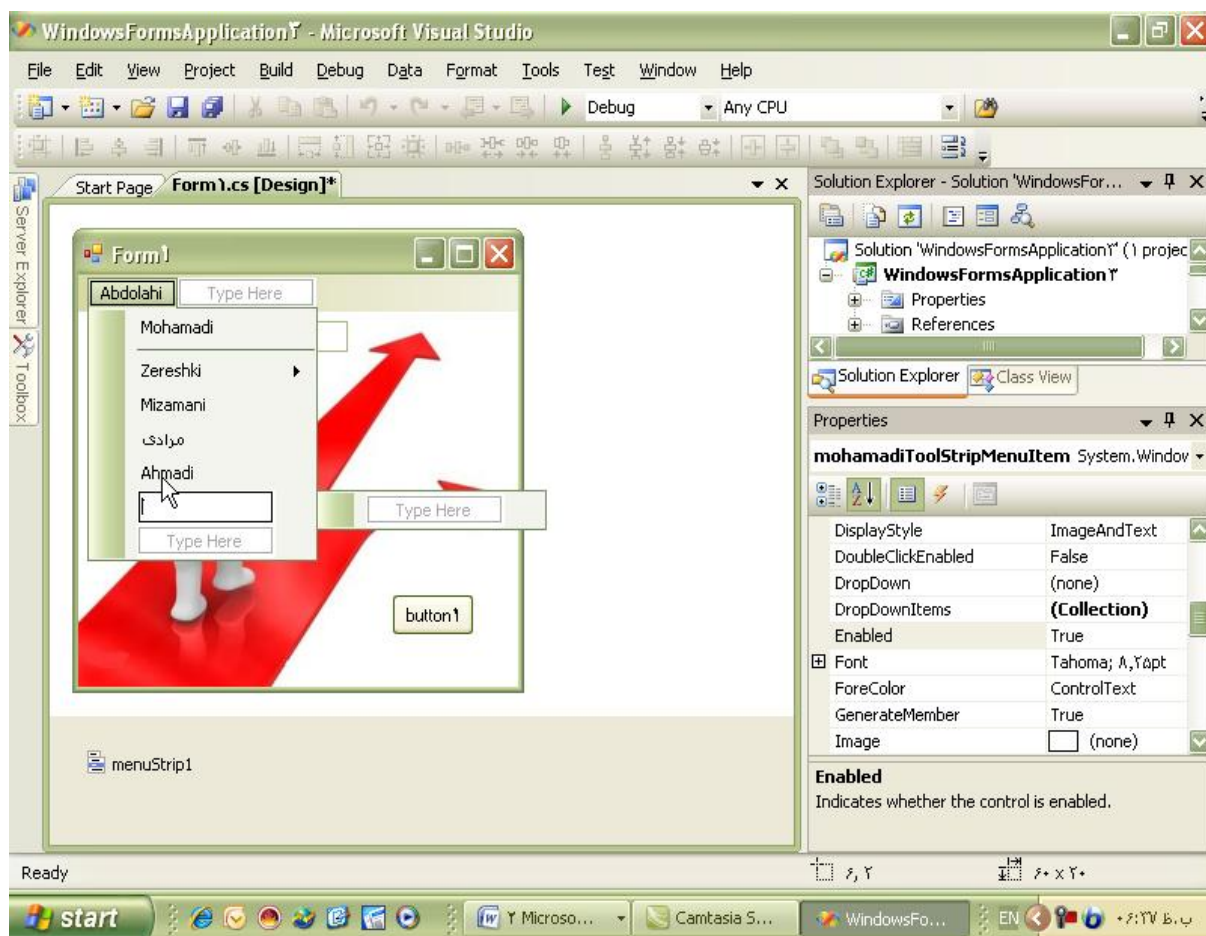
\* برای هر یک از فهرست ها و زیر فهرست ها می توانیم دستورات دلخواه را داشته باشیم .

برای این کار روی هر یک از آنها دو بار کلیک می کنیم ، با ورود به محیط دستورات و استفاده از کلید های `Ctrl+ Space bar` دستورات را مشاهده و از میان آنها `Text Box` را انتخاب می کنیم و دستور مورد نظر را به این ترتیب می نویسیم :

```
textBox1.Text="Hello Visual C++";
```

با اجرای این دستور مشاهده می کنیم که هر زمان که روی فهرست مورد نظر کلیک کنیم ، در درون `TextBox` متن مشخص شده نمایش داده می شود .

برای مثال یک منو با زیر فهرست های آن به این ترتیب درست می کنیم :



برای این که هر وقت روی زیر فهرست Ahmadi کلیک کردیم در TextBox متن Mola Ahmadi نمایش داده شود ، روی BOX مربوط به Ahmadi دوبار کلیک می کنیم و در دستورات مربوط به آن دستور زیر را می نویسیم :

```
private void ahmadiToolStripMenuItem_Click(object sender, EventArgs e)
{
    textBox1.Text = "Molla Ahmadi";
}
```

textBox1.Text=" Mola Ahmadi ";

به این ترتیب با کلیک بر روی Ahmadi در TextBox متن Mola Ahmadi را مشاهده می کنیم .



کار دیگری که می توانیم انجام دهیم این است که می توانیم با کلیک بر روی هر یک از زیر فهرست ها یا فهرست ها ، وضعیت Enable یکی دیگر از آنها را به یکی از حالت های False و یا True ( به دلخواه ) تغییر دهیم ، برای این کار ، به طور مثال وضعیت Enable زیر فهرست Ahmadi را در ابتدا False قرار می دهیم ، سپس مثلاً" روی زیر فهرست Mohamadi قرار می گیریم (باید نام دقیق زیر فهرست Ahmadi را داشته باشیم ) سپس با دو بار کلیک در میان دستورات ، دستور زیر را می نویسیم .

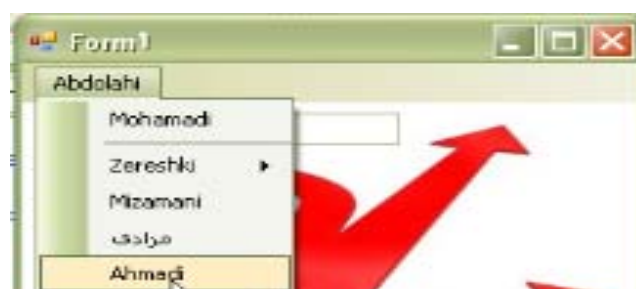
```
private void mohamadiToolStripMenuItem1_Click(object sender, EventArgs e)
{
    ahmadiToolStripMenuItem.Enabled = true;
}
```

Ahmadi Tool strip Menu Item. Enable=True;

با اجرای این دستور در ابتدا Enable زیر فهرست Ahmadi غیر فعال است و با کلیک بر روی آن هیچ اتفاقی نمی افتد ،



اما بلافاصله پس از کلیک بر روی Mohamadi زیر فهرست Ahmadi فعال شده و در این زمان با کلیک بر روی Ahmadi دستوری که برای آن نوشته ایم اجرا می شود ،



و به این ترتیب در TextBox متن Mola Ahmadi نمایش پیدا می کند.

\*ما در محیط Windows جاهای زیادی داریم که عنصر های متفاوتی در ابتدا فعال یا غیر فعال هستند و با اجرای دستور یا دستوراتی و یا وقوع اتفاقی حالت های آنها تغییر می کند .

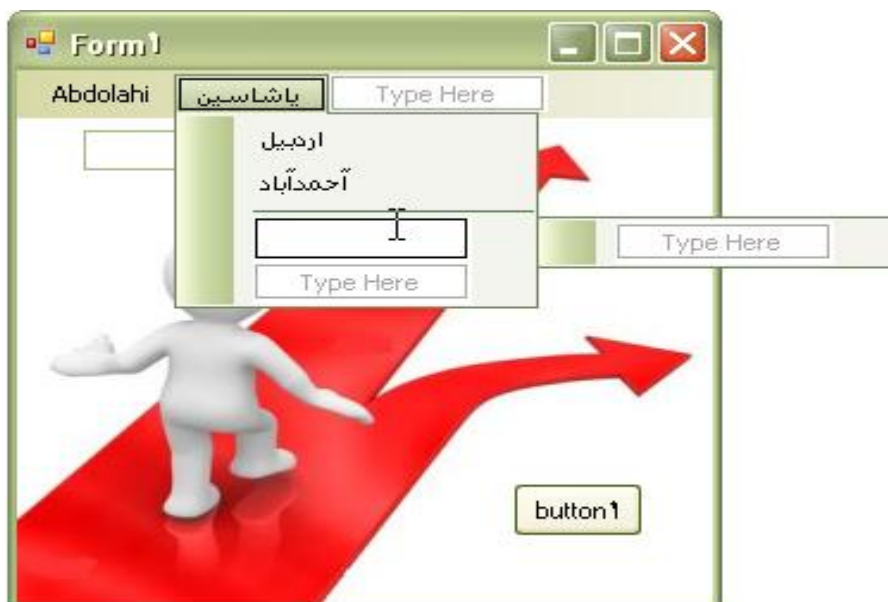


مثلا" عنصر های Cut ، Copy ، Past و . . . در ابتدا غیر فعال هستند و پس از انتخاب شدن متن یا چیزی آنها فعال می شوند و آمادگی یا امکان انجام دستور یا دستوراتی را خواهند داشت .



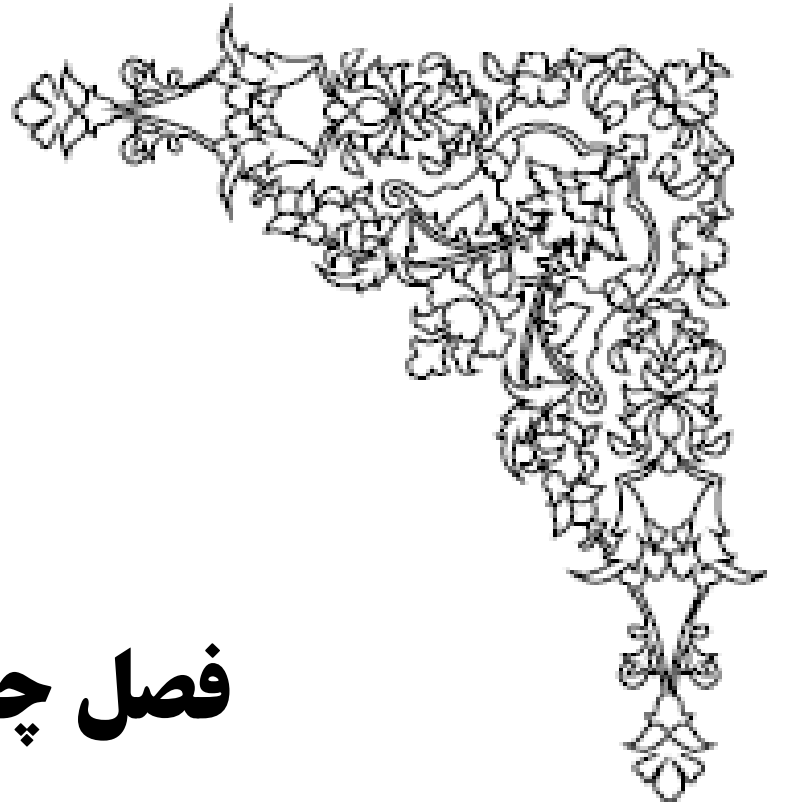
فعال و غیر فعال بودن Menu ها امکانی است که توسط برنامه نویس تعیین می گردد و بستگی به وقوع و یا انجام چیز یا چیزهایی دارد.

\*برای ادامه دادن به ساختن منو ، کافیهست که روی نوار Menu کلیک کره و به تعداد دلخواه منو بسازیم ، به محض شروع به تایپ نام فهرست ، امکان ایجاد زیر فهرست ها هم ایجاد می شود و به همین ترتیب تا اتمام ایجاد تعداد دلخواه می توانیم کار را ادامه دهیم .



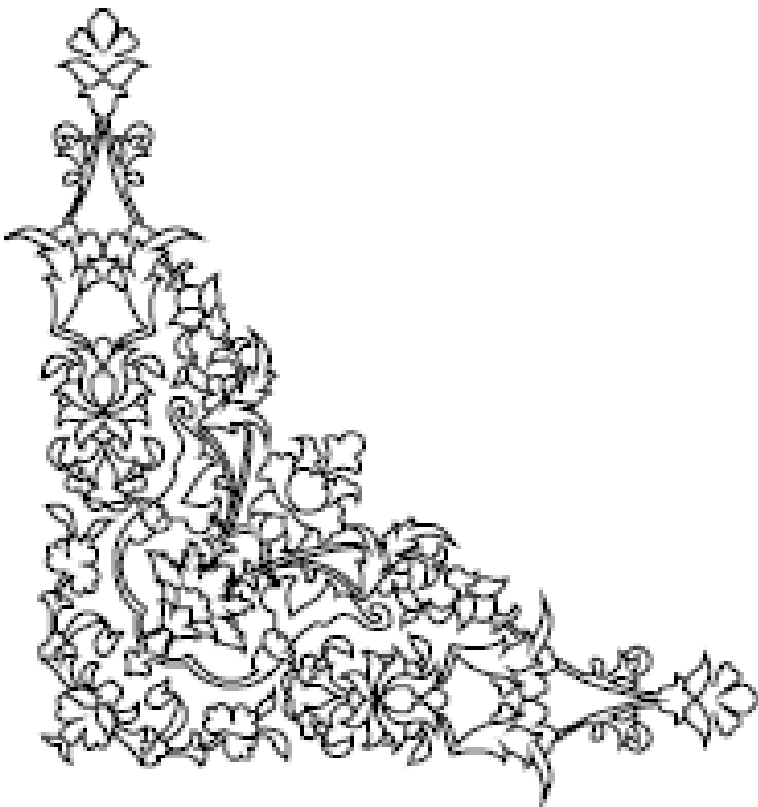
\*برای ایجاد خط جدا کننده زیر فهرست ها کافیهست که از یک خط استفاده و با رفتن به زیر فهرست بعدی بصورت اتوماتیک خط کامل ترسیم می گردد و به این ترتیب زیر فهرست ها هم از هم جدا می گردند.





## فصل چہارم

# بررسی Menu



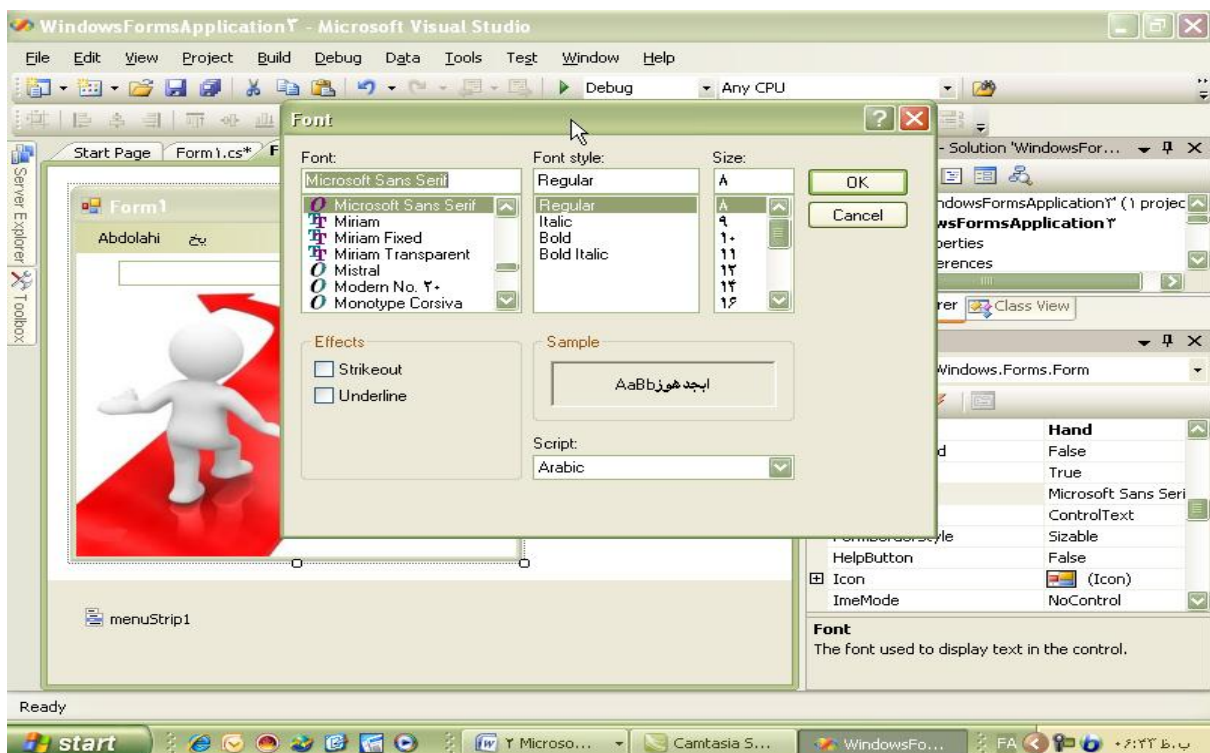
## ۴-۱- ویژگیهای Menu :

در ویژگیهای Menu خصوصیات بسیاری را خواهیم داشت ، مثلاً:

امکان داشتن Shortcut ، عکس ، تیک ، عکس زمینه ، امکان رنگ آمیزی و بسیاری از قابلیت های دیگر که نمونه های بسیاری از آنها را در محیط های مختلف منوهای Windows می توانیم مشاهده کنیم .

۴-۱-۱- Font : صفت بعدی ، صفت Font است که همان ویژگیهای Font در محیط های دیگر را مثل

نوع ، سایز ، سبک و... را دارا می باشد و قلم نوشته ها و متن ها را می توان با استفاده از آن تعیین نمود.



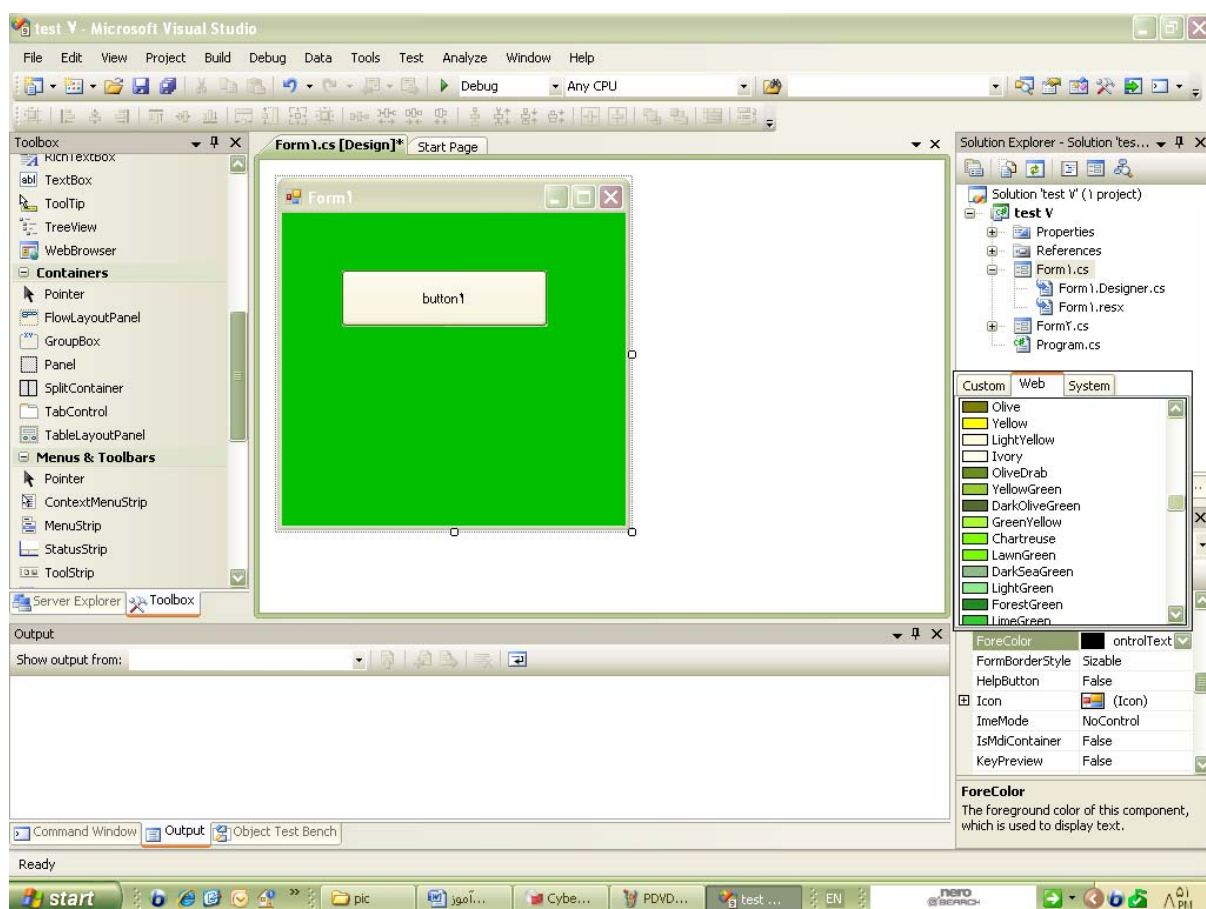
توجه داریم که اگر در یک فرم Font را تعیین کنیم ، تمام عناصر موجود در آن فرم از تنظیمات انجام شده ارث می برند .

مثلاً وقتی سایز یک Font را در یک فرم تغییر می دهیم ، تمام عناصر موجود در آن فرم هم تغییر سایز می دهند.

در صورت تمایل می توانیم Font را در هر یک از عناصر جداگانه و به دلخواه تغییر دهیم و این تغییرات مختص همان عنصر خاص باشد.

\*این موضوع همان بحث ارث بری در شیء گرای می باشد.

۲-۱-۴ - **ForeColor** : با استفاده از این صفت می توانیم رنگ نوشته ها را تعیین کنیم .

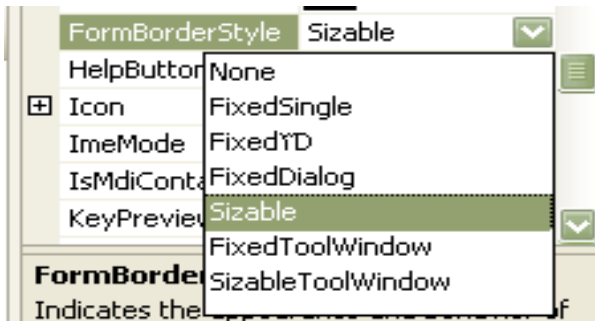


C# از این نظر که معماری لایه ای را حمایت می کند و چون هم Application و هم Web دارد ، بسیار مهم و قدرتمند است و با وجود اینکه محیط php یک محیط Open Source می باشد ، ولی چون معماری لایه ای را حمایت نمی کند ، این C# است که می تواند بسیار مفید باشد و به جرأت می توان گفت که آینده حرفه ای ، خصوصا" از نظر Security مختص به C# می باشد و با استفاده از امکانات بسیار C# می توان از ویژگی Open Source بودن php هم صرفنظر نمود.

۳-۱-۴ - **Image** : صفت دیگر Image می باشد ، این صفت امکان گذاشتن یک عکس را روی یک عنصر برای ما فراهم می کند.

توجه داریم که این امکان با امکان Background Image متفاوت است و عکس زمینه را ایجاد نمی کند ، بلکه یک عکس را می توانیم روی عنصر دلخواه بگذاریم .  
مثلا" می توانیم روی دکمه هایمان عکس بگذاریم .

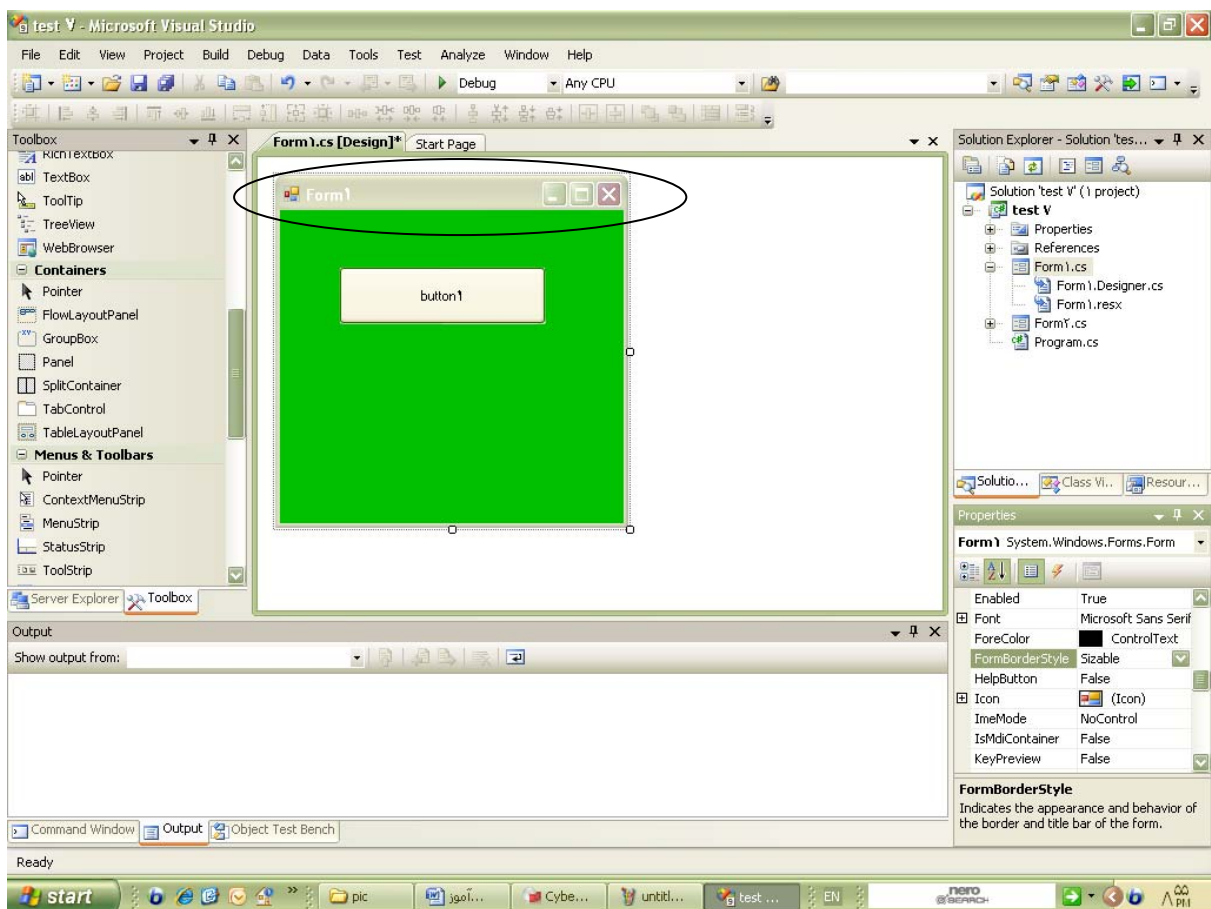
۴-۱-۴ - **FormBorderStyle**: با استفاده از این صفت می توانیم سبک فرم حاشیه فرم را تعیین کنیم .



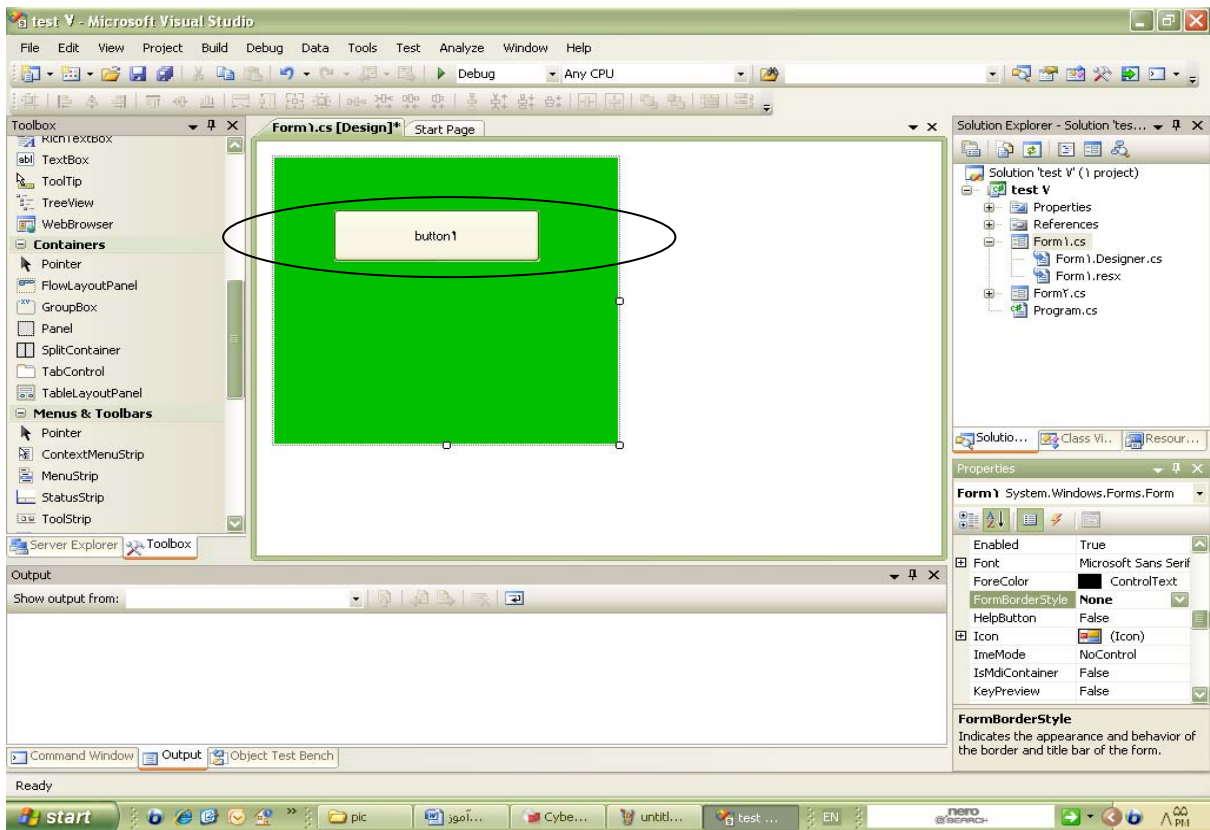
با باز کردن دکمه کشویی مقابل آن می توانیم یکی از انواع سبک های دلخواه را برای فرم خودمان انتخاب کنیم .

این صفت امکان این را فراهم می کند که در قسمت Title ، ابزار کناری فرم مثل Close ، Maximize & Minimize وجود داشته باشد یا خیر.

مثلاً در حالت Sizable فرم ما از تمام امکانات برخوردار است ،



ولی اگر حالت None را انتخاب کنیم ، در فرم هیچ یک از امکانات را مشاهده نخواهیم کرد.

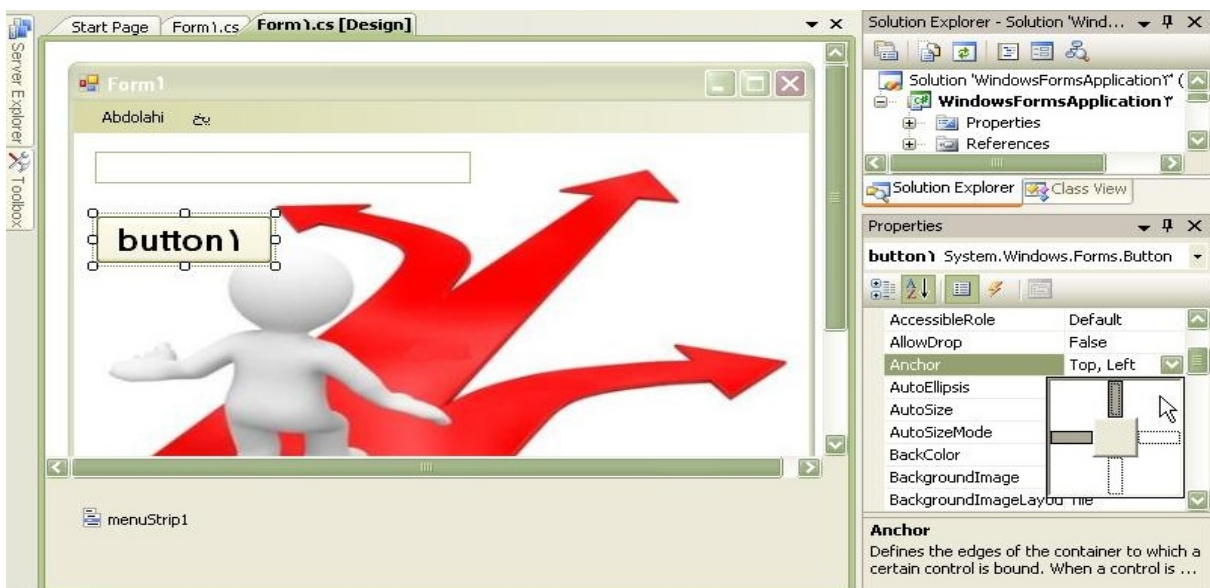


با انتخاب حالت‌های دیگر بنا به سلیقه و دلخواه ، می توانیم Title بالای فرم را هم تعیین کنیم .

\*پیش فرض Form Border Style در حالت Sizeable قرار دارد.

۵-۴-۱-Anchor : این صفت امکان این را فراهم می کند که در حالت Resize کردن فرم ، عناصر

موجود در فرم هم از تغییرات ارث ببرند.





۶-۱-۴-Icon : امکان تغییر Icon کنار فرم را برای ما فراهم می سازد.

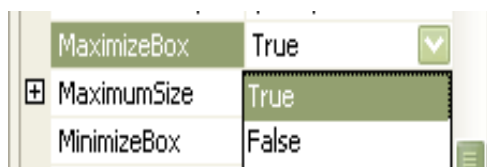


۷-۱-۴-Language : امکان تغییر و انتخاب زبان را در اختیار ما می گذارد.

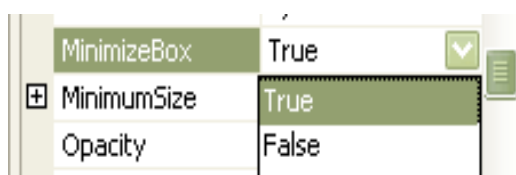


۸-۱-۴-Maximize Box : در دو حالت True و False امکان فعال بودن و یا غیر فعال بودن

Maximize Box را برای ما فراهم می سازد.



۹-۱-۴-Minimize Box : هم دقیقاً" به همین صورت برای حالت Minimize می باشد.



۱۰-۱-۴-Right To Left : در ۳ حالت Yes ، No و Inherit می باشد که اگر :

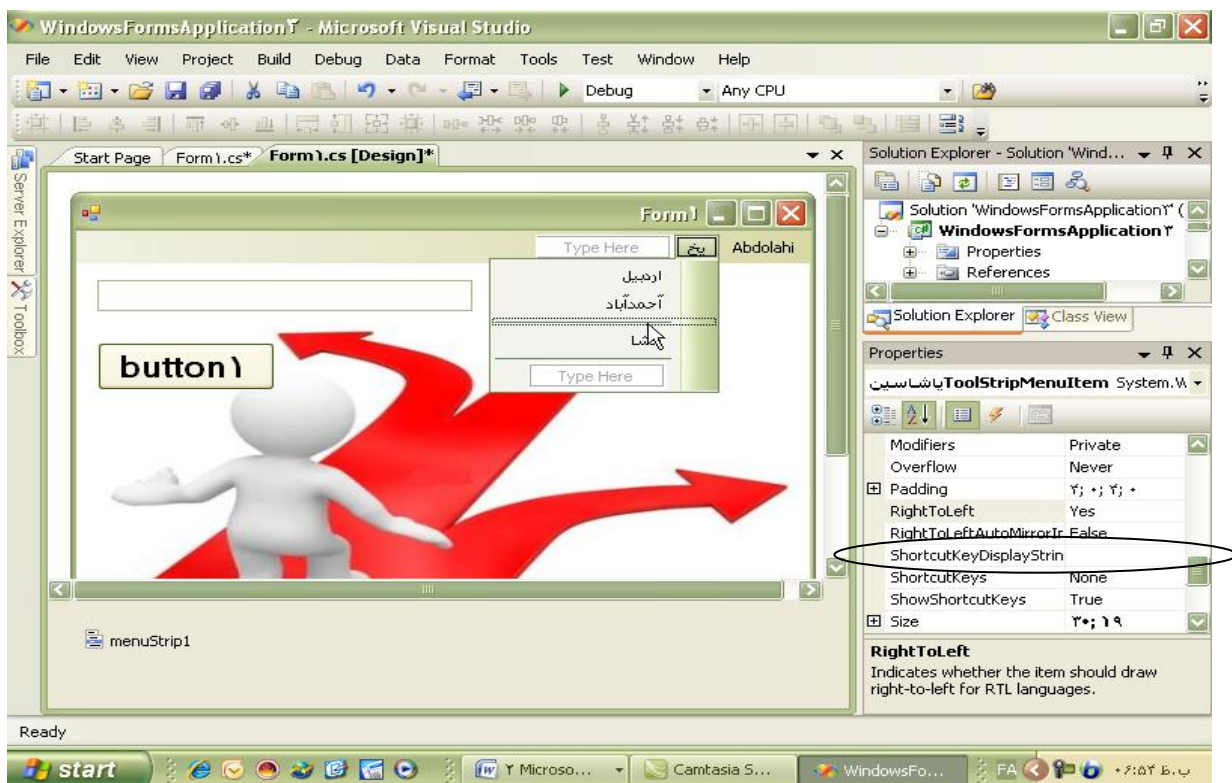
- Right To Left در حالت Yes باشد ، کلیه عناصر از راست به چپ چیده خواهند شد.
- Right To Left در حالت No باشد ، کلیه عناصر از چپ به راست چیده خواهند شد.
- Right To Left در حالت Inherit باشد ، باعث می شود که ارث بری داشته باشیم .

این حالت برای فرم ها معنا پیدا نمی کند ، بلکه برای عناصر و یا آیتم هایی که روی فرم قرار می گیرند

معنا پیدا می کند.

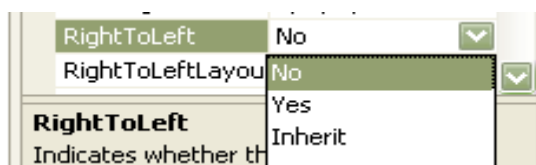
۱-۱۰-۴-RightToLeft : اگر Yes را انتخاب کنیم تمام عناصر فرم از راست به چپ

چیده می شود،



حالت Inherit یعنی اینکه ارث ببرد، این حالت در مورد فرم معنا پیدا نمی کند ، بلکه فقط در مورد عناصر و آیتم های روی فرم معنا پیدا می کند.

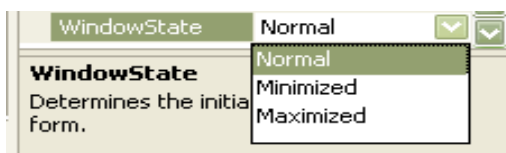
۱۱-۴-۱-RightToLeft Layout : این صفت در مورد چیدمان Layout ها می باشد.



۱۲-۴-۱-Text : این صفت بسیار مهم است و عنوان روی عنصر می باشد ، نوع آن رشته (String) است .

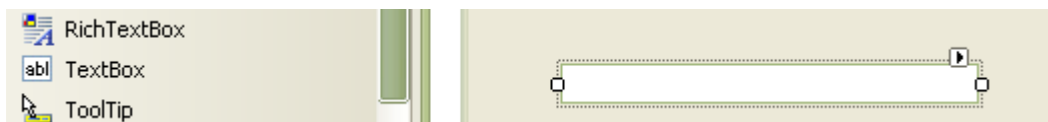
۱۳-۴-۱-WindowState : این صفت دارای ۳ حالت Normal ، Minimized و Maximized

است .



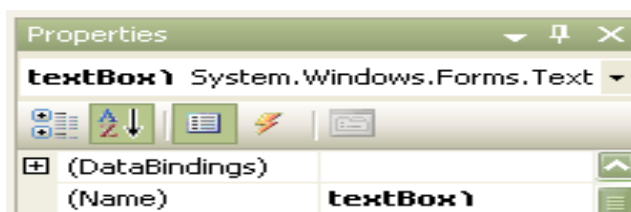
این صفت حالت فرم را در زمان اجرا تعیین می کند.

اینکه تمام صفحه یا Maximized، سایز استاندارد یا Normal و یا سایز کوچک یا Minimized باشد .  
**۴-۱-۱۴ - TextBox** : این عنصر جهت دریافت اطلاعات از ورودی و نمایش اطلاعات در خروجی و همچنین اتصال به یک فیلد از جدول بانک اطلاعاتی به کار می رود.

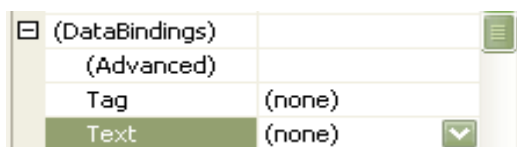


مهم ترین صفات آن عبارتند از :

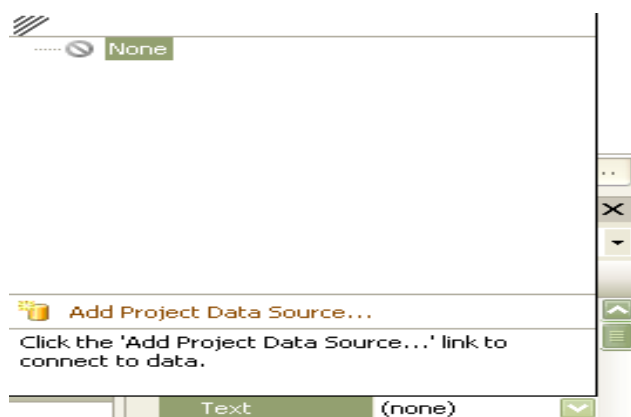
**۴-۱-۱۵ - Name** : که نام TextBox در آن تعیین می گردد و مهم ترین صفت عنصر است و همانطوریکه گفته شد ، امکان دسترسی به صفات را در اختیار ما می گذارد.



**۴-۱-۱۶ - DataBindings** : جهت اتصال به یک فیلد از جدول بانک اطلاعاتی به کار می رود.

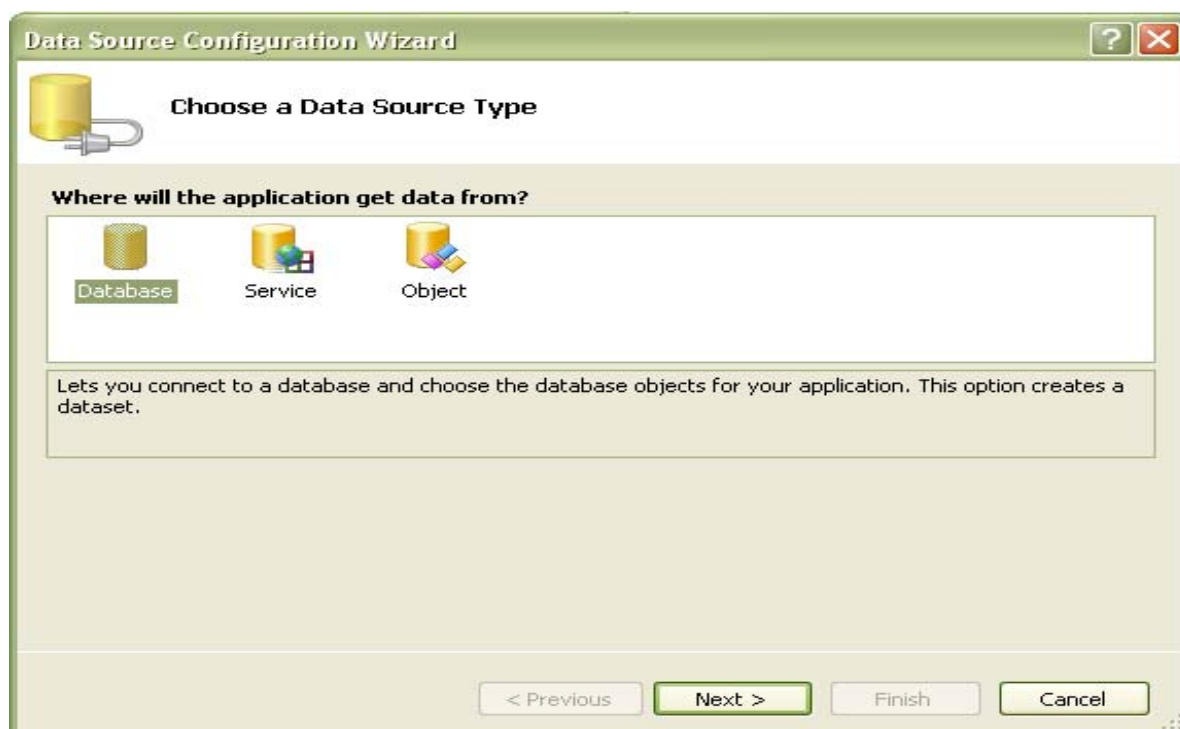


با باز کردن کلید کشویی مقابل Text در زیر مجموعه DataBindings پنجره جدیدی باز می شود ،

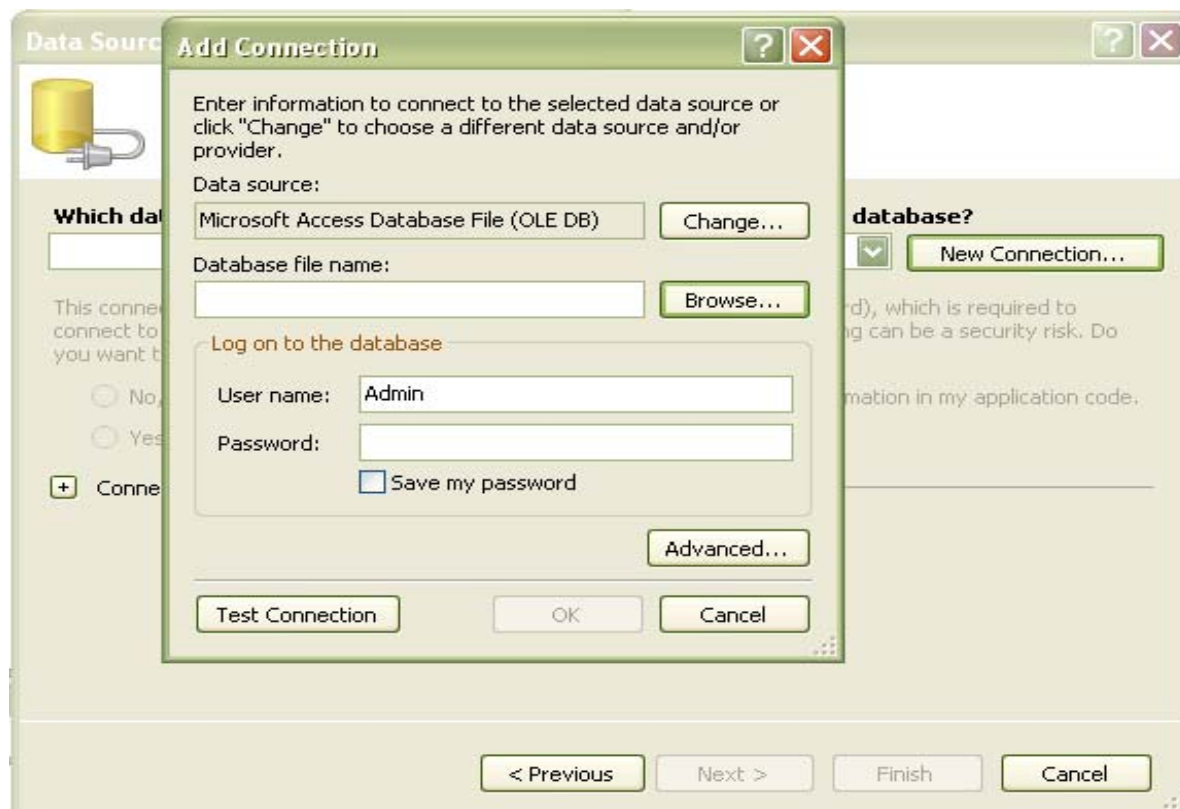


که گزینه ای به نام Add Project Data Source ..... دارد ،

و با کلیک بر روی آن ، پنجره جدید دیگری باز می شود که بانکهای اطلاعاتی را در اختیار ما می گذارد ،



مثلاً" با انتخاب یک پایگاه داده و کلیک بر روی Next مرحله به مرحله در یک Wizard پیش می رویم،



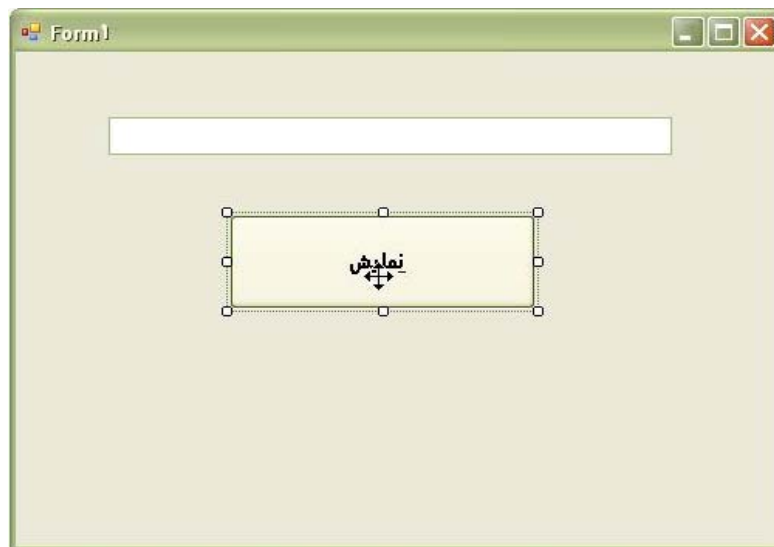
و در نهایت می توانیم به یک فیلد در یک بانک اطلاعاتی متصل می شویم .

**Text** : محتویات درون عنصر می باشد و نوع آن رشته (String) است .

مثال : می خواهیم برنامه ای بنویسیم که با کلیک بر روی یک دکمه پیغام " Hello Visual C# " را نمایش دهد:

برای این کار پس از ایجاد یک فرم که پیش فرض می باشد یک عنصر TextBox ترسیم می کنیم ، سپس یک Button رسم می کنیم که صفت Text آن را که همان عنوان روی دکمه می باشد ، عنوان مورد نظر یا مناسب قرار می دهیم ،

\* نمایش یا تایپ علامت & قبل از هریک از حروف ، متن نوشته شده آن حرف را عملیاتی می کند و با تایپ آن حرف کاری که با کلیک بر روی دکمه قرار است انجام شود ، انجام می شود.



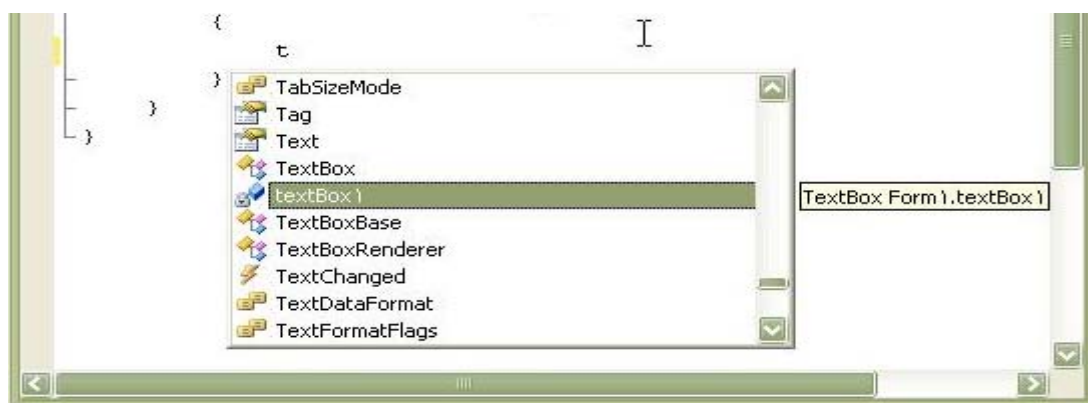
با ۲ بار کلیک بر روی دکمه مورد نظر و ورود به محیط دستورات کد زیر را می نویسیم :

```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = "Hello Visual c#";
}
}
```

با در نظر گرفتن نام عنصر که (textBox) است و محتویات آن که Text است می توانیم به محتویات درون textBox دسترسی پیدا کنیم .

\*نکته : بهتر است همیشه از **Ctrl+ Spacebar** استفاده کنیم ، چراکه **Help** را در اختیار ما می گذارد و از بروز غلط های املایی پیشگیری می کند.

با **Ctrl+ Spacebar** تمام دستورات و آیتم هایی که ما لازم داریم در اختیار ما قرار می گیرد ، با تایپ اولین حرف از آنچه که می خواهیم ، روی دستور یا متن مورد نظرمان قرار می گیریم .



با نوشتن دستور زیر :

```
textBox1.Text="Hello Visual C#";
```

در `textBox` ، نوشته `Hello Visual C#` را مشاهده می کنیم .



\*توجه داریم که اگر تایپ دستورات ما درست باشد ، با زدن " . " صفات در اختیار ما قرار می گیرند ، در غیر اینصورت به دستورات دسترسی نخواهیم داشت .

با اجرای برنامه ( **Debug** ) مشاهده می کنیم که یک دکمه در فرم داریم که با کلیک بر روی آن متن مشخص شده در `textBox` نمایش داده می شود.

## ۲-۴- تمرین نوشتن برنامه ماشین حساب Windows:

ابتدا یک فرم ایجاد می کنیم ،

سپس یک عنصر textBox (جعبه متن) ترسیم می کنیم ،

به تعداد ارقام صفحه کلید ماشین حساب یا به تعداد دکمه های دلخواه Button یا دکمه فرمان روی فرم

ترسیم می کنیم ،

ترتیب چیدمان ، شکل ، سایز و سایر صفات مربوط به Button ها کاملا" دلخواه بوده و در روند برنامه

هیچ تغییری ایجاد نخواهد کرد.

صفت Text هریک از دکمه ها را برابر آنچه که می خواهیم نمایش دکمه ها باشد قرار می دهیم ، مثلا"

برای اعداد برابر با اعداد مورد نظر و برای Function ها برابر با عملگر های مورد نظر و به همین ترتیب الی

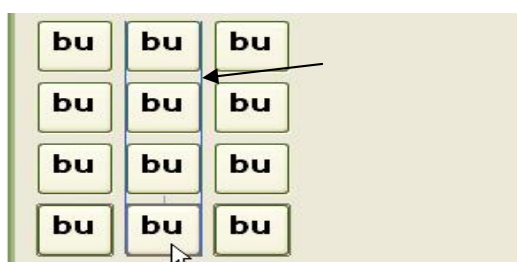
آخر .....



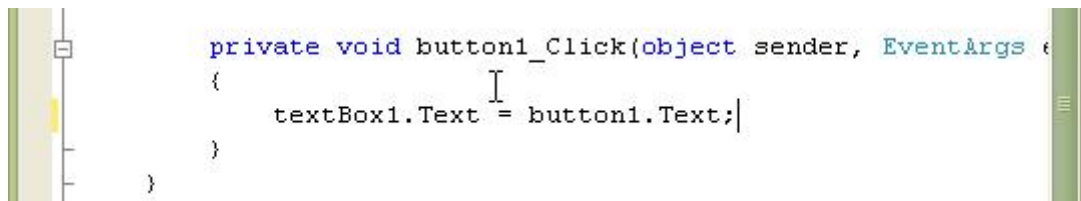
با ۲ بار کلیک بر روی هر یک از دکمه ها در وضعیت دستورات قرار گرفته و کدهای لازم را می نویسیم .

برای زیبا تر شدن و راحت تر شدن مراحل ترسیم می توانیم از امکان Copy و Past استفاده کنیم و از

خطوط Guide Line هم برای اینکه ترسیم بهتر و دقیق تری داشته باشیم می توانیم کمک بگیریم .



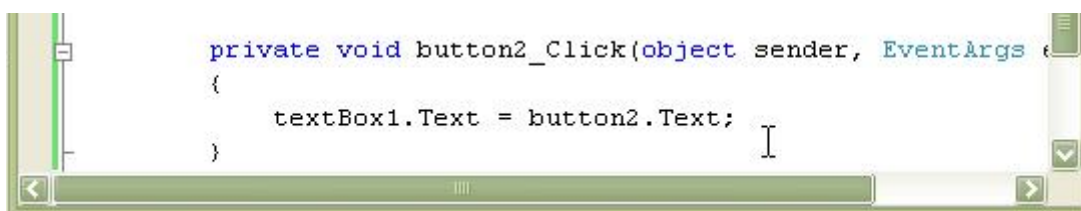
برای اینکه وقتی که روی هر یک از دکمه های اعداد کلیک می کنیم ، عنوان آن دکمه ها در `textBox` قرار بگیرد ، کفایت که روی هر یک از دکمه های اعداد ، دستور زیر را بنویسیم :



```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = button1.Text;
}
```

`textBox).Text=button).Text;`

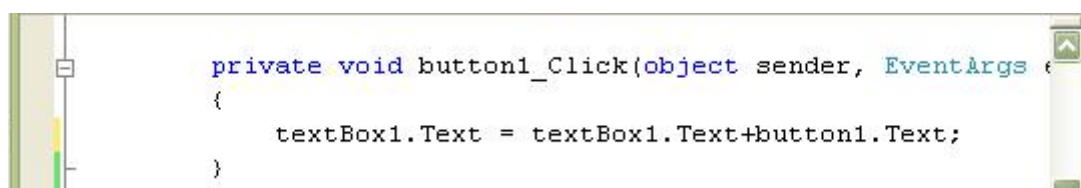
برای دکمه های بعدی ، بنا به اینکه نام آنها چه باشد ، دستور بالا تغییر می کند ، مثلاً :



```
private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = button2.Text;
}
```

`textBox).Text=button۲.Text;`

ولی با اجرای این برنامه مشاهده می کنیم که وقتی اولین دکمه را می زنیم ، مثلاً عدد ۷ در `textBox` قرار می گیرد و وقتی که دکمه بعدی را می زنیم ، مثلاً عدد ۸ ، عدد ۷ از بین می رود و عدد ۸ جایگزین عدد ۷ می گردد ، با زدن دوباره عدد ۸ علیرغم آن که می بایست در `textBox` عدد ۸۸ نمایش داده شود ، مشاهده می کنیم که این اتفاق نمی افتد و تنها عدد ۸ به تنهایی نمایش داده می شود ، برای رفع این اشکال کفایت که دستوراتمان را به این ترتیب اصلاح کنیم :



```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text+button1.Text;
}
```

`textBox).Text= textBox).Text +button).Text;`

یعنی مقدار جدید `textBox` را برابر می گیریم با مقدار قدیمی `textBox` به اضافه ( جمع رشته ای ) بمقدار `button` .

به این ترتیب هر بار که دکمه ای را می زنیم ، مقدار آن دکمه در `textBox` قرار می گیرد و با زدن دکمه بعدی مقدار آن هم در کنار مقدار قبلی قرار می گیرد و جمع رشته ای می شود.



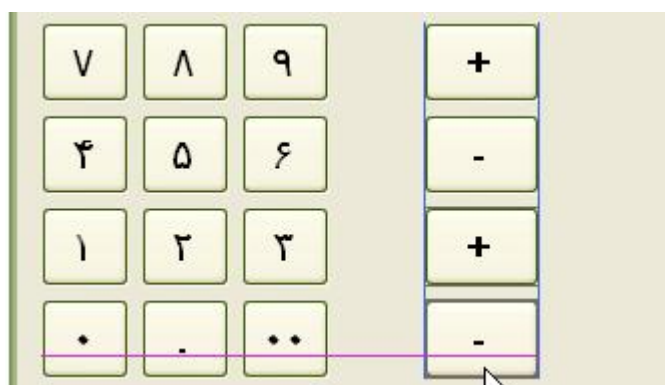
این کار را برای هر یک از دکمه‌ها انجام می‌دهیم، با توجه به اینکه برای هر یک از دکمه‌ها عنوان آنها تغییر می‌کند و کد زیر را برای همه دکمه‌ها می‌نویسیم:

`textBox1.Text = textBox1.Text + button1.Text;`

برای هر یک از دکمه‌ها تغییر می‌کند که در خط مربوط به Button عنوان آن را می‌توانیم مشاهده کنیم.

```
private void button3_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + button3.Text;
}
```

پس از تعریف کلیه دکمه‌ها عملگرها را تعریف می‌کنیم.



مثلاً دکمه جمع: برای این کار ابتدا یک Button ترسیم می‌کنیم و مقدار Text آن Button را برابر با

عملگر + قرار می‌دهیم، همانند قبل با دوبار کلیک در محیط دستورات قرار می‌گیریم.

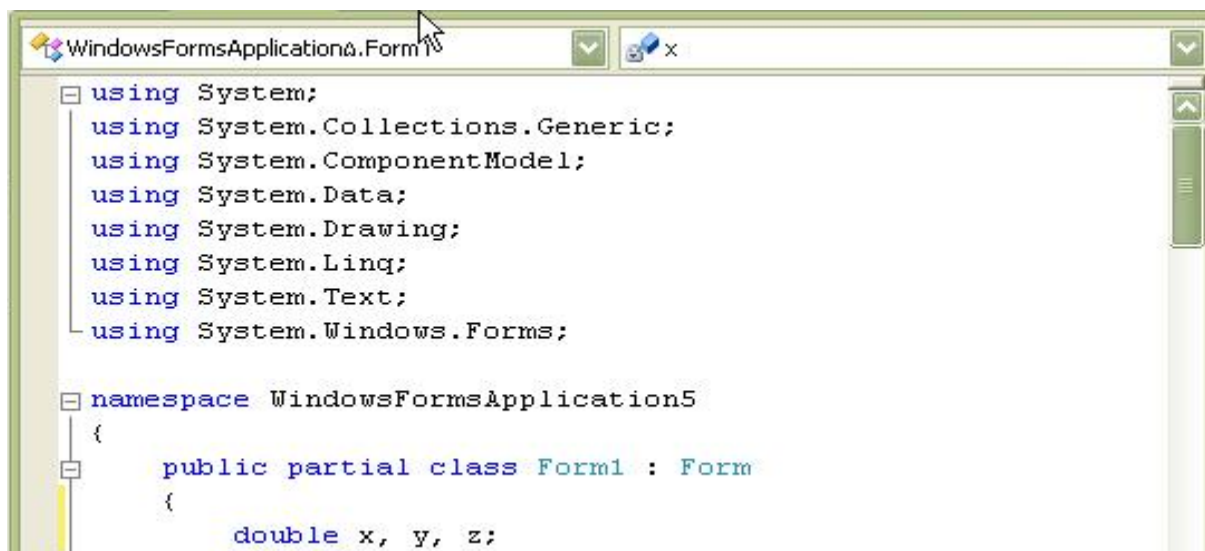
\*توجه داریم که وقتی یک عدد را وارد می‌کنیم، با زدن هر یک از عملگرها می‌بایست عدد مورد نظر در

یک متغیر در حافظه قرار بگیرد و textBox خالی شود و منتظر باشد عدد دوم را وارد کنیم.



اینجاست که بحث متغیرهای سراسری (global) و محلی (local) و Private مطرح می شود، اگر ما متغیر را در دکمه جمع تعریف کنیم، بعد در حالت = متغیر شناخته شده نیست، چون آن متغیر فقط برای دکمه جمع محلی و شناخته شده است.

بنابراین می بایست متغیرهایمان را سراسری تعریف کنیم، برای این کار کافیهست که در قسمت Public فرم (مانند زبان C در قسمت تعریف متغیرها) شروع به تعریف متغیرها می کنیم:



```

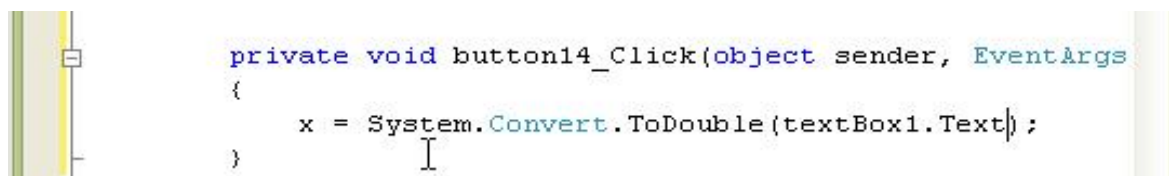
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        double x, y, z;
    }
}

```

### Double x,y,z

\*به این دلیل متغیرهایمان را به صورت Double تعریف می کنیم که محدوده آن بزرگ باشد. متغیر X را برای مقدار اول، متغیر Y را برای مقدار دوم و متغیر Z را برای جواب تعریف می کنیم. می بینیم که محتویات textBox رشته است و X محتویاتش Double می باشد، بنابراین اگر بخواهیم محتویات \ textBox را در متغیر X قرار دهیم، می بایست عمل تبدیل را انجام دهیم، بنابراین کد زیر را می نویسیم:

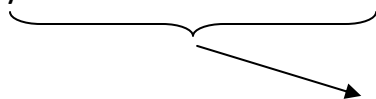


```

private void button14_Click(object sender, EventArgs e)
{
    x = System.Convert.ToDouble(textBox1.Text);
}

```

x=System.Convert.ToDouble(textBox \.Text);



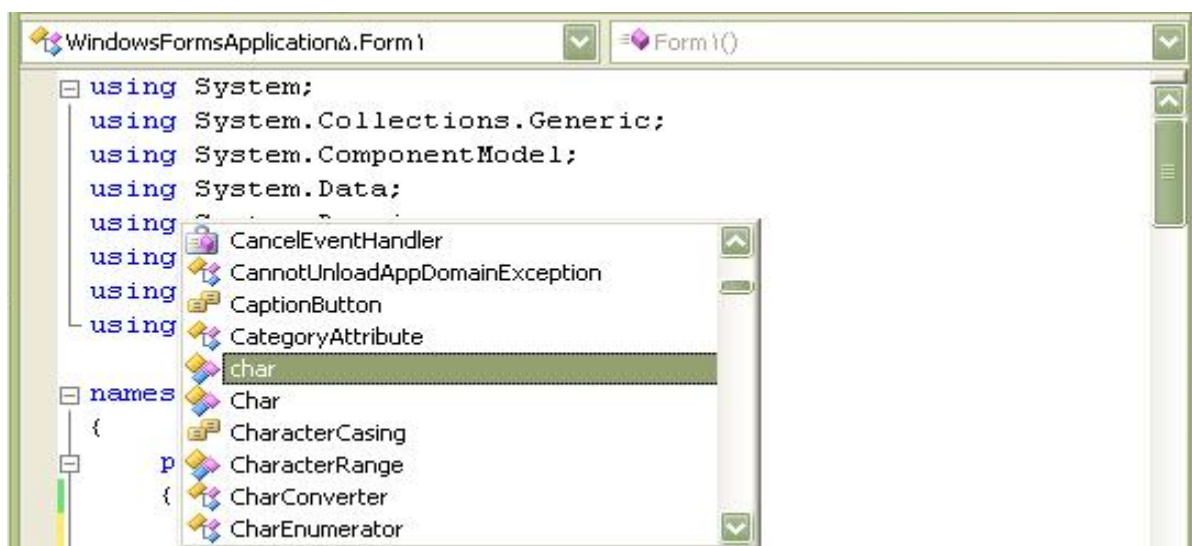
دستور System.Convert.ToDouble حالت String را به Double تبدیل می کند.

با این دستور می گوئیم که محتویات textBox را تبدیل به Double نماید و بعد در X قرار دهد ، چراکه textBox محتویاتش String است و بدون Convert نمی توان آن را در X قرار داد.  
 حال باید Textbox را خالی کنیم ، برای این کار دستور زیر را می نویسیم :

```
{
    x = System.Convert.ToDouble(textBox1.Text);
    textBox1.Text = "";
}
```

textBox).Text= "";

پس از این کار باید عملگر + را تعریف کنیم ، در همان قسمت Public که متغیرها را تعریف می کنیم ، یک متغیر کاراکتر هم تعریف می کنیم ،



مثلاً "متغیر op به این ترتیب که :

```
{
    double x, y, z;
    char op;
```

char op;

حال در قسمت کلید جمع ، دستوراتمان را به این ترتیب کامل می کنیم :

```
private void button14_Click(object sender, EventArgs)
{
    x = System.Convert.ToDouble(textBox1.Text);
    textBox1.Text = "";
    op = '+';
}
```

x=System.Convert.ToDouble(textBox\).Text);

textBox\).Text= "";

Op='+';

به این ترتیب عدد اول را که می زنیم آن عدد در textBox قرار می گیرد ، سپس با زدن دکمه عملگر عدد اول در حافظه قرار می گیرد، پس از زدن عدد دوم آن عدد در textBox قرار می گیرد ، با زدن علامت مساوی عدد دوم در حافظه قرار می گیرد و پس از چک کردن عملگر نتیجه می بایست در textBox قرار گیرد ، فقط می بایست توجه داشته باشیم که چون عدد اول در X و عدد دوم در Y و نتیجه آنها در Z قرار می گیرد، Z یک عدد می باشد و چون آنچه در textBox قرار می گیرد می بایست رشته باشد ، پس حتما" می بایست آن عدد تبدیل به رشته شود، بنا براین در دستورات دکمه مساوی خواهیم داشت که :

```
{
    y = double.Parse(textBox1.Text);
}
struct System.Double
Represents a double-precision floating-point number.
```

Y=double.Parse(textBox\).Text);

که با :

```
private void button14_Click(object sender, EventArgs)
{
    x = System.Convert.ToDouble(textBox1.Text);
}
```

System.Convert.ToDouble(textBox\).Text);

به یک معناست و دقیقا" یک کار را انجام می دهند و String را تبدیل به Double می کنند.

حال می بایست اپراتورمان را کنترل کنیم :

```
{
    y = double.Parse(textBox1.Text);
    if (op == '+') z = x + y;
    if (op == '-') z = x - y;
    if (op == '*') z = x * y;
    if (op == '/') z = x / y;
}
```

If (op == '+') z = x + y;

If (op == '-') z = x - y;

If (op == '\*') z = x \* y;

If (op == '/') z = x / y;

⋮

حال می بایست آنچه درون Z قرار گرفته است را تبدیل به String کرده و در textBox قرار دهیم ، برای

این کار دستور لازم را می نویسیم :

```
{
    y = double.Parse(textBox1.Text);
    if (op == '+') z = x + y;
    if (op == '-') z = x - y;
    if (op == '*') z = x * y;
    if (op == '/') z = x / y;
    textBox1.Text = z.ToString();
}
```

textBox1.Text = z.ToString();

Double را تبدیل به String می کند

\*برای اینکه اعداد بصورت رشته در textbox قرار گیرند می بایست Double را تبدیل به String کنیم .

حال یک سری تعاریف کلی را مرور می کنیم و خلاصه ای از عملکردهای انجام شده را به این ترتیب خواهیم داشت :

نحوه تعریف متغیرهای سراسری :

در پنجره کد نویسی در قسمت form \: form Public practical class با نوشتن کدهای :

Double x,y,z

Char op;

متغیرها و اپراتورهای مورد نظر را تعریف می کنیم .

برای کلیه عملگرهای دیگر هم به همین ترتیب عمل می کنیم .

حال می خواهیم ماشین حسابمان را مهندسی کنیم ، مثلاً " می خواهیم دکمه فاکتوریل (x!) ، Sin ، Cos ، Tang ، \* \log ، sqrt و... را به آن اضافه کنیم .

برای فاکتوریل متغیر X عدد ورودی ما می باشد ، می بایست یک متغیر کمکی تعریف کنیم ، مثلاً " F و آن را برابر با ۱ قرار می دهیم ، یک حلقه تکرار می سازیم و یک متغیر i برای حلقه تعریف می کنیم که هر بار یک واحد به آن اضافه شود تا به X برسد.

سپس جواب را که در F است در textBox قرار می دهیم .

برای اجرای این مراحل در کلید مربوط به فاکتوریل دستورات زیر را خواهیم داشت :

```
private void button23_Click(object sender, EventArgs e)
{
    double f = 1;
    x = double.Parse(textBox1.Text);
    for (int i = 1; i <= x; i++) f = f * i;
    textBox1.Text = f.ToString();
}
```

double f= 1;

x=double.Parse(textBox \.Text);

for(int i = 1; i<=x;i++) f=f\*i;

textBox \.Text=f.ToString();

نتیجه این دستورات فاکتوریل را محاسبه می کند.

برای محاسبه لگاریتم در پایه ۱۰:

عدد دلخواهتان را در متغیر X قرار می دهیم .

با استفاده از کلاس `math` که تمام توابع ریاضی را در اختیار ما قرار می دهد لگاریتم در پایه ۱۰ را انتخاب می کنیم و آن را در Z قرار می دهیم و نتیجه تبدیل شده به `String` را در `textBox` قرار می دهیم و برای انجام مراحل فوق دستوراتمان را به این ترتیب می نویسیم :

```
(  
    x = System.Convert.ToDouble(textBox1.Text);  
    z = Math.Log10(x);  
    textBox1.Text = z.ToString();  
)
```

`x=System.Convert.ToDouble(textBox \.Text);`

`z=Math.Log \*(x);`

`textBox \.Text=z.ToString();`

نتیجه این دستورات لگاریتم بر پایه ۱۰ عدد انتخابی X را در اختیار ما قرار می دهد.

\*از کلاس `Math` که کلیه عملیات ریاضی را دارد می توانیم توابع دلخواهتان را انتخاب کنیم .

مثلاً "Sin" .

```
private void button19_Click(object sender, EventArgs e)  
{  
    double degrees = double.Parse(textBox1.Text);  
    double angle = degrees * (Math.PI / 180);  
    textBox1.Text = Math.Sin(angle).ToString();  
}
```

، Cos

```
private void button9_Click_1(object sender, EventArgs e)  
{  
    double degrees = double.Parse(textBox1.Text);  
    double angle = degrees * (Math.PI / 180.0);  
    if (canconvert(textBox1.Text) == false)  
        textBox1.Text = "";  
    else  
    {  
        if ((degrees % 90 == 0) && degrees >= 90)  
            textBox1.Text = "0";  
        else  
            textBox1.Text = Math.Cos(angle).ToString();  
    }  
}
```

Tang ، \Tang که می شود Cotang و... .

```
private void button13_Click_1(object sender, EventArgs e)
{
    double degrees = double.Parse(textBox1.Text);
    double angle = degrees * (Math.PI / 180);
    if ((degrees % 90 == 0) && degrees > 90)
        textBox1.Text = "invalid value";
    else
        textBox1.Text = Math.Tan(angle).ToString();
}
```

و اما توان ، برای محاسبه توان ابتدا می دانیم که همانند + و - و یا \* و .. نیاز به ۲ عملوند داریم .

توجه داریم که آن عملگر هائیکه دارای ۲ عملوند هستند ، نیاز به = دارند ، چراکه باید عدد یا عملوند اول را بگیرد و در یک متغیر قرار دهد و پس از گرفتن عدد دوم و قرار دادن آن در متغیر دوم ، با استفاده از مساوی ، نتیجه را در متغیر جواب قرار داده و سپس آن را مشاهده می کنیم .

بعضی از عملگر ها فقط نیاز به یک عملوند دارند ، مثلاً "فاکتوریل که دستورات لازم را در خود دکمه مربوط به آن می نویسیم ، همینطور Sin ، Cos ، Tang، لگاریتم و... .

ولی توان چون دارای دو عملوند می باشد ، همان کاری که برای + یا - و مانند اینها انجام می دهیم ، برای توان هم انجام می شود.

یعنی ابتدا در دکمه توان آنچه که در textBox است را پس از تبدیل در X قرار می دهیم و سپس op برابر با عملگر توان می شود، پس از آن textBox خالی می شود.

دستورات دکمه توان به شرح زیر است :

```
private void button24_Click(object sender, EventArgs e)
{
    x = double.Parse(textBox1.Text);
    op = '^';
    textBox1.Text = "";
}
```

سپس در دکمه مساوی ، یک if خواهیم داشت که اگر op برابر با توان بود ، تابع توان را استفاده کند ،

```
if (op == '^') z = Math.Pow(x, y);
```

یعنی خواهیم داشت :

If(op=='^')=z=Math.Pow(x,y);

و به این ترتیب عدد ورودی X به توان عدد ورودی Y خواهد رسید و به همین ترتیب دستورات دلخواه دیگر...

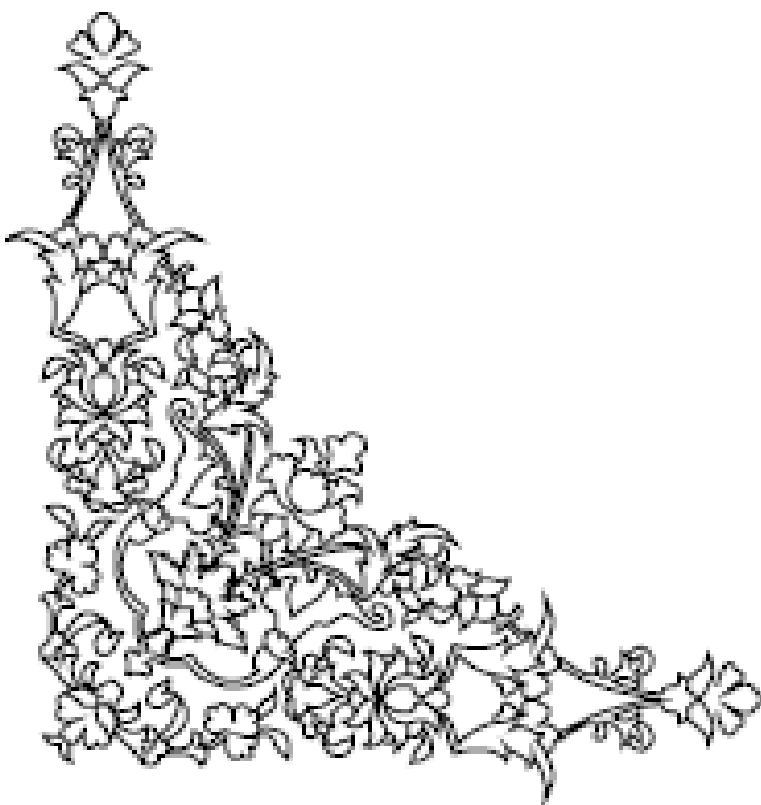






## فصل پنجم

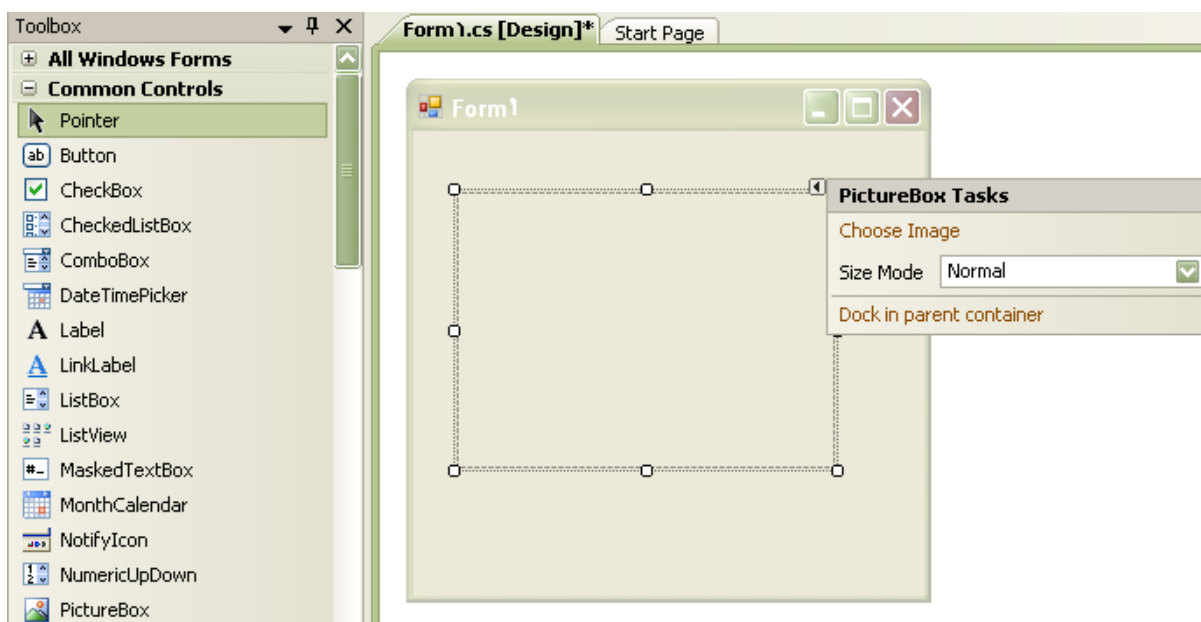
# استفاده از عکس



## ۱-۵- Load عکس :

در ادامه بحث می خواهیم در مورد Load عکس صحبت کنیم ، پس از بازکردن یک فرم از میان عناصر در جعبه ابزارمان عنصر Picture Box را انتخاب می کنیم .(ازجعبه ابزار از قسمت Common control یعنی کنترل عناصر معمولی )

۱-۱-۵- PictureBox : با انتخاب این عنصر و ترسیم یک قاب فرضی فضای مربوط به Picture مورد نظرم را انجام می دهیم .

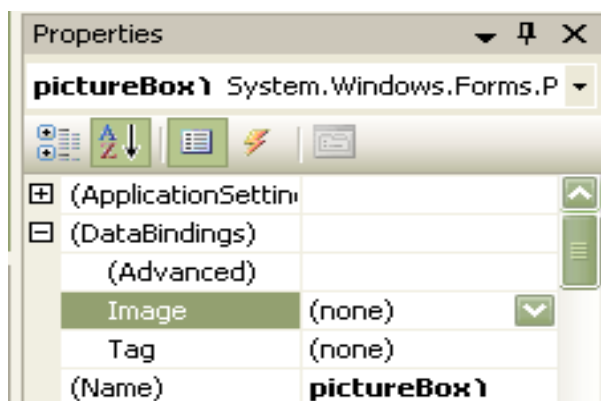


این عنصر جهت فراخوانی و نمایش عکس و همچنین اتصال به یک فیلد عکس از پایگاه داده ( مانند فیلد Image در SQL Server ، فیلد نوع Graphical در Paradox ، ( binary... ) به کار می رود. صفات مهم Picture Box عبارتند از :

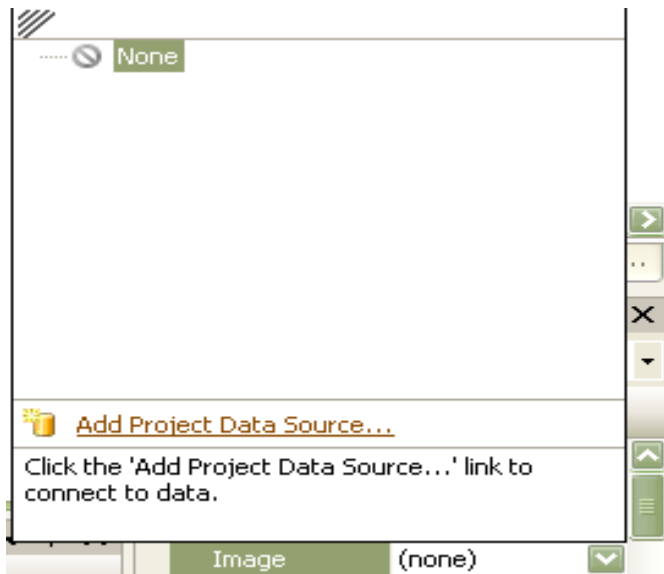
در Databinding زیر صفت Image : جهت اتصال مثلاً" به یک عکس از یک جدول بانک اطلاعاتی به

کار می رود. با باز کردن Databinding

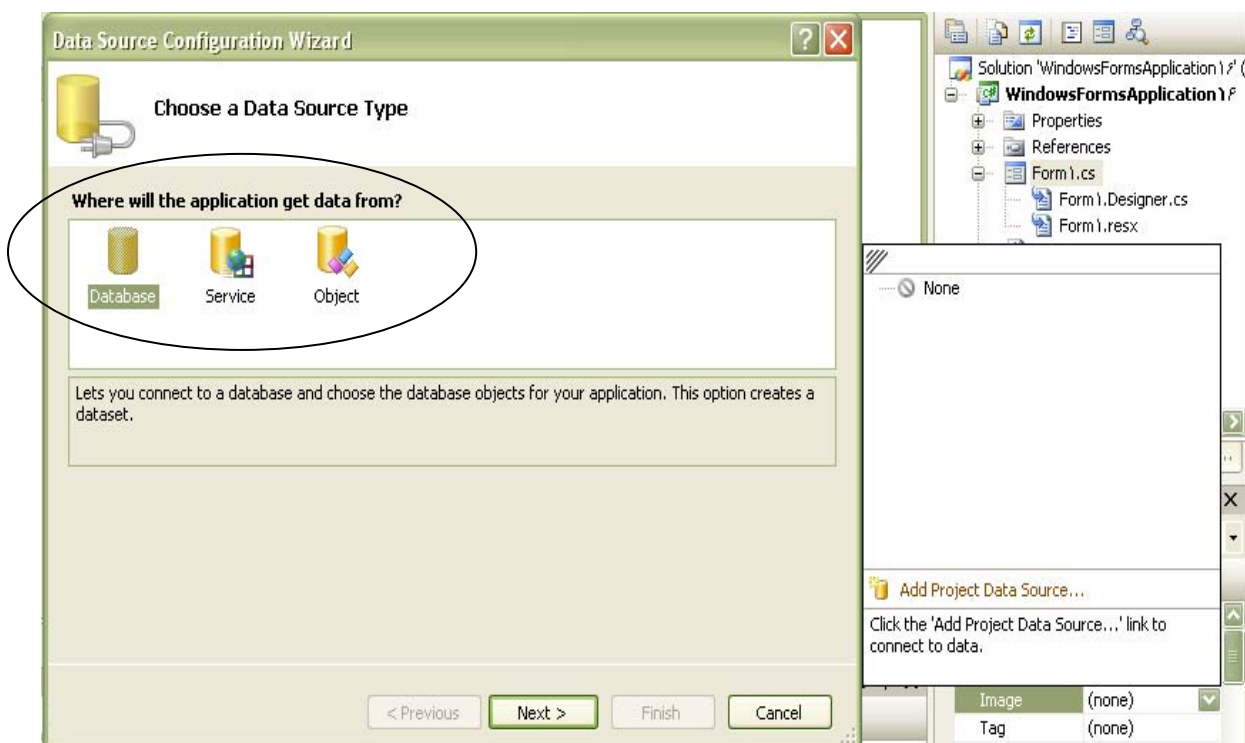
زیر صفت Image را مشاهده می کنیم ،



با باز کردن آن ، پنجره ای داریم که با انتخاب ..... Add Project Data Source



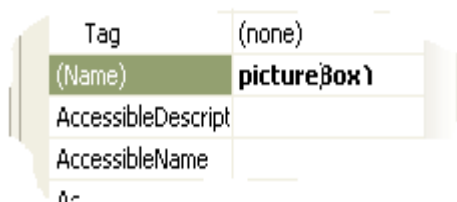
از ما می پرسد که می خواهیم به چه چیزی وصل شویم ،



که می توانیم با انتخاب هریک از آنها به Image دلخواه دسترسی داشته باشیم و آن فیلد پایگاه داده عکس به Picture Box وصل می شود و می توانیم عکس های آن پایگاه داده را نمایش داشته باشیم .  
یک تکنیک بهتر اینست که URL فایل عکس را Save کنیم ، یعنی فیلدمان را متنی بگیریم و مسیر و پسوند فایلمان را Save کنیم که این موضوع باعث می شود که افزونگی کمتری داشته باشیم .

۲-۱-۵ - Name : صفت نام که نام PictureBox در آن ذخیره می شود.

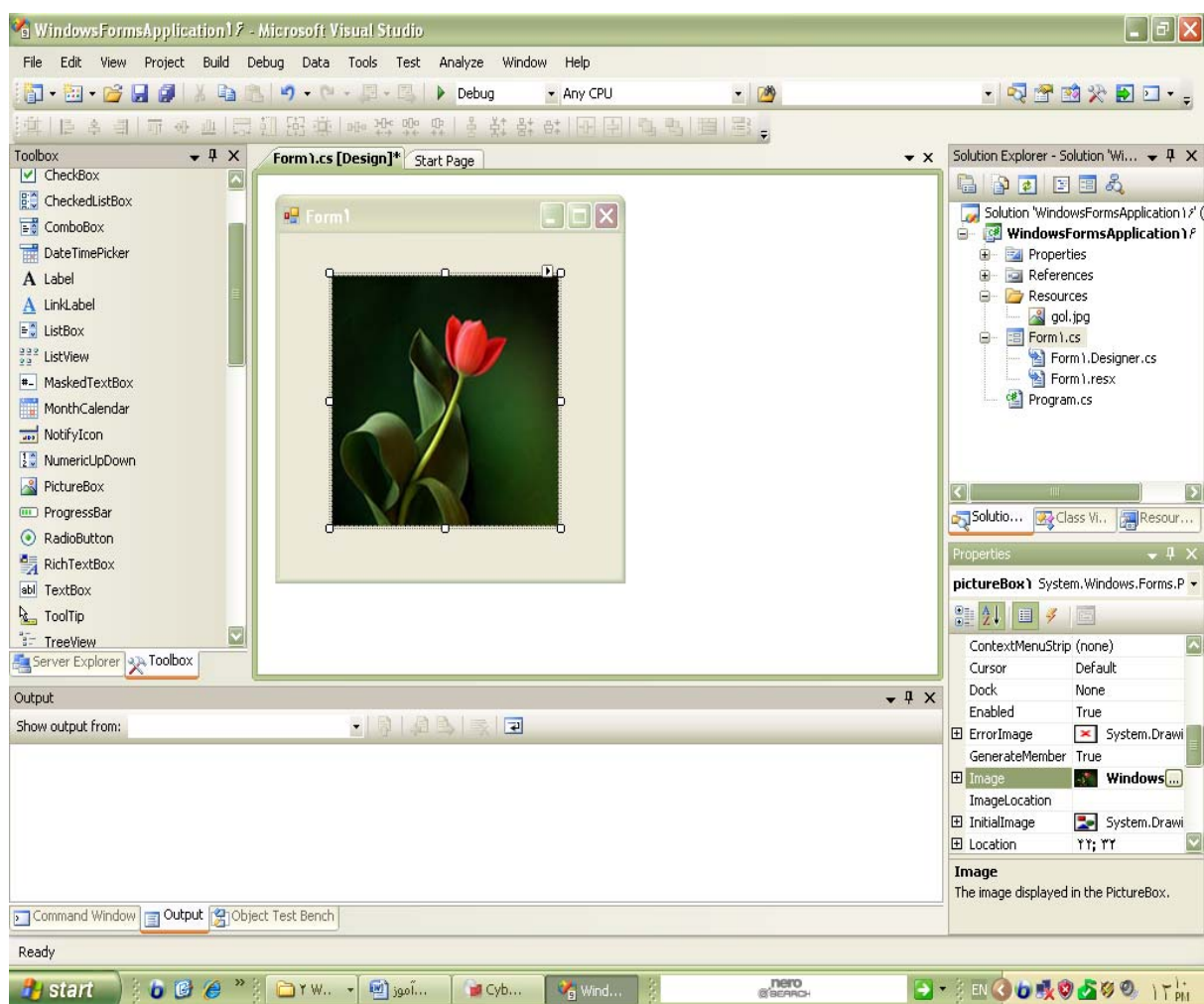
۳-۱-۵ - Image : صفت Image یکی دیگر از صفت های مهم PictureBox می باشد

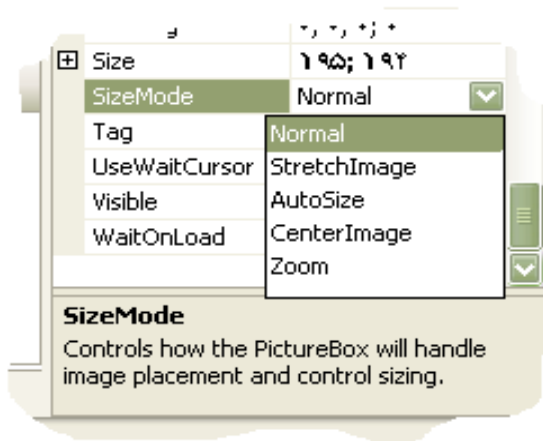


( صفت Image با زیر فهرست Image متفاوت است )

URL فایل عکس و یا خود فایل عکس به این صفت اختصاص می یابد.

وقتی دکمه مقابل Image را می زنیم به ما اجازه Import کردن ( واردکردن ) را می دهد و ما عکس دلخواه را از مسیر عکس بر می داریم و از آن استفاده می کنیم .





۴-۱-۵ - **SizeMode** : این صفت جهت تعیین

سایز عکس استفاده می شود که حالت های متفاوتی

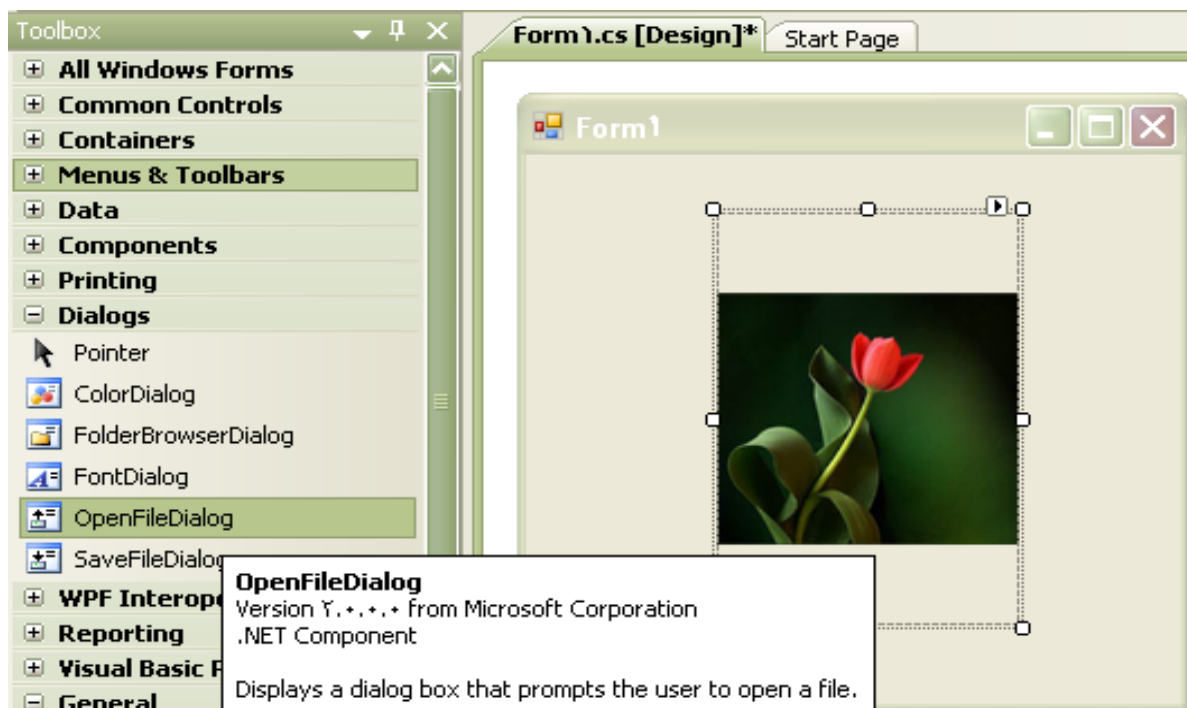
دارد ،

مثلاً"

- **StragelImage** : که در این حالت عکس مورد نظر خود را به اندازه سایز عنصر PictureBox در می آورد. در این حالت می توان ابعاد و حالت عکس مورد نظر را به دلخواه تغییر داد.
- **Autosize** : در این حالت PictureBox خود را به اندازه عکس فراخوانی شده در می آورد.
- **Zoom** : در این حالت ابعاد را بزرگ می کند.

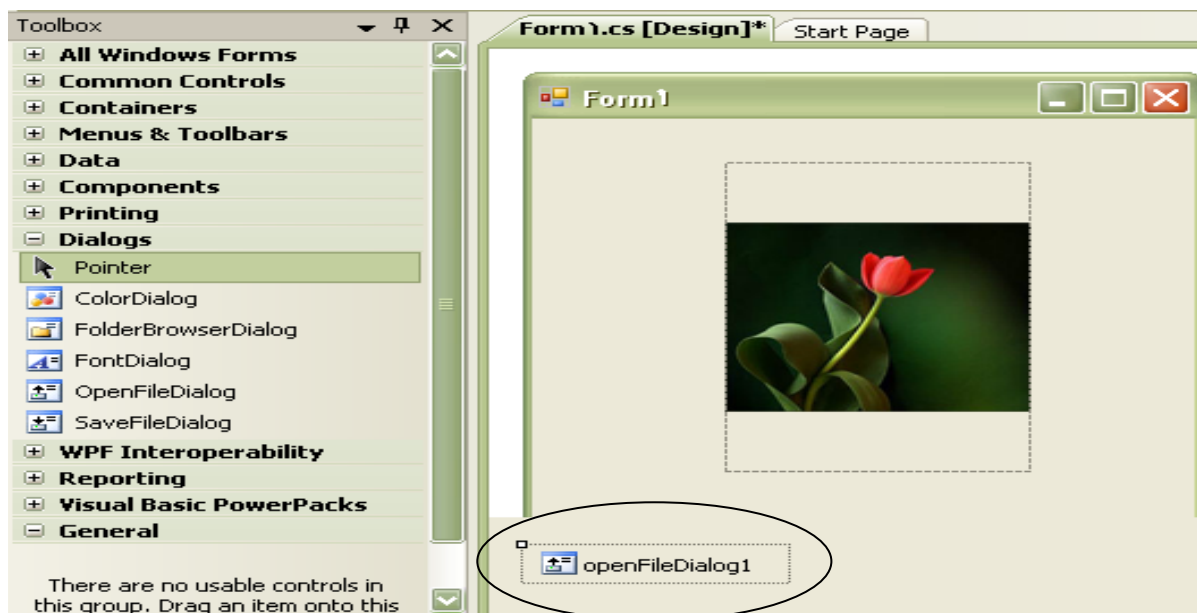
.....

تا به حال با تعدادی از امکانات Common control آشنا شدیم ، حال می خواهیم با گوشه ای از امکانات قسمت Dialogs آشنا شویم :



در قسمت ابزار Dialogs عنصر OpenFileDialog را انتخاب می کنیم .

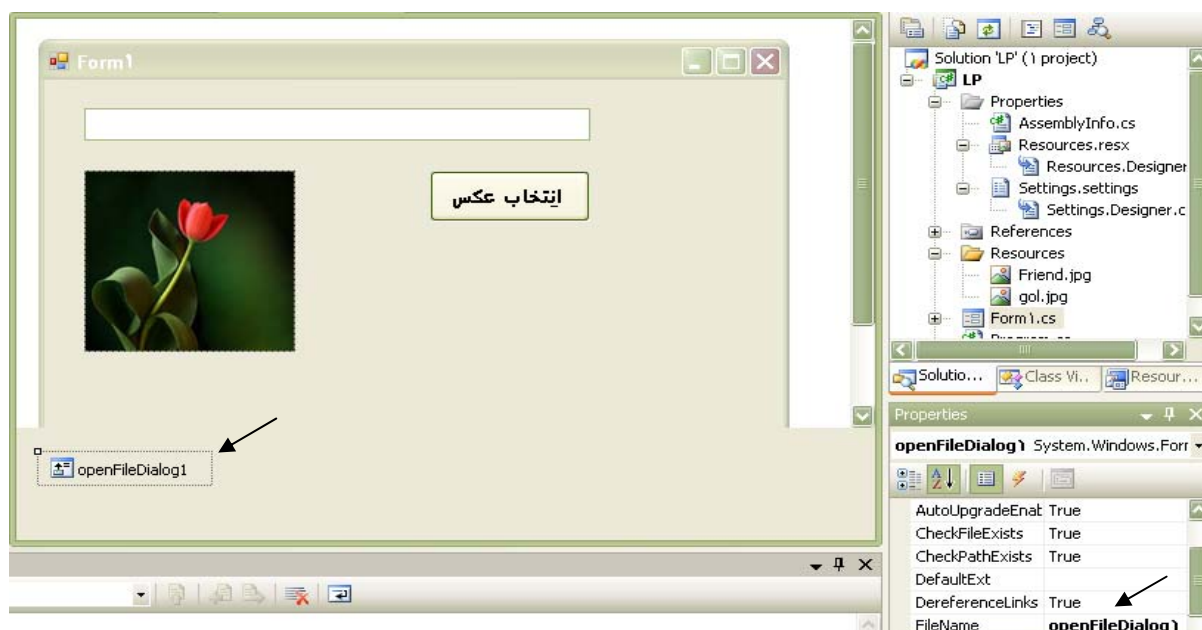
\*توجه داریم که این عنصر غیر Visual است ، عناصر Visual روی فرم قرار دارد ولی عناصر غیر Visual روی فرم قرار نمی گیرد و در اجرا دیده نمی شوند و در پائین فرم قابل مشاهده هستند.



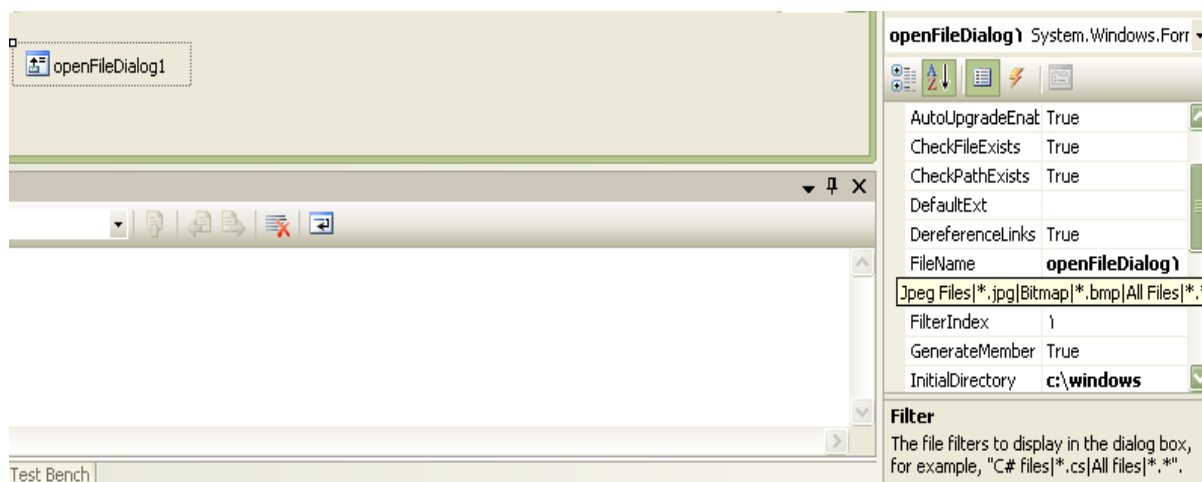
این عنصر جهت ایجاد امکان Browse (مرور کردن) با امکان انتخاب فایل از پنجره باز شد ، به کار می رود و دارای صفات مهم زیر می باشد:

۵-۱-۵ - FileName ( نام فایل ) : فایلی را که در پنجره باز شده ، نمایش می یابد و انتخاب می شود

توسط این صفت قابل دسترسی می باشد.

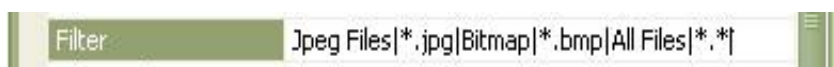


۶-۱-۵- Filter ( تعیین کننده ) : تعیین اینکه چه نوع فایل هایی در پنجره باز شده قابل دسترس باشد توسط فیلتر انجام می شود.



یک فیلتر مناسب به صورت زیر تعیین می شود:

پسوند فایل مورد نظر.\*|نام فیلترn|.....|پسوند فایل مورد نظر.\*|نام فیلتر ۱



Jpeg File | \*.jpg | Bitmap | \*.bmp | ALL Files | \*. \* | video clip | \*.avi | video cd | \*.dat

InitialDirectory ( مقداردهی کردن پوشه ) : تعیین مسیر پیش فرض را با این صفت انجام می دهیم که در زمان Browse کردن ، پیش فرض ما کدام درایو باشد.



مثلاً " اگر ما C:\windows را بنویسیم، وقتیکه Browse می کنیم ،مسیر پیش فرض ما C:\windows خواهد بود و برای رفتن به پوشه مورد نظر ، مسیر آن پوشه را عیناً در این قسمت وارد می کنیم .

۷-۱-۵- Title : عنوانی که در بالای پنجره باز شده نمایش پیدا می کند به این صفت اختصاص می یابد ، این عنوان می تواند به هر زبانی نوشته شود.





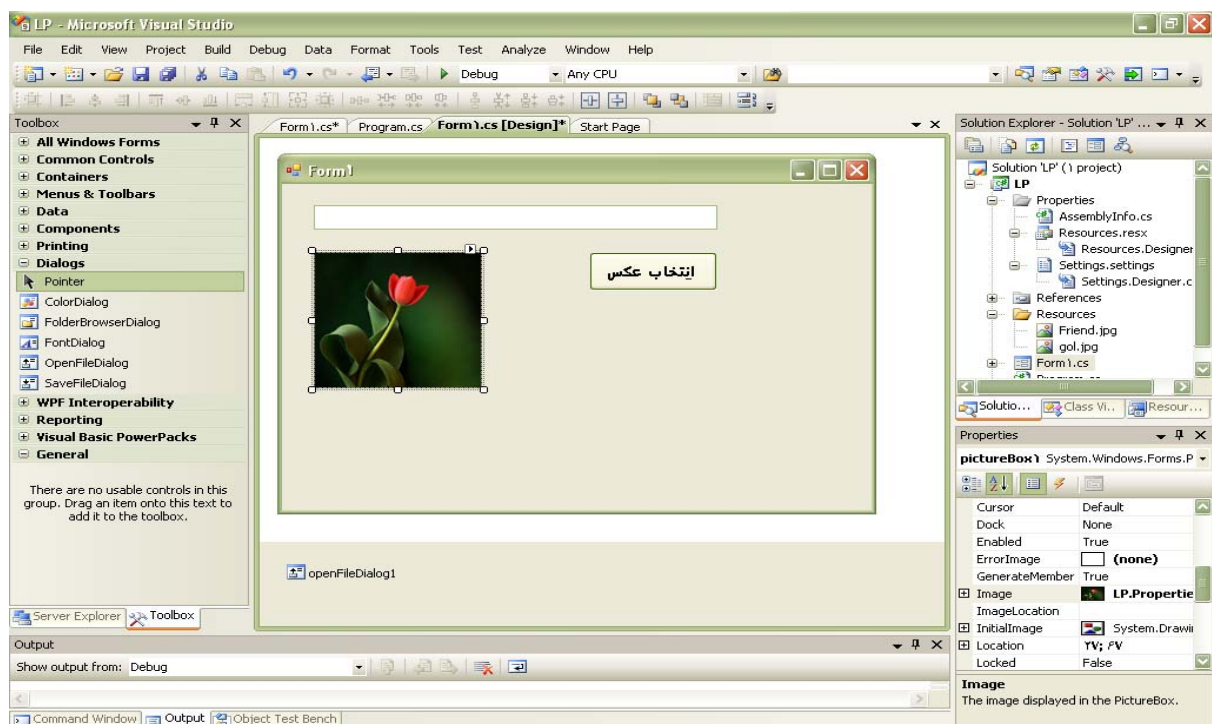
## ۲-۵- یک تمرین ساده :

برنامه ای بنویسید که با کلیک بر روی دکمه انتخاب عکس پنجره ای باز شود و با انتخاب فایل از پنجره باز شده عکس مورد نظر نمایش یابد و URL آن نیز در یک عنصر جعبه متن نمایش داده شود.

ابتدا پس از ترسیم فرم یک عنصر `textBox` ایجاد می کنیم ، سپس یک عنصر `Picture Box` ایجاد و صفت `Size mod` آن را برابر با `Stretch Image` قرار می دهیم .

سپس یک عنصر `Open file dialog` که صفت `فیلتر آن` را `فیلتر مناسب برای عکس` مقدار دهی می کنیم و صفت `Title` آن را یک پیغام مناسب قرار می دهیم ، نام این عنصر `openFileDialoge` است که توسط نام آن می توانیم به صفات آن دسترسی پیدا کنیم .

در قسمت `InitialDirectory` هم مسیر پیش فرض را تعیین می کنیم ، یک عنصر دکمه فرمان روی فرم قرار می دهیم (از قسمت `Common control` یک دکمه `Button` انتخاب می کنیم ) و صفت `Text` آن را برابر با مقدار " انتخاب عکس " قرار می دهیم ، سپس روی عکس ۲ بار کلیک می کنیم و کد مربوطه لازم را می نویسیم ، توجه داریم که برای نوشتن کد های مورد نظر، نام `openFileDialoge` نام `Image` ، `Picture Box` ، `Text` و `Textbox` را برای دسترسی به صفات آنها لازم داریم ، چرا که محتویات آن که قرار است URL شود در `Text` آن می باشد.



حال کدهای مورد نظر را به این ترتیب می نویسیم :

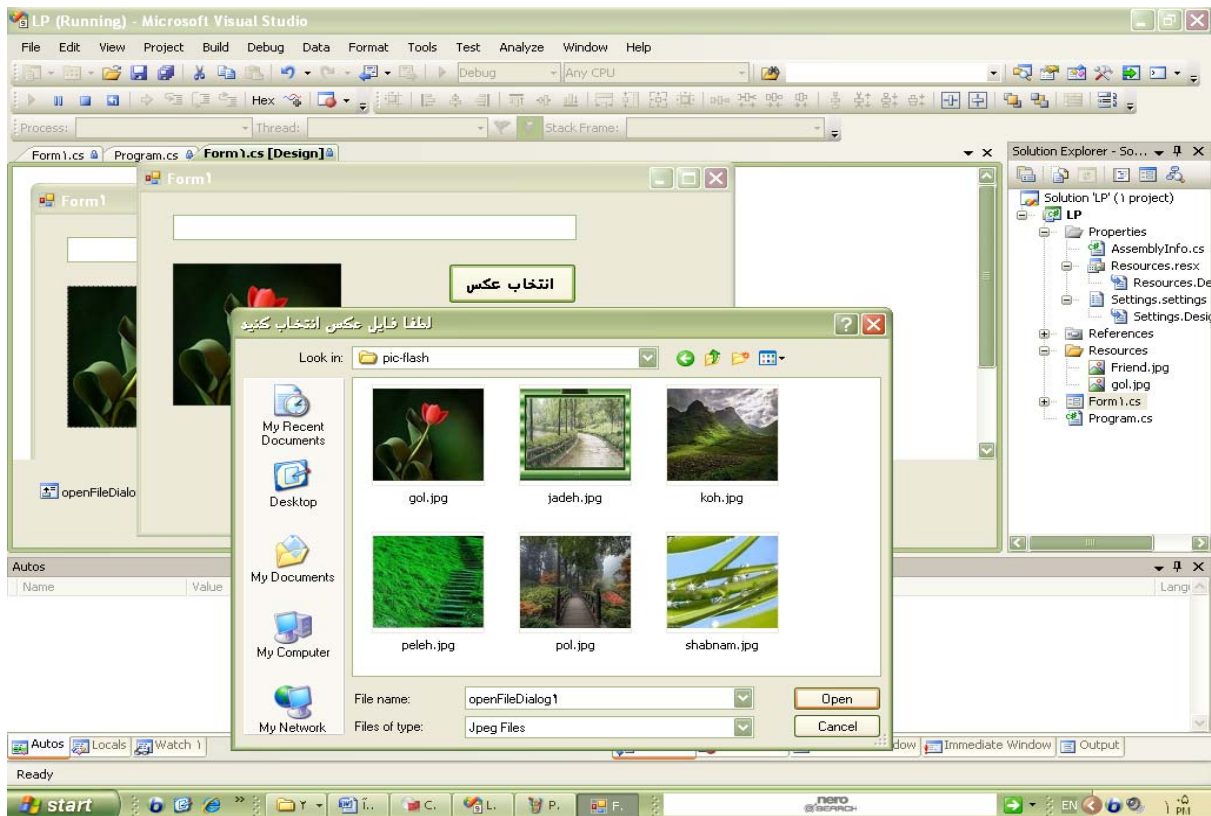
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

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

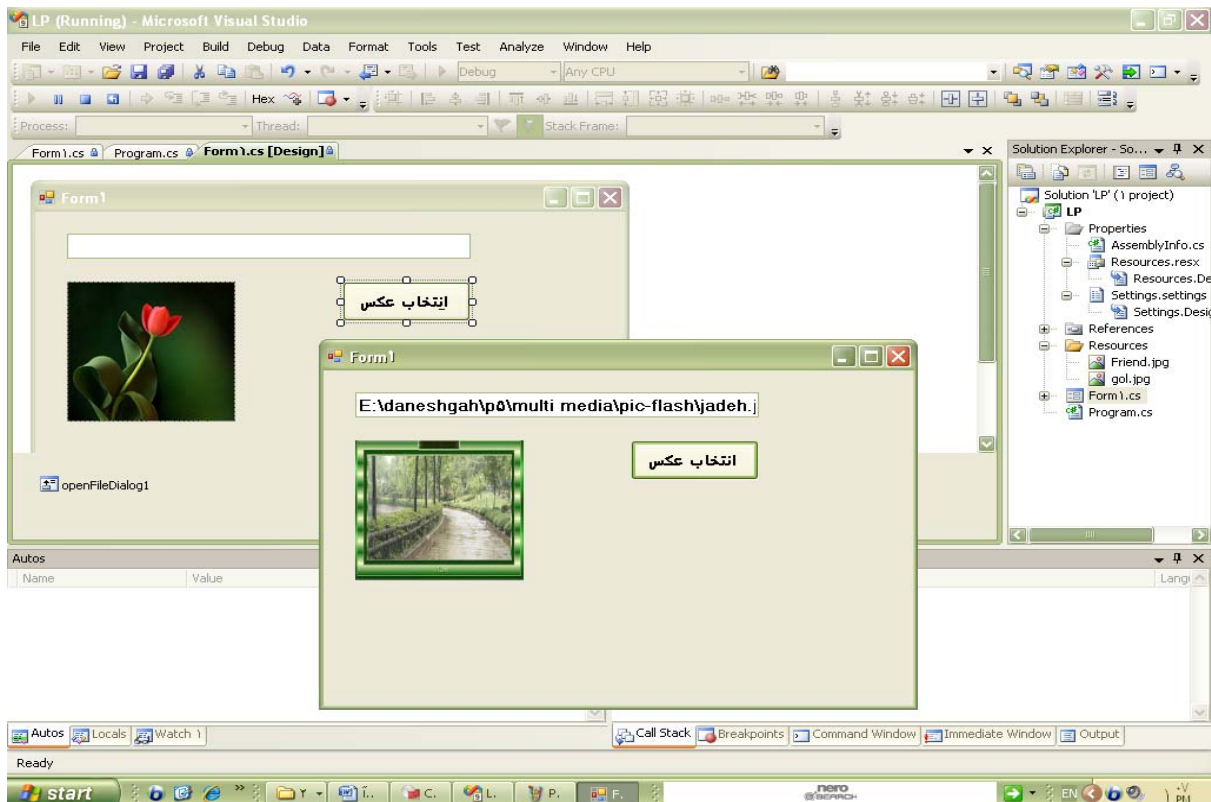
        private void button1_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                pictureBox1.Image =
                    Image.FromFile(openFileDialog1.FileName);
                textBox1.Text = openFileDialog1.FileName;
            }
        }

        private void pictureBox1_Click(object sender,
            EventArgs e)
        {
        }
    }
}
```

با اجرای این دستورات تصویر دلخواه از مسیر تعریف شده انتخاب ،



و در Picture Box و آدرس آن در Textbox قرار می گیرد.





**فصل ششم**

# **بانک اطلاعاتی**

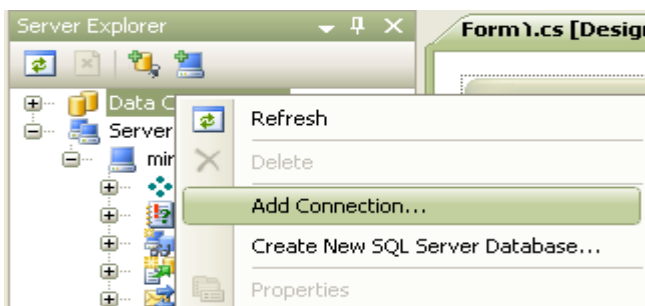


## ۱-۶- آشنایی بانک اطلاعاتی :

حال می خواهیم در مورد بانک اطلاعاتی و هم چنین در مورد ابزاری که به صورت Wizard این کار را انجام می دهند صحبت کنیم و هم چنین یک سری کلاس ها مثل Dataset، Table Adapter و..... را معرفی کرده و به روش کلاس نویسی بحث را ادامه دهیم .

ابتدا روش Wizard که خیلی سریع می توان از تباط با پایگاه داده را برقرار نمود :

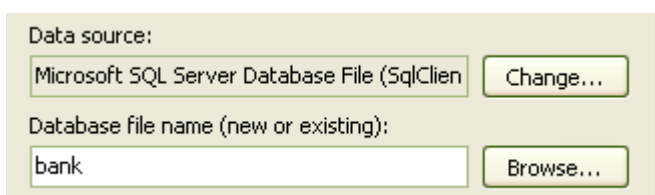
برای برقراری ارتباط با پایگاه داده، نیاز به یک بانک اطلاعاتی داریم ، در قسمت Server Explorer رایت کلیک می کنیم ،



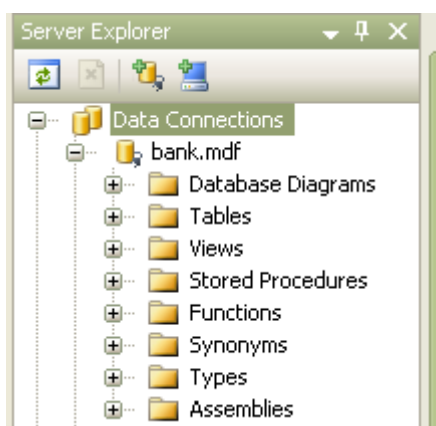
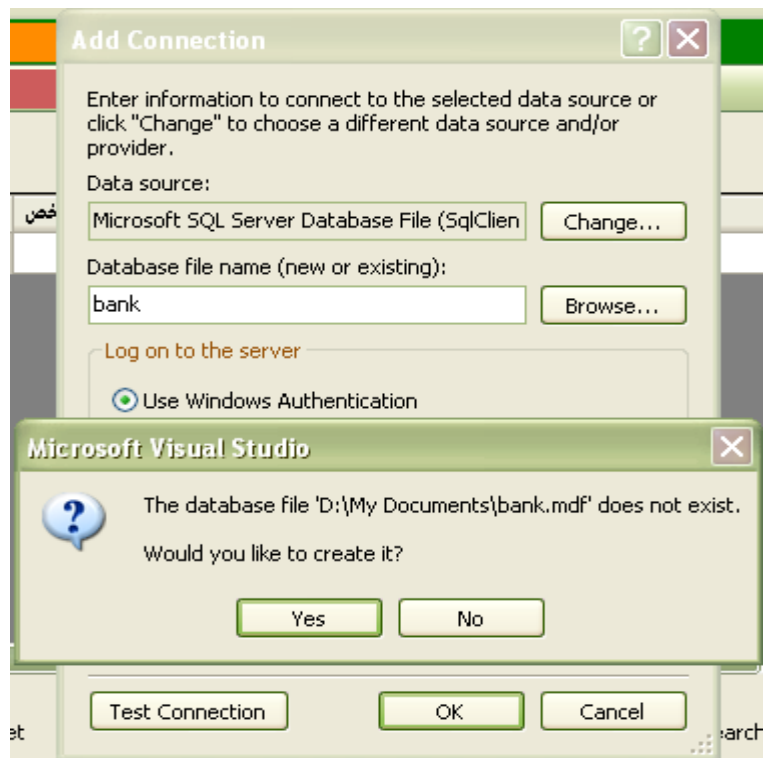
و گزینه Add connection را انتخاب کرده ، سپس پایگاه داده مورد نظرم را انتخاب ،



و بانک مورد نظرم را در Microsoft SQL Server Database file انتخاب و ok می کنیم و یک نام هم برای آن انتخاب می کنیم ،



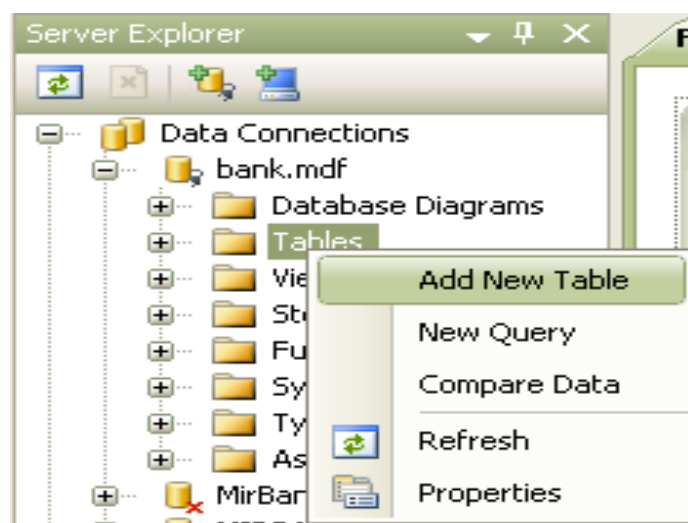
در پنجره **Add Connection** در قسمت **Database file name (new or existing)** نام **Database** مورد نظرمان را تایپ می کنیم و یا با استفاده از **Browse** پایگاه داده مورد نظرمان را انتخاب می کنیم. این زمان سؤال می کند که این فایل با این نام در بانک اطلاعاتی وجود ندارد ، آیا مایل هستید که آن را ایجاد کنید ، با کلیک بر روی **Yes** تأیید می کنیم که آن فایل را ایجاد کند.



به این ترتیب بانک مورد نظرمان ایجاد می گردد،  
بانک ایجاد شده با پسوند **mdf** ایجاد می گردد.

\*نکته : اگر فایل مورد نظرمان موجود باشد ، توسط **Browse** فایل مورد نظرمان را فراخوانی می نمایم ،  
در غیر این صورت مراحل فوق را اجرا و بانک مورد نظر را ایجاد می کنیم .

پس از ایجاد بانک در قسمت Table توسط Add new table یک Table درست می کنیم ،



۱-۱-۶ - طریقهٔ ایجاد جدول بانک اطلاعاتی جدید :

با رایت کلیک بروی Table گزینهٔ Add new table را انتخاب می کنیم ، در این زمان پنجرهٔ

آن ظاهر می شود:

A screenshot of the 'Table Designer' window in SQL Server Enterprise Manager. The window title is 'MIRZAMANI.ma...dbo.Table\_1\*'. It shows a table with three columns: 'Column Name', 'Data Type', and 'Allow Nulls'. The first row has 'id' in the 'Column Name' column, 'numeric(18, 0)' in the 'Data Type' column, and an unchecked checkbox in the 'Allow Nulls' column. The second row is empty, with an unchecked checkbox in the 'Allow Nulls' column.

Column Name	Data Type	Allow Nulls
id	numeric(18, 0)	<input type="checkbox"/>
		<input type="checkbox"/>

- در قسمت Column Name نام دلخواه فیلد ها را ( با رعایت قوانین نام گذاری ) وارد می کنیم .

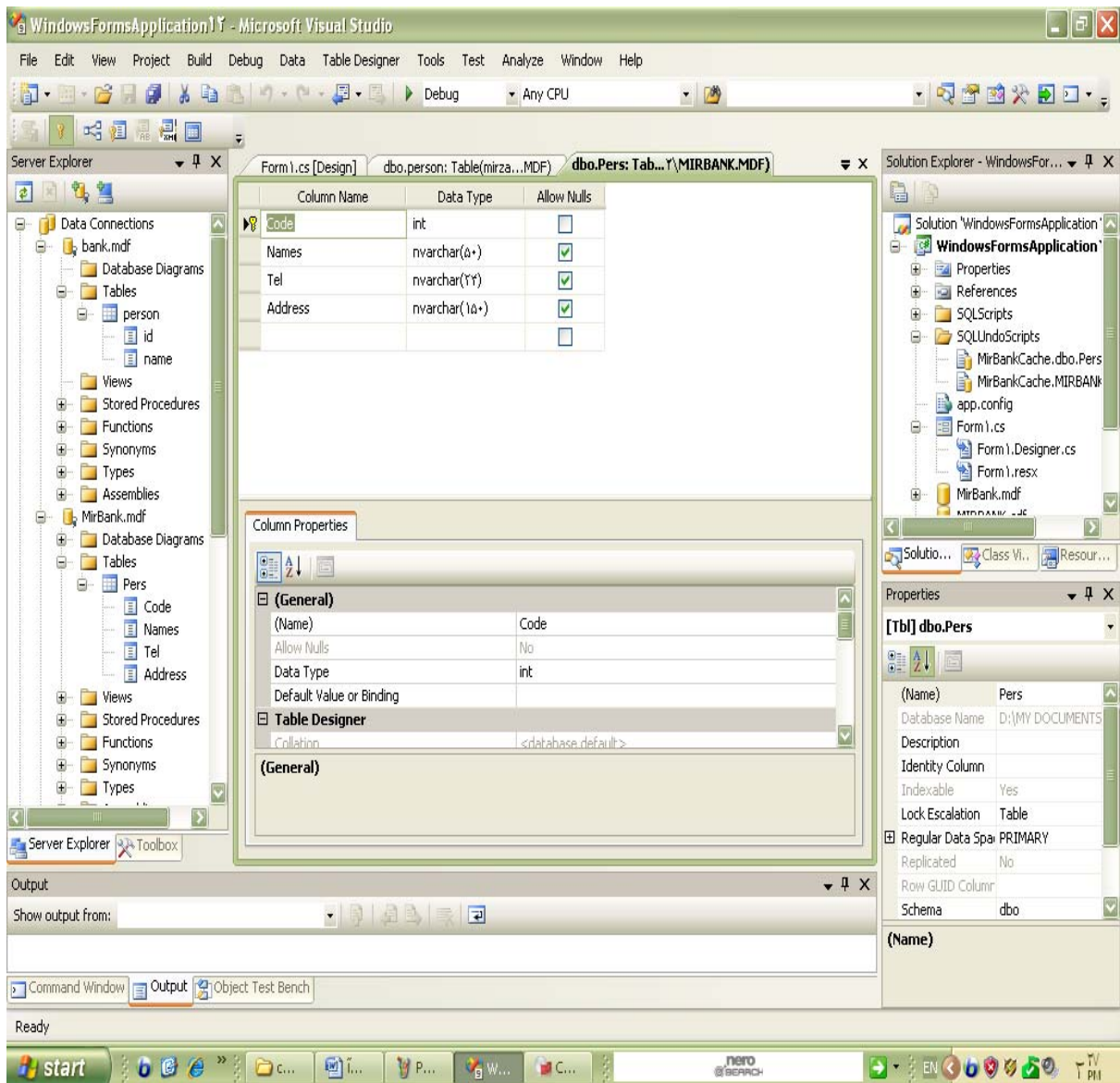
- در قسمت Data Type نوع فیلدها را مشخص می کنیم .

- قسمت Allow Nulls به ما اجازه می دهد که انتخاب کنیم که می خواهیم فیلد ها خالی باشند یا خیر.

و جهت تبدیل یک فیلد خالی به کلید اصلی ، کفایت که در کنار آن فیلد مورد نظر رایت کلیک کرده و

فرمان Set Primary Key را انتخاب کرده و در نهایت جدول را با یک نام save می کنیم .

مثلاً "Table ما شامل موارد زیر در یک بانک مجازی می باشد.



Code وام ( ۱۰ ) int

Code مشتری ( ۱۰ ) int

Date وام ( ۸ ) Char

و یا مبلغ وام ، کارمزد وام ، تعداد اقساط ، مبلغ اقساط ، نوع وام ، نام بانک و شعبه و.....



## ۲-۱-۶ - انواع فیلدها در SQL Server :

: bigint

عدد صحیح ۸ بایتی ، در محدوده  $-2^{63} < \text{bigint} < +2^{63}-1$  - بیت

: int

عدد صحیح ۴ بایتی ، در محدوده  $-2147483648 < \text{int} < +2147483647$  - بیت

: smalint

عدد صحیح ۲ بایتی ، در محدوده  $-32768 < \text{smalint} < +32767$  - بیت

: tinyint

عدد صحیح ۱ بایتی، در محدوده  $+127 < \text{tinyint} < -128$  - بیت (مثل تعداد واحد درس)

: binary(۵۰)

جهت نگهداری داده های دودویی مثل عکس، موسیقی، فیلم و... با سایز ثابت بکار می رود.

: bit

جهت نگهداری فیلدهای دو حالتی و فقط هم ۱ بیت را اشغال می کند.

( مثل True or false )

و یا در برنامه های آژانس املاک ، مثل زدن تیک در قسمت آسانسور دارد ، یا نه .

: date & datetime

فیلد نوع تاریخ و زمان ، ۸ بایت فضا اشغال می کند ، تاریخ را میلادی در نظر می گیرد،

بنابر این مناسب برای تاریخ هجری شمسی نمی باشد ، زیرا همانطوریکه می دانیم ، مثلاً"

ماه ۲ میلادی ۲۸ روز و ماه ۴ و ۶ هم ۳۰ روزه می باشد ،

بنابراین اجازه ورود ۲۹,۳۰,۳۱ / ۲ و ۴/۳۱ و ۶/۳۱ را نمی دهد.

: char(۱۰)

جهت نگهداری فیلدهای اسکی ، متنی ( رشته ای ) حد اکثر تا سایز ۴۰۰۰ کاراکتر، با

سایز ثابت .

: varchar

داده نوع متنی با سایز متغیر است و به اندازه ورودی فضا اشغال می نماید.

### نکته :

انواع داده هایی که نام آنها به صورت : نوع داده + n می باشند مانند nchar ، nvarchar و.....  
Code page الفبای غیر انگلیسی مانند فارسی را ارائه می نمایند در زبانهای برنامه نویسی میزبان نیز معمولاً در نمایش داده های غیر انگلیسی بدون مشکل کار می کنند.

: xml

داده هایی هستند که برای فرمت web استفاده می شوند.

: image

برای داده های binary و عکس استفاده می شود.

: money

برای داده های ارزی ، پولی و از این قبیل استفاده می شود.

### نکته :

اگر بخواهیم هر کجا auto number داشته باشیم و یا بصورت اتوماتیک شماره بخورد ، در قسمت صفات ، مقدار گزینه (is Identity) را در بخش Identity Specification برابر با yes قرار می دهیم .

Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

پیش فرض شروع و فاصله شماره ها برابر با 1 است که به دلخواه می توانیم آنها را تغییر دهیم .

### نکته :

وقتی فیلدی را با سایز ثابت انتخاب می کنیم ، در نظر داریم که در صورت خالی ، نصفه و یا پر بودن (در هر حال ) فیلد مورد نظر به اندازه فضای ثابت تعریف شده فضا اشغال می کند.

و.....

سپس در قسمت Show Table Date شروع به Data Entry می کنیم .

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane shows the 'person' table under the 'MirBank.mdf' database. A context menu is open over the 'person' table, with 'Show Table Data' selected. The main window displays a data grid with the following data:

Code	Names	Tel	Address
۱	فخر زمان	۲۲۲۲۲۴۵۵	تهران - درب ۱
۲	فخر جهان	۶۶۷۷۸۸۹۹	تهران طبقه ۴
۳	آقای مرادی	۱۱۲۲۳۳۴۴	تهران جردن ...
	خانم حسین زاده	۷۷۸۸۹۹۰۰	تهران رسالت
	خانم عبدالله وند	۷۷۷۷۶۶۵۵	تهران شرق
	NULL	NULL	NULL

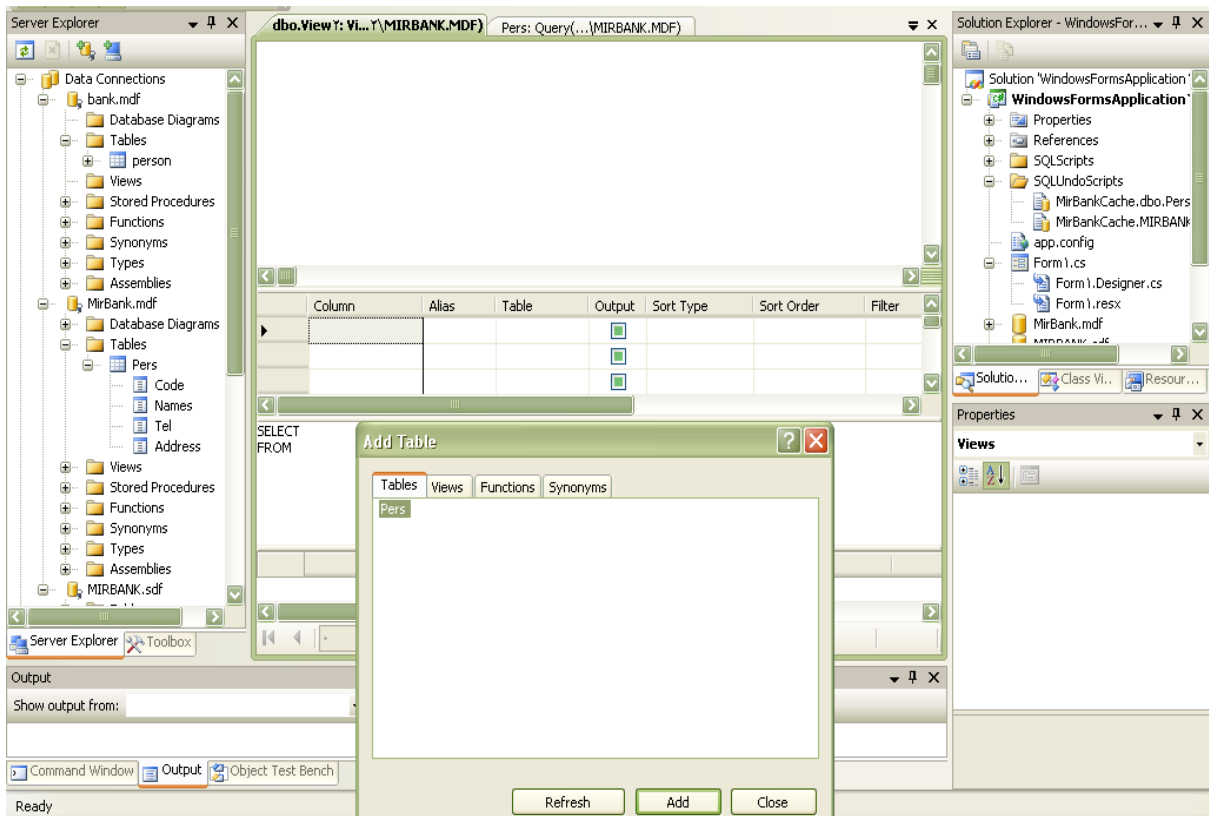
Callout Box 1: باز کردن جدول در نمای طراحی :  
در این حالت به قسمت تعریف فیلدها و مشخصات آنها دسترسی داریم .

Callout Box 2: نمایش جدول در نمای نمایش و ورود اطلاعات :  
در این حالت امکان ویرایش و ایجاد را هم در اختیار ما قرار می دهد.

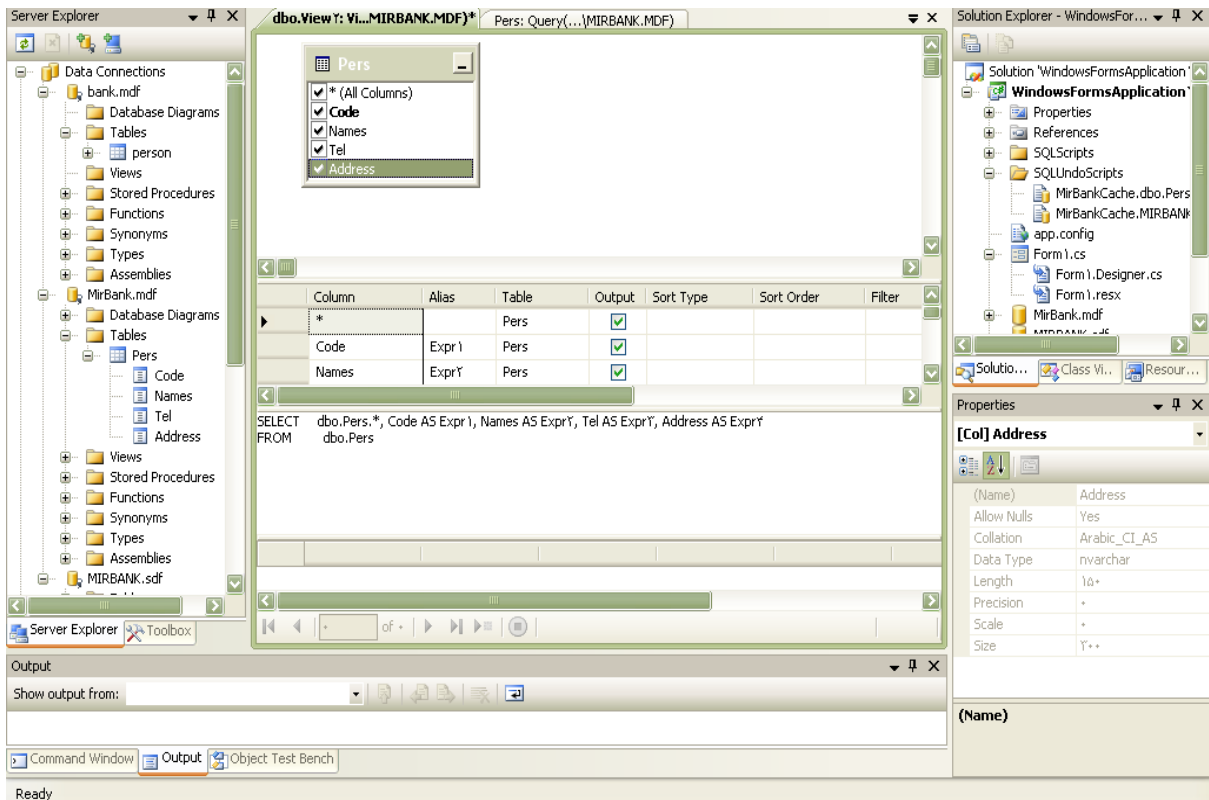
سپس با رایت کلیک بر روی views و انتخاب گزینه Add New View ،

The screenshot shows the SQL Server Enterprise Manager interface. In the 'Server Explorer' pane, the 'Views' folder is selected. A context menu is open over the 'Views' folder, with 'Add New View' selected. The main window displays the same data grid as in the previous image.

سپس پنجره جدیدی در اختیار ما قرار می گیرد که با انتخاب نام Tables ،

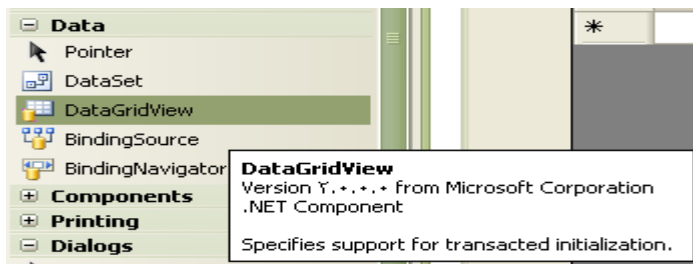


و نام فیلد های دلخواه می توانیم select های دلخواه را بنویسیم ،

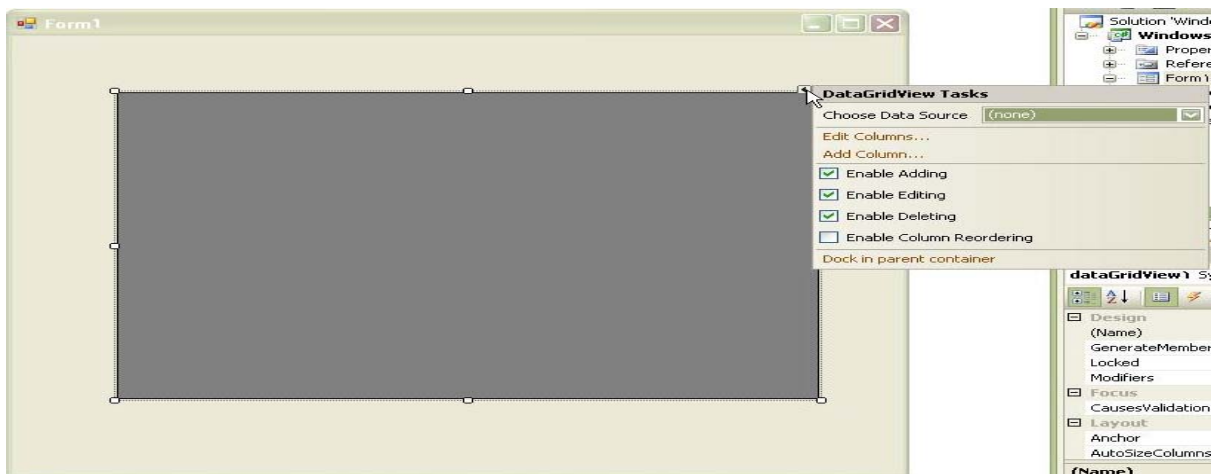


## ۲-۶- اتصال به بانک اطلاعاتی :

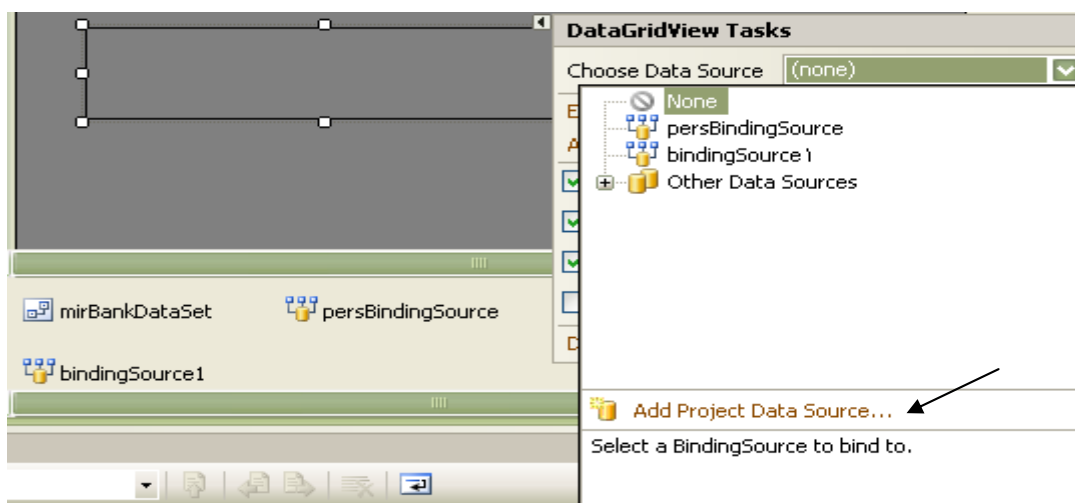
حال می خواهیم به بانک اطلاعاتی وصل شویم ، در ابزار Data با عناصر دیگری آشنا می شویم :



۱-۲-۶- **DataGridview** (اتصال به یک منبع داده ) : این عنصر جهت اتصال به یک Table ، view ، StoredProcedure یا Function به کار می رود، با قرار دادن این عنصر روی فرم ، در فرم مورد نظر منویی ظاهر می شود،

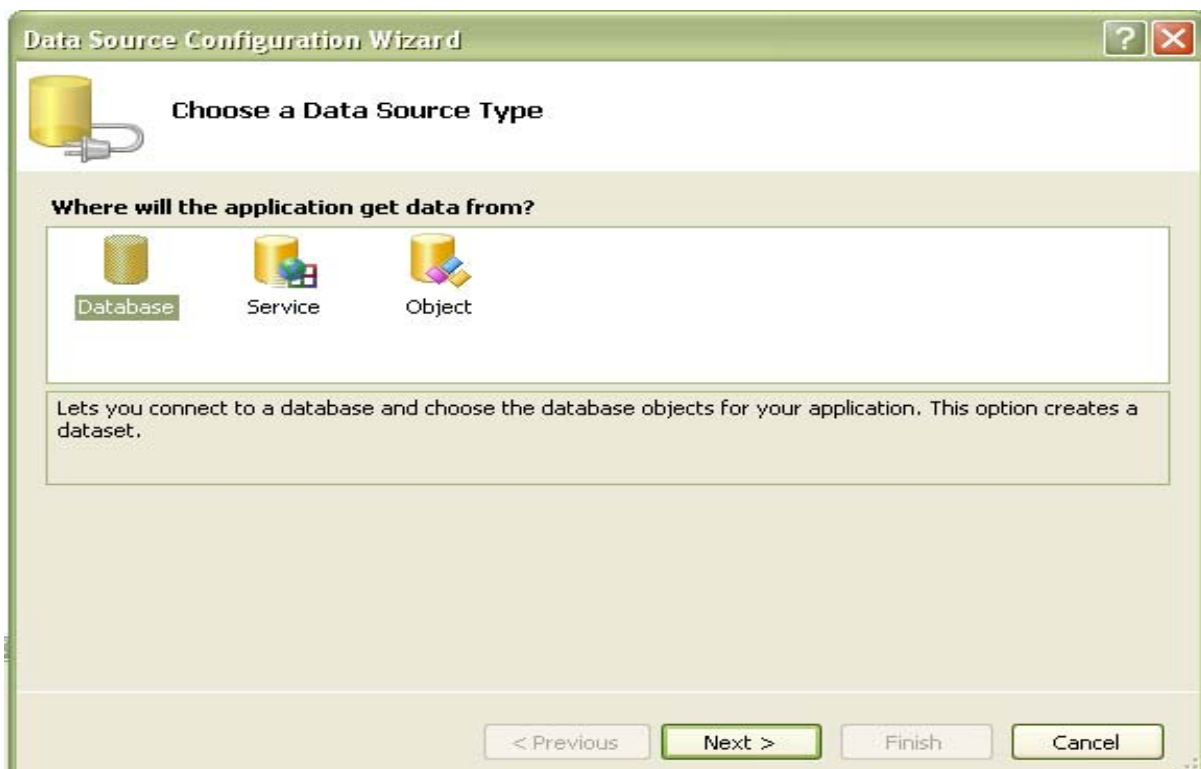


به نام DataGridview Tasks که اگر این Menu نبود ، کافیست که روی علامت کلیک کنیم . در این منو در قسمت Choose Data Source ( انتخاب منبع داده ) از لیست باز شو ،

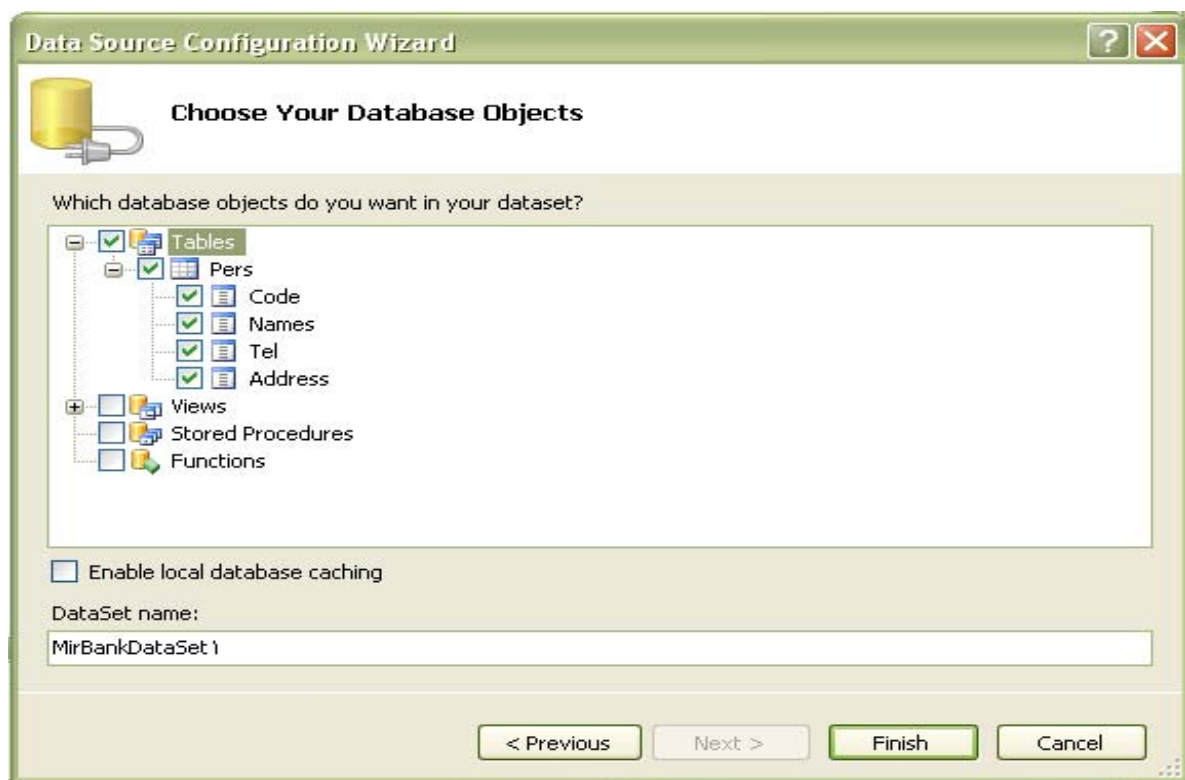


گزینه Add project data source را انتخاب می کنیم .

پنجره Data Source Configuration wizard ظاهر می شود ،

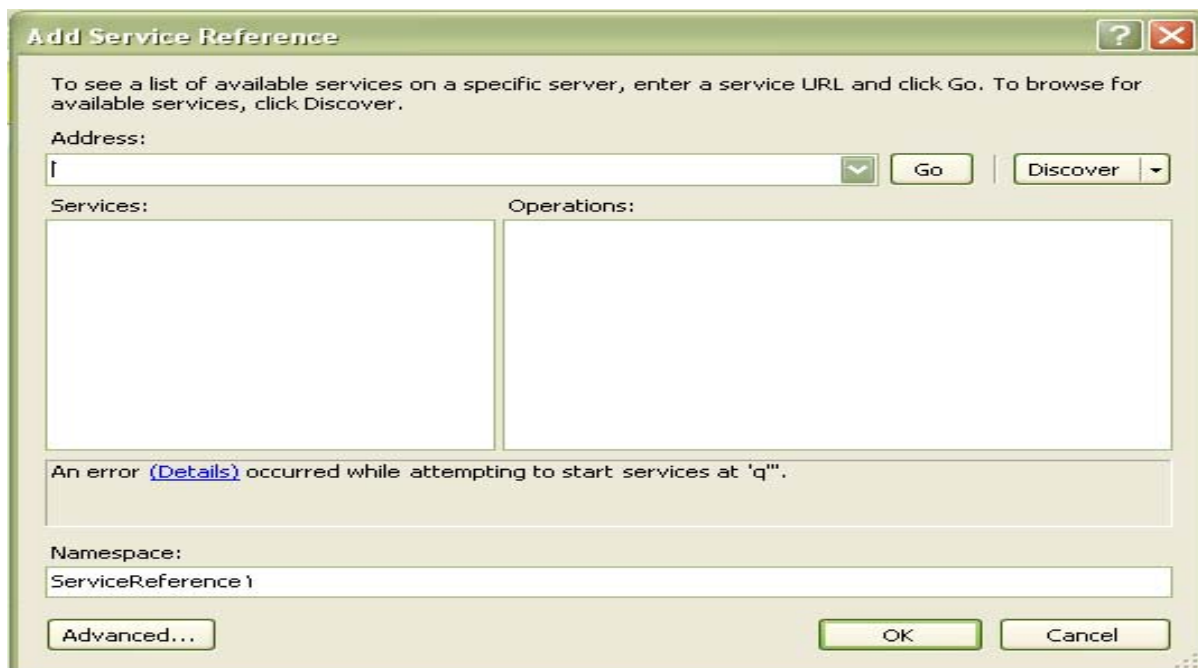


که در آن ۳ انتخاب وجود دارد که اولین گزینه Database است که اجازه می دهد به پایگاه داده متصل شده و شیء ای از نوع پایگاه داده را برای برنامه کاربردی انتخاب کنیم ،

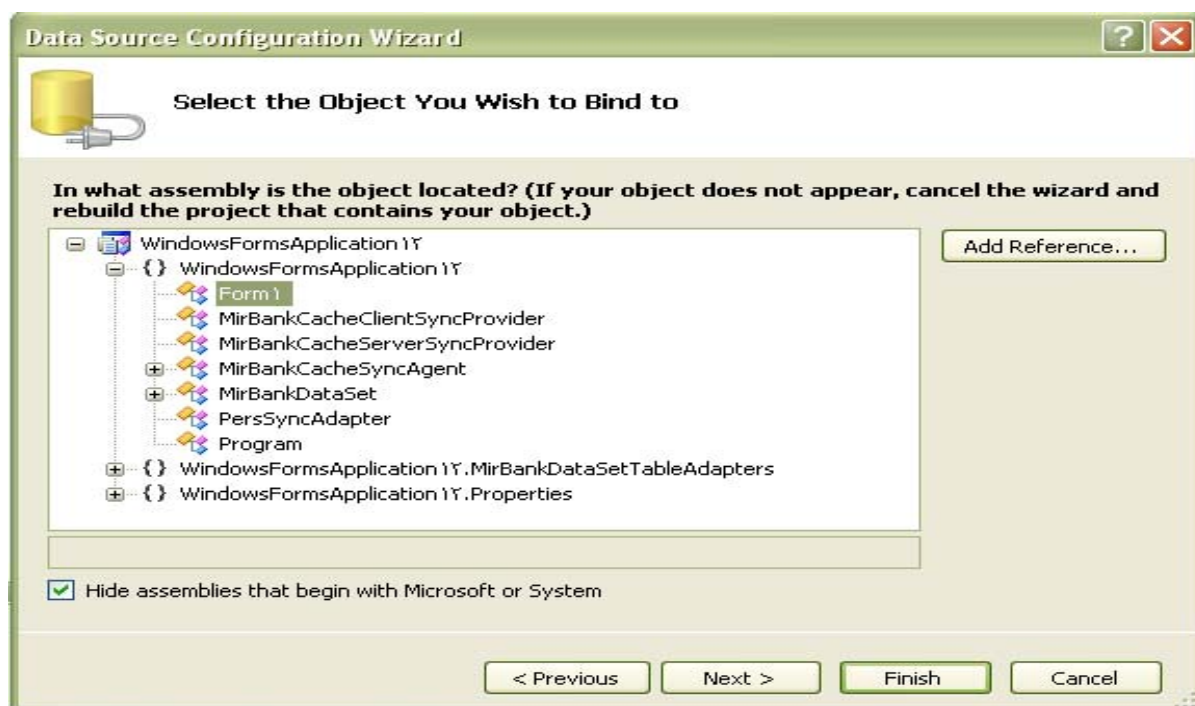


و این انتخاب یک عنصر Database ای را ایجاد می کند.

۲-۲-۶- Web service : این سرویس پنجره گفتگوی اضافه نمودن مرجع وب را فراخوانی می کند و به ما اجازه می دهد که یک ارتباط با سرویس وبی که داده را برای برنامه کاربردی مان باز می گرداند ایجاد کنیم . (ازانواع کاربردهای وب سرویس می توان پرداخت وجه اینترنتی ، قرار دادن اطلاعات نرخ ارز و طلاو.....را در یک سایت اطلاعاتی نام برد.)



۳-۲-۶- Object : این گزینه به ما اجازه می دهد که شیءای را که می تواند بعداً برای تولید کنترل های بانک داده استفاده شود انتخاب کنیم .



### ۳-۶- Wizard اتصال به بانک اطلاعاتی :

در اینجا مراحل را Step by step پیش می رویم :

ابتدا در قسمت Server explorer رایت کلیک می کنیم و سپس گزینه Add connection را انتخاب می کنیم ، در قسمت Data source نوع پایگاه داده را انتخاب می کنیم که آن انتخاب در این جا Microsoft SQL server data base file می باشد و سپس یک نام برای آن انتخاب می کنیم و Ok می کنیم، در پنجره بعدی از ما می پرسد که با این نام بانکی وجود ندارد ، آیا می خواهید ایجاد شود در جواب به این سؤال هم Yes می زنیم و در نهایت Ok می کنیم به این ترتیب Database ما ایجاد می شود. با رایت کلیک و Add new table شروع به ایجاد جداول مورد نظر می کنیم ، ترتیب و چگونگی ایجاد جداول شبیه به سایر پایگاه داده های دیگر می باشد.

ورود اطلاعات در جداول هم به همین ترتیب شبیه دیگر پایگاه های داده می باشد.

برای ساخت View هم با رایت کلیک و Add new view می توانیم View داشته باشیم .

سپس در قسمت Data Grid view table با انتخاب Data source و گزینه Add project می توانیم نوع Data source مورد نظرم را انتخاب کنیم .

با انتخاب Database و Next و انتخاب بانک مورد نظر و مجدداً " Next و پاسخ به سؤالات بعدی به دلخواه ، لیستی از جداول ، View ها و ..... را مشاهده می کنیم و در آخر فرم نهایی را مشاهده می کنیم ،

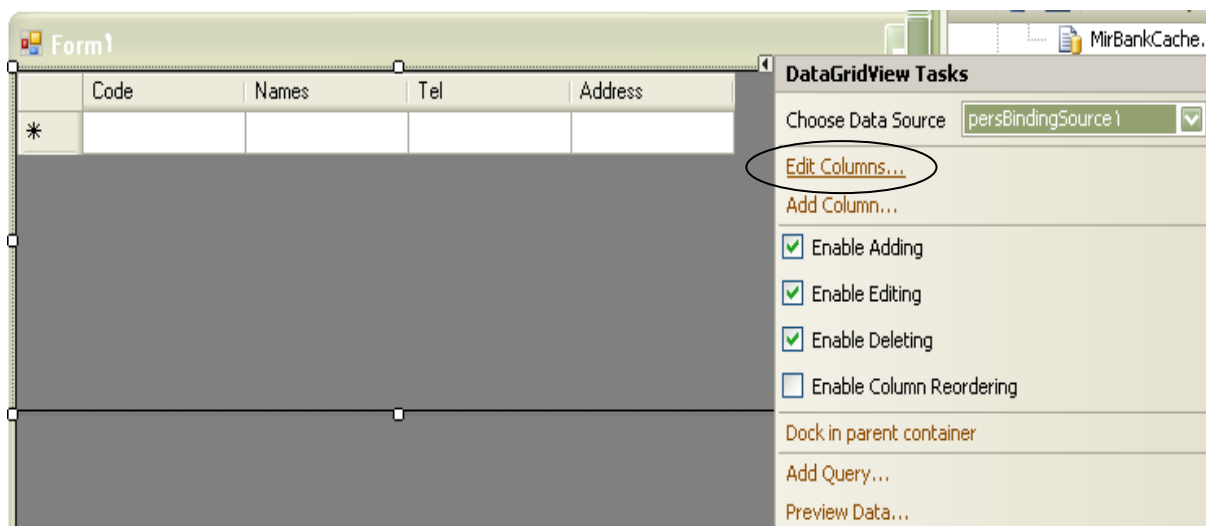


	Code	Names	Tel	Address
*				

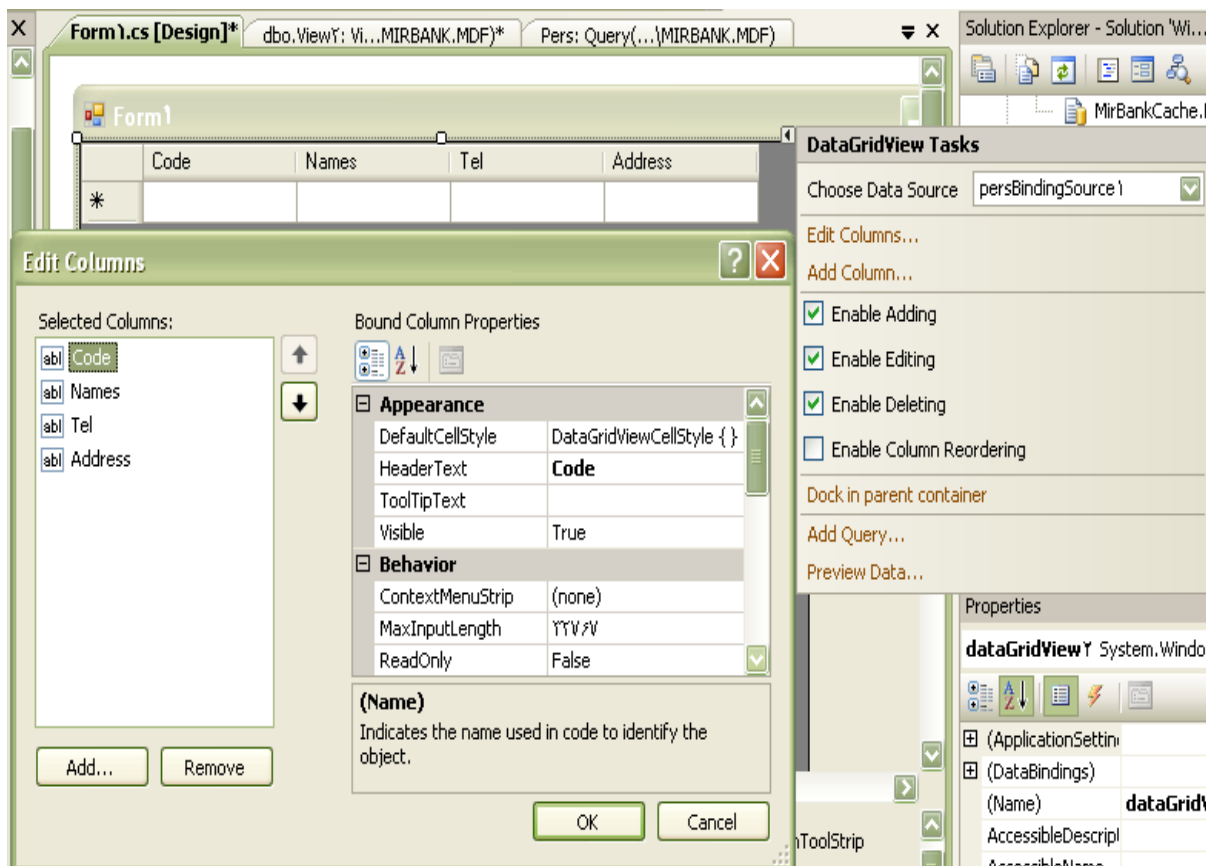
که با اجرای فرم ، اطلاعاتی را که وارد کرده ایم را هم مشاهده می کنیم و این اطلاعات به دلخواه قابلیت Sort شدن را هم دارا هستند.



با رایت کلیک بر روی نشانه فرم و انتخاب گزینه Edit columns می توانیم Edit های دلخواه را روی هر یک از انتخاب ها انجام دهیم و همچنین ترتیب قرار گرفتن آنها را هم تغییر دهیم و..... سپس با کلیک بر روی فلش بالای Data grid view و انتخاب فرمان Edit columns پنجره ای باز می شود ،



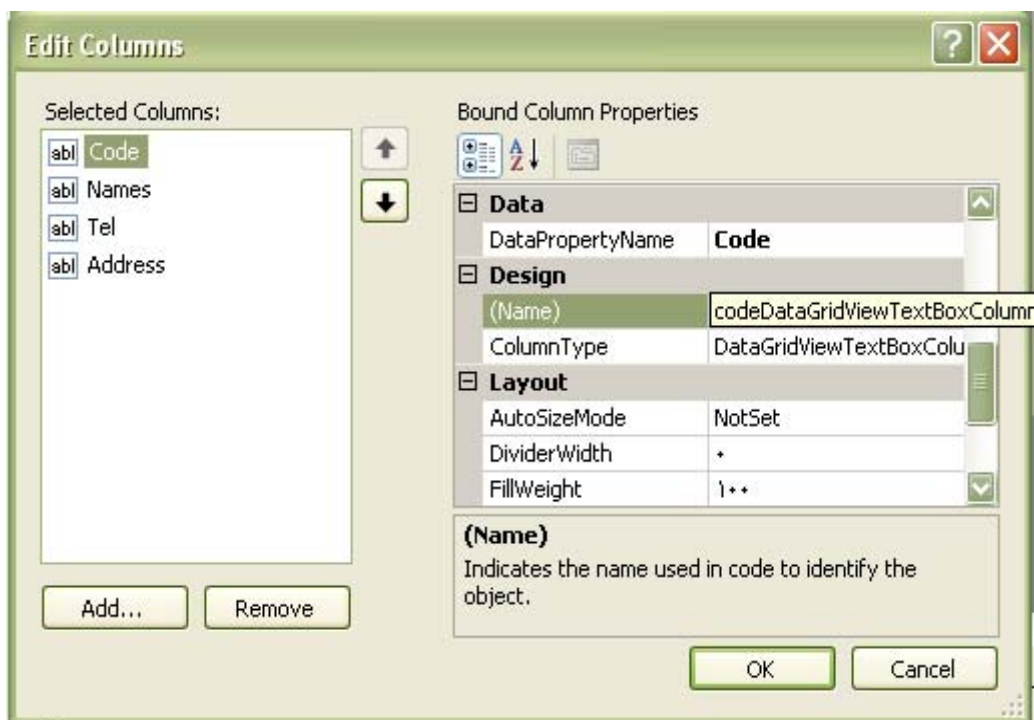
که امکان برخی از ویرایش ها و تنظیمات را در اختیار ما می گذارند.



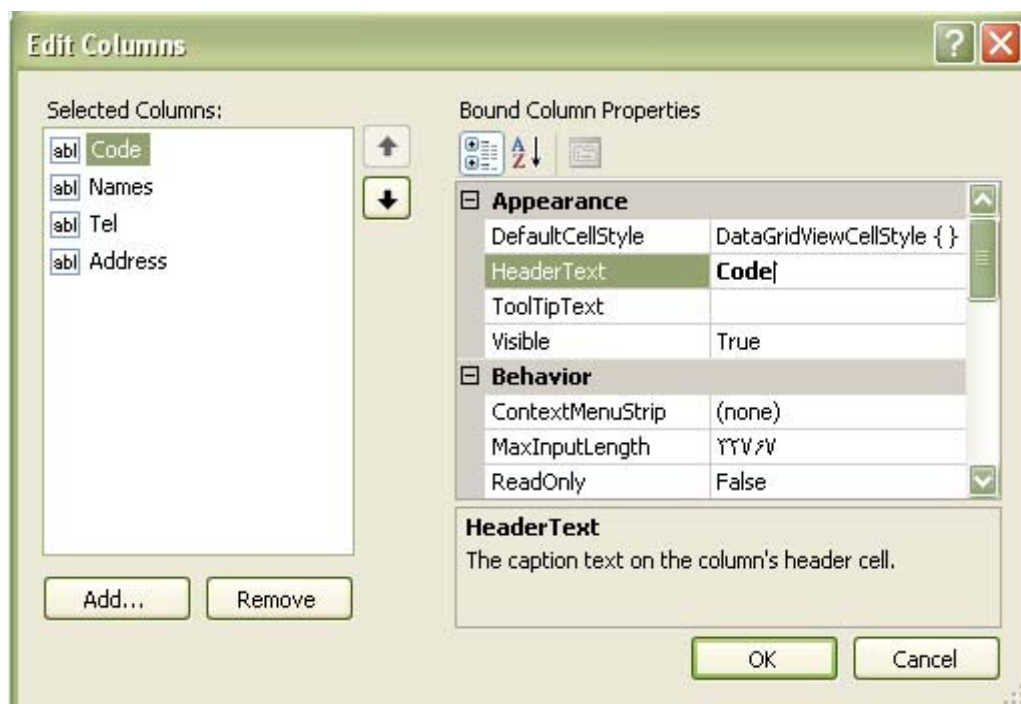
## ۴-۶- معرفی امکانات :

تعدادی امکانات ویرایش و تنظیمات عبارتند از :

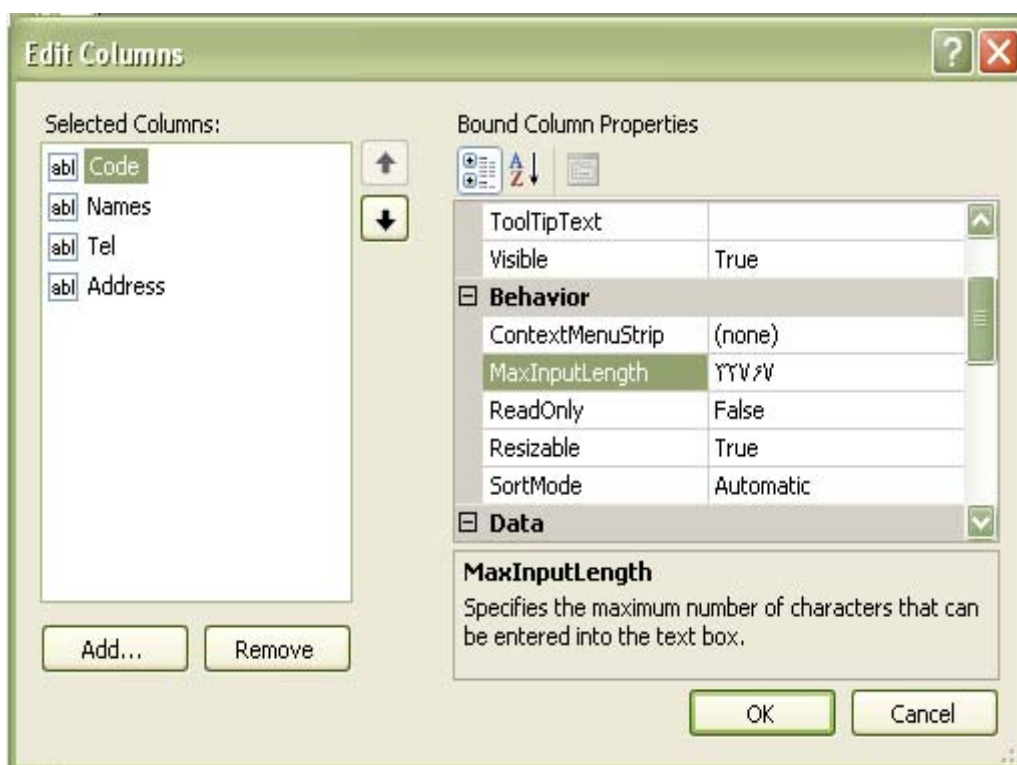
۱-۴-۶- **Name** : که می توانیم از آن در کد نویسی استفاده کنیم .



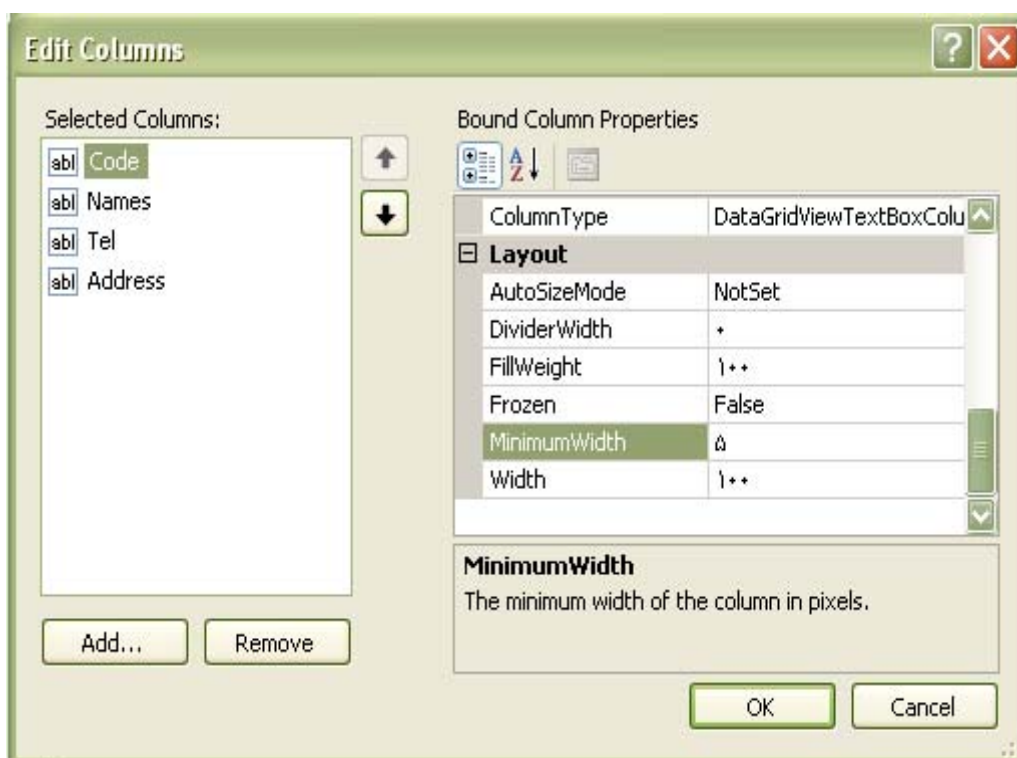
۲-۴-۶- **Header Text** : که عنوان یا متن سرستون را تعیین می کند .



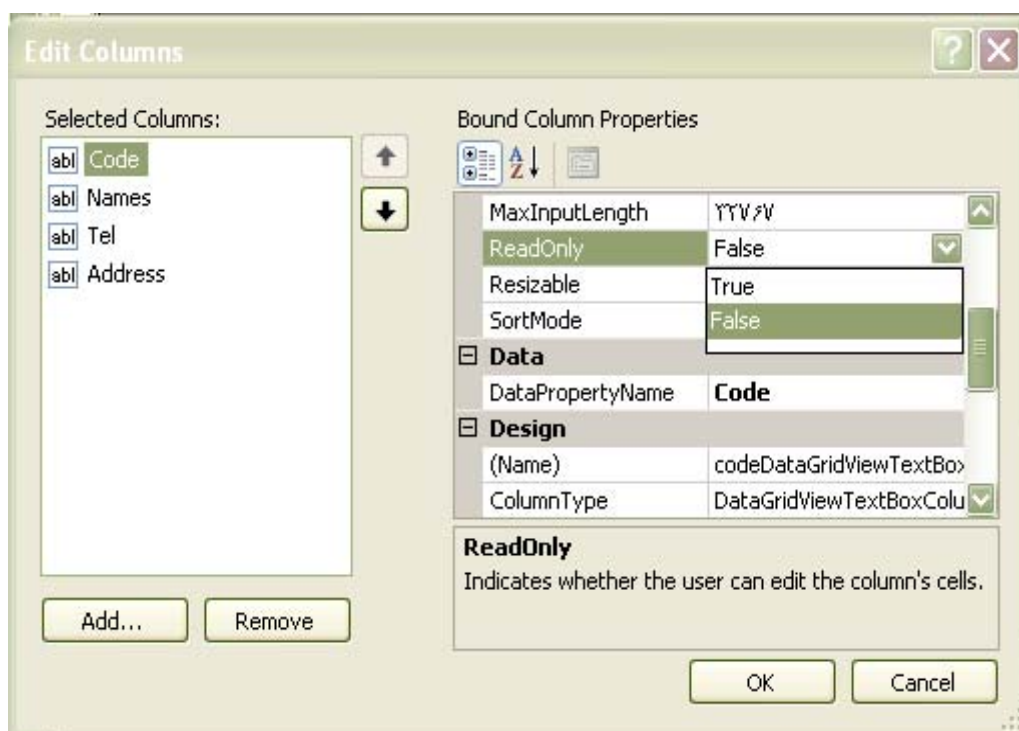
۳-۴-۶ - **MaxLength**: طول حداکثر کاراکتر ورودی را تعیین می کند .



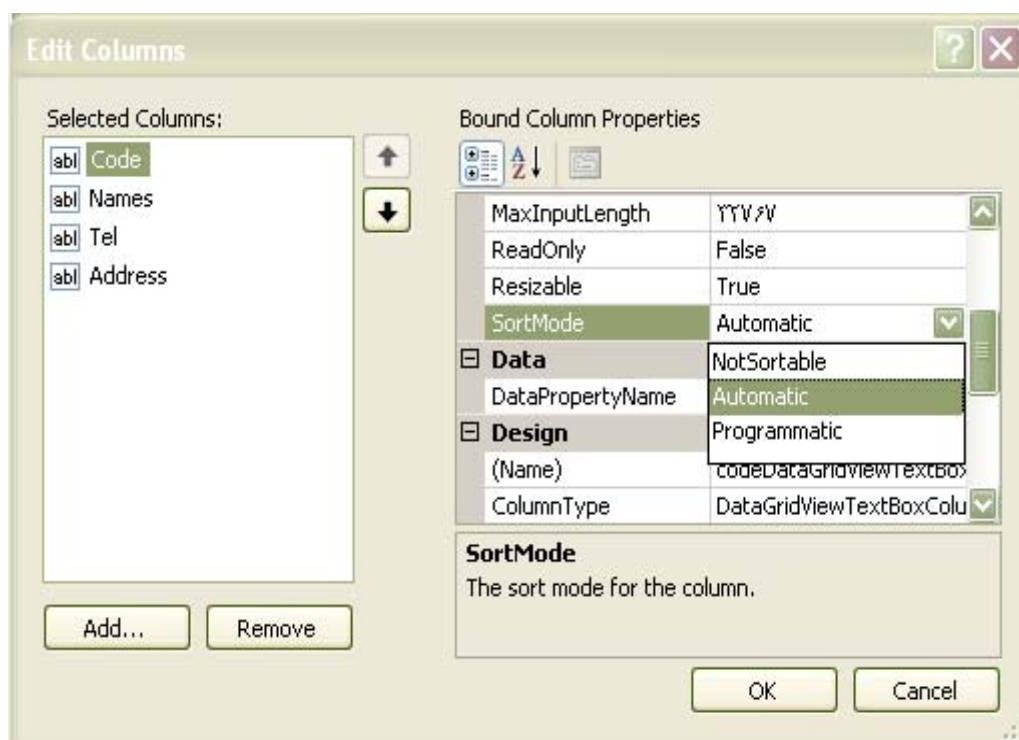
۴-۴-۶ - **MinimumWidth**: طول حداقل پهنای ستون را تعیین می کند .



۵-۴-۶-ReadOnly : یک ستون را فقط خواندنی می کند و کسی نمی تواند آن را Edit کند که اگر مقدار آن True شود ، یعنی فقط خواندنی است .

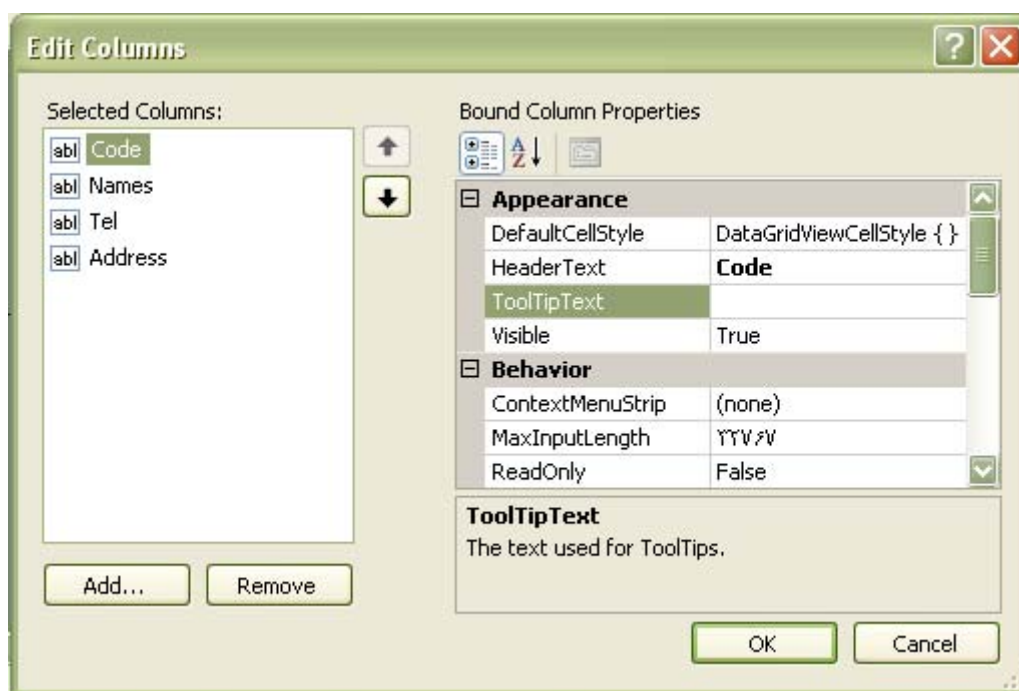


۶-۴-۶-SortMode : حالت‌های مختلف Sort را خواهیم داشت .



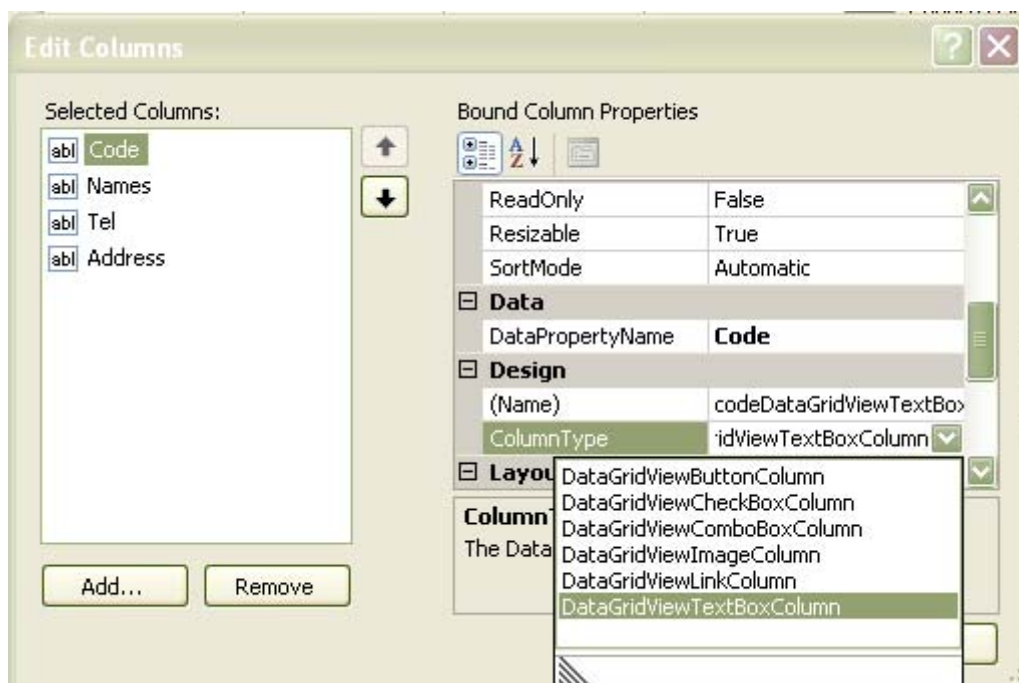
۷-۴-۶ - **ToolTipText** : متنی که به این صفت اختصاص داده شود در زمان قرار گرفتن ماوس بر روی

آن ظاهر می شود و پس از مدتی هم دیگر نمایش ندارد.

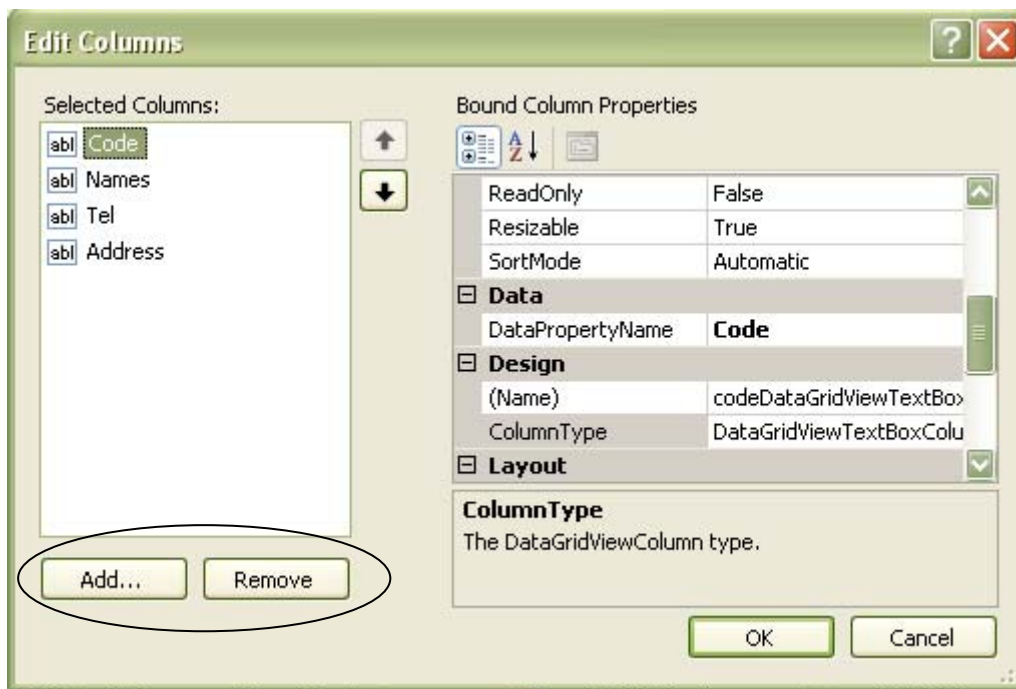


۸-۴-۶ - **ColumnType** : تعیین نوع عنصر برای نمایش ستون ، اینکه به صورت **Textbox** ،

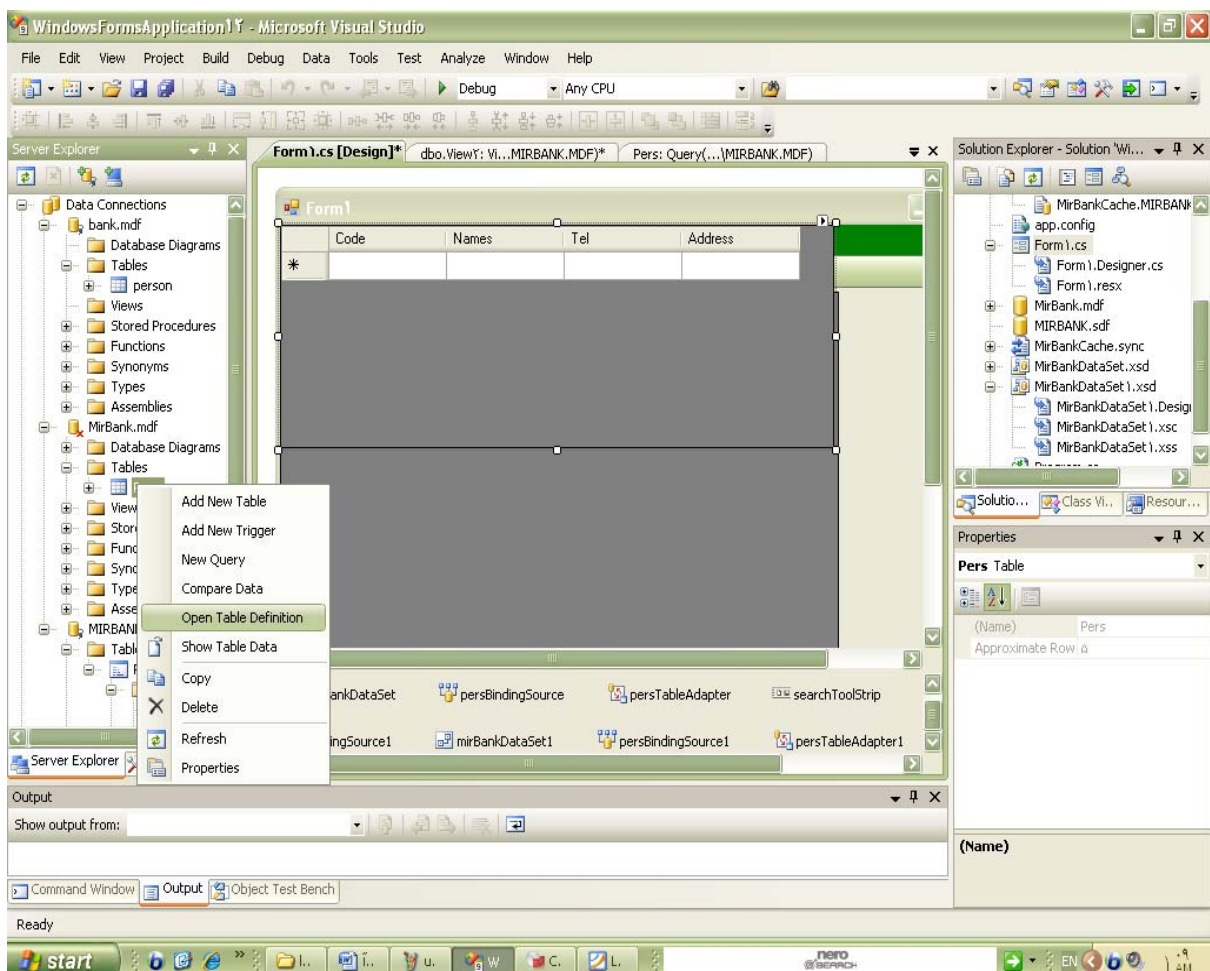
**Checkbox** ، **Combo box** و .... باشد را انجام می دهد و پیش فرض آن **Textbox** است .



با Add و Remove می توانیم ستون های دلخواه را کم یا زیاد کنیم .

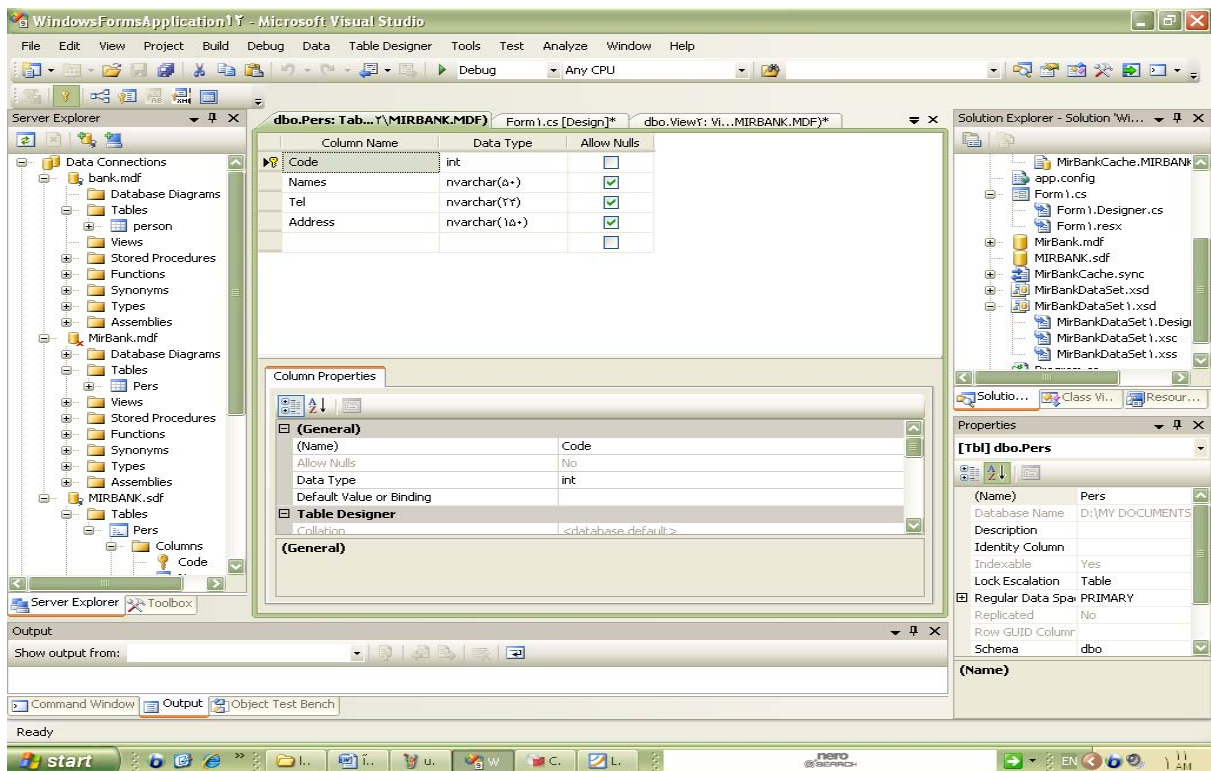


برای گرفتن گزارش در Tables مورد نظر گزینه Open Table Definition را انتخاب می کنیم .

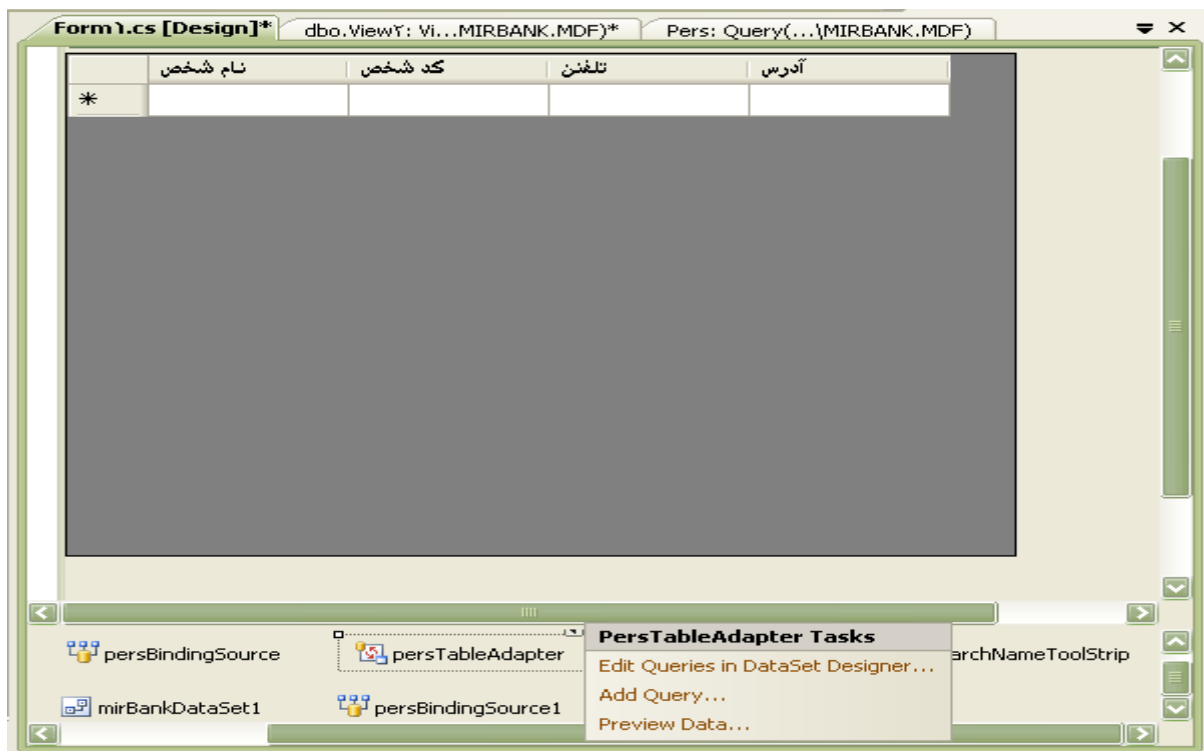


## ۵-۶- جستجو :

پنجره جدید امکان دیگری را در اختیار ما می گذارد،

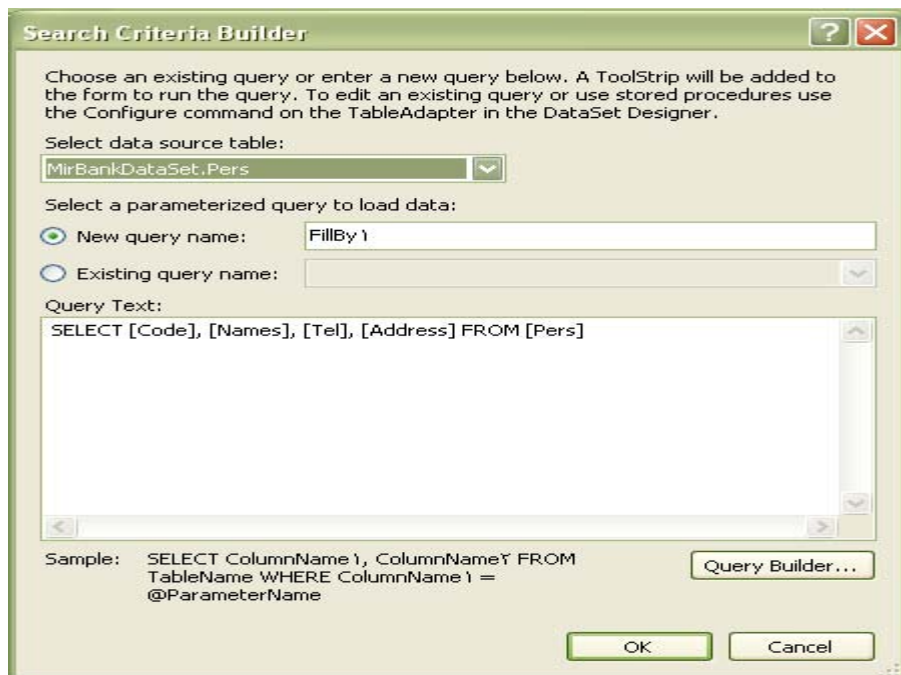


حال در فرم با کلیک بر روی نشانه گزینه PersTableAdapter Tasks و انتخاب Add Query..... محیط نوشتن Query وارد می شویم ،



در پنجره جدید به نام Search Criteria Builder در قسمت Query Text آن امکان Select نویسی

به شکل مورد نظر را برای ما فراهم می سازد.



مثلاً" برای اینکه بتوانیم براساس Code جستجو کنیم ، دستور زیر را خواهیم داشت :

```
SQL Search (@code)
SQL SearchName (@Name) SELECT [Code], [Names], [Tel], [Address] FROM [Pers] where code=@code
```

Select[Code],[Names],[Tel],[Address]FROM[Pers]where code=@code

و یا برای اینکه بتوانیم براساس Name جستجو کنیم ، دستور زیر را خواهیم داشت :

```
SQL Search (@code)
SQL SearchName (@Names) SELECT [Code], [Names], [Tel], [Address] FROM [Pers] where Names=@Names
```

Select[Code],[Names],[Tel],[Address]FROM[Pers]where Names=@Names

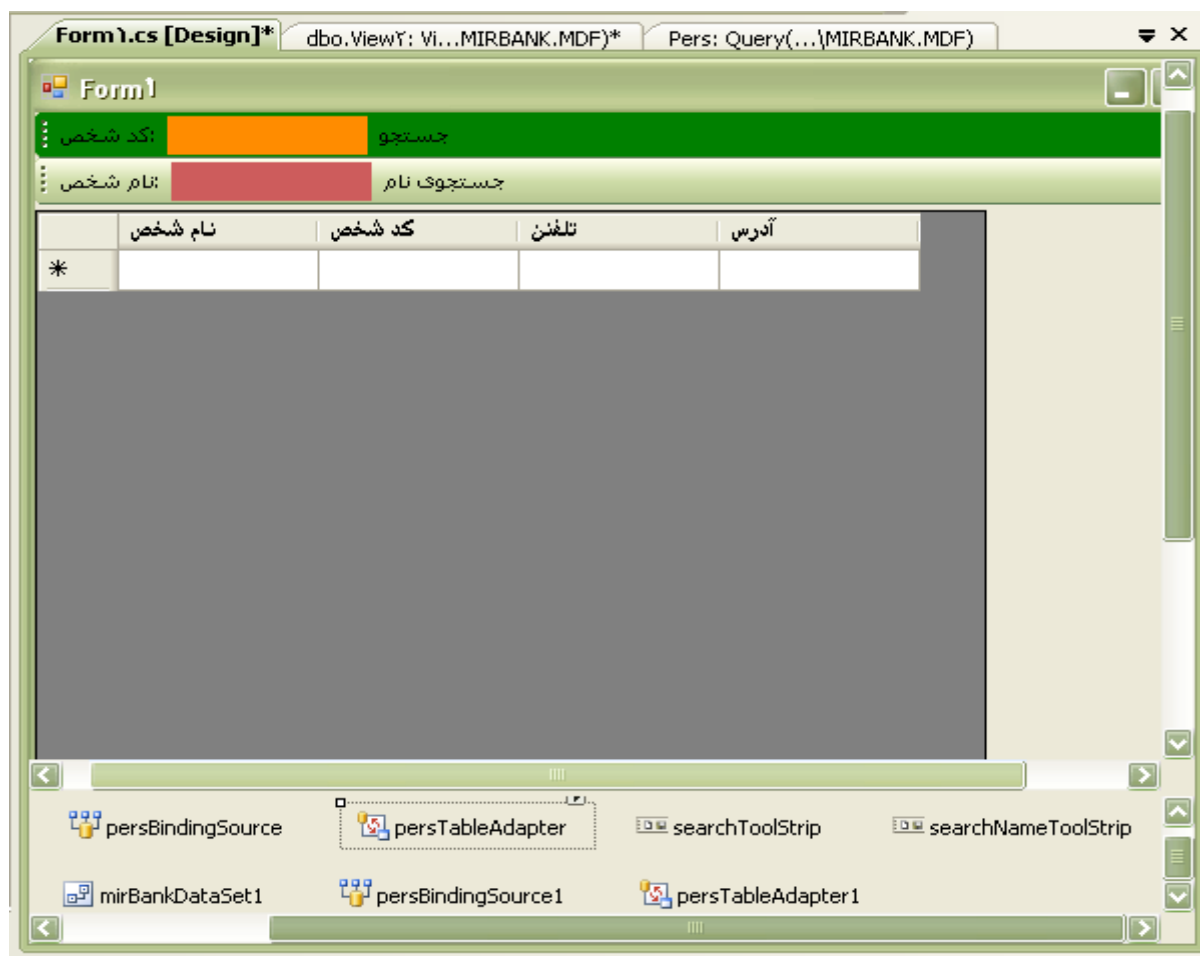
و یک نام هم برای Query مورد نظر انتخاب می کنیم .

در نهایت برای هر کد جستجو یک Box مربوط به آن جستجو برای فرم ما ایجاد می شود .





\*هرکجا که خواهیم می توانیم از زبان فارسی هم استفاده کنیم .



بنابراین برای جستجو کافیست که روی عنصر `persTableAdapter` کلیک و روی فلش بالای آن کلیک و در منوی ظاهر شده گزینه `Add Query` را انتخاب می کنیم تا پنجره `Search criteria Builder` ظاهر شود.

سپس `Select` مورد نظر را نوشته و دقت می کنیم که برای اعمال شرط متغیر در دستور `Where` از علامت `@` و سپس نام متغیر را استفاده می کنیم .  
مثلاً :

Select column name \,column name\from Table name where column name \=@ parameter name

سپس `Ok` می کنیم .

فونت ، رنگ ، سایز و سایر مشخصات به دلخواه در `BOX` های داده شده قابل تغییر و سایر قسمت ها هستند.





## فصل هفتم

# دسترسی به اطلاعات



## ۷-۱-۱ Component های دسترسی به اطلاعات :

می خواهیم در مورد Component های دسترسی به اطلاعات صحبت کنیم ،

مثل Dataset مثل Binding source و.....

در Visual C# برای دسترسی به اطلاعات و نمایش آنها ۳ ، Component مهم و اصلی وجود دارد که

عبارتند از :

Dataset

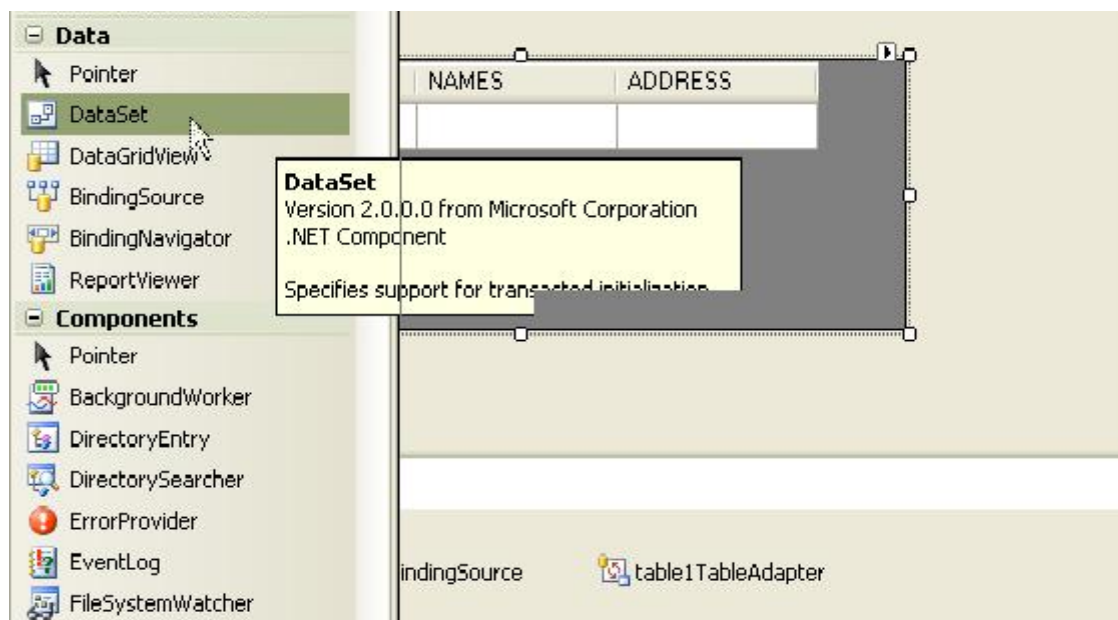
Binding source

TableAdapter &

دو Component مهم Dataset و Binding source در جعبه ابزار وجود دارند ولی برای داشتن

TableAdapter کفایت که براساس مسیری که برای دسترسی به اطلاعات درون بانک اطلاعاتی و

نمایش آن طی می کنیم پیش رویم ، تا بصورت اتوماتیک ایجاد شود.

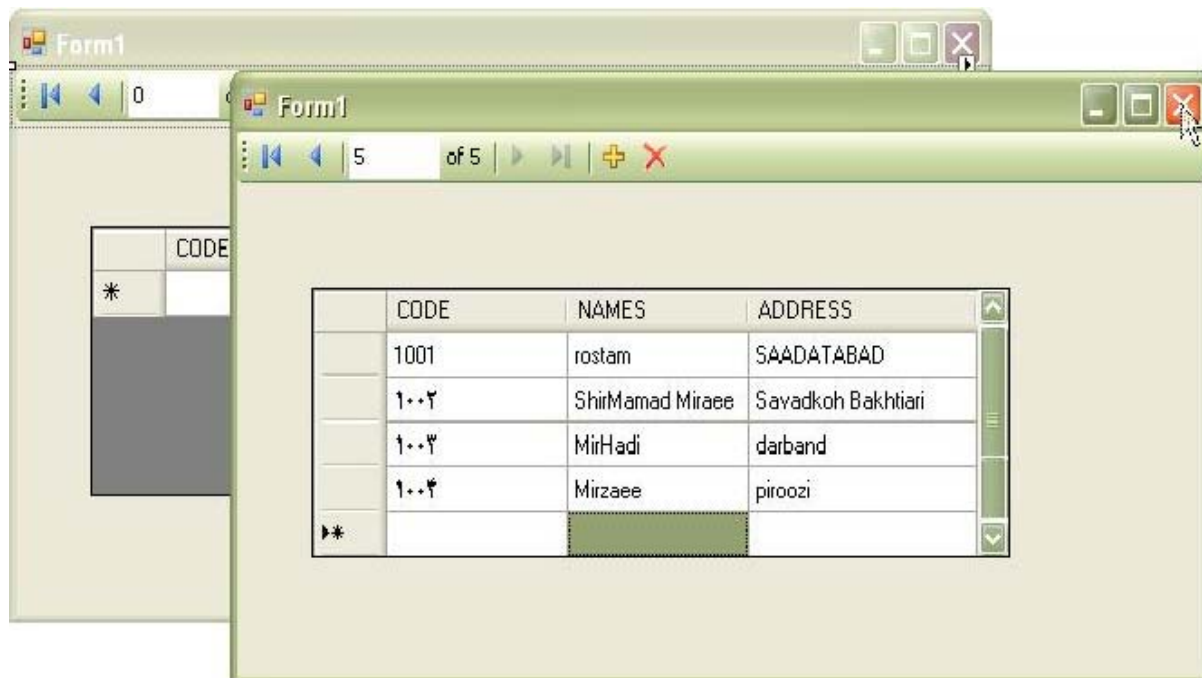


## ۷-۱-۱-۱ : Dataset

این کامپوننت همانند یک مخزن است که داده های مورد نیاز را در حافظه کامپیوتر نگهداری نموده و همانند

یک موتور بانک اطلاعاتی کوچک عمل می کند ، که داده های مورد نیاز را در حافظه نگهداری می کند .

علاوه بر توانایی ذخیره داده ها در جداول حجم زیادی از متا دیتا ( اطلاعاتی در مورد داده های موجود ) را نگهداری می کند.



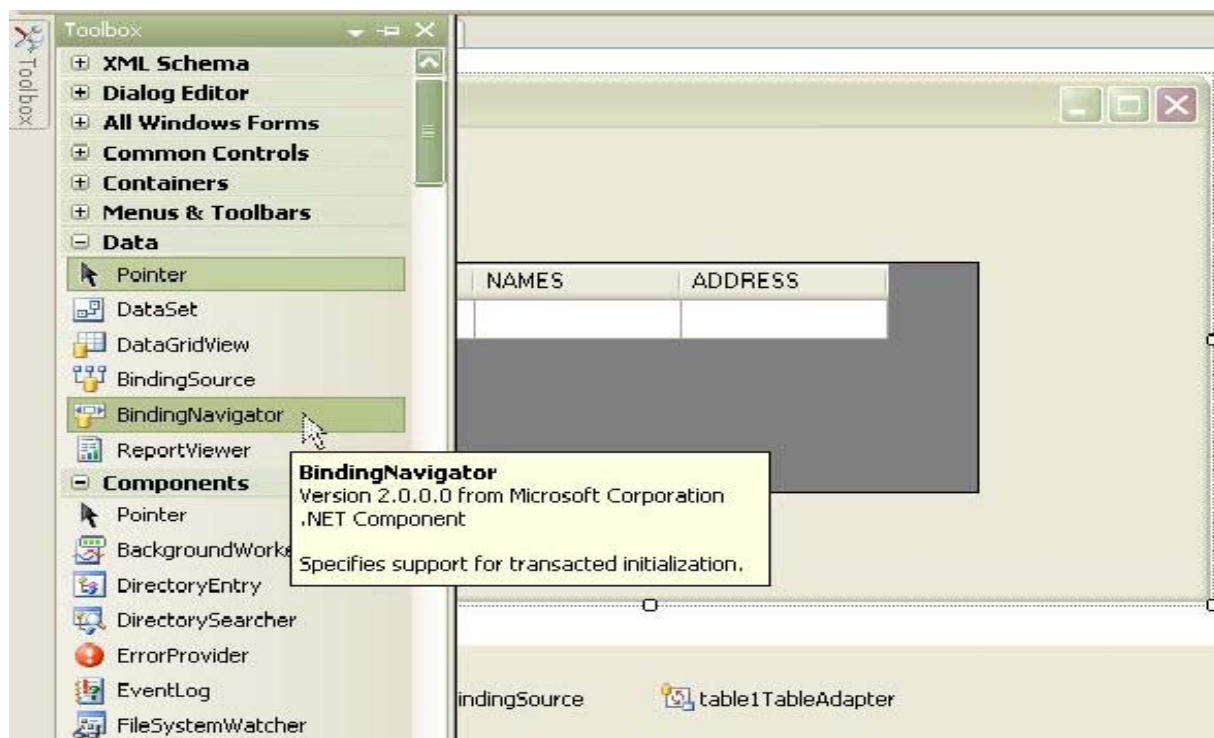
این اطلاعات شامل مواردی مانند : نام جداول و فیلدها ، نوع داده ها و..... می باشد.

نکته : تمام اطلاعات در قالب XML در حافظه نگهداری (ذخیره) می شود، یعنی می تواند به سادگی در قالب XML در دیسک ذخیره شده و یا از قالب XML از دیسک به حافظه آورده شود . همانطوریکه می دانیم به کمک XML می توان از طریق شبکه های مختلف مانند اینترنت اطلاعات را مبادله نمود.

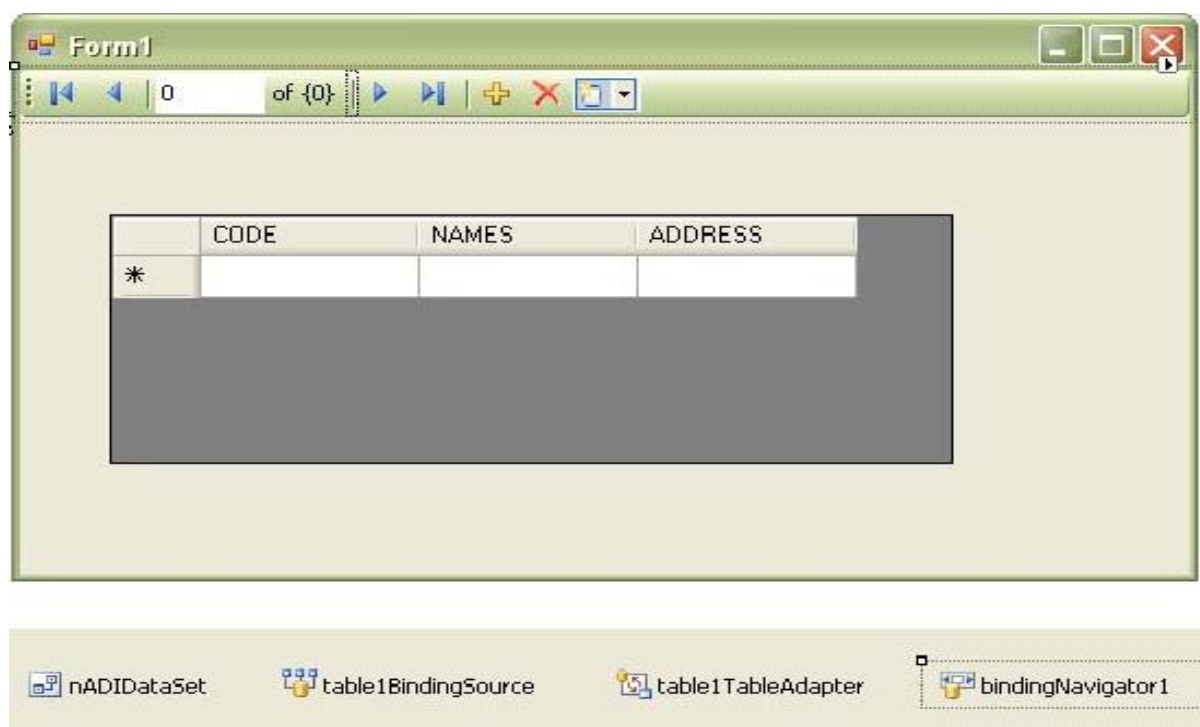
نکته : به علت اینکه داده های یک کامپونت Dataset در حافظه قرار دارند ، بنابراین به سادگی می توان بین آنها به جلو و عقب حرکت نموده و تغییرات ایجاد نمود ، که این تغییرات در داده های موجود در حافظه RAM اعمال می شود و تا زمانیکه مشخص نکنیم به داده های موجود در بانک اطلاعاتی منعکس نخواهد شد.

## :BindingNavigator - ۷-۱-۲

این عنصر جهت ناوبری ، راه بری و حرکت بر روی رکوردهای جدول بانک اطلاعاتی به کار می رود،

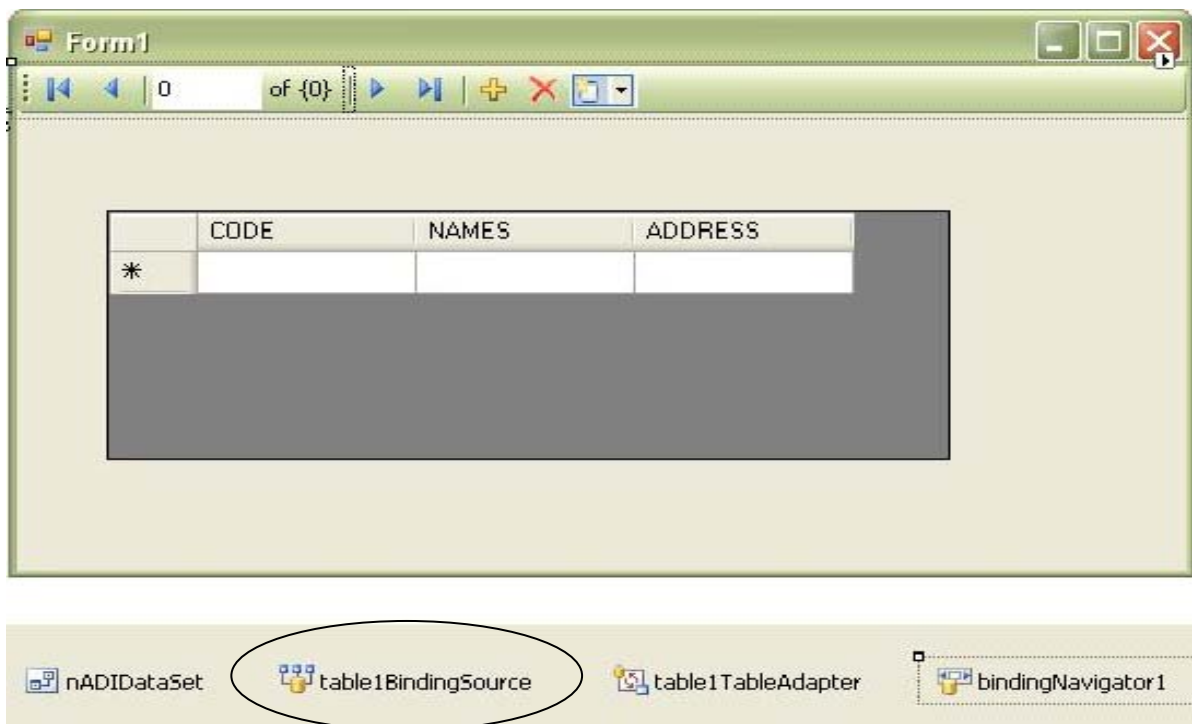


آن را روی فرم قرار داده و صفت Binding source آن را به عنصر Binding source مورد نظر مقدار دهی می کنیم .



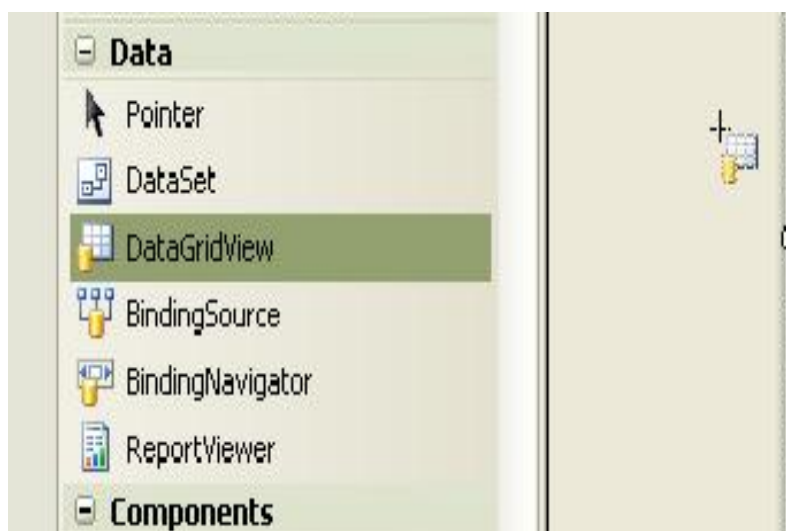
### ۳-۱-۷ : BindingSource

این Component ( که خودش یک عنصر است ) همانند پلی برای ایجاد ارتباط بین داده های موجود در منبع داده ای شما ( Dataset ) و نیز کنترل هایی که برای نمایش داده ها مورد استفاده قرار می گیرند به کار می روند.

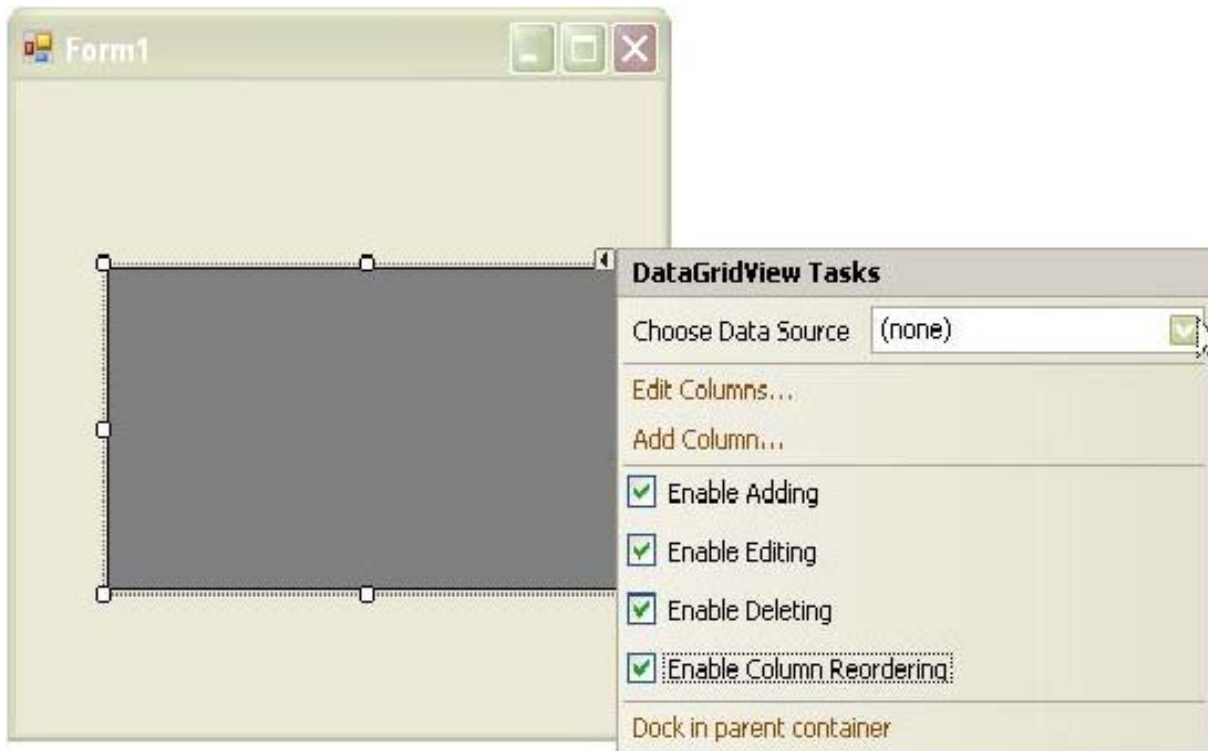


### ۴-۱-۷ : TableAdapter

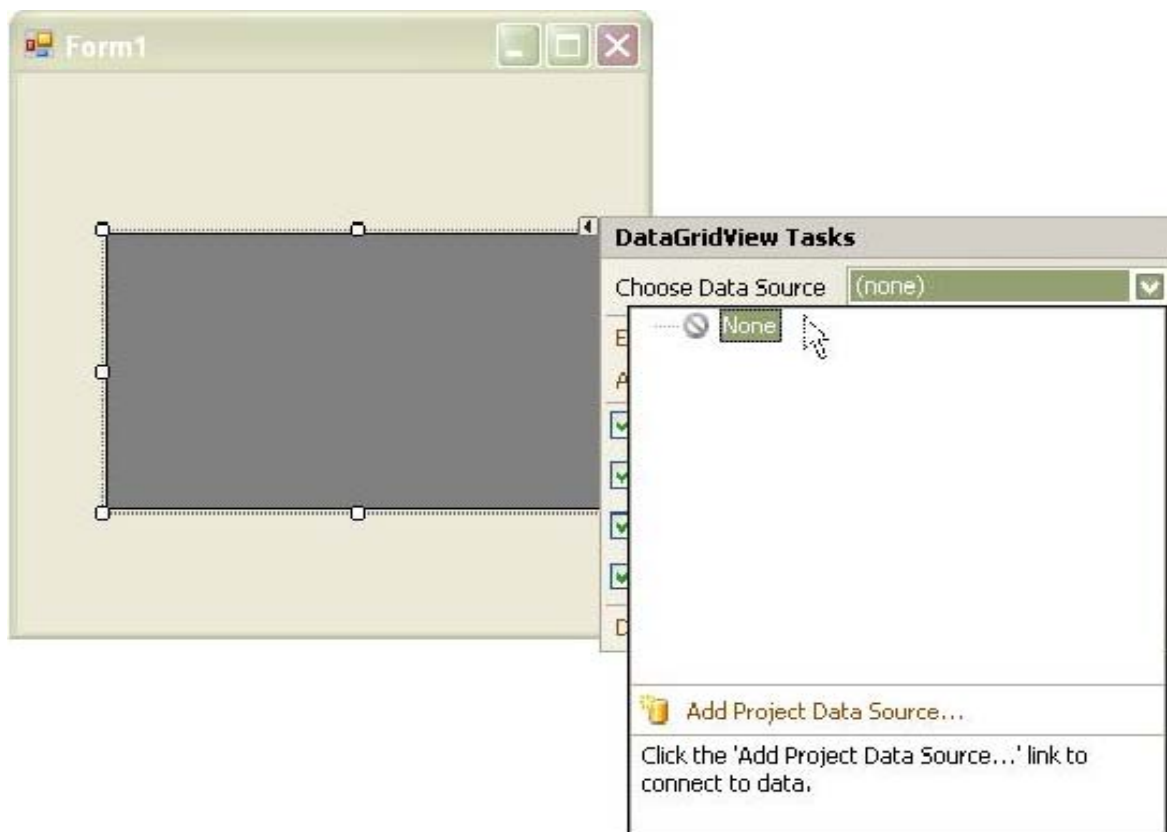
این Component به صورت اتوماتیک تولید می شود ، و مراحل ایجاد آن به ترتیب زیر است که پس از انتخاب DataGridView ،



و انتخاب Data Source مورد نظر ،

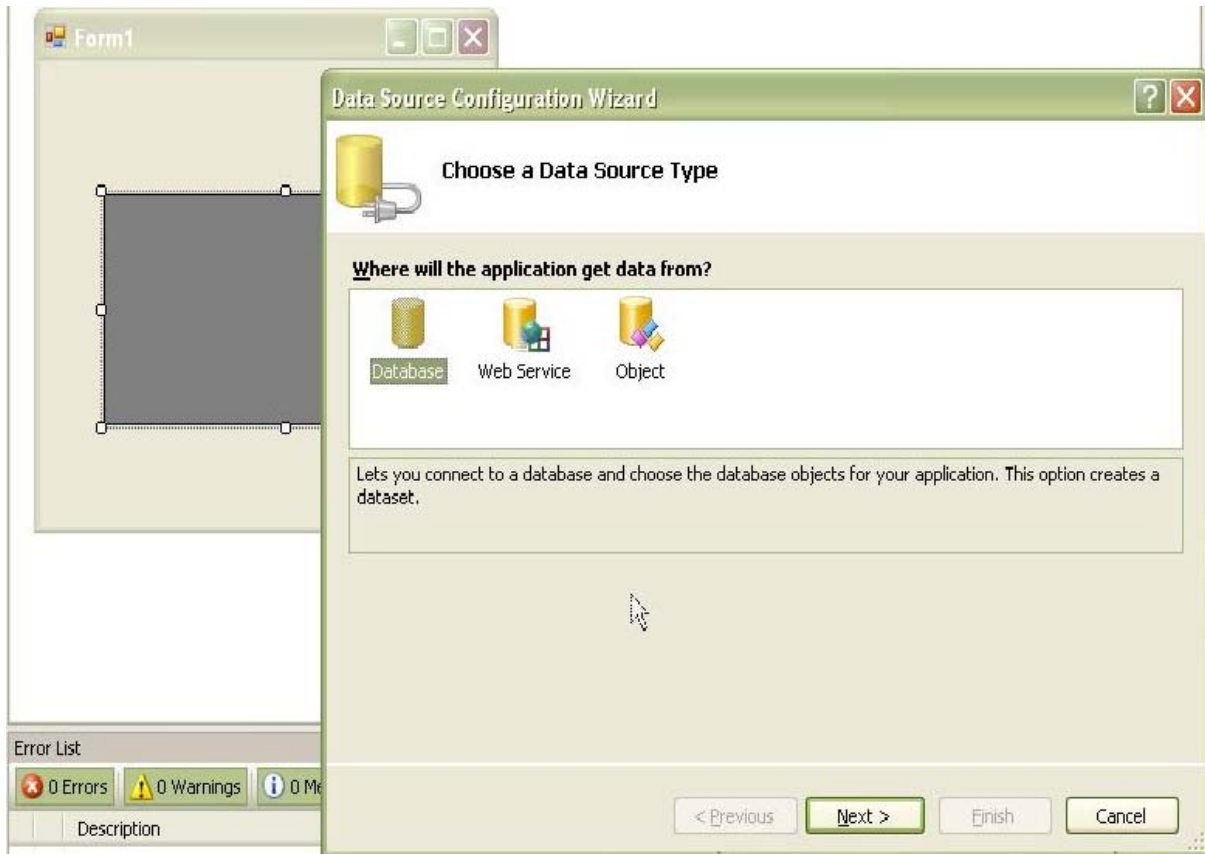


افزودن Data Source مورد نظر ،

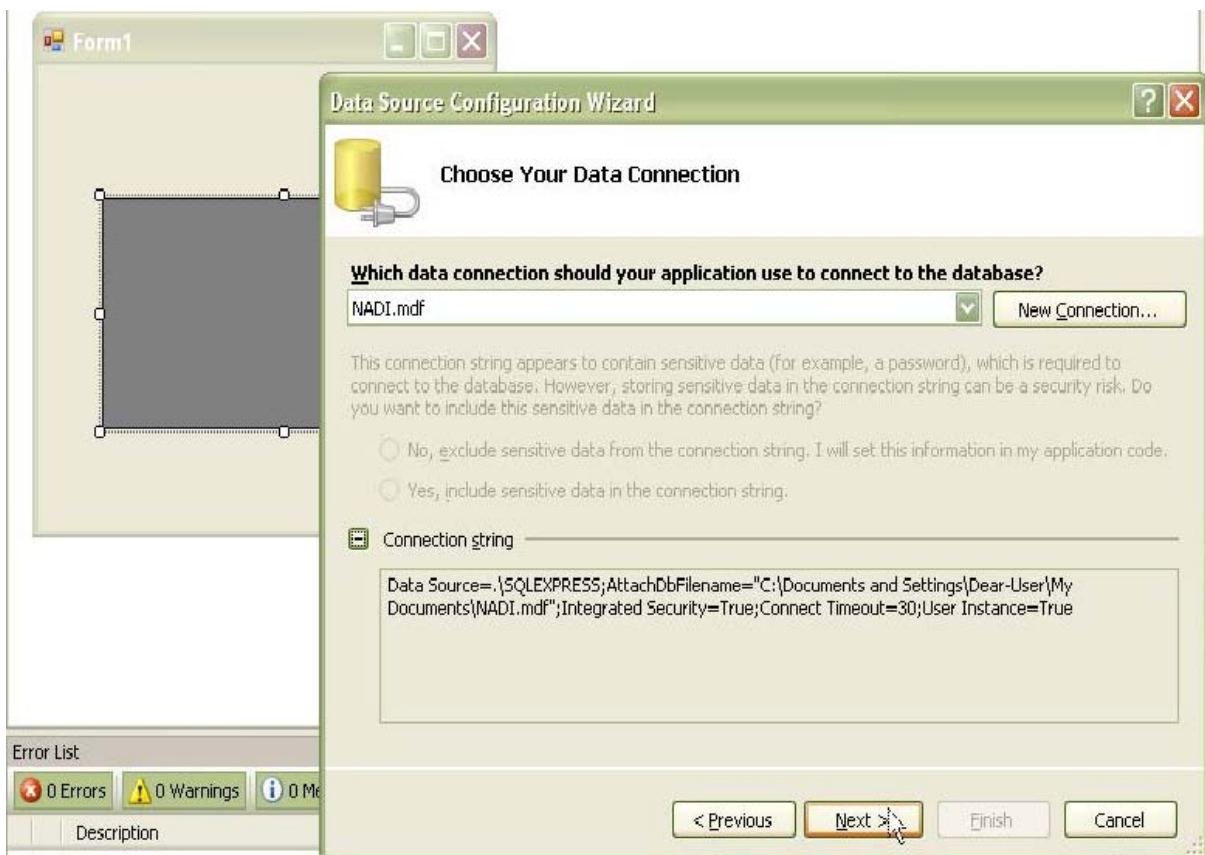




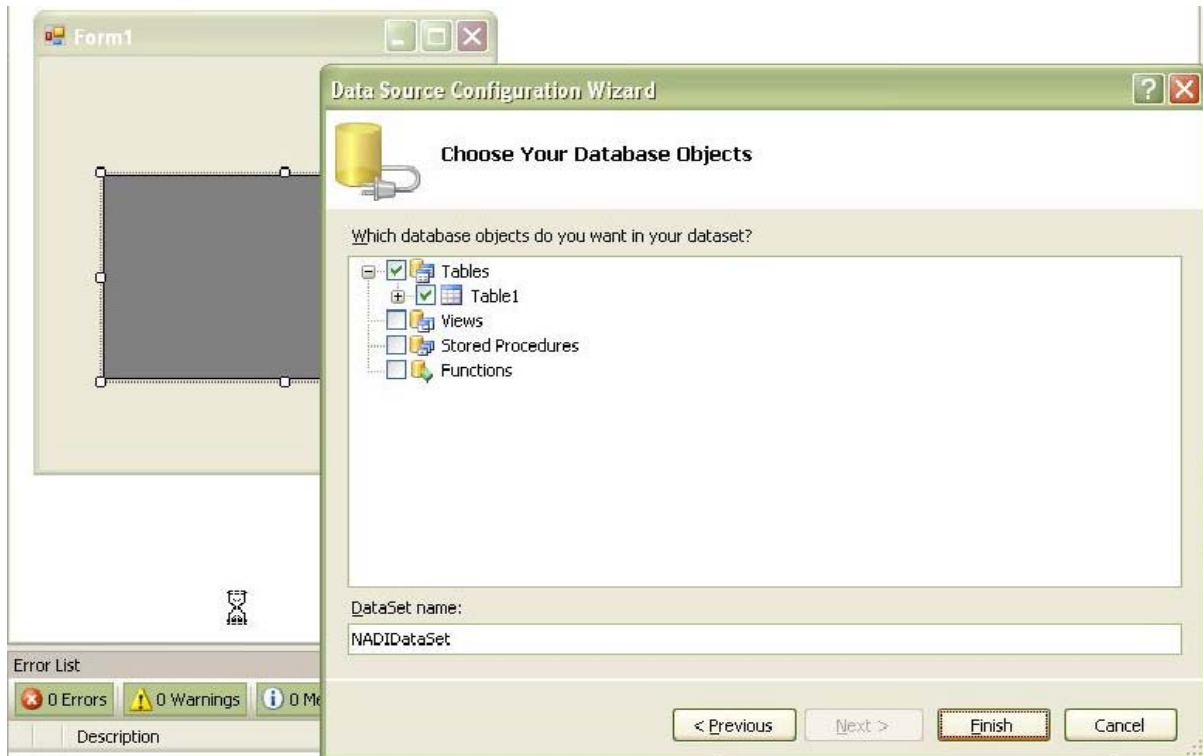
و انتخاب نوع آن ،



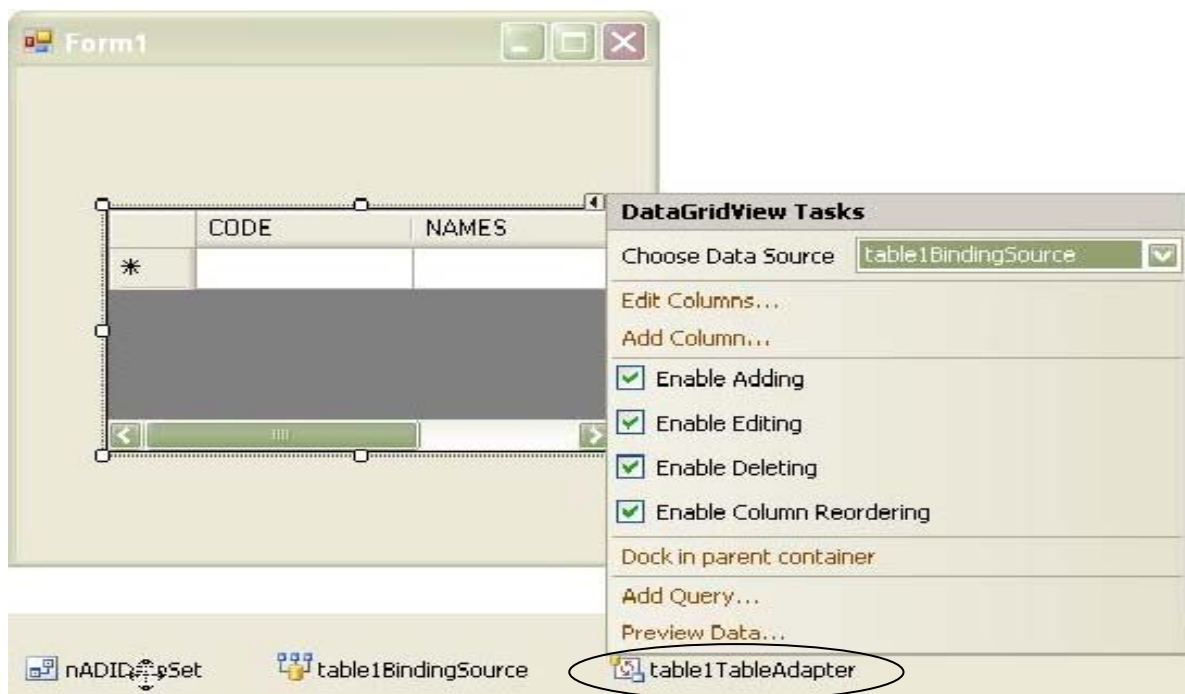
و با طی مراحل Wizard ،



و به پایان رساندن Wizard مربوط به آن ،



آیتم های : nADIDatadset ، table \BindingSource ، و table \TableAdapter به صورت خودکار ایجاد می شوند،

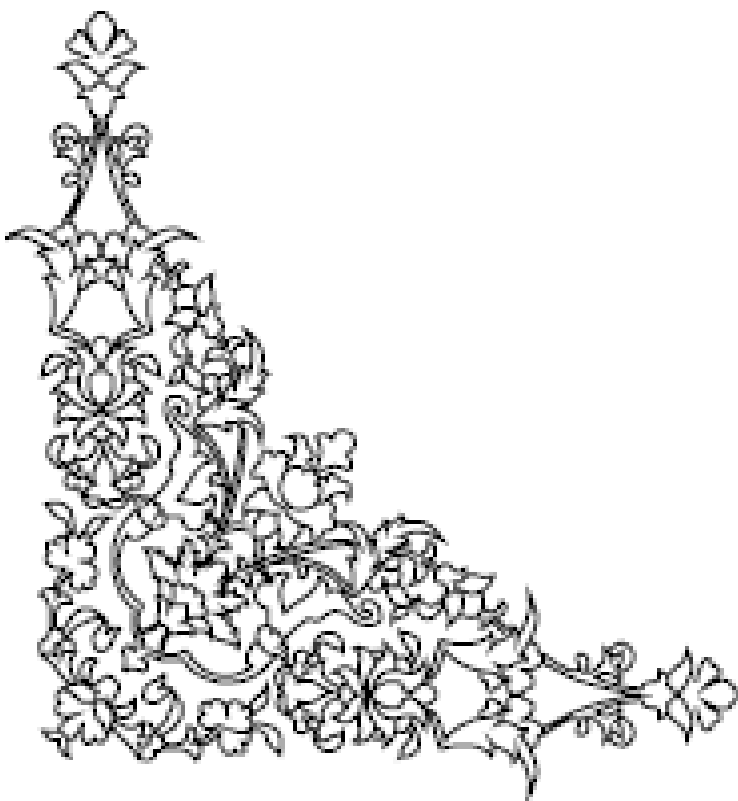


TableAdapter ، حاوی پرس و جوهایی برای انتخاب داده های موجود در بانک اطلاعاتی و نیز اطلاعاتی درباره نحوه اتصال برنامه به بانک است .



## فصل هشتم

# کلاس نویسی



## ۱-۸-ADO.NET :

به مجموعه Component هایی که برای دسترسی به داده های یک بانک اطلاعاتی در .NET استفاده می شود ADO.NET ( ActiveX Data Object ) گفته می شود که برای دسترسی به داده ها از معماری غیر متصل استفاده می کند .

\*معماری غیر متصل یعنی اینکه به بانک وصل می شود و عملیات را انجام می دهد و پس از آن ارتباط را قطع می کند .

حسن این کار این است که هم زمان تعداد زیادی به بانک وصل نیستند و تعداد Connection وصل به سرور به صورت هم زمان اضافه می شود و محدودیت های تعریف شده خللی در ارتباط با سرور برای دیگر کاربردها ایجاد نمی کند .

بنابراین معماری غیر متصل به این معناست که ابتدا برنامه به موتور بانک اطلاعاتی مورد نظر متصل شده و داده های مورد نیاز خود را از بانک اطلاعاتی دریافت نموده و آنها را در حافظه کامپیوتر ذخیره کرده ، سپس ارتباط قطع می شود و تغییرات مورد نظر را در داده های موجود در حافظه انجام می دهد و هر زمان که لازم باشد ، تغییرات ایجاد شده در بانک اطلاعاتی ذخیره می شوند ، برنامه یک اتصال جدید را به بانک اطلاعاتی ایجاد کرده و از طریق این اتصال تغییراتی را که در داده اعمال کرده بود را در جدول اصلی اعمال نموده و مجدداً ارتباط را قطع می کند .

نکته : استفاده از معماری غیر متصل مزایای زیادی دارد که مهم ترین آن افزایش توانایی برنامه در سرویس دادن به چندین کاربر بصورت هم زمان می باشد ، یعنی تعداد افرادی که می توانند به طور هم زمان از برنامه استفاده کنند را از دهها نفر به صد ها نفر و بیشتر افزایش می دهد .

## ۲-۸- کلاس نویسی :

کلاس های اصلی ADO.NET در فضای نام System.Data قرار دارد ، این فضای نام نیز خودش شامل چند فضای نام دیگر است که برخی از مهم ترین آنها عبارتند از System.Data که حاوی کلاس هایی است که برای دسترسی به بانکهای اطلاعاتی OLEDB از نوع OLE مانند Access می باشد .

SQL Client : این حالت شامل کلاس هایی است که برای دسترسی به بانکهای اطلاعاتی ساخته شده به وسیلهٔ SQL Server به کار می رود که خود این حالت دارای کلاس هایی می باشد ، مانند :

SQL Connection

SQL Command

SQL DataAdapter

.....و SQL Parameter

Oracle Client : شامل کلاس هایی است که برای دسترسی به بانکهای اطلاعاتی ساخته شده توسط Oracle می باشد .

Odbc : حاوی کلاس هایی است برای دسترسی انواع بانکهای اطلاعاتی ، خصوصا " بانکهای اطلاعاتی قدیمی تر که تکنولوژی OleDb را پشتیبانی نمی کنند به کار می رود .

### ۳-۸- کلاس های موجود در SQL Client :

SQL Connection

SQL Command

SQL DataAdapter

.....و SQL Parameter

SQL Connection : این کلاس وظیفه برقراری ارتباط بین برنامه و بانک اطلاعاتی را برعهده دارد، هنگامی که بخواهیم یک نمونه از این کلاس را ایجاد کنیم باید پارامتری به نام Connection String را به آن ارسال کنیم ، این پارامتر متغیری از نوع رشته ( String ) است که شامل تمام داده های مورد نیاز برای برقراری اتصال به یک بانک اطلاعاتی می باشد .

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

```
using System.Data;
```

```
SqlConnection objConnection = new
```

```
SqlConnection("Server=localhost;Database=Pubs;User" +
```

```
" ID=sa;Password=csdotnet;");
```

SQL Command : این کلاس ها فقط برای ارتباط با بانکهای اطلاعاتی SQL Server به کار می روند و برای ارتباط با بانکهای دیگر باید از کلاس های متناظر استفاده نمود .

```
SqlCommand objCommand = new SqlCommand();
```

#### ۸-۴- متصل شدن و قطع کردن اتصال به یک بانک اطلاعاتی :

برای این کار می بایست کد های زیر را اجرا کنیم :

```
// Open the database connection
objConnection.Open();
// ... Use the connection
objConnection.Close();
// Close the database connection
```

#### ۸-۵- خاصیت Parameter :

```
SqlConnection objConnection = new
SqlConnection("Server=localhost;Database=Pubs;User "
+ "ID=sa;Password=csdotnet;");
SqlCommand objCommand = new SqlCommand();
objCommand.Connection = objConnection;
objCommand.CommandText = "INSERT INTO authors " +
"(au_id, au_lname, au_fname, contract) " +
"VALUES(@au_id, @au_lname, @au_fname, @au_contract)";
```

و :

```
SqlConnection objConnection = new
SqlConnection("Server=localhost;Database=Pubs;User
ID=sa;Password=csdotnet;");
SqlCommand objCommand = new SqlCommand();
objCommand.Connection = objConnection;
objCommand.CommandText = "INSERT INTO authors " +
"(au_id, au_lname, au_fname, contract) " +
"VALUES(@au_id, @au_lname, @au_fname, @au_contract)";
objCommand.Parameters.AddWithValue("@au_id", txtAuId.Text);
objCommand.Parameters.AddWithValue("@au_lname",
txtLastName.Text);
objCommand.Parameters.AddWithValue("@au_fname",
txtFirstName.Text);
objCommand.Parameters.AddWithValue("@au_contract",
chkContract.Checked);
```

#### ۸-۶- خاصیت CommandText :

```
SqlConnection objConnection = new
SqlConnection("Server=localhost;Database=Pubs;User " +
"ID=sa;Password=csdotnet;");
SqlCommand objCommand = new SqlCommand();
objCommand.Connection = objConnection;
objCommand.CommandText = "INSERT INTO authors " +
"(au_id, au_lname, au_fname, contract) " +
"VALUES('۱۲۳-۴۵-۶۷۸۹', 'Barnes', 'David', \)";
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace persons {
    public class PersonDAL
    {
        public string CNNstr = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\\Documents and
Settings\\am_aghvami.iranet.net\\My Documents\\Visual Studio
۲۰۰۸\\Projects\\person\\PersonDB.mdf;Integrated
Security=True;Connect Timeout=۳۰;User Instance=True";
        public System.Data.SqlClient.SqlConnection CNN = new
System.Data.SqlClient.SqlConnection();
        public int Insert_Person(int PersonID, string
PersonName, string PersonLname)
        {
            CNN.ConnectionString = CNNstr;
            System.Data.SqlClient.SqlCommand cmd\ = new
System.Data.SqlClient.SqlCommand();
            cmd\.CommandType = System.Data.CommandType.Text;
            cmd\.CommandText = "insert into tbl_Persons
Values (@PersonID,@PersonName,@PersonLname) ";
            cmd\.Parameters.AddWithValue("@PersonID", PersonID);
            cmd\.Parameters.AddWithValue("@PersonName", PersonName);
            cmd\.Parameters.AddWithValue("@PersonLname", PersonLname);
            cmd\.Connection = CNN;
            CNN.Open();
            cmd\.ExecuteNonQuery();
            CNN.Close();
            return ۰;
        }
    }

```

```

        public int Update_Person(int PersonID, string
PersonName, string PersonLname)
        {
            CNN.ConnectionString = CNNstr;
            System.Data.SqlClient.SqlCommand cmd\ = new
System.Data.SqlClient.SqlCommand();
            cmd\.CommandType = System.Data.CommandType.Text;
            cmd\.CommandText = "Update tbl_Persons set
PersonName=@PersonName, PersonLname=@PersonLname Where
PersonID=@PersonID, ";
            cmd\.Parameters.AddWithValue("@PersonID", PersonID);
            cmd\.Parameters.AddWithValue("@PersonName", PersonName);
            cmd\.Parameters.AddWithValue("@PersonLname", PersonLname);
            cmd\.Connection = CNN;
            CNN.Open();
            cmd\.ExecuteNonQuery();
            CNN.Close();
            return ·;
        }
        public int Delete_Person(int PersonID)
        {
            CNN.ConnectionString = CNNstr;
            System.Data.SqlClient.SqlCommand cmd\ = new
System.Data.SqlClient.SqlCommand();
            cmd\.CommandType = System.Data.CommandType.Text;
            cmd\.CommandText = "Delete From tbl_Persons where
PersonID=@PersonID) ";
            cmd\.Parameters.AddWithValue("@PersonID", PersonID);
            cmd\.Connection = CNN;
            CNN.Open();
            cmd\.ExecuteNonQuery();
            CNN.Close();
            return ·;
        }
    }
}

```



در فرم بر روی دکمه ها insert , delet

```
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;
namespace person
{
    public partial class frmMain : Form
    {
        public persons.PersonDAL objPerson = new
persons.PersonDAL();
        public frmMain()
        {
            InitializeComponent();
        }
        private void btnOK_Click(object sender, EventArgs e)
        {
objPerson.Insert_Person(System.Convert.ToInt32(txtPersonID.Text), txtPersonName.Text, txtPersonLName.Text);
        }
        private void btnDelete_Click(object sender, EventArgs e)
        {
objPerson.Delete_Person(System.Convert.ToInt32txtPersonID.Text));
        }
        private void btnupdate_Click(object sender, EventArgs e)
        {
objPerson.Update_Person(System.Convert.ToInt32(txtPersonID.Text), txtPersonName.Text, txtPersonLName.Text);
        }
    }
}
```

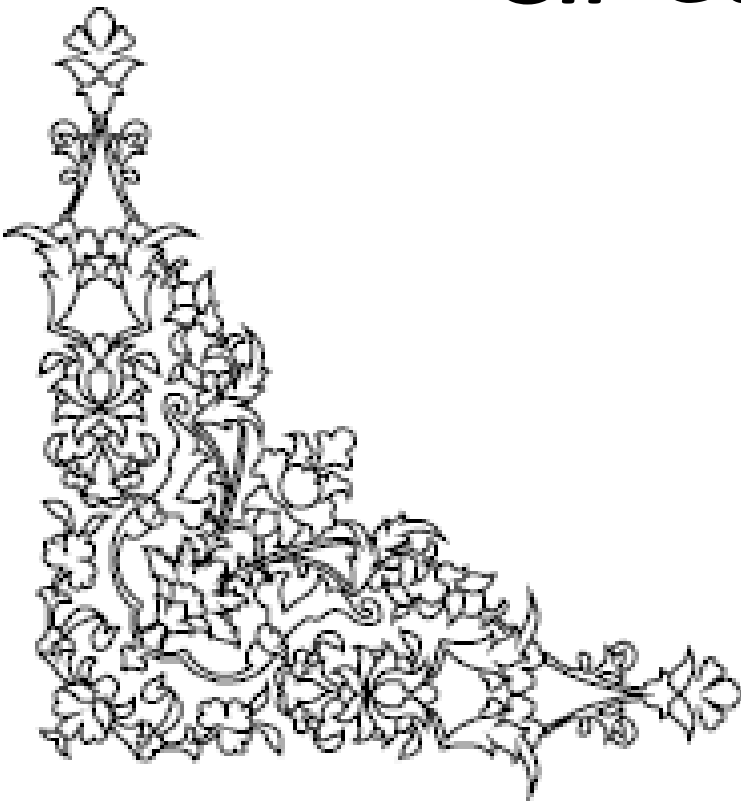




فصل نهم

مزایا و معایب

محیط C#



## ۹-۱- مزایای Visual C# :

- کاملاً رایگان ( Freeware )
- نسخه کامل و نصب به صورت آفلاین ( offline ) - بر خلاف مواردی که فایل کم حجم نصب دانلود شده و نصب به صورت آنلاین انجام می شود-
- شباهت بسیار زیاد به محیط Studio Visual
- رابط کاربری بسیار قدرتمند
- قابلیت Intellisense

( نمایش لیستی از کلمات با تایپ چند حرف اول کلمه و زدن Ctrl+Space )

- امکان باز کردن پروژه های ایجاد شده بوسیله Visual Studio ( زبان C# )
- امکان نصب Microsoft MSDN Express Edition و Microsoft SQL Server
- Edition Express ( قابل نصب به صورت اختیاری )

این زبان برنامه نویسی به عنوان یک زبان قدرتمند در زمینه وب نیز می تواند نقش آفرین باشد. کاربران و استفاده کنندگان محصولات میکروسافت این زبان را به عنوان اولین و قدرتمند ترین زبان برنامه نویسی مبتنی بر وب انتخاب نموده و اکثر وب سایتهای طراحی شده به صورت حرفه ای با این زبان طراحی شده اند. بهره مندی از مزایای تکنولوژی شی گرای (Object oriented) و شباهت این زبان به زبان ++C از دیگر دلایل توجه و اقبال به این زبان برنامه نویسی است.

## ۹-۲- معایب Visual C# :

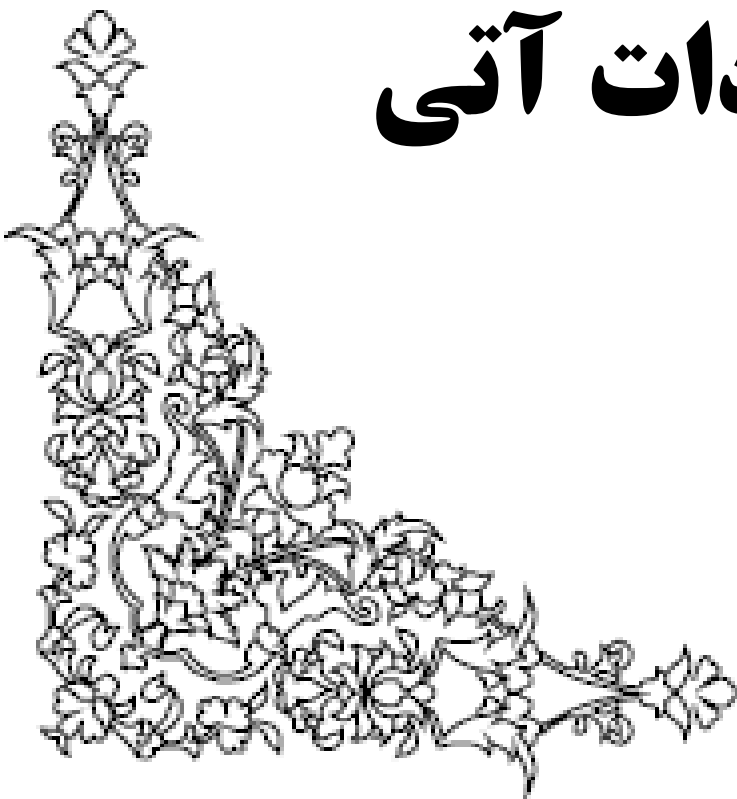
از آنجائیکه این محیط نرم افزاری نسبت به محیط های دیگر مشابه از توانمندیها و قابلیت های بسیاری برخوردار است و جذابیت های ویژه ای دارد که آن را بسیار بسیار کارآمد تر از موارد مشابه نموده است ، لذا از پرداختن به معایب آن صرفنظر می کنیم . یکی از معایب آن عدم امکان برنامه نویسی MFC می باشد.



## فصل دهم

# خلاصه، نتیجه گیری

# پیشنهادات آتی



## ۱-۱- خلاصه :

همانطوریکه گفتیم نرم‌افزار Visual Studio، یک مجموعه از برنامه‌هایی است که ارتباط بسیار نزدیک با هم دارند و MicroSoft آن را به توسعه دهندگان و برنامه نویسان برنامه‌های کاربردی اهدا نمود تا آنها را وادار نماید در محیطی توسعه یافته بر روی پلت فرم‌های ویندوز و دات نت به ساخت برنامه‌های خود بپردازند .

Visual Studio می‌تواند برای نوشتن برنامه‌های کنسولی، ویندوزی، سرویس‌های ویندوز، برنامه‌های کاربردی موبایل، برنامه‌های کاربردی ASP.NET و سرویس‌های وب ASP.NET بنا به انتخاب شما همراه با زبان‌هایی مانند C++, C#, VB.NET, J# استفاده شود . اینکه با Visual Studio واقعا" چه کارهایی می‌توان انجام داد ؟ و چه استفاده‌هایی می‌توان از آن کرد و چگونگی انجام آنها مواردی بود که مختصرا" به آن پرداختیم .

همچنین با معرفی ویژوال استدیو گفتیم که این برنامه دارای ابزاری قدرتمند برای پیاده سازی برنامه های ASP. NET است . برخلاف نسخه قبل که صرفا" مختص یک نسخه خاص frame work بود ( به عنوان نمونه در ویژوال استودیو ۲۰۰۳ ، نسخه ۱.۱ ASP.NET و در ویژوال استودیو ۲۰۰۵ نسخه ASP. NET ۲.۰ ) ، از ویژوال استودیو ۲۰۰۸ می‌توان به همراه چندین نسخه frame work استفاده کرد .

در ویژوال استودیو ۲۰۰۸، در محیط طراحی نیز تغییراتی در جهت قدرتمند تر شدن آن ایجاد شده است . افزودن امکاناتی نظیر هوشمندی در تایپ و اشکال زدائی کدهای جاوا اسکریپت و قابلیت مشاهده و حتی توقف در هسته کد frame work دات نت در حین اشکال زدائی نمونه هائی در این زمینه می باشند .

در یک نگاه اجمالی از نظر رابط کاربری و قابلیت‌های کاربردی Visual Studio ، خود تحول زیادی داشته و در همین راستا تعداد زیادی از کنترل‌های جدید مربوط به بانک اطلاعاتی به این نرم‌افزار اضافه شده است. البته با توجه تجاری که در استفاده از این ابزارهای جدید داشته‌ایم به نظر می‌رسد این کنترل‌ها و ابزارهای جدید در قسمت Windows Application به نسبت Web Application ها ، بسیار پخته‌تر و کامل‌تر می‌باشد.

## ۲-۱۰- نتیجه گیری :

با استفاده از این آموزش گام به گام می توانیم با بهره گیری از امکانات محیط Visual Studio برنامه های مورد دلخواهمان را که از Data base هم برخوردار هستند تهیه کنیم . برنامه نویسان در سطوح ابتدایی با داشتن اندکی اطلاعات در مورد محیط های مشابه می توانند برنامه هایی را که می خواهند طراحی و از امکانات وب آن استفاده و در وب قرار دهند. تهیه کتابخانه الکترونیکی (پیوست ۴) ، تهیه سایت های ثبت نام ، سایت های خرید ، اطلاعات آنلاین ، ..... و بسیاری از برنامه های دیگر با بهره گیری از این محیط به راحتی انجام پذیر است . آنچه آموختیم خلاصه ای بود از چگونگی ایجاد پایگاه های داده ، دسترسی به آن و ایجاد مبانی و بسترهای ابتدایی برای ورود اطلاعات و زیباسازی و طراحی محیط و در برخی قسمت ها انجام عملیات محاسباتی .

## ۳-۱۰- پیشنهادات آتی :

چند سالیست که میکروسافت بر روی چند تکنولوژی جدید از جمله WCF ، WF ، WPF ، و LINQ و ... کار می کند که این اواخر توضیحات و تبلیغات بر روی این تکنولوژیها را به اوج رسانده است.

در نسخه هایی از ویژوال استودیو، زبان LINQ به نسخه جدید زبانهای C# و Basic Visual اضافه شده است.

تکنولوژی LINQ به کمک آندسته از برنامه نویسانی که سر و کار زیادی با کارکردن با داده های جدول گونه دارند، آمده است. از این پس برنامه نویسان می توانند به کمک LINQ آنگونه که به نوشتن Query code ها در محیط های SQL و غیره می پرداختند، در سورس کد برنامه خود و تحت NET. بر راحتی از مزایای آن استفاده کنند.

زبان LINQ دارای ساختاری شبیه به زبان SQL می باشد. LINQ به ما اجازه می دهد که با زبان های VB و C# کوئری های خود را بنویسیم. ویژوال استودیو ۲۰۰۸ به برنامه نویسان امکان می دهد تا با استفاده از ابزارهای بسیار پیشرفته برنامه نویسی راحت تر از قبل برنامه هایی با کارایی بالا تهیه کنند.

با استفاده از ویژوال استودیو ۲۰۰۸ ابزارهای متنوعی در اختیار برنامه نویسان قرار می گیرد که می توان با استفاده از آن ابزارها، برنامه های تحت وب و ویندوز را به راحتی تولید کرد.

یکی دیگر از قابلیت‌های ویژوال استودیو ۲۰۰۸ امکان انتخاب فریم‌ورک‌های مختلف است.

در نسخه‌های قبلی، تنها امکان استفاده از یک فریم‌ورک وجود داشت، ولی همان‌طور که در شکل مشاهده می‌کنید، در این نسخه می‌توانید فریم‌ورک‌های مختلف را انتخاب کنید.

در ویژوال استودیو ۲۰۰۸ امکان دیدن **Split view** وجود دارد که به برنامه‌نویسان اجازه می‌دهد سورس‌کدهای HTML را به همراه طرح صفحه (**Design view**) همزمان مشاهده کنند (کاربرانی که با نرم‌افزارهایی مانند **Front Page** و یا **Dream Viewer** کار کرده باشند با **Split view** آشنایی دارند).

همچنین در نسخه‌های جدید ویژوال استودیو ابزار جدیدی در IDE به نام **Manage Styles** وجود دارد که تمام **Style**های **CSS** در یک صفحه را مدیریت می‌کند.

یکی دیگر از امکاناتی که برای برنامه‌نویسان در ویژوال استودیو قرار دارد و بسیار هم مفید است پشتیبانی این نسخه، از **Using Statement**ها در **C#** است.

با استفاده از این قابلیت می‌توانیم چند **Statement Using** را انتخاب کرده و با راست کلیک کردن، می‌توانیم این **Statement**ها را مرتب و **Using Statement**هایی را که در برنامه استفاده نشده‌اند، حذف کنیم.

از دیگر امکانات جدیدی که می‌توان در این نسخه از ویژوال استودیو مشاهده کرد، امکان تولید برنامه‌های تحت وب جذاب با استفاده از فناوری **Ajax** و **ASP.NET** است. با دو کلیک روی کنترل HTML مانند **Button**، یک **Click Event** به صورت خودکار تولید می‌شود و همزمان، یک **Skelton** از **Function** جاوا اسکریپت نیز تولید می‌شود.

با اطلاع از امکانات ویژه بالا و همچنین با مروری بر دانسته‌هایمان، ما در قدم بعدی می‌توانیم دستورات **Edit**، **Sort**، ..... و پیمایش را مورد بررسی قرار داده و به انجام برسانیم.

می‌توانیم از خود ویژوال استودیو برای ساخت توابعی جدید و قابل اضافه شدن به خود ویژوال استودیو استفاده نماییم. کاربردهای دیگر ویژوال استودیو همچنین شامل پروژه‌هایی برای توسعه برنامه‌های کاربردی، کار با **databases**، ساخت گزارشها و ... می‌باشد.



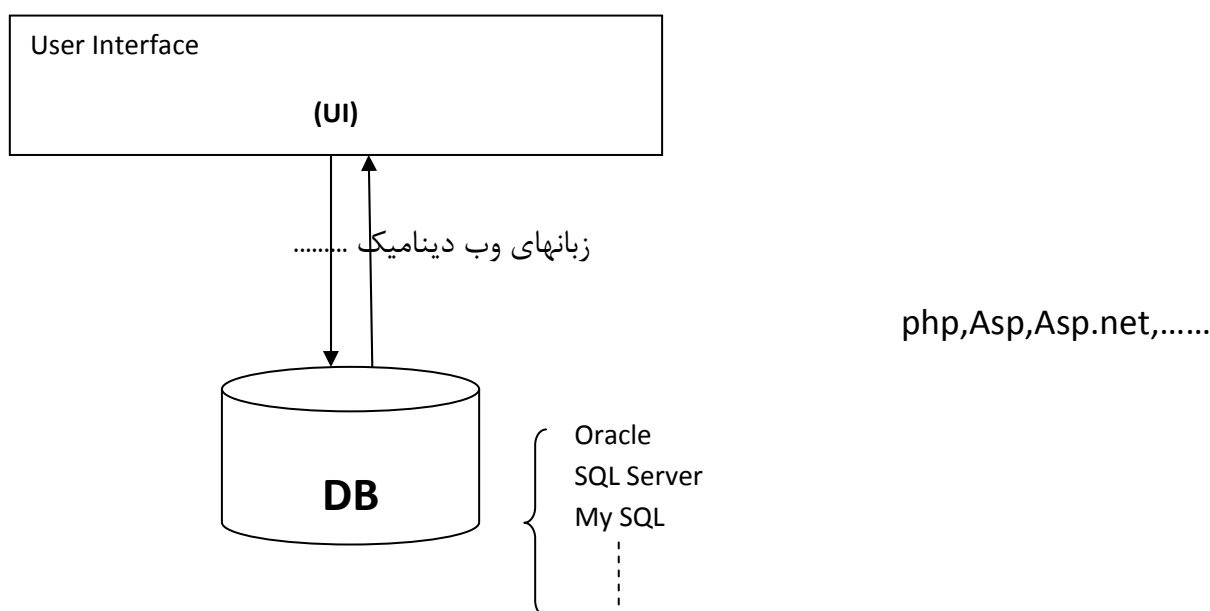
## پیوست ۳ : C++

### معماری لایه ای :

#### معماری نرم افزارهای بانک اطلاعاتی تحت وب از گذشته تا کنون

- معماری نرم افزارهای بانک اطلاعاتی تحت وب در گذشته

توسط Http روی کامپیوتر کاربر قرار می گیرد



در این نوع معماری Security پایین است و به راحتی می توان به اطلاعات موجود در بانک دسترسی داشت

حمله های SQL ای و هک کردن به راحتی انجام می شود.

#### • منظور از لایه در برنامه نویسی چیست ؟

برای پاسخ به این سؤال یک پروژه تحت وب را در نظر بگیرید ، در این پروژه قرار است یک عملیات بانک اطلاعاتی مانند درج و به روز رسانی و دریافت اطلاعات را انجام شود ، میتوان تمامی کدهای مربوط به این عملیات را در یک صفحه (CodeBehind) نوشت ویا میتوان این عملیات را به چند بخش تقسیم کرده و آنها را بر مبنای خاصی دسته بندی نمود ، این مبنا را لایه می نامند .

## • مبنای دسته بندی چیست و این دسته بندی به چه صورت انجام میشود؟

باید دانست که یک روش ثابت برای دسته بندی وجود ندارد ولی برنامه نویسان و توسعه دهندگان طی تجربه به این نتیجه رسیده اند که یک پروژه را به سه یا چهار یا... لایه تقسیم کنند ، لایه اول Presentation Layer یا User Interface نام دارد ، این لایه نهایی ترین لایه است و در دسترس کاربر نهایی قرار می گیرد ، کمترین کد در این لایه نوشته می شود (مثلا Default.aspx و Default.aspx.cs جزئی از این لایه هستند)

لایه دیگری به نام Data Access Layer (یا DAL) وجود دارد ، این لایه ارتباط با دیتابیس را برقرار می کند ، تمامی کد های مربوط به عملیات بانک اطلاعاتی (دریافت ، درج ، حذف و به روز رسانی و...) و کد های مربوط به ارتباط و قطع ارتباط در این لایه پیاده سازی می شود .

لایه ای به نام Business Logic Layer ( یا BLL ) نیز وجود دارد که بین DAL و UI قرار میگیرد ، این لایه وظایف متعددی را میتواند ایفا کند ، چک کردن شرط ها ، برخی اعتبار سنجی ها ، فراخوانی متدهای DAL و همچنین Mapping OR (مپ کردن فیلدهای بانک اطلاعاتی به کلاس) و ... را می توان از وظایف این لایه نام برد .

لایه دیگری نیز میتوان به لایه های فوق اضافه کرد ، Cache Layer لایه ذخیره اطلاعات در حافظه کش است ، همانطور که میدانید کش از حافظه کوتاه مدت استفاده می کند بنابراین بسیار سریعتر از دیتابیس( که بر روی هارد دیسک ذخیره می شود ) قابل دسترسی است . برخی اطلاعات پر استفاده که نیازی ندارد به روز باشند را میتوان در cache ذخیره کرد . این لایه بین BLL و UI قرار میگیرد تا UI به جای استفاده از اطلاعات تازه از cache اطلاعات را واکشی نماید .

میتوان لایه های دیگری را نیز بر حسب نیاز اضافه کرد و یا از برخی لایه های فوق استفاده نکرد ، هیچ الزامی وجود ندارد .

برخی از افراد خود دیتابیس را هم یک لایه در نظر میگیرند و برخی خیر ، بنابراین اگر اسمی از برنامه نویسی سه لایه یا چهارلایه یا پنج به میان آمد باید دانست که به کمی توضیح بیشتر نیاز دارد .

معمول ترین روش برنامه نویسی چند لایه برنامه نویسی سه لایه ( tier۳ ) است « DAL , UI , BLL

## • چه نیازی به لایه بندی پروژه است؟

شرکتی را در نظر بگیرید که ۴۰ کارمند دارد ، این شرکت میتواند یک آپارتمان ۲۰۰ متری را اجاره کرده و تمام کارمندان خود را در این واحد مستقر کند . با این کار همه کارمندان واحد های مختلف

(بازرگانی ، اداری ، مالی ، فنی و...) در کنار هم خواهند بود ، هم همه ، ازدحام رفت و آمد مشتری و... از ابتدایی ترین مشکلات این روش خواهد بود .

اما این شرکت میتواند ۳ طبقه ۸۰ متری تهیه کند و واحد های مختلف را در طبقات مختلف مستقر نماید ، با این کار شاید شرکت هزینه بیشتری را متقبل شود و دسترسی واحد ها به یکدیگر به سادگی قبل نباشد ولی در عوض آرامش ، افزایش بازدهی کار ، برقرار شدن نظم و سکوت ، رضایت مشتری و... از نتایج آن خواهد بود . کد نویسی هم از این قاعده مستثنی نیست ، در حقیقت با این کار هر قطعه کد در سر جای خود قرار می گیرد .

علاوه بر نظم دهی به کد نویسی و خوانا شدن کد ، روش های چند لایه مزایای دیگری نیز دارند :

- هر لایه در نهایت به DII تبدیل شده و مجزا می شود بنابراین قابلیت استفاده از هر لایه در پروژه های مختلف یا حتی چند پروژه در آن واحد (مثلا پنل مدیریت و نمایه یک وب سایت به صورت اشتراکی از لایه BLL و DAL استفاده می کنند ) را در اختیار توسعه دهنده قرار می دهد .
- جداسازی لایه ها و تمیز شدن کد ها به تیم برنامه نویس کمک می کند که بتوانند در کنار هم راحت تر کار کنند .
- در هنگام تغییر بانک اطلاعاتی کل پروژه تحت الشعاع قرار نمی گیرد و تغییرات فقط در لایه ارتباط با بانک اطلاعاتی اعمال می شود .
- روند خطایابی (Debuging) تسهیل می گردد .
- برنامه نویسی شی گرا OOP در روش چند لایه نمود پیدا کرده و قابل پیاده سازی می گردد .
- از تکرار کد در صفحات مختلف جلوگیری می شود .
- کامپایل شدن و بارگذاری پروژه بسیار ساده تر خواهد شد .
- احتمال بروز خطای انسانی در هنگامی کد نویسی به حداقل کاهش می یابد .

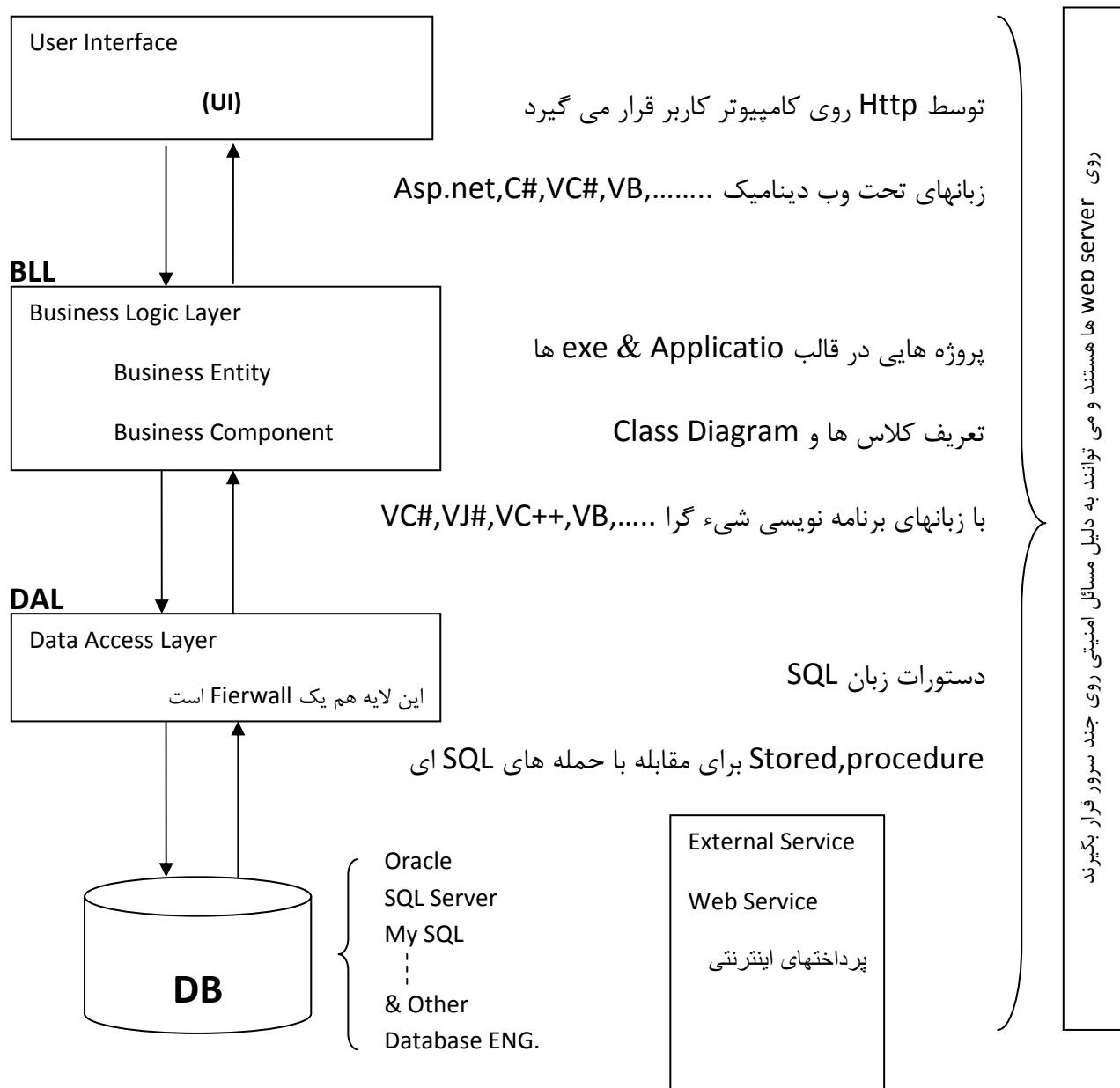
...و

## **لایه بندی پروژه در سرعت توسعه پروژه و همچنین سرعت لود شدن صفحات چه تاثیری دارد ؟**

در هنگام لایه بندی پروژه به ویژه در دفعات اول به نظر می رسد که از سرعت توسعه و پیشرفت پروژه کاسته می شود ولی واقعیت این است که این کار با نظم دهی در کار ها و سهولت خطا زدایی سرعت شما را موارد بسیاری افزایش خواهد داد .

میتوان گفت تاثیری خاصی در بارگذاری صفحات ندارد مگر اینکه از لایه CL یا Cache Layer استفاده کنید که این کار مسلماً باعث افزایش سرعت و بازدهی پروژه شما خواهد شد .

- معماری نرم افزارهای بانک اطلاعاتی تحت وب ( معماری لایه ای ، مثلا " ۳ لایه )



در این نوع معماری Security بالا است و به راحتی نمی توان به اطلاعات موجود در بانک دسترسی داشت

حمله های SQL ای و هک کردن به راحتی انجام نمی شود و باید از لایه های محافظتی بسیاری عبور نمود

که این کار هم بسیار سخت است .

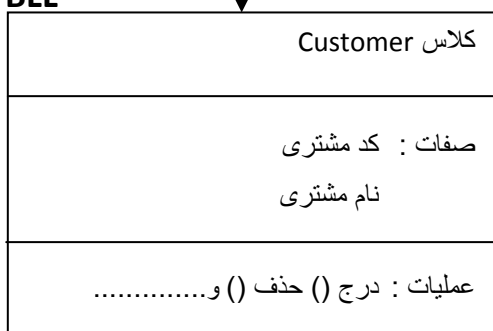
- مثالی برای معماری نرم افزارهای بانک اطلاعاتی تحت وب ( معماری لایه ای ، مثلاً " ۳ لایه )

### Web form or UI

<input type="text"/>	نام مشتری *
<input type="text"/>	تلفن
	⋮

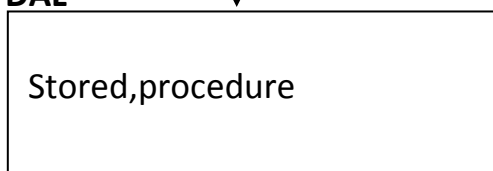
اطلاعات از این لایه به لایه BLL وارد می شود

### BLL



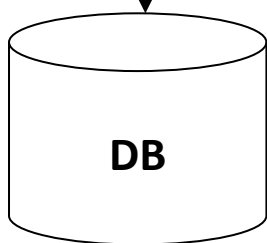
اطلاعات در Web Server خودمان کنترل می شود

### DAL



در اینجا کنترل می کنیم که آیا حمله SQL ای وجود دارد

یا نه ، اگر حمله ای در کار نبود ، اطلاعات به Database وارد می شود.



Oracle  
SQL Server  
My SQL  
⋮  
& Other  
Database ENG.

## Object Oriented Programing ( OPP ) :

مثال : کلاس انسان	
صفات	رفتار
نام	بازکردن دست
وزن	بستن دست
قد	راه رفتن
رنگ چشم	خوابیدن
رنگ مو	خوردن
⋮	⋮

هدف : مدل سازی دنیای واقعی درون کامپیوتر می باشد .

اولین مفهوم در بحث شیء گرایی مفهوم کلاس است .

یک کلاس دارای دو قسمت می باشد :

○ یک سری صفات

○ یک سری رفتار ( عملیات )

نکته : Object ( شیء ) یک نمونه از یک کلاس است .

## Inheritance ( ارث بری ) :

انتقال یک سری صفات از یک یا چند Object به یک یا چند Object دیگر که می تواند یگانه یا چند گانه باشد.

\*ارث بری یگانه یعنی از یک جا ارث میبرد و ارث بری چندگانه یعنی از چند جهت ارث بری داشته باشد.

## شکل کلی یک کلاس در C++ :

Class نام کلاس {

محل تعریف صفات و توابع خصوصی ( Private اختصاصی )

Public:

محل تعریف صفات و توابع عمومی

Private:

محل تعریف صفات و توابع خصوصی ( Private اختصاصی )

\*چیزی که در اینجا تعریف شود و بلافاصله بعد از نام کلاس تعریف شود ، باهم فرقی ندارد .

Protected:

محل تعریف صفات و توابع محافظت شده

\*چیزهایی که می تواند ارث داده شود و خاص است می تواند در Protected تعریف شود.

( در دنیای واقعی دارایی های پدر برای فرزندش در Protected تعریف می شود )

اگر دوست داشته باشیم ( ; ) می گذاریم اگر نه شیء را تعریف می کنیم ، سپس ( ; ) می گذاریم .

( یکی از محل های تعریف Object از نوع Class )

## مثال : کلاس دانشجو

Class Student

{

Public:

Char \*stid ;  
 Char \*stname;

یا char \*stid [11];  
 رشته با طول کاراکتری نامشخص

یا char \*stname[50];  
 اشاره گر

Void print ()

{

Cout<<"\n student identification is : "<<stid;

Or {

برو به خط بعد

Printf("\n student identification is : % s " , stid);

}

Pritf = cout : note  
 Scanf = cin

Private :

Char \*tel;

} st \, st ٢;

student دو object از کلاس St \&st ٢

## Encapsulation ( کپسوله سازی ) :

قرار دادن صفات در داخل خودش ، به عبارت دیگر قرار دادن یک سری صفات و عملیات و ایجاد آن به عنوان یک شیء کپسوله سازی می باشد که این عملیات به فرآیند پنهان سازی کمک می کند.

## Polymorphism ( پولی مورفیسم یا پولی مور فیزم ) :

این موضوع به مفهوم چند ریختی اشاره دارد و به این مفهوم است که اشیای مختلف در قبال یک عملیات خاص می توانند رفتارهای متفاوتی داشته باشند.

این موضوع همان هوش مصنوعی است ، به عبارت دیگر یعنی اینکه رفتارها و عکس العمل ها برای حل یک مشکل مختلف ممکن است که متفاوت باشد.

## گفتیم که :

Class (Class Name) نام کلاس {  
محل تعریف صفات و توابع خصوصی ( Private اختصاصی )  
Public:  
محل تعریف صفات و توابع عمومی  
Private:  
محل تعریف صفات و توابع خصوصی ( Private اختصاصی )  
Protected:  
محل تعریف صفات و توابع محافظت شده  
} obj\,obj۲,.....objn

## نکته :

نام گذاری کلاس ها و object ها مطابق با قوانین نام گذاری متغیرهاست که می تواند ترکیبی از حروف الفبای انگلیسی کوچک و بزرگ ، ارقام ۰ تا ۹ و علامت \_ باشد ، به شرطی که :

- با یک عدد شروع نشود
- فاصله خالی مجاز نیست
- جزء کلمات رزو شده زبان نباشد ( if,for,..... )

## نکته :

با تعریف یک کلاس ، حافظه ای از RAM برای آن اشغال نمی شود و با تعریف یک شیء از نوع کلاس است که حافظه RAM برای آن رزو می گردد.

## حوزه تعریف متغیر ها و توابع :

### Public ( عمومی ) :

دسترسی به صفات و توابع عمومی از طریق تمام توابع و کلاس های دیگر مجاز می باشد .

### Private ( خصوصی ) :

این صفات بلافاصله پس از نام کلاس یا پس از Private هستند. تعریف صفات و توابع از نوع خصوصی می باشد و دسترسی به آنها را فقط توسط توابع کلاس خودشان مجاز می شمارد.

\*اگر صفتی Private تعریف می شود، تمام توابعی که در آن کلاس تعریف می شود ، به آن صفت دسترسی دارد. \*در صفات Private تمام توابع کلاس های خودش به آن دسترسی دارند ولی یک متغیر Local فقط در تابع خودش اعتبار دارد و توابع دیگر به آن دسترسی ندارند.



## Protected (محافظت شده) :

تعریف صفات و توابع از این نوع ، دسترسی را فقط برای کلاس های ارث برنده مجاز می شمارد .

## نحوه تعریف اشیاء از نوع کلاس :

### ○ روش اول :

```
Class نام کلاس {  
    .....  
};
```

### ○ روش دوم :

```
Class نام کلاس ;  
* تعریف شیء پس از اعلان کلاس در هر جای دلخواه از برنامه ( C++ & VC++ )
```

در VC# :

VC#;

VC# : solution Explorer پنجره → نام پروژه → کلیک → Add → New Class

Class نام کلاس (Class Name)

```
{  
  
    نام صفت      نوع داده .....      یا Private یا Public ;  
  
    نام صفت ۱      نوع داده      Public ;  
  
    نام صفت ۱      نوع داده      Private ;  
  
    (پارامتر ورودی) نام تابع      نوع خروجی      Public ;  
  
    {  
        .....  
        بدنه تابع  
    }  
}
```

## تعریف Object (شیء) :

تعریف شیء جدید از نوع کلاس :

نام کلاس      New = نام شیء      نام کلاس

مثالی از یک کلاس در visual c# برای عملیات درج در بانک اطلاعاتی (جدول بانک اطلاعاتی در SQL Server)

Person	int	nvarchar[۳۰]
Id		-----
	Name	

Using ها را باید

⋮

بنویسیم

class Person

{

public int id;

public string name;

⋮

ارتباط با پایگاه داده یا Local Host

SqlConnection objcon = new SqlConnection("رشته ارتباطی")

Public Void Insert()

{

Sql command.objcom=new Sql command();

Objcom.connection= objcom;

Objcom.commandText= "insert into PersonValues(@id,@

name,...)";

Objcom.Parameters.AddWithValue Values(@id,id)";

Objcom.Parameters.AddWithValue Values(@name,name)";

Objcon.open();

Objcom. ExecuteNonQuery();

Objcon.close();

}

⋮

}

## نکته :

برای دسترسی به صفات و توابع عمومی از عملگر "." استفاده می شود ( نام صفت یا تابع "." نام شیء ) و دسترسی به صفات و توابع خصوصی با "." امکان پذیر نمی باشد .

## مثال :

```
Class test {
    Private :
        Int a ;
    Public:
        Int b ;
} obj);
Void main() {
    Obj\b=۱۰;
    Obj\a=۲۰;
};
```

با در نظر گرفتن تعریف عمومی بودن متغیر ، مجاز است

با در نظر گرفتن Private بودن a امکان پذیر نمی باشد، پس خطاست

## نکته :

دسترسی به صفات خصوصی کلاس با استفاده از توابع عضو کلاس و توابع دوست امکان پذیر است .  
مثال ساده ای از دسترسی به صفات Private :

```
#include<iostream.h>
```

io=input/output

```
Class test {
```

```
    Private :
```

```
        Int x ;
```

```
    Public:
```

```
    Void input(int x) {
```

```
        X=x;
```

```
    }
```

```
    Void Print() {
```

```
        Cout<<x;
```

```
    }
```

مقدار دهی صفت خصوصی X

مقدار X1 را در X2 قرار می دهد

مقدار X را در خروجی نمایش می دهد

X را در خروجی نشان می دهد

// پایان تعریف کلاس ( End of class )

// استفاده می کنیم

```
}; //
```

```
Void main(){
```

```
    Test obj;
```

```
    Obj.input(۵۰);
```

```
    Obj.print();
```

```
}
```

Obj ( Object ) تمام صفات کلاس test را دارد.

## " :: " حوزه یا میدان :

پس از اعلان class برای تعریف توابع عضو کلاس از عملگر " :: " ( حوزه یا میدان ) استفاده

می شود که جایگاه آن قبل از نام تابع و بعد از نام کلاس می باشد.

مثال :

کلاسی به نام Circle تعریف شده است

```

Class circle {
    Int x,y ;
    Public:
    Void valueset (int,int,double) {
    Void Print();
    Private :
    double radius;
} cir\); // انتهای تعریف کلاس

```

وقتی عنوان تابع را می نویسیم ، می توانیم فقط نوع متغیرها را ذکر کنیم و دیگر اسم پارامتر ها را نیاوریم و یا اینکه داشته باشیم .... (intx \,inty \,doubler \)

یک شیء از نوع کلاس circle

// انتهای تعریف کلاس

بعد از اتمام تعریف کلاس ، بدنه کلاس را می نویسیم

```

Void circle::valueset(intx \,inty \,doubler \)
{
    x=x \);
    y=y \);
    radius=r \);
}

Void circle::Print()
Cout<<"\n x=" << x <<"\n y= " <<y;
Cout<<"\n radius =" << radius;
}

```

کار تابع valueset مقدار دهی صفات خصوصی کلاس است

در اینجا باید حتماً عنوان پارامترها را هم ذکر کنیم

x \) را در x قرار می دهد

y \) را در y قرار می دهد

r \) را در radius قرار می دهد

مقادیر x,y را چاپ می کند ، هربار هم به خط بعد می رود

به خط بعد رفته و مقدار شعاع را چاپ می کند

## تمرین :

کلاسی به نام دایره تعریف نمائید که دارای صفت شعاع ، مساحت و محیط باشد و سپس یک تابع به نام input برای مقدار دهی شعاع تعریف نمائید ، همچنین یک تابع به نام calculate جهت محاسبه مساحت و محیط دایره تعریف کرده و پس از آن تابعی به نام print برای نمایش طول ، مساحت و محیط دایره در خروجی تعریف نمائید . در نهایت یک شیء از نوع کلاس تعریف نموده و توابع مذکور را فراخوانی و اجرا نمائید :

```
#include<iostream.h>
#include<stdio.h>
class circle
{
    private: float radius , area , circuit ;
    public: void input (float Vallength)
        { radius= Vallength; }
        void Calculate() ;
        void Print() ;
}; // end of class definition
void circle::Calculate()
{
    area= ۳.۱۴* radius* radius;
    circuit= ۲*۳.۱۴* radius;
}
void circle::Print()
{
    printf("\nsquare length: " %f", radius);
    printf("\nsquare area : " %f", area);
    printf("\ncircuit area : " %f", circuit);
}
void main()
{
    circle ObjCls;
    ObjCls.input(۴);
    ObjCls.Calculate();
    ObjCls.Print();
}
```

در داخل public تعریف شد

## تمرین :

کلاسی به نام مربع تعریف نمائید که دارای صفت طول ، مساحت و محیط باشد و سپس یک تابع به نام input برای مقدار دهی طول تعریف نمائید ، همچنین یک تابع به نام calculate جهت محاسبه مساحت و محیط مربع تعریف کرده و پس از آن تابعی به نام print برای نمایش طول ، مساحت و محیط مربع در خروجی تعریف نمائید. در نهایت یک شیء از نوع کلاس تعریف نموده و توابع مذکور را فراخوانی و اجرا نمائید:

```
#include<iostream.h>
#include<stdio.h>
class square
{
    private: float length , area , circuit ;
    public: void input (float Vallength)
        { length= Vallength; }
    void Calculate ();
    void Print ();
}; // end of class definition
void square::Calculate()
{
    area= length*length;
    circuit= length*۴;
}
void square::Print()
{
    printf("\nsquare length: " %f", length);
    printf("\nsquare area : " %f", area);
    printf("\ncircuit area : " %f", circuit);
}
void main()
{
    square ObjCls;
    ObjCls.input (۴);
    ObjCls.Calculate ();
    ObjCls.Print ();
}
```

## تابع سازنده ( Constructor ) :

برای مقدار دهی اولیه به اعضای کلاس از تابع سازنده استفاده می شود که می تواند در صورت نیاز

، آرگومان ( ورودی ) نیز دریافت کند ولی هیچ مقداری را به عنوان خروجی باز نمی گرداند.

این تابع دقیقا" هم نام کلاس است و در داخل کلاس تعریف می شود.

مثال :

```
Class test {
```

```
    Int a ;  
    test (int b) {  
        a=b;}  
};
```

تابع سازنده است و حتما" هم نام کلاس است و هیچ خروجی ای هم ندارد  
یعنی حتی خروجی آن Void هم نیست .

\* نکته : وقتی می گوئیم خروجی تابعی Void هم نیست ، یعنی هیچ خروجی ای Return نمی شود ، ولی

وقتی می گوئیم خروجی ندارد ، یعنی هیچ چیزی خارج نمی شود.

\* نکته : نوع بازگشتی تابع سازنده حتی از نوع Void نیز نمی تواند باشد.

\* نکته : تابع سازنده در هنگام تعریف شیء جدید از نوع کلاس به طور اتوماتیک اجرا می شود.

\* نکته : چنانچه تابع سازنده را برای کلاسی ایجاد نکنیم ، کامپایلر به طور پیش فرض ، یکی را در نظر می

گیرد که دارای آرگومان نبوده و کاری را هم انجام نمی دهد.

```
#include <iostream.h>
```

```
Class test {
```

```
    Int a,b ;  
    Public:  
        test (int a \,int b \) {  
            a=a ;  
            b=b \;}  
    Void print(){  
        Cout<<"\n a =" << a;  
        Cout<<"\n b =" << b;  
    } }; // انتهای تعریف کلاس
```

در اینجا کار تابع سازنده مقدار دهی صفات خصوصی کلاس است

تعریف شیء t1 و فراخوانی تابع سازنده با آرگومان های ورودی :

```

Void main(){
    test t1= test(0,5)
    t1.print();

    test t2= test(10,20)
    t2.print();
}

```

\* نکته : آرگومان های تابع سازنده می تواند مقادیر پیش فرض داشته باشند ، تا در صورت عدم ارسال مقدار به تابع سازنده مقدار دهی اولیه صفات اشیاء انجام شود.  
مثال :

```
#include <iostream.h>
```

```
Class test {
```

```

    Int a,b ;
    Public:
    test (int a = 10, int b = 7);
    a=a ;
    b=b ;}
    Void print(){
    Cout<<"\n a =" << a;
    Cout<<"\n b =" << b;
    }
}; // انتهای تعریف کلاس

```

```
Void main(){
```

تعریف شیء از نوع کلاس و اجرای اتوماتیک :

```

test t1;
t1.print();

test t2(8,15)
t2.print();
}

```

\* نکته : اعضای یک کلاس نمی توانند شیء تعریف شده از نوع همان کلاس باشند ، البته تعریف اشاره گر به شیء از نوع کلاس مجاز است .



## مثال :

الف)

Class test

```
{  
    Int a ;  
  
    Public :  
  
    test b ;  
};
```



عضو یک کلاس نمی تواند شیء از نوع کلاس باشد ، بنابراین این حالت غیر مجاز است .

ب)

Class test

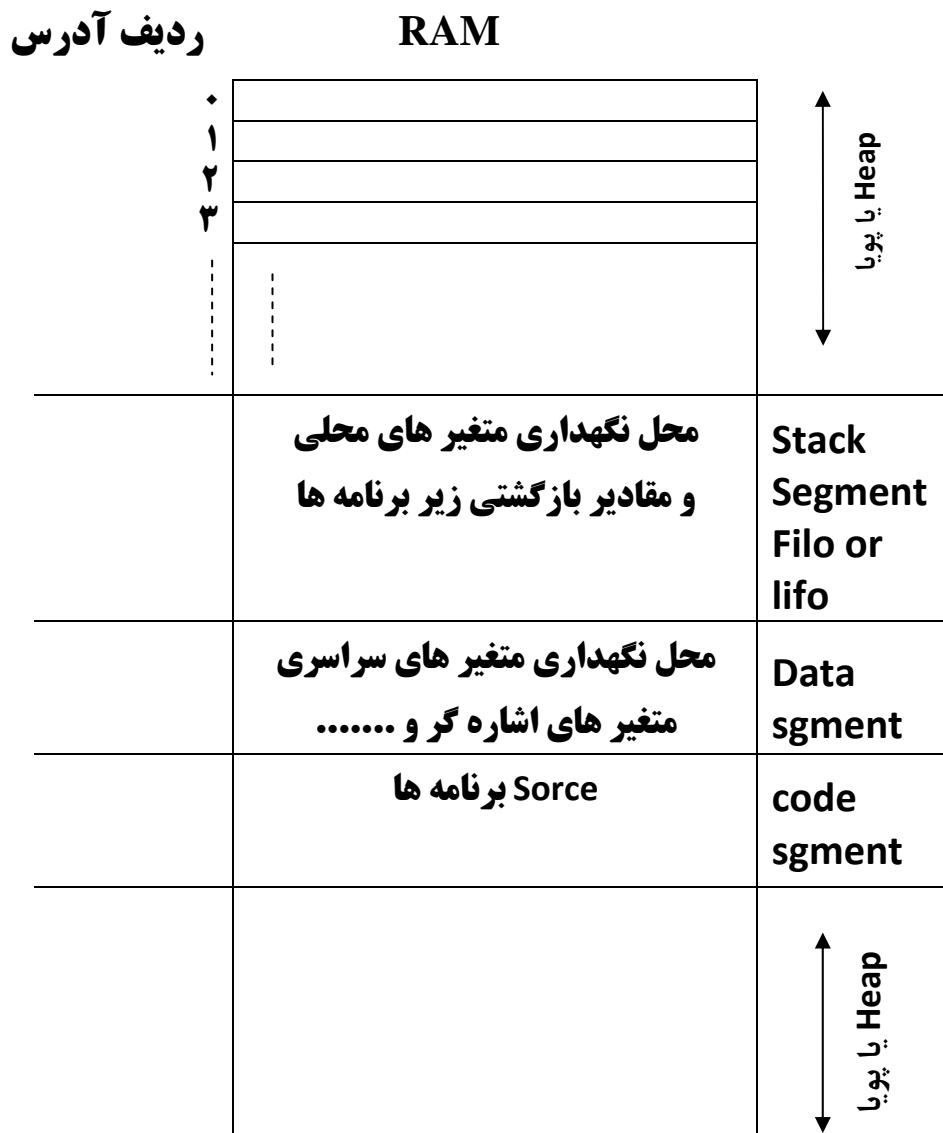
```
{  
    Int a ;  
  
    Public :  
  
    test *b ;  
};
```

اشاره گر



چون از اشاره گر استفاده شده است ، بنابراین ، این حالت مجاز است .  
\* اشاره گر در زبان C++ حداکثر ۲ بایت اشغال می کند و آدرس شروع مکان مورد نظر را نشان می دهد.

## از دیدگاه کامپایلر ( Compiler ) مثل VC# ، VC++ ، C++ و.....



\*نکته : وقتی یک متغیر را تعریف می کنیم ، از ابتدای برنامه تا پایان برنامه آن متغیر فضا را اشغال می کند ، مثلاً "وقتی می گوئیم `int x=0` همواره برای آن متغیر فضای مورد نظر اشغال شده است .

ولی وقتی از دستور `int *p` به همراه مثلاً `new` و `delete` استفاده می کنیم به صورت پویا حافظه را رزو می کنیم . در این حالت `object` را تعریف می کنیم و حافظه را به صورت پویا استفاده می کنیم ، یعنی در زمان اجرا آن را رزو می کنیم و پس از پایان اجرا آن را آزاد می نمائیم .

## اشاره گر ( pointer ) :

### نحوه تعریف اشاره گر :

نام متغیر \* نوع داده ;

```
int *p;
```

```
float*f;
```

```
char*ch;
```

### نکته :

**نام متغیر &** : آدرس متغیر در حافظه Ram را باز می گرداند.

**نام متغیر اشاره گر \*** : محتویات جایی که اشاره گر به آن اشاره دارد را بیان می کند.

### نکته :

متغیر هایی که به طور معمول ( غیر اشاره گر ) تعریف می شوند ، از ابتدا تا انتهای برنامه ، خواه استفاده بشوند و خواه استفاده نشوند ، حافظه اصلی Ram را اشغال می نمایند ، با تعریف متغیر از نوع اشاره گر می توان در طول اجرای برنامه ، در هنگام نیاز فضا از حافظه پویا ( heap ) رزرو نموده ، با آن کار کرد و در همان زمان اجرا نیز آن را آزاد نمود ، که این استفاده بهینه از حافظه می باشد.

### دستورات رزرو و آزاد سازی حافظه به روش پویا در برخی زبانها عبارتند از :

زبان	دستور رزرو	دستور آزاد سازی
C	malloc	free
C#	new	dispose
C++	new	delete
Pascal	getmem	freemem

### نکته :

در زبان c++ اشاره گر از هر نوعی که باشد ، فقط ۲ بایت از حافظه اصلی Ram را اشغال می نماید که در آن ۲ بایت ، آدرس متغیر دیگری را نگهداری می کند.

```
float*f , z=۱۸.۵;
```

مثلاً :

```
f=&z;
```

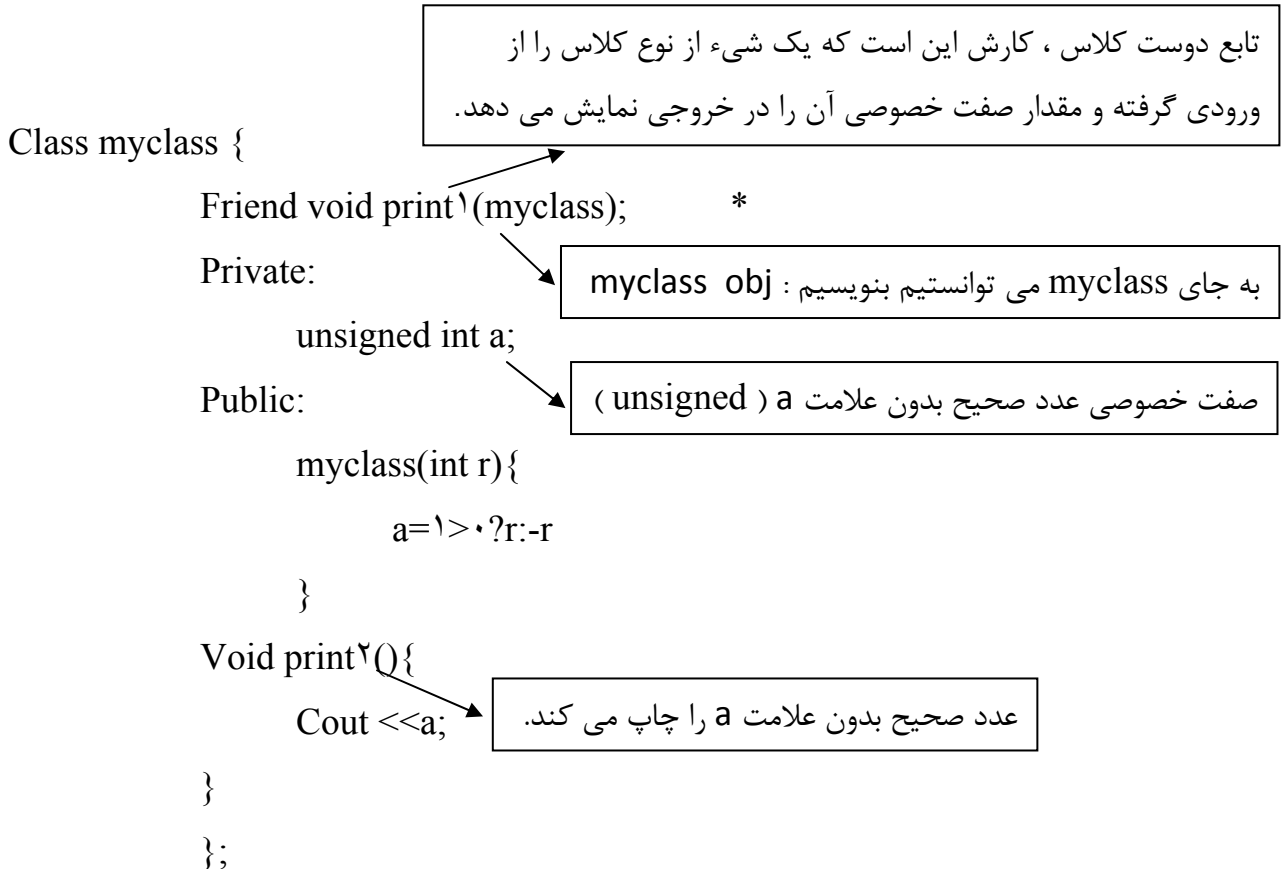
که چون اشاره گر از نوع Float است ۴ بایت ۴ بایت اشاره می کند.

## کلاس ها و توابع دوست ( Friend ) :

برای دسترسی به تمامی داده ها و توابع خصوصی ( Private ) و محافظت شده ( Protected ) از

یک کلاس ، خارج از حوزه اعلان آن کلاس از توابع و کلاس های دوست می توان استفاده نمود ، برای پیاده

سازی از کلمه کلیدی Friend استفاده می شود .



### نکته :

چون print¹ عضو کلاس نیست و دوست کلاس است ، نیازی به عملگر حوزه و میدان ندارد و

نوشتن این دستور به این صورت غلط است :

```
void myclass::print¹(myclass obj){ ..... }
```

### نکته :

دوست کلاس دسترسی بیشتری نسبت به خود کلاس به صفات خصوصی کلاس دارد .

خود کلاس با ( . ) به صفات خصوصی دسترسی ندارد ، حال آنکه دوست کلاس با این امکان به صفات

خصوصی کلاس دسترسی دارد.

```

void print\ (myclass obj) { *
    Cout <<obj.a; }
Void main() {
    myclass mc\ (۹); ( a=۹)
    print\ (mc\);
}

```

تعریف یک شیء از نوع کلاس است با ورودی ۹ برای تابع سازنده

mc\ یک شیء از نوع کلاس است ، بنابراین خروجی Print\ برابر با ۹ می شود.

\* زمانیکه عنوان یک تابع را می گوئیم ، می توانیم فقط نوع آن را تعریف کنیم .

\* Print\ عضو کلاس نیست ، بلکه دوست کلاس است .

\* a صفت خصوصی کلاس از نوع عدد صحیح بدون علامت است که محدوده آن ۰-۶۵۵۳۵ می باشد.

\* myclass یک تابع دقیقا" هم نام کلاس است و هیچ خروجی ندارد و تابع سازنده است و کارش این است که یک عدد را از ورودی می گیرد و اگر شرط برقرار باشد ، آن عدد را در a قرار می دهد و اگر آن عدد مثبت نباشد ، آن را مثبت می کند و در r می گذارد.

```

; عبارت ۲ : عبارت ۱ ؟ شرط = متغیر
Int r { a=r > ۰ ? r : -r ; }

```

یعنی مقدار دهی اولیه توسط تابع سازنده انجام می شود .

مثال :

```

class A {
friend class B;
private:
    int v\ ;
}; // end of class A
class B {
public:
    void initA()
    {
    A obj\ ;
    obj\ .v\ = ۱۰ ;
    }
};

```

معرفی کلاس B به عنوان دوست کلاس A

صفت خصوصی کلاس A

یک شیء از نوع کلاس A تعریف می کند

دسترسی به صفت خصوصی شیء تعریف شده از کلاس a توسط تابع عضو کلاس B که دوست کلاس a است .

\* ما با تعریف دوست می توانیم به کلاس خصوصی دست پیدا کنیم .

## تابع مخرب (Destructor) :

تابع مخرب یک تابع عضو هم نام با کلاس است که با کاراکتر ~ شروع می شود که شیء تعریف شده از نوع کلاس ، با تمام اجرای توابع حاوی آن از بین می رود ، تابع مخرب فراخوانی شده حافظه اشغال شده را به سیستم باز می گرداند.

مثال :

```
class test {  
    friend class B;
```

```
public:
```

```
int x ;
```

یک عدد از ورودی می گیرد

```
test (int a) { x=a+1; }
```

آن را + 1 می کند و در x قرار می دهد

تابع مخرب

```
~test () {}
```

فقط حافظه را برمی گرداند

```
} t(7);
```

تعریف شیء یا object جدید با ورودی 7 برای تابع سازنده

عدد 7 به عنوان ورودی تابع سازنده وارد می شود .

روش های دیگر تعریف شیء :

```
test t1 (10);
```

```
test t2 = test (15)
```

## تمرین :

کلاسی به نام فروشنده تعریف نمائید ، شامل صفات نام فروشنده ، کد فروشنده و سمت (صفات عمومی ) و همچنین شامل صفاتی چون آدرس ، تلفن منزل و تاریخ تولد ( صفات خصوصی ) و یک تابع سازنده تعریف نمائید جهت مقدار دهی صفات عمومی کلاس و تابعی به نام Input جهت مقدار دهی صفات خصوصی کلاس و کلاس مشتری را به عنوان دوست کلاس فروشنده تعریف نمائید که کلاس مشتری شامل صفات نام ، تلفن و آدرس باشد و تابعی به نام نمایش را عضو کلاس مشتری تعریف نمائید که نام فروشنده ، آدرس و تلفن و تاریخ تولد او را هنگام دریافت یک شیء از نوع کلاس فروشنده به عنوان ورودی در خروجی نمایش دهد.

```
#include<iostream.h>
class seller
{
    friend class customer;
private:char* address;long telhome; char* datebirth;
public:char* nameseller; int codeseller; char* post;
seller()
{ nameseller="foroshandeh";
codeseller=۱۱۴۵;
post="sells-manager";
}
~seller()
{};
void input()
{ address="tehran";
telhome=۸۵۴;
datebirth="۱۳۹۰";
}
}obj۱;
class customer
{
char* nm; long tel; char* add;
public:
void input\()
{nm="moshtari";
tel=۱۸۵;
add="tehran";
}
void print()
{
cout<<"\n =====seller dateil =====";
cout<<"\n nameseller:"<<obj۱.nameseller;
cout<<"\n address:"<<obj۱.address;
cout<<"\n telhome:"<<obj۱.telhome;
cout<<"\n datebirth:"<<obj۱.datebirth;
cout<<"\n\n ===== costomer dateil =====";
cout<<"\n namecustomer:"<<nm;
cout<<"\n telcustomer:"<<tel;
cout<<"\n address:"<<add;
}
}obj۲;
void main()
{ obj۲.input\();
obj۱.input();
obj۲.print();
}
```

## برای یک کاراکتر در address :

```
#include<iostream.h>
//using namespace std;
#include<time.h>
#include<stdio.h>
#include<stdlib.h>
#include <string.h>
class Seller{
    friend class Customer;
public:
    char SName;
    int SCod;
    char SPosition;
private:
    char SAddress;
    int STelHome;
    int SBirthDay;
    Seller(char sname,int scode,char sposition){
        SName=sname;
        SCod=scode;
        SPosition=sposition;
    };
    Seller(){
        SName='N';
        SCod=0;
        SPosition='P';
    };
    ~Seller(){};
public: void Input (char address,int stelHome,int sbirthDay);
};
class Customer {
private:
    char CName;
    int CTel;
    char CAddress;
public: void Display (char address,int telhome,int sbirthDay);
};
void Seller :: Input (char address,int stelHome,int
sbirthDay){
    SAddress=address;
    STelHome=stelHome;
    SBirthDay=sbirthDay; }
void Customer :: Display (char address,int telhome,int
sbirthday){
    Seller seller;
    seller.Input(address,telhome,sbirthday);
    cout<<"seller Name="<<seller.SName<<endl;
    cout<<"seller Address="<<seller.SAddress<<endl;
    cout<<"seller telhome="<<seller.STelHome<<endl;
    cout<<"seller birthday="<<seller.SBirthDay<<endl; }
int main (){
    char address;
    cout<<"Address=";
    cin >>address;
    cout<<"Telhome=";
    int telhome;
    cin >>telhome;
    int birthday;
    cout<<"Birthday=";
    cin >>birthday;
    Customer.Display (address,telhome,birthday);
    return 0; }
}
```



```

#include<iostream.h>
//using namespace std;
#include<time.h>
#include<stdio.h>
#include<stdlib.h>
#include <string.h>
class Seller {
    friend class Customer;
public:
    char* SName;
    int SCod;
    char* SPosition;
private:
    char* SAddress;
    int STelHome;
    char* SBDay;
    Seller(char* sname,int scode,char* sposition){
        SName=sname;
        SCod=scode;
        SPosition=spostion;    };
    Seller(){
        SName="Mehdi Ebrahimi";
        SCod=100;
        SPosition="p";    };
    ~Seller(){};
public: void Input (char* saddress,int stelHome,char* sb);    };
class Customer {
private:
    char* CName;
    int CTel;
    char* CAddress;
public: void Display (char* address,int telhome,char*
sbirthDay);
}Customer;
void Seller :: Input (char* saddress,int stelHome,char* sbi){
    SAddress=saddress;
    STelHome=stelHome;
    SBDay=sbi;    }
void Customer :: Display (char* address,int telhome,char* s){
    Seller seller;
    seller.Input (address,telhome,s);
    cout<<"-----"<<endl;
    cout<<"seller Name="<<seller.SName<<endl;
    cout<<"seller Address="<<seller.SAddress<<endl;
    cout<<"seller birthday="<<s<<endl;
    cout<<"seller telhome="<<seller.STelHome<<endl;    }
int main (){
    char* address;
    cout<<"Address=";
    cin >>address;
    char* b;
    cout<<"Birthday=";
    cin >>b;
    int telhome;
    cout<<"Telhome=";
    cin >>telhome;
    Customer.Display (address,telhome,b);
    int i;
    cin>>i;
    return 0;
}

```

## عضو داده ای استاتیک در کلاس :

برای اشتراک یکی از اعضای داده ای کلاس ، بین اشیاء تعریف شده از نوع آن کلاس از کلمه کلیدی Static استفاده می شود ، در این وضعیت هر گونه تغییر عضو استاتیک توسط هر یک از اعضاء برای اعضاء دیگر قابل دسترس است .

### مثال :

توضیح عملکرد متغیر استاتیک :

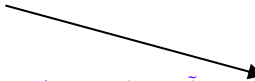
```
class my calss
{
    public:
        void get count() {count << count;}
        void setcount(int a) {count =a;}
        private:
            static int count;
};

int my calss ::count;

void main() {
    my calss obj۱,obj۲;
    obj۱.setcount(۵);
    obj۱.getcount();           خروجی ۵
    obj۲.getcount();           خروجی ۵
    obj۲.setcount(۲۰);
    obj۲.getcount();           خروجی ۲۰
    obj۱.getcount();           خروجی ۲۰
};
```

تعریف صفت استاتیک

اختصاص حافظه به متغیر استاتیک توسط نام حوزه یا میدان



## نکته :

با معرفی عضو استاتیک در داخل کلاس ، حافظه به آن اختصاص نمی یابد، بلکه با تعریف این متغیر در خارج از کلاس با استفاده از نام کلاس و عملگر تعیین کننده حوزه (::) یا میدان ، حافظه به آن تخصیص می یابد.

## نکته :

به همین دلیل قبل از ایجاد شیء از نوع کلاس که دارای عضو داده ای استاتیک می باشد ، می توان به آن عضو ، مستقیماً" از طریق خود کلاس دسترسی داشت .

## نکته :

یکی از کاربرد های اعضای استاتیک مقدار دهی اولیه اعضای Public کلاس قبل از تعریف شیء از نوع کلاس می باشد.  
مثال :

```
class test
{
    public:
        static int count;          صفت استاتیک
};

int test :: count;                اختصاص حافظه

void main() {
    test :: count=۱۰۰;            دسترسی مستقیم به صفت استاتیک توسط خودکلاس
    count << test::count;        خروجی ۱۰۰

    test obj;
    obj.count=۲۰۰;
    count << obj.count;         خروجی ۲۰۰
}
```

## توابع عضو استاتیک :

توابع عضوی که به اعضای غیر استاتیک دسترسی نداشته باشند ، می توانند به صورت استاتیک

تعریف شوند .

## روش های استفاده از توابع عضو استاتیک :

با استفاده از اشیاء تعریف شده از نوع کلاس

به صورت مستقل از اشیاء توسط نام کلاس و عملگر تعیین کننده حوزه (::)

## مثال :

```
class my calss{
    static int count;           صفت استاتیک
public:
    static int setcount(int x) {count =x;}
                                setcount به صورت استاتیک تعریف شده است
    void getcount() {count << count;
                                getcount مقدار count را چاپ می کند
}obj\;
int my calss ::count;         اختصاص حافظه به عضو استاتیک
void main(){
```

فرم دوم :

```
my calss :: setcount(۱۰);
obj\ .getcount();           خروجی ۱۰
```

چون getcount استاتیک تعریف نشده است ، پس نمی توانیم با نام کلاس و عملگر حوزه آن را صدا کنیم .

فرم اول :

```
obj\ .setcount(۲۰);
obj\ .getcount();           خروجی ۲۰
};
```

## آرایه ای از اشیاء:

در زبان C++ امکان تعریف آرایه ای از اشیاء وجود دارد که تمام قوانین آرایه در مورد آن صادق

است .

### مثال:

```
class student{
public:
    int id;
    char name[۳۰];
    char reshteh[۵۰];
    void input(){
    cin>> id>> name>>reshteh;
    }
}
st[۴۰]
void main(){
    St[۰].id=۱
        .
        .
        .
```

یا

```
For (int i=۰ ;i<۴۰ ;i++)
{
Cin >> st [i] .id;
Cin >> st [i].name;
.
.
.
}
```

id	name	reshteh	input()					id	name	reshteh	input()
St[۰]				st[۱]				st[۲۹]			

### تمرین:

کلاسی به نام انتخاب واحد تعریف نمائید ، شامل ، صفات شماره دانشجویی ، کد درس ، نام درس ،

تعداد واحد ، نوع درس ، استاد درس ، روز و ساعت درس ، یک تابع برای مقدار دهی صفات

یک تابع برای نمایش صفات (چاپ ) ، سپس یگ آرایه ۵ تایی از اشیاء از نوع کلاس تعریف نموده ، با آن

کار نمائید.

## مقدار دهی اولیه به آرایه ای از اشیاء :

در صورتیکه کلاس مورد نظر دارای تابع سازنده باشد و تابع سازنده آرگومان ورودی دریافت کند ، آرایه معرفی شده از اشیاء را می توان به صورت زیر مقدار دهی اولیه نمود.

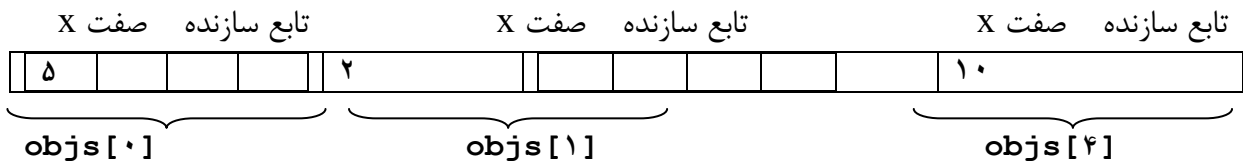
**مثال :**

```
class myclass{
int x;

public:
    myclass (int a){x=a}
};
Myclass obj[5]={5,2,0,-3,10};
```

آرایه از نوع اشیاء  
 ورودی تابع سازنده  
 obj[0]  
 ورودی تابع سازنده  
 obj[1]

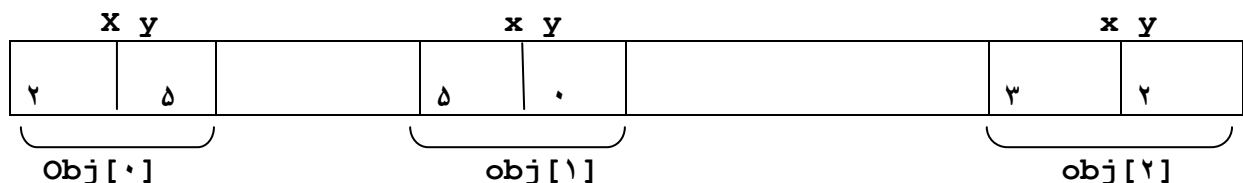
تابع سازنده هم نام کلاس بود ولی خروجی آن حتی void هم نبود.



## نحوه مقدار دهی اولیه :

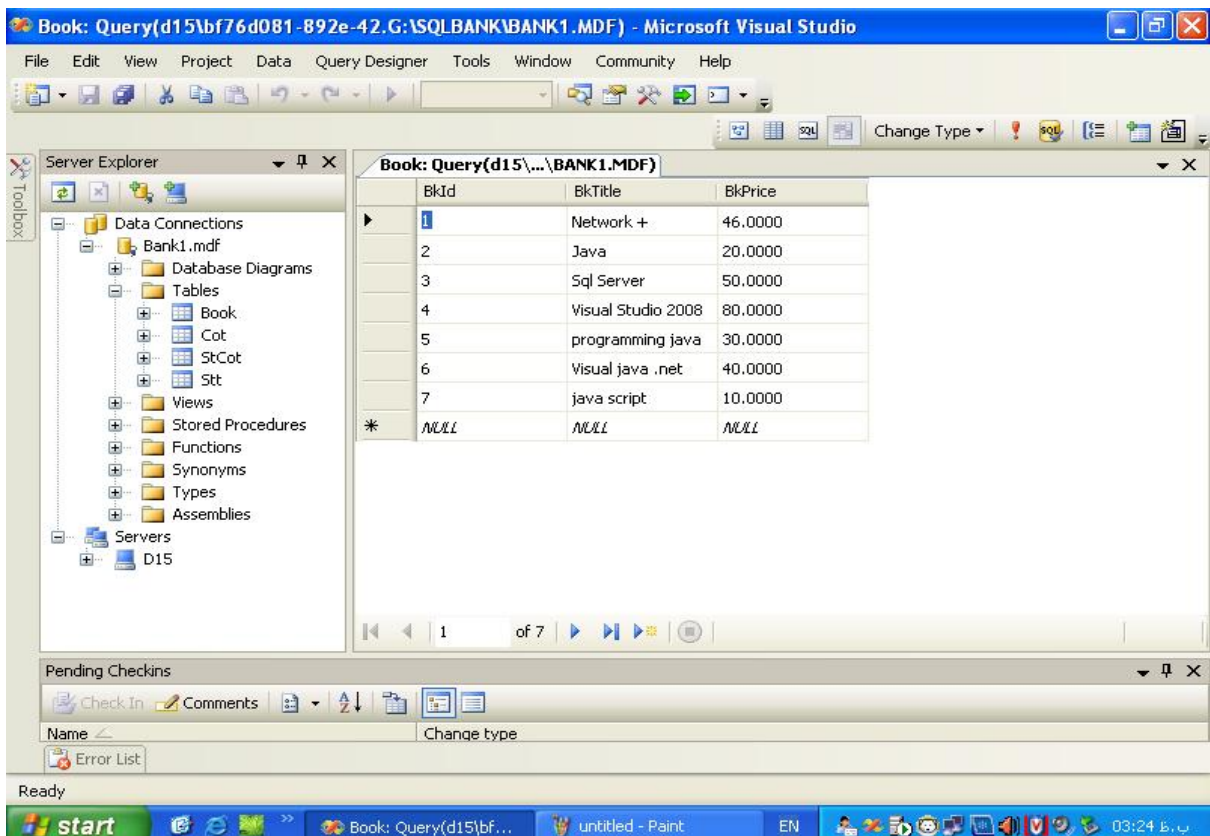
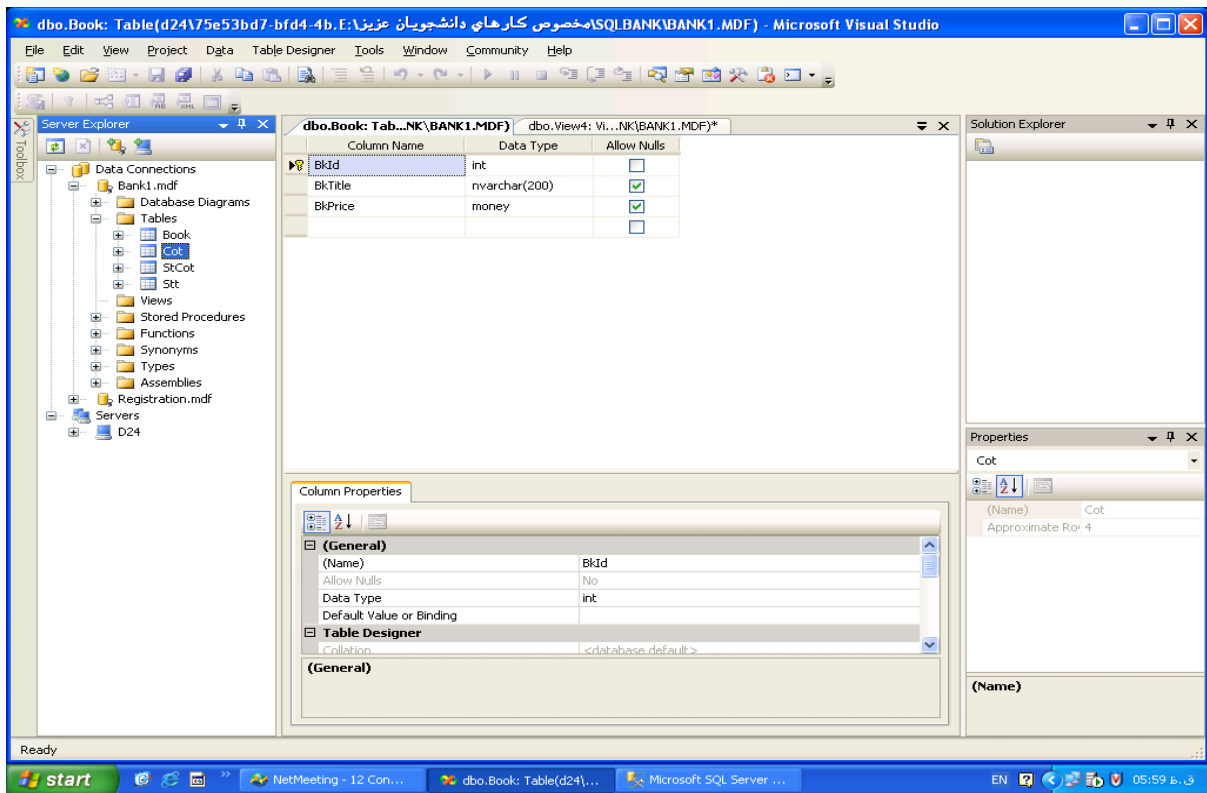
نحوه مقداردهی اولیه به آرایه ای از اشیاء در صورتیکه تابع سازنده کلاس مربوطه بیش از یک آرگومان ورودی داشته باشد به شکل زیر است :

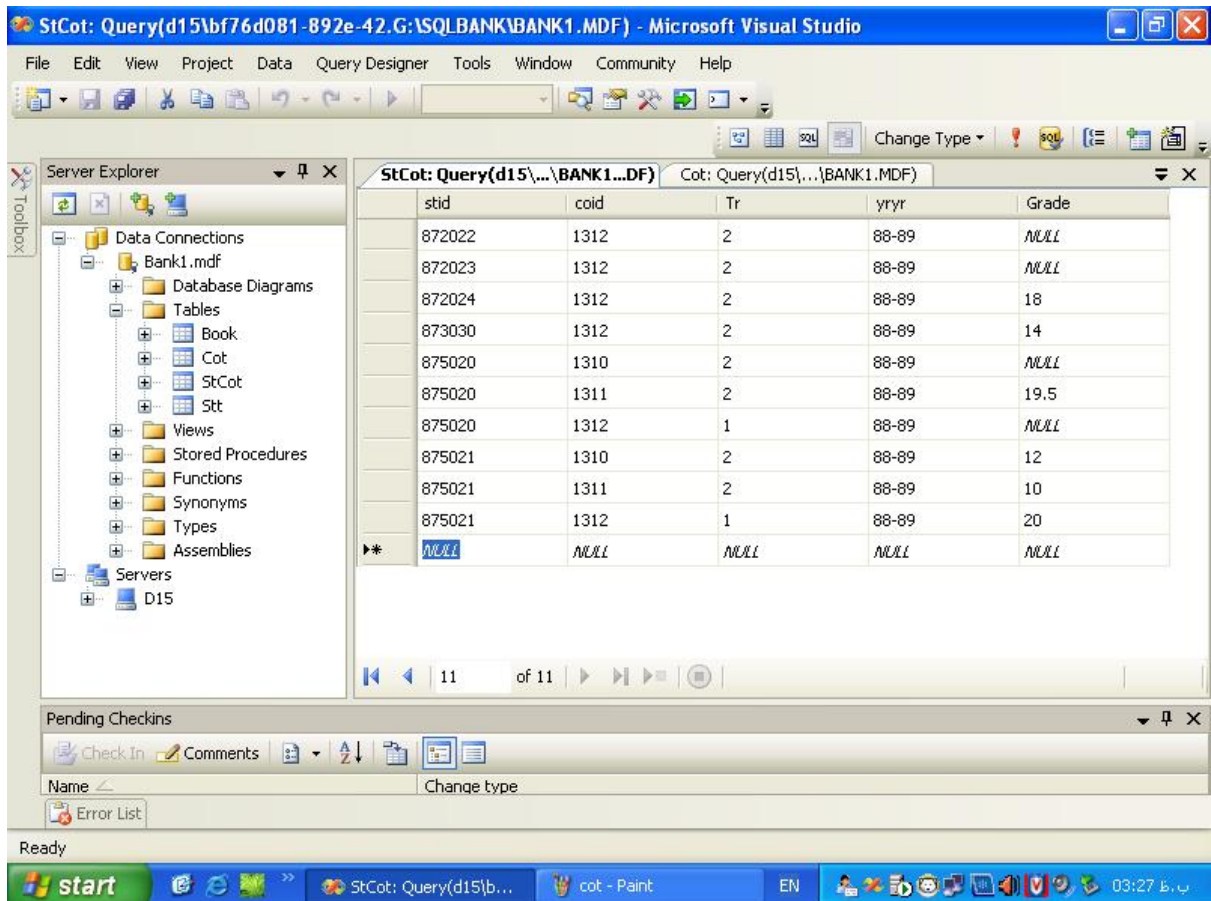
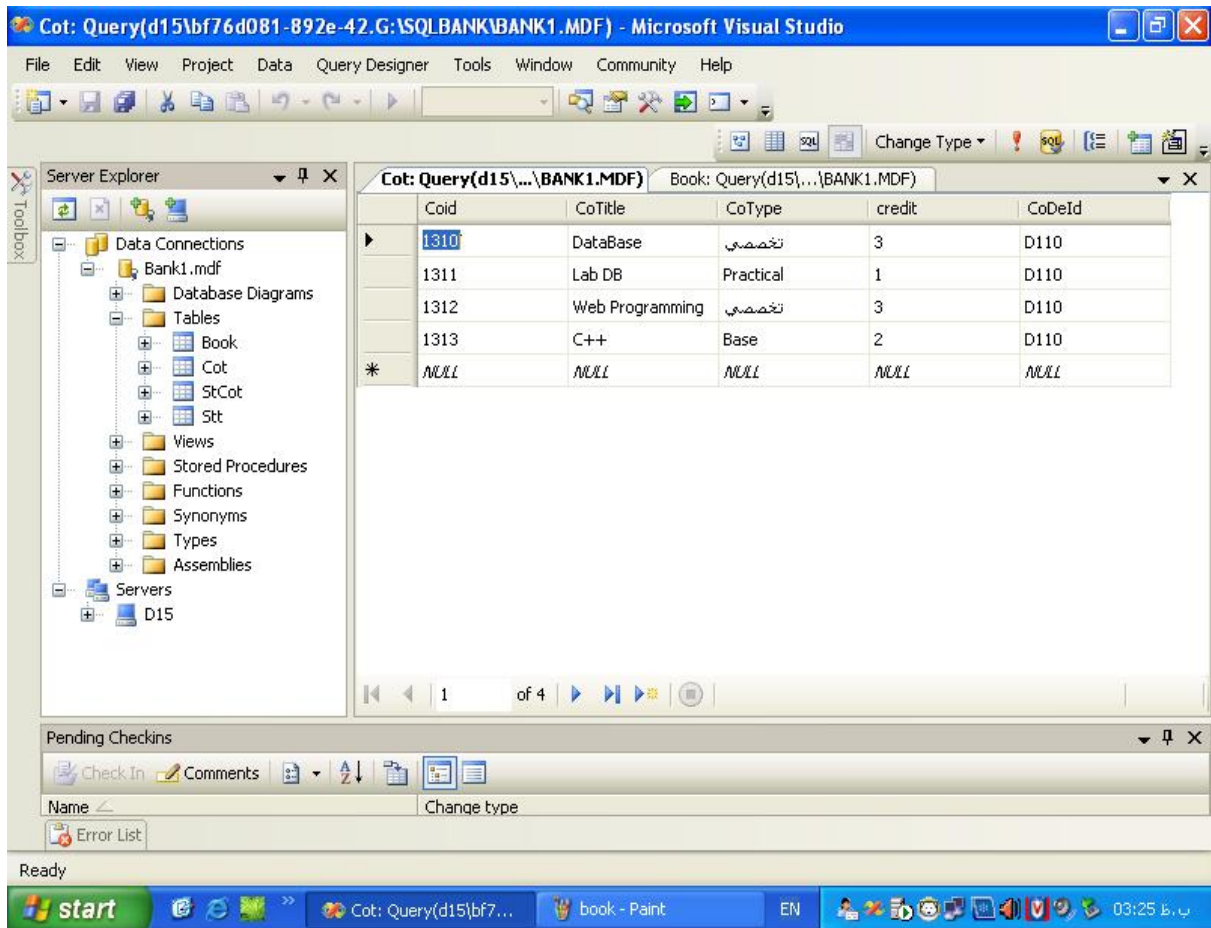
```
class my calss{
    int x;
    int y;
public:
    my calss (int a,int b){ x=a ; y=b;}
};
my class obj[3]={
myclass(2,5) , myclass(5,0) ,myclass(3,2)
};
```



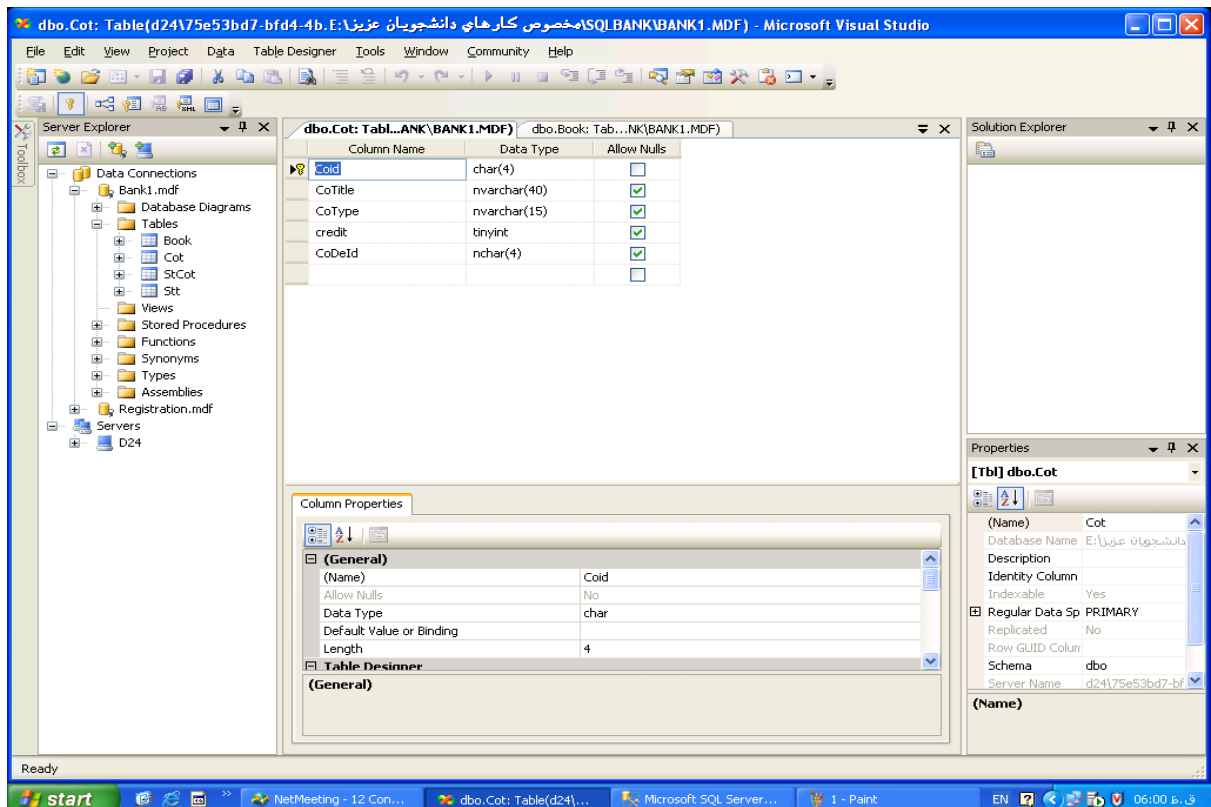
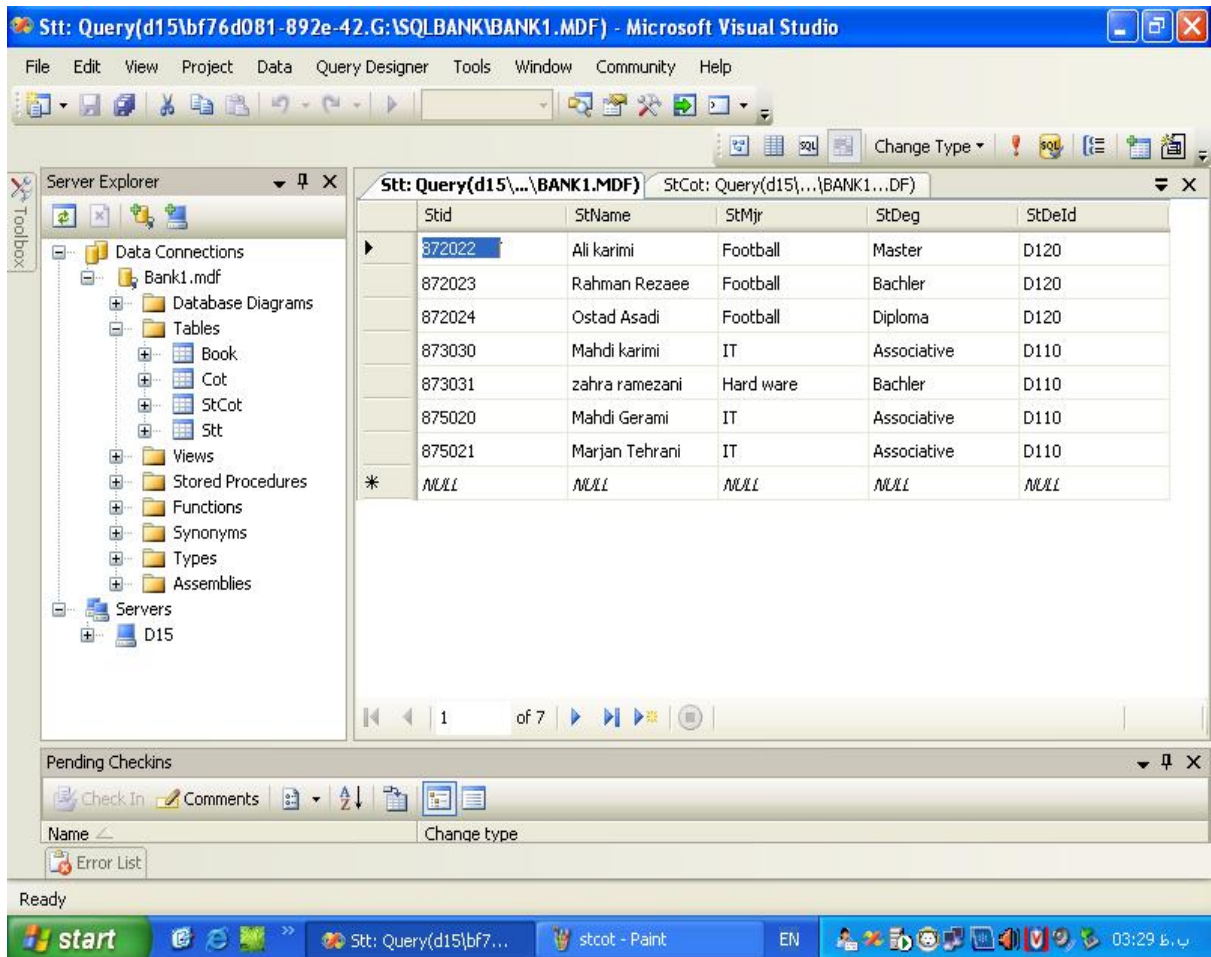
# پیوست ۲ : SQL

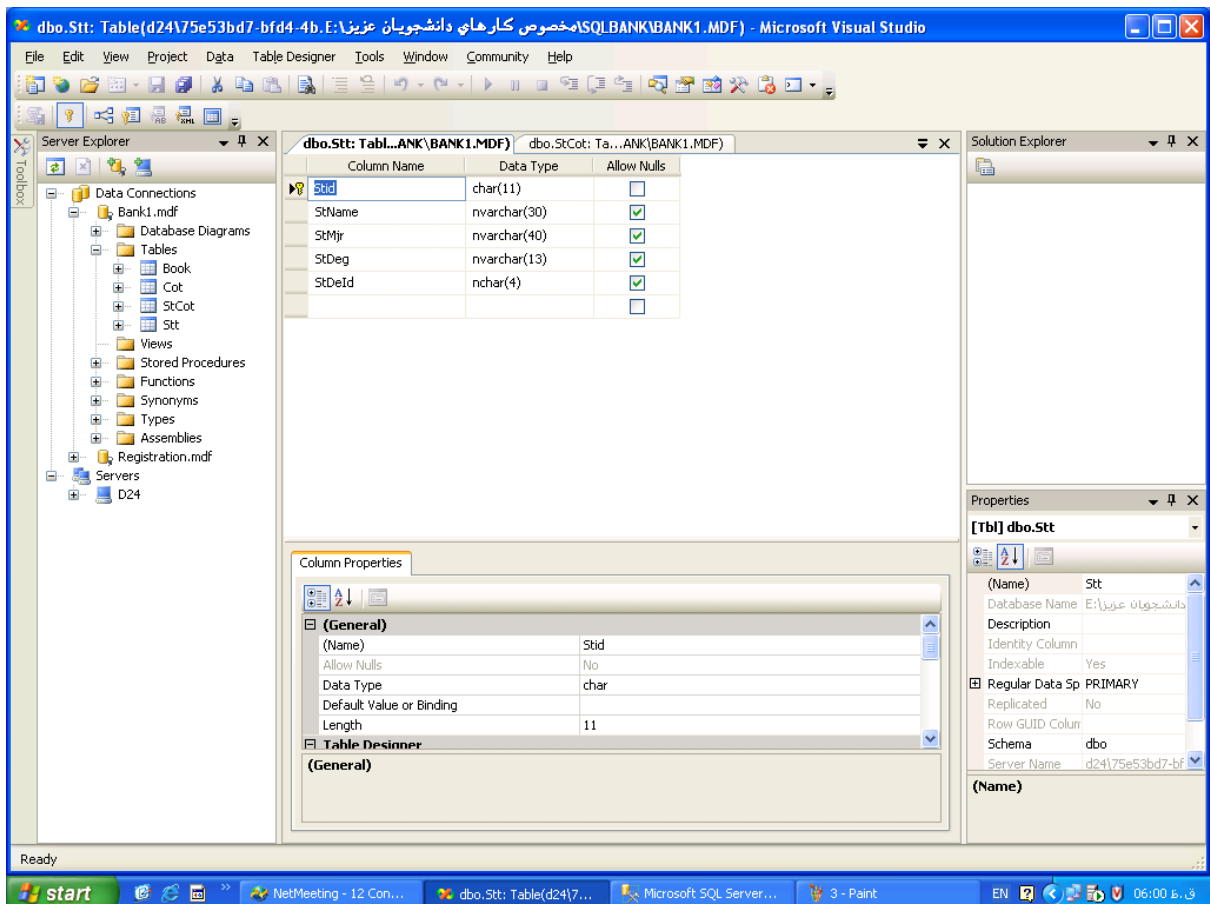
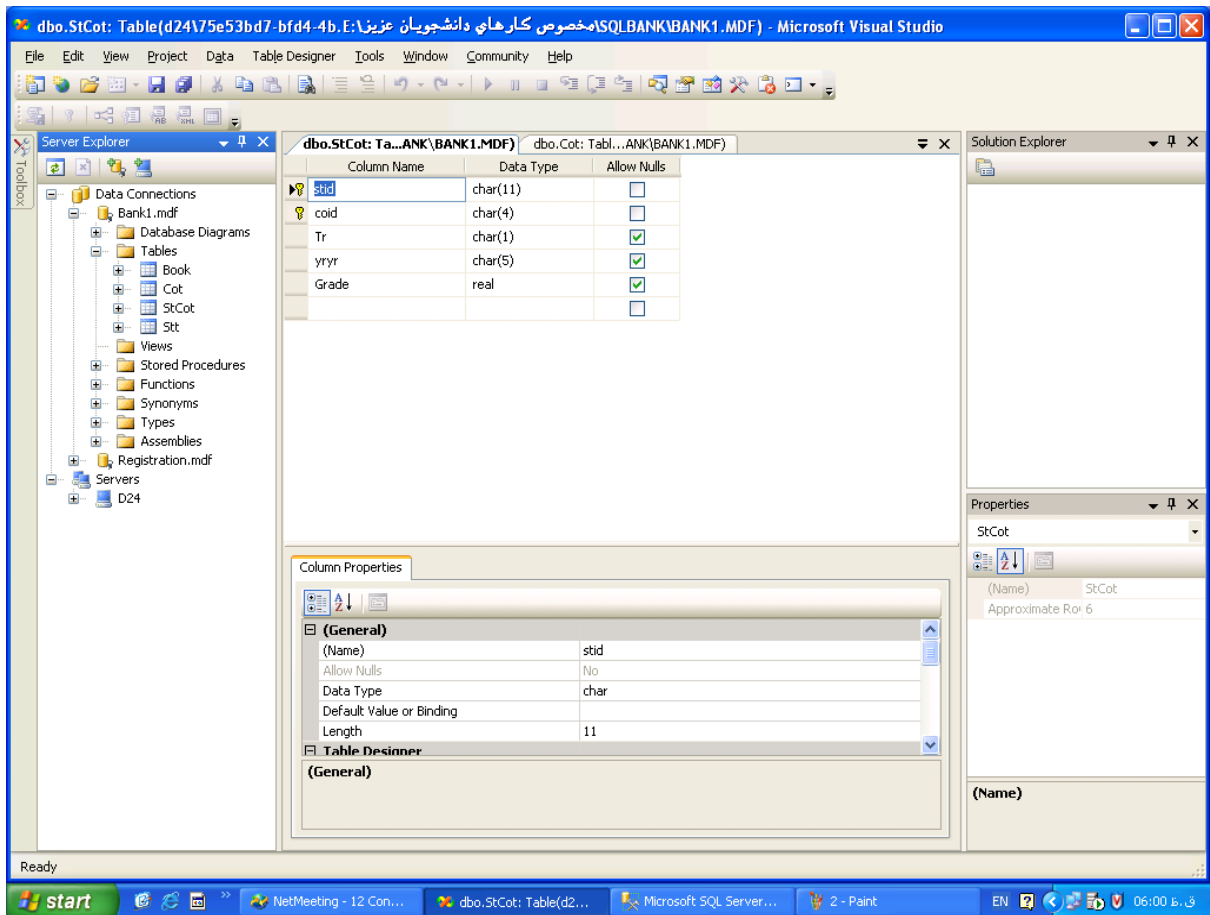
ایجاد table های اولیه :











dbo.View3: View(d24\75e53bd7-bfd4-4b.E:دانشجویان عزیزا\مخصوص SQLBANK\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

- Data Connections
  - Bank1.mdf
    - Database Diagrams
    - Tables
      - Book
        - BkId
        - BkTitle
        - BkPrice
      - Cot
      - StCot
      - Stt
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
    - Registration.mdf
    - Servers
      - D24

dbo.View4: Vi...NK\BANK1.MDF)\*

dbo.View3: Vi...NK\BANK1.MDF)\*

Book

- \* (All Columns)
- BkId
- BkTitle

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
BkId		Book	<input checked="" type="checkbox"/>			
BkTitle		Book	<input checked="" type="checkbox"/>			
BkPrice	Bkprice...	Book	<input checked="" type="checkbox"/>			
BkPrice * 10030	BkPrice...		<input checked="" type="checkbox"/>			

```
SELECT BkId, BkTitle, BkPrice AS BkpriceDolar, BkPrice * 10030 AS BkPriceRials
FROM dbo.Book
```

BkId	BkTitle	BkpriceDolar	BkPriceRials
1	Network +	46.0000	461380.0000
2	Java	20.0000	200600.0000
3	Sql Server	50.0000	501500.0000
4	Visual Studio 2008	80.0000	802400.0000
*	NULL	NULL	NULL

Properties

[View] dbo.View3

- (Name) View3
- Bind To Schema No
- Database Name E:\دانشجویان عزیزا\
- Description
- Deterministic Yes
- Distinct Values No
- GROUP BY Exter <None>
- Output All Column No
- Schema dbo
- Server Name d24\75e53bd7-bf

Ready

dbo.View4: View(d24\75e53bd7-bfd4-4b.E:دانشجویان عزیزا\مخصوص SQLBANK\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

- Data Connections
  - Bank1.mdf
    - Database Diagrams
    - Tables
      - Book
        - BkId
        - BkTitle
        - BkPrice
      - Cot
      - StCot
      - Stt
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
    - Registration.mdf
    - Servers
      - D24

dbo.View4: Vi...NK\BANK1.MDF)\*

dbo.View3: Vi...NK\BANK1.MDF)\*

StCot

- \* (All Columns)
- stid
- cod
- Tr
- yr

Column	Alias	Table	Output	Sort Type	Sort Order	Group By
*	TedadD...		<input checked="" type="checkbox"/>			Count
stid		StCot	<input type="checkbox"/>			Where

```
SELECT COUNT(*) AS TedadDars
FROM dbo.StCot
WHERE (stid = '875020')
```

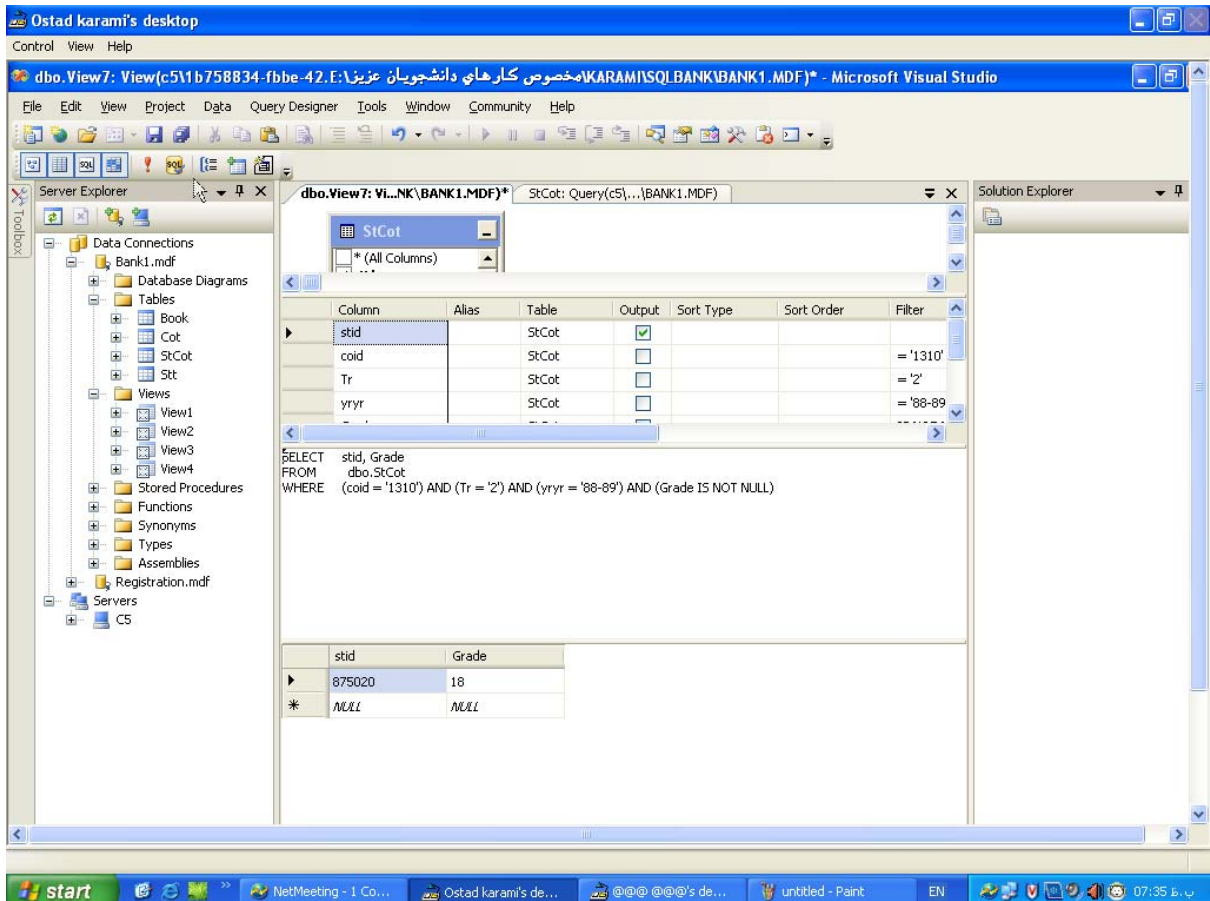
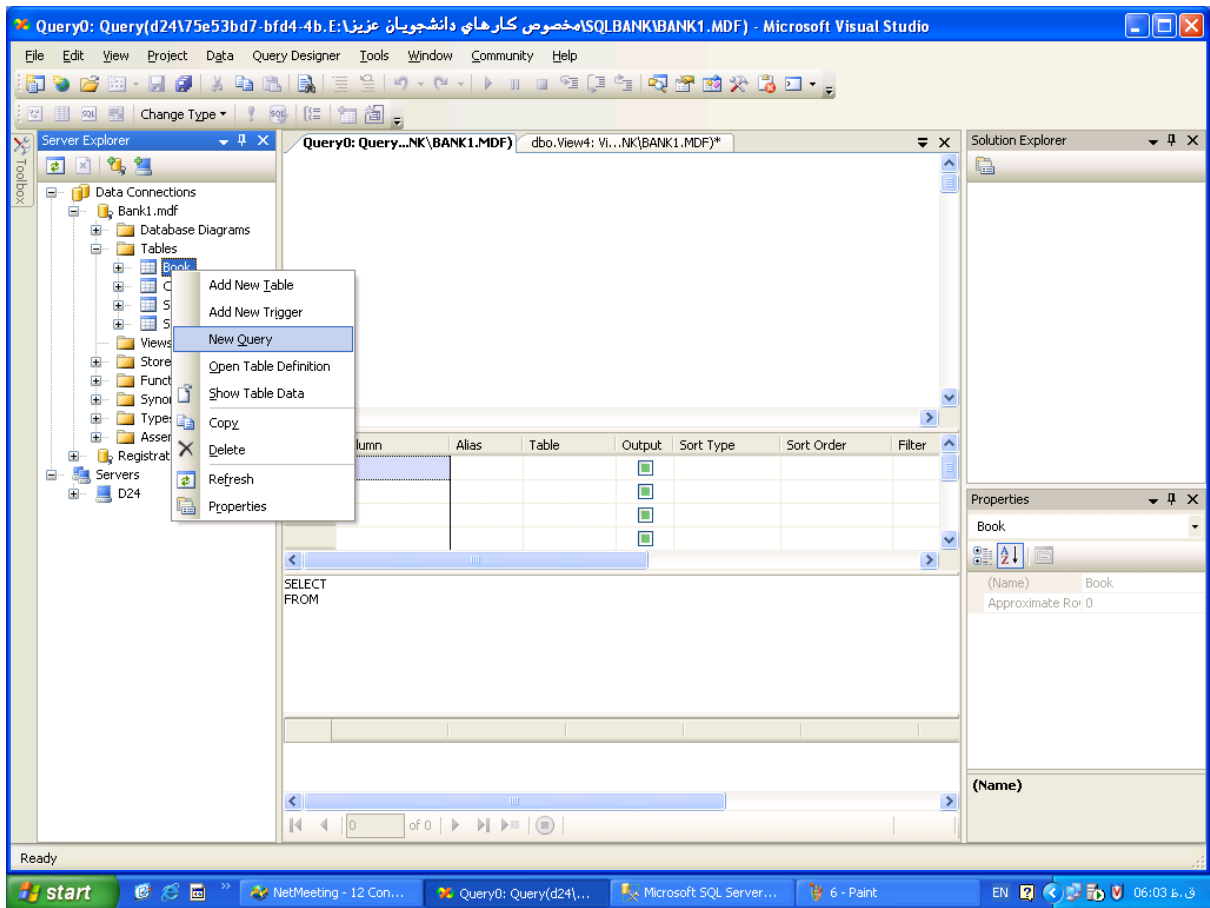
TedadDars
3

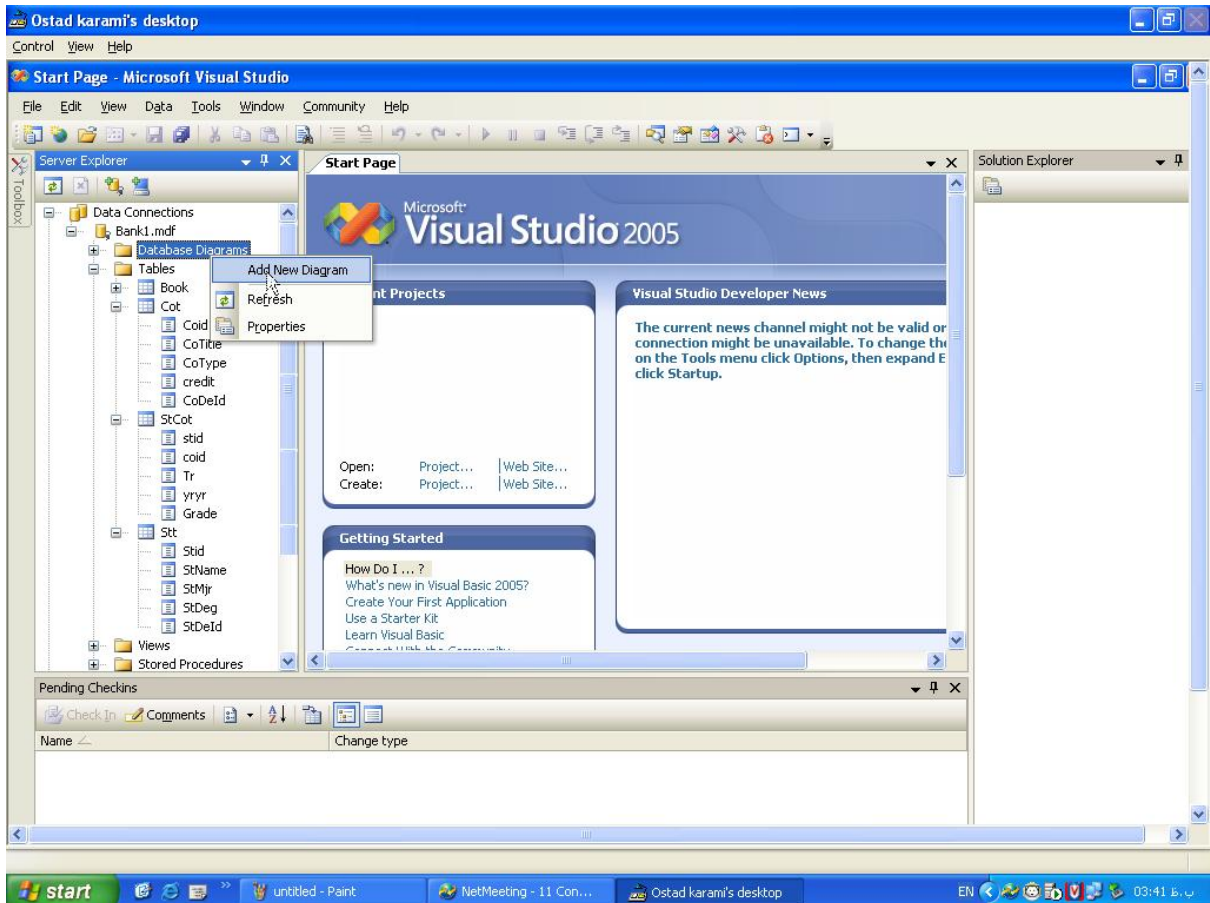
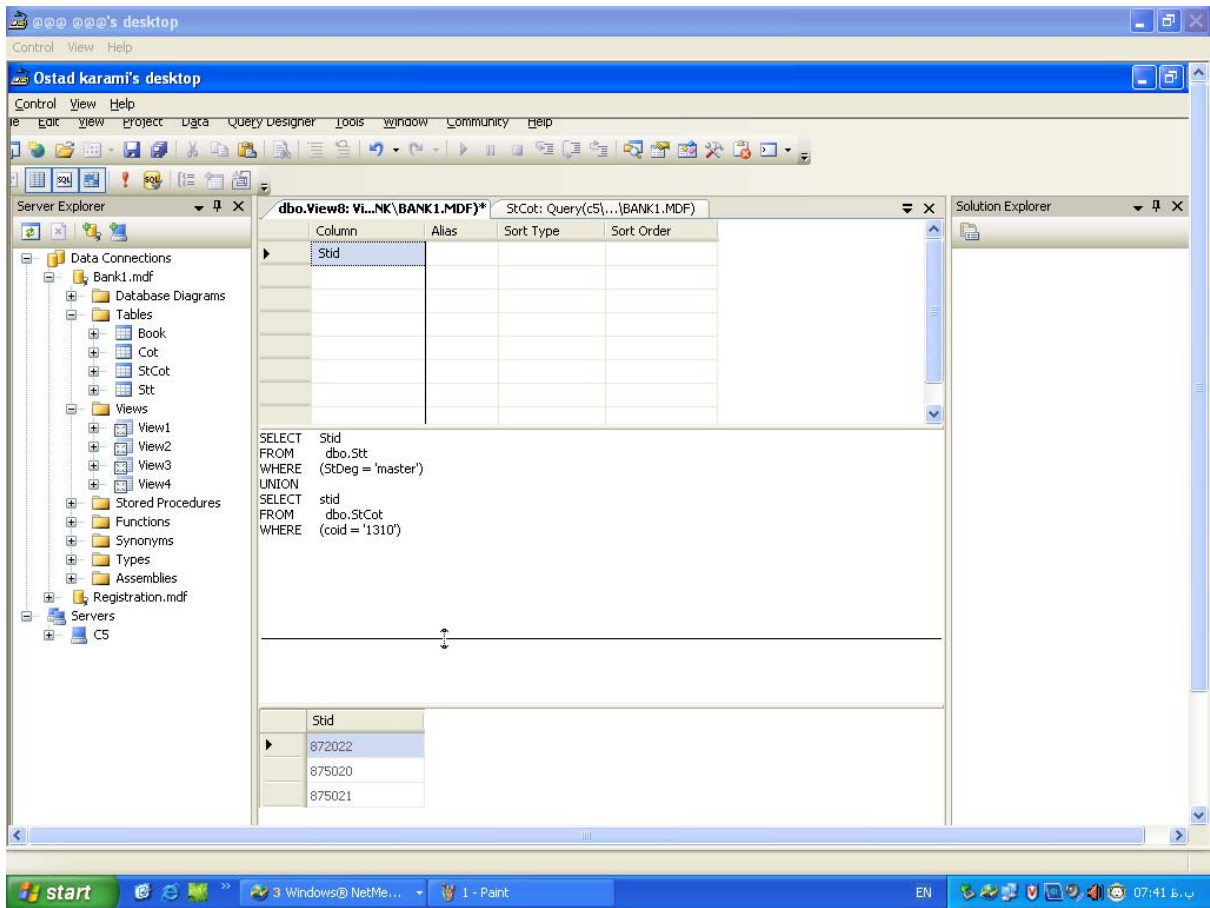
Properties

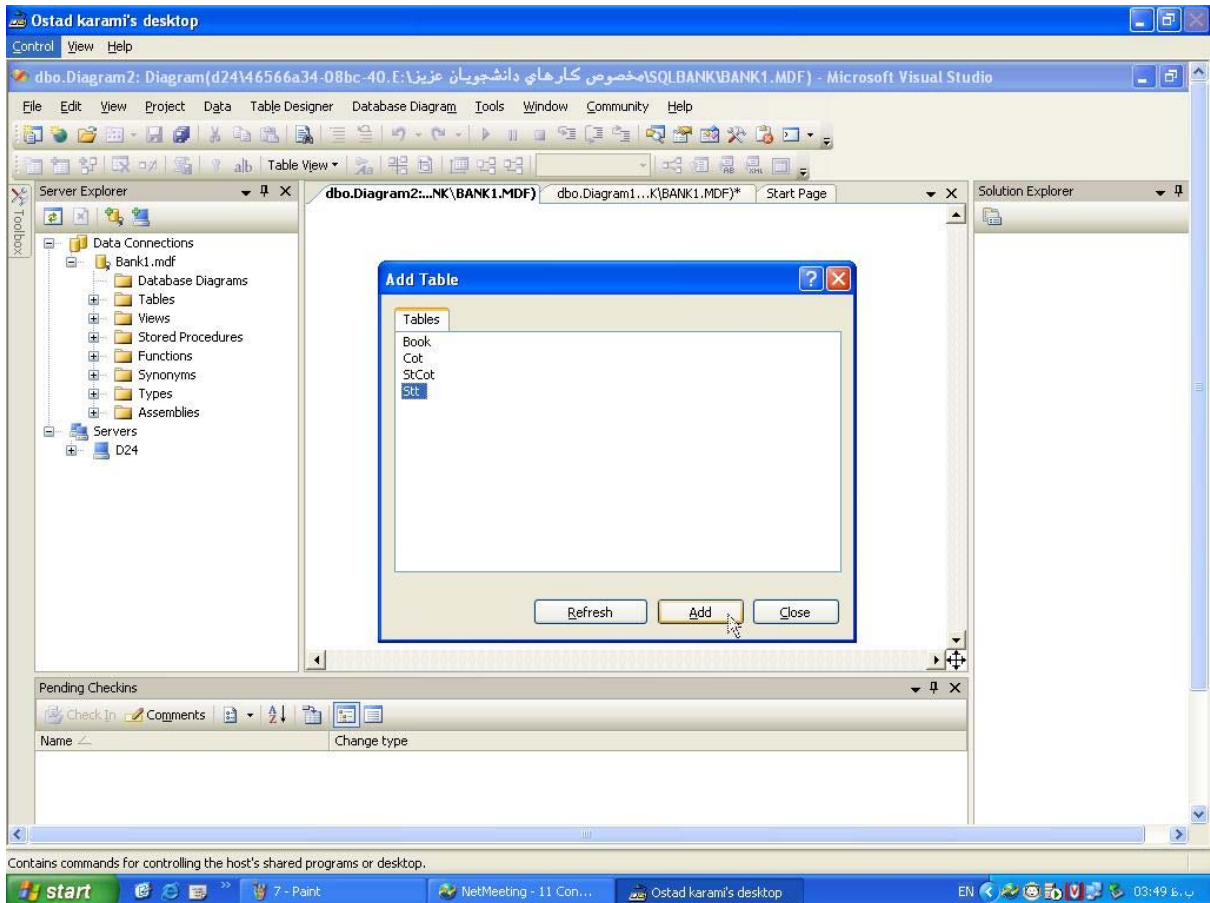
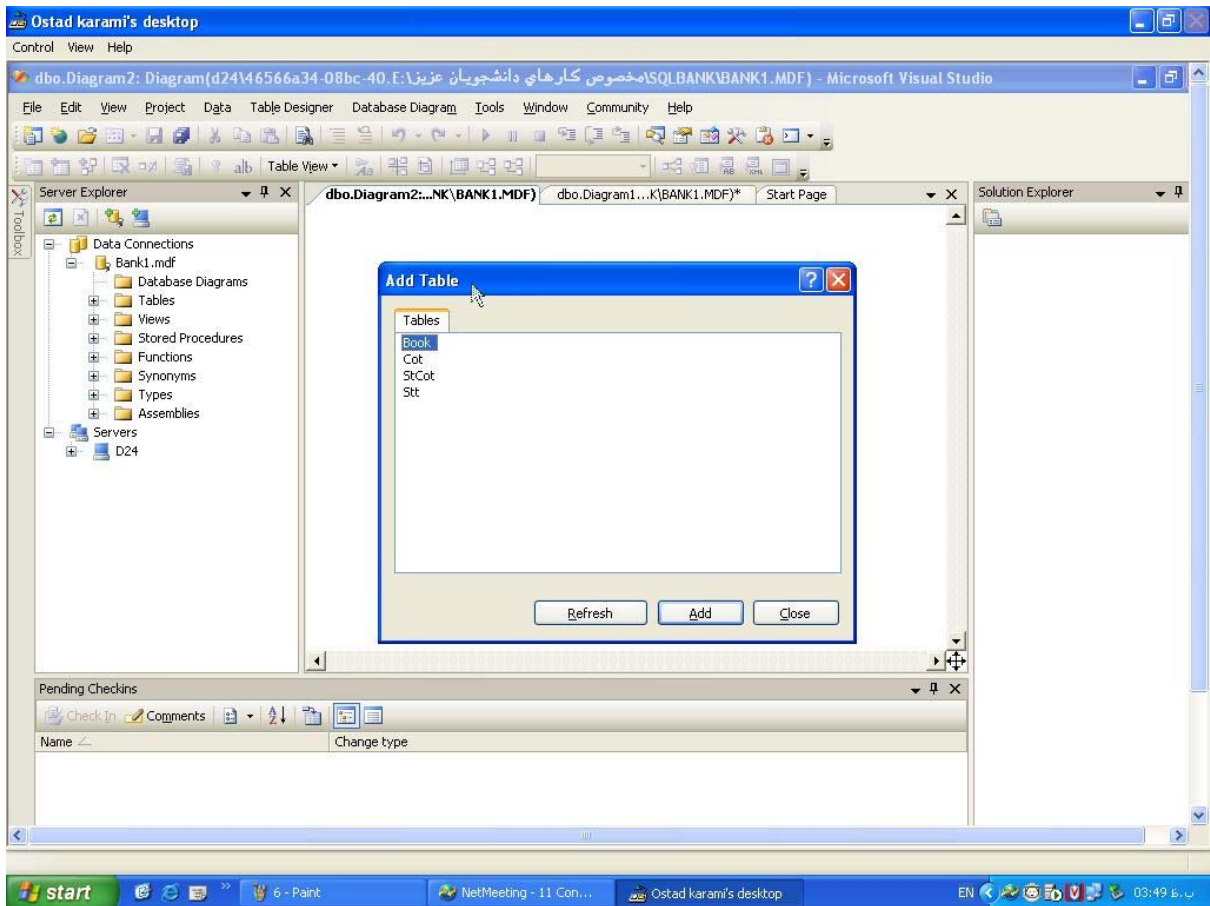
[View] dbo.View4

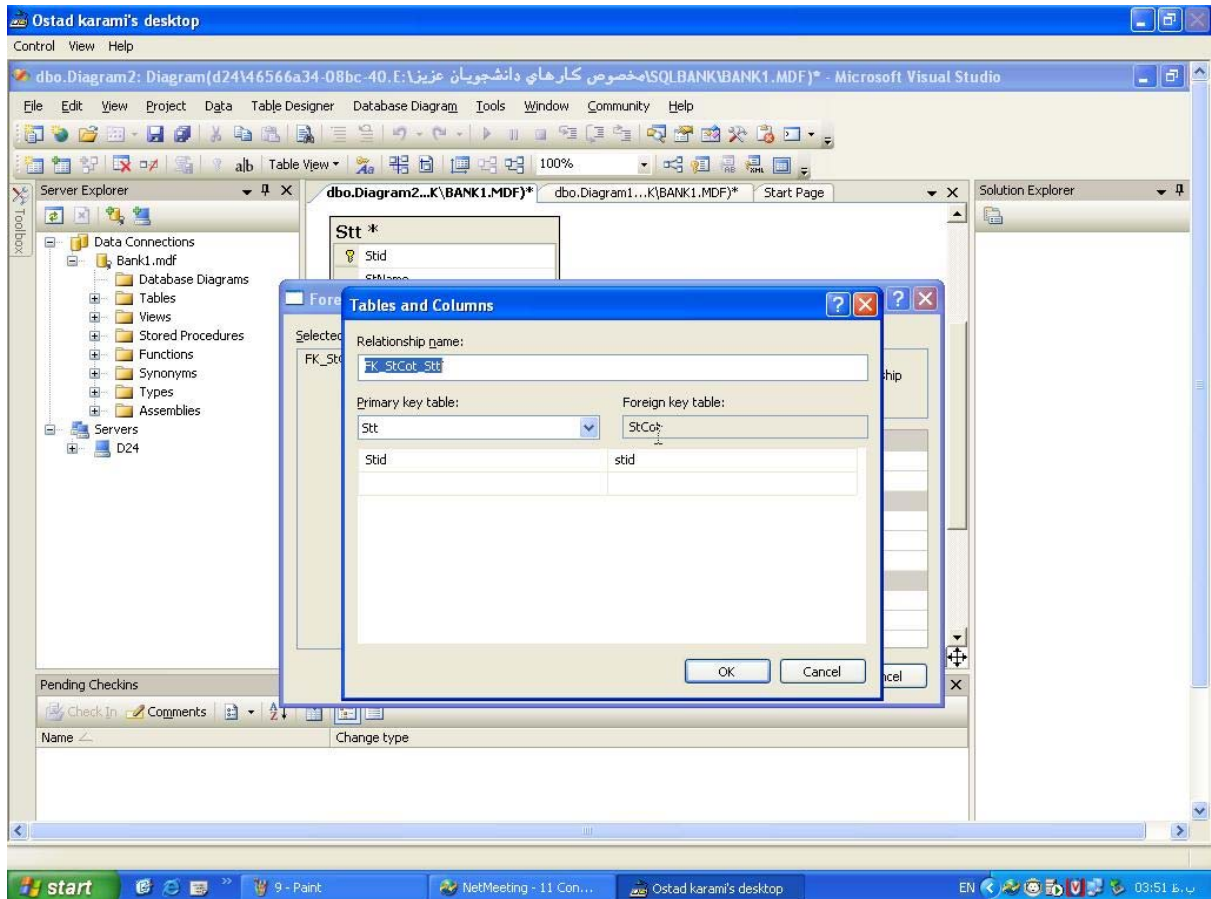
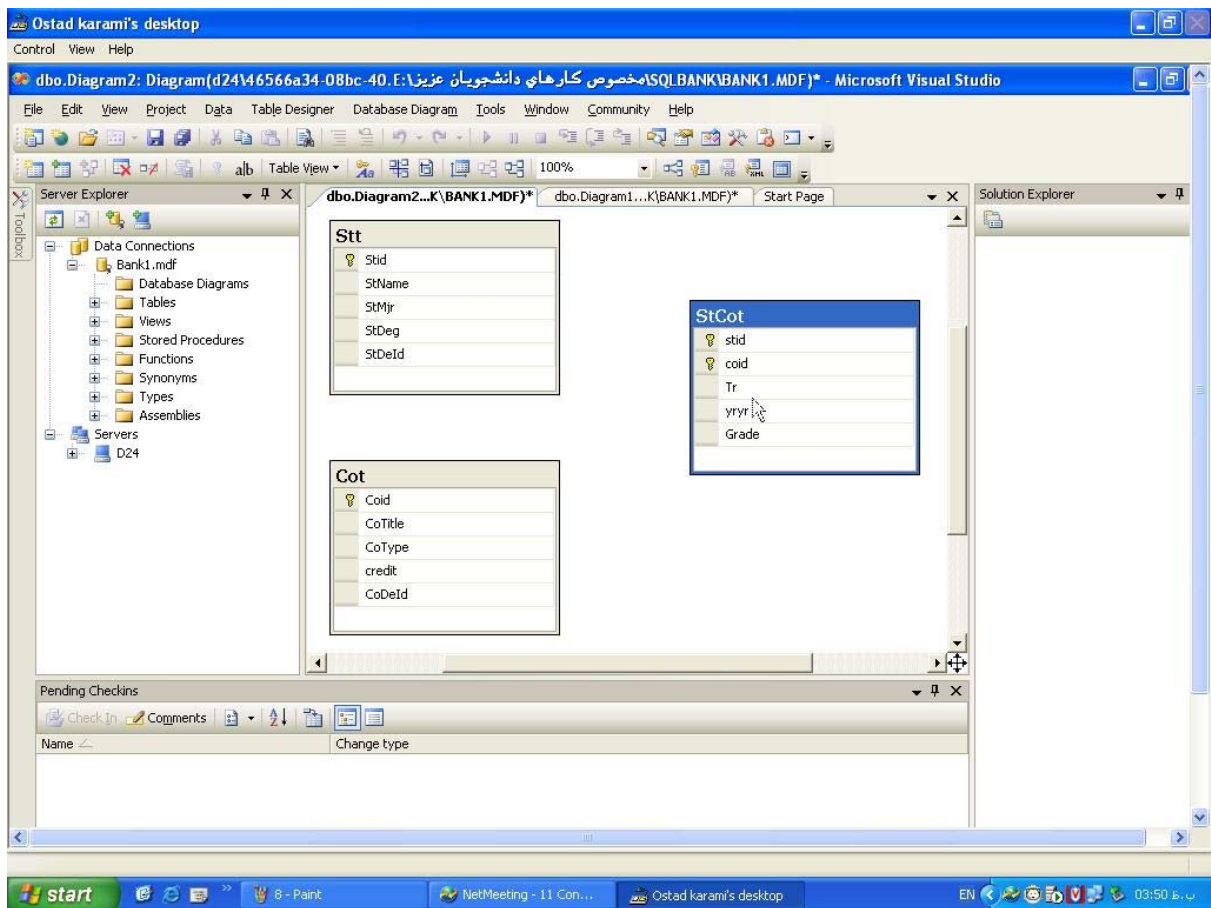
- (Name) View4
- Bind To Schema No
- Database Name E:\دانشجویان عزیزا\
- Description
- Deterministic Yes
- Distinct Values No
- GROUP BY Exter <None>
- Output All Column Yes
- Schema dbo
- Server Name d24\75e53bd7-bf

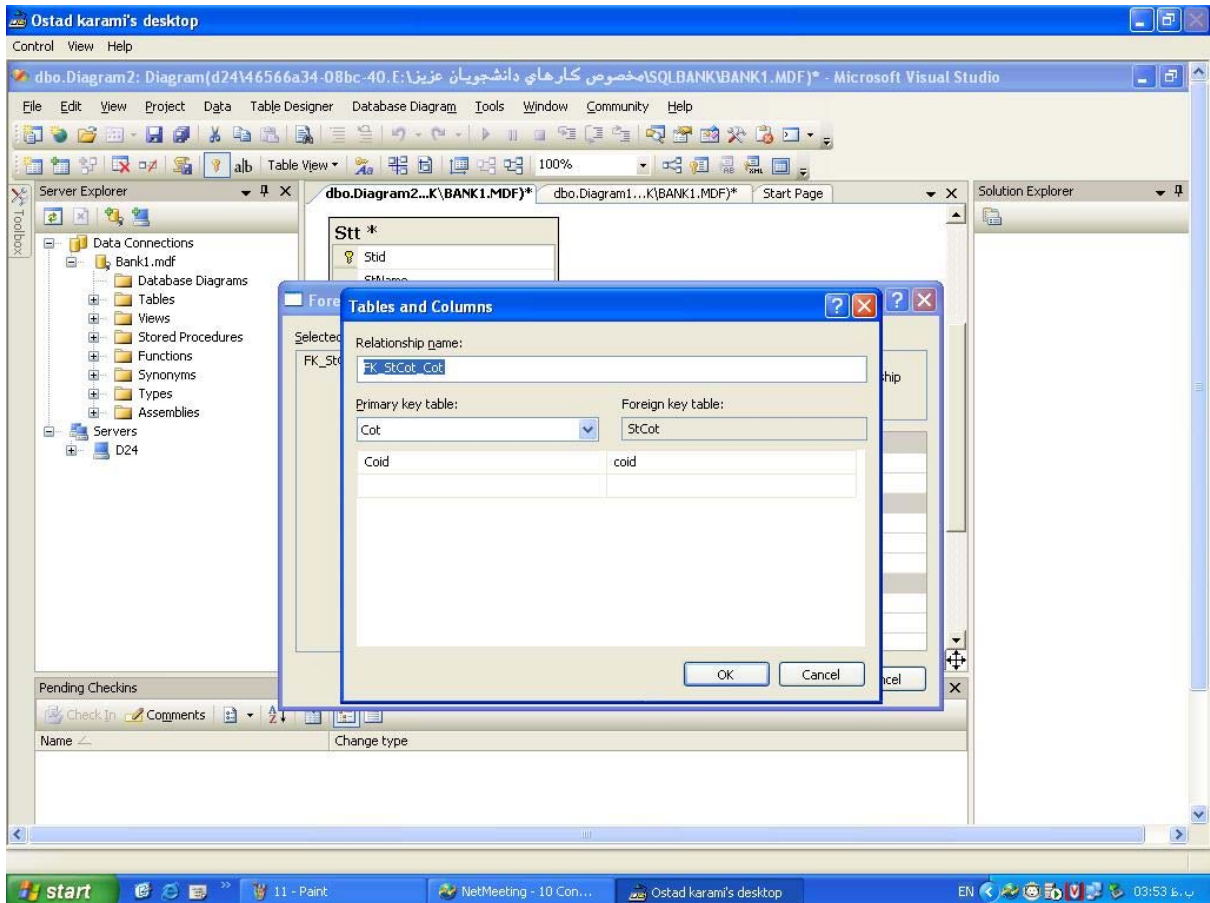
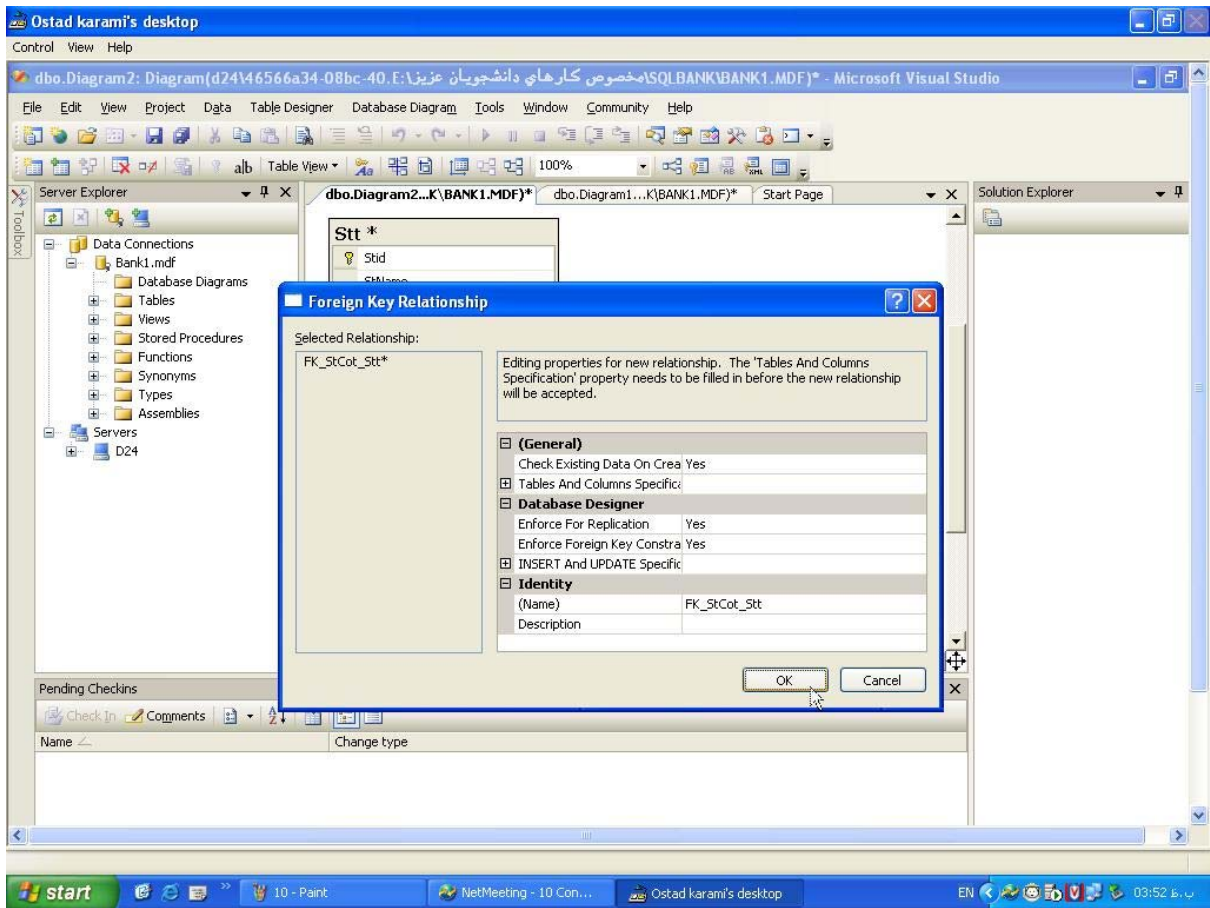
Ready



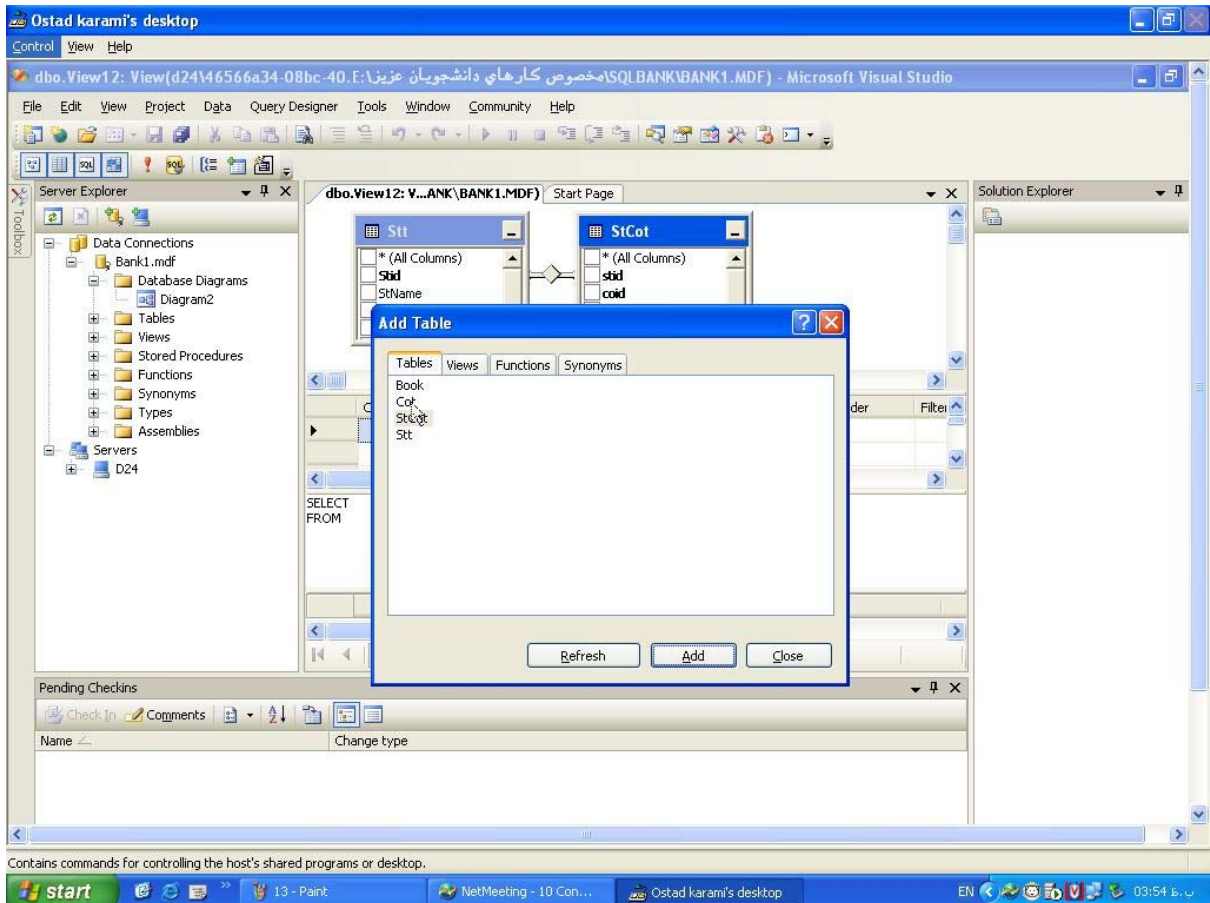
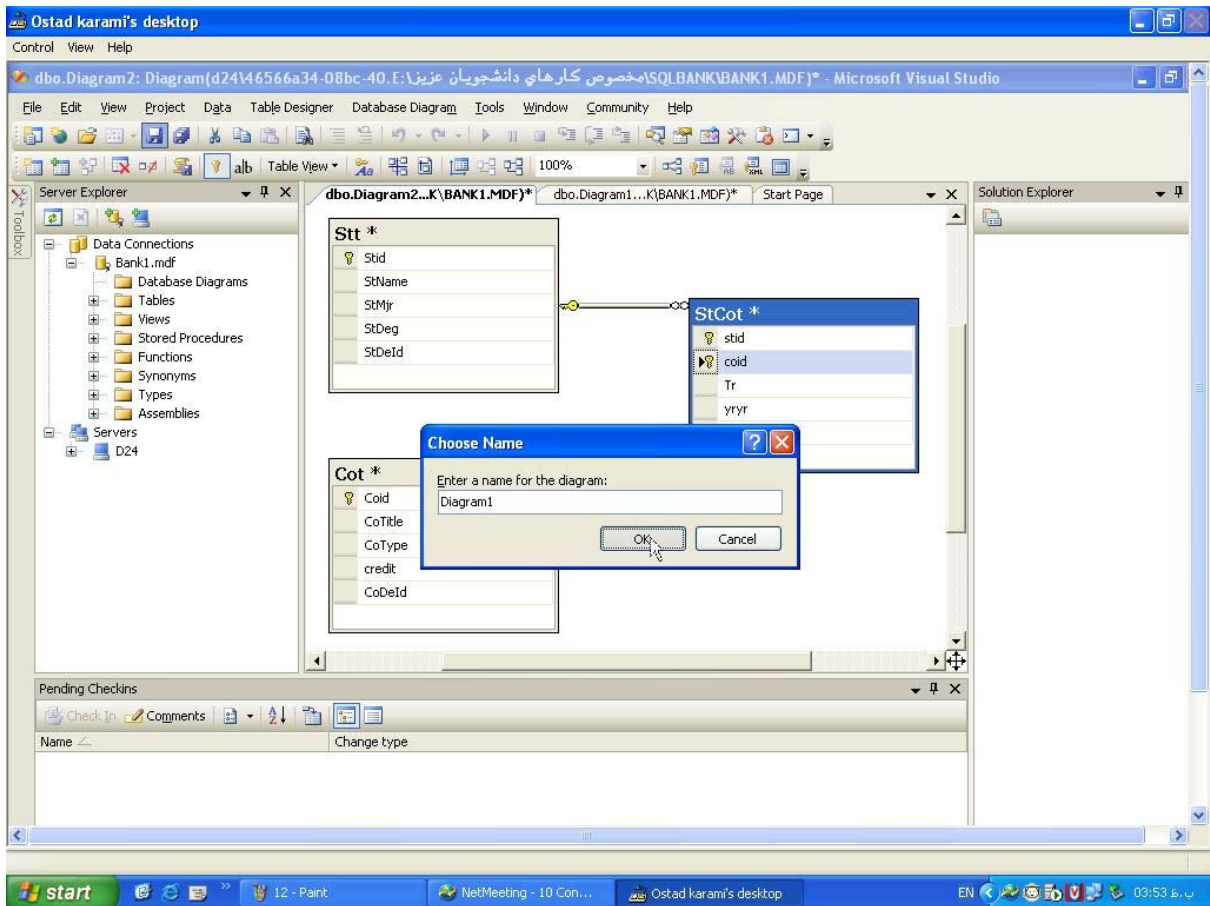




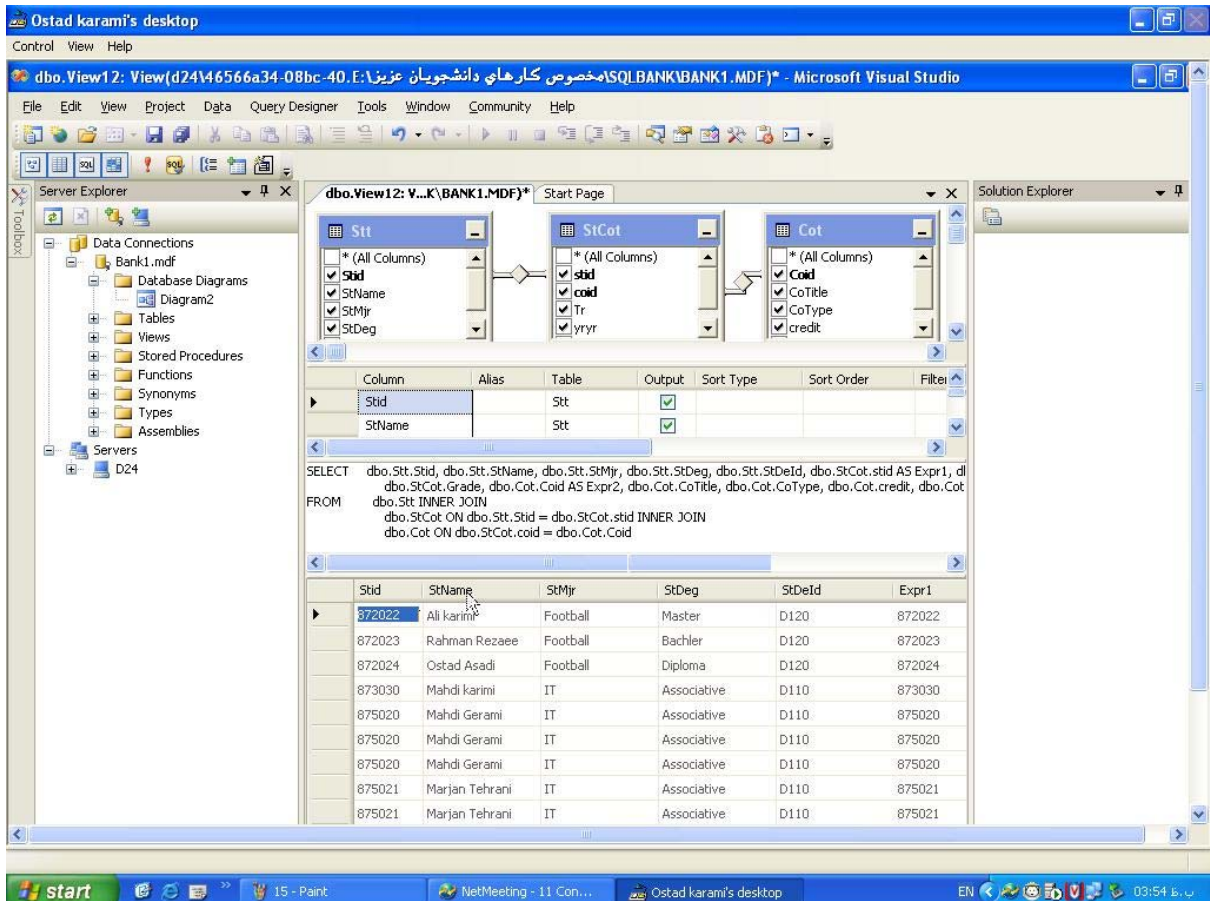
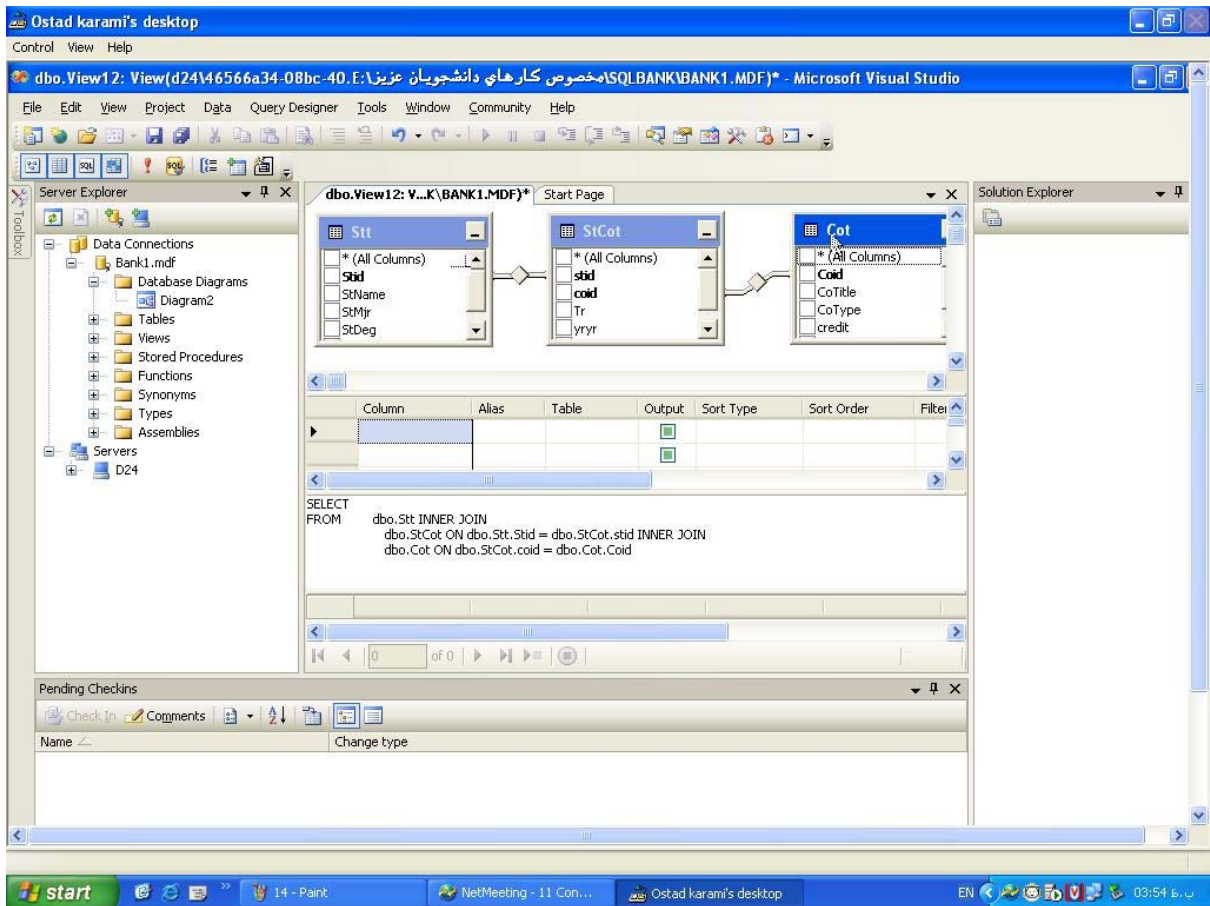








Contains commands for controlling the host's shared programs or desktop.



Ostad karami's desktop

Control View Help

dbo.View12: View(d24146566a34-08bc-40.E: \کارهای دانشجویان مخصوص SQL\_BANK\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

Bank1.mdf

Database Diagrams

Diagram2

Tables

Views

Stored Procedures

Functions

Synonyms

Types

Assemblies

Servers

D24

dbo.View12: V...K\BANK1.MDF)\* Start Page

Stt

\* (All Columns)

Stid

StName

StMjr

StDeg

StCot

\* (All Columns)

stid

coid

Tr

yrYr

Cot

\* (All Columns)

Coid

CoTitle

CoType

credit

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
Stid		Stt	<input checked="" type="checkbox"/>			= '875020'		
StName		Stt	<input checked="" type="checkbox"/>					

```

SELECT  dbo.Stt.Stid, dbo.Stt.StName, dbo.Stt.StMjr, dbo.Stt.StDeg, dbo.Stt.StDeId, dbo.StCot.stid AS Expr1, dbo.StCot.coid, dbo.StCot.Tr, dbo.StCot.CoDeId
FROM    dbo.Stt INNER JOIN
        dbo.StCot ON dbo.Stt.Stid = dbo.StCot.stid INNER JOIN
        dbo.Cot ON dbo.StCot.coid = dbo.Cot.Coid
WHERE   (dbo.Stt.Stid = '875020')

```

Stid	StName	StMjr	StDeg	StDeId	Expr1	coid	Tr	yrYr	Grade	Expr2	CoTitle	CoType	credit
875020	Mahdi Gerami	IT	Associative	D110	875020	1310	2	88-89	NULL	1310	DataBase	تخصصی	3
875020	Mahdi Gerami	IT	Associative	D110	875020	1311	2	88-89	19.5	1311	Lab DB	Practical	1
875020	Mahdi Gerami	IT	Associative	D110	875020	1312	1	88-89	NULL	1312	Web Programming	تخصصی	3

start

16 - Paint

NetMeeting - 11 Con...

Ostad karami's desktop

EN

03:56

Ostad karami's desktop

Control View Help

dbo.View13: View(d24146566a34-08bc-40.E: \کارهای دانشجویان مخصوص SQL\_BANK\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

Bank1.mdf

Database Diagrams

Diagram2

Tables

Views

Stored Procedures

Functions

Synonyms

Types

Assemblies

Servers

D24

dbo.View13: V...K\BANK1.MDF)\*

Stt

\* (All Columns)

Stid

StName

StMjr

StDeg

dbo.View12: V...K\BANK1.MDF)\* Start Page

StCot

\* (All Columns)

stid

coid

Tr

yrYr

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
			<input checked="" type="checkbox"/>					
			<input checked="" type="checkbox"/>					
			<input checked="" type="checkbox"/>					

```

SELECT
FROM    dbo.Stt INNER JOIN
        dbo.StCot ON dbo.Stt.Stid = dbo.StCot.stid

```

start

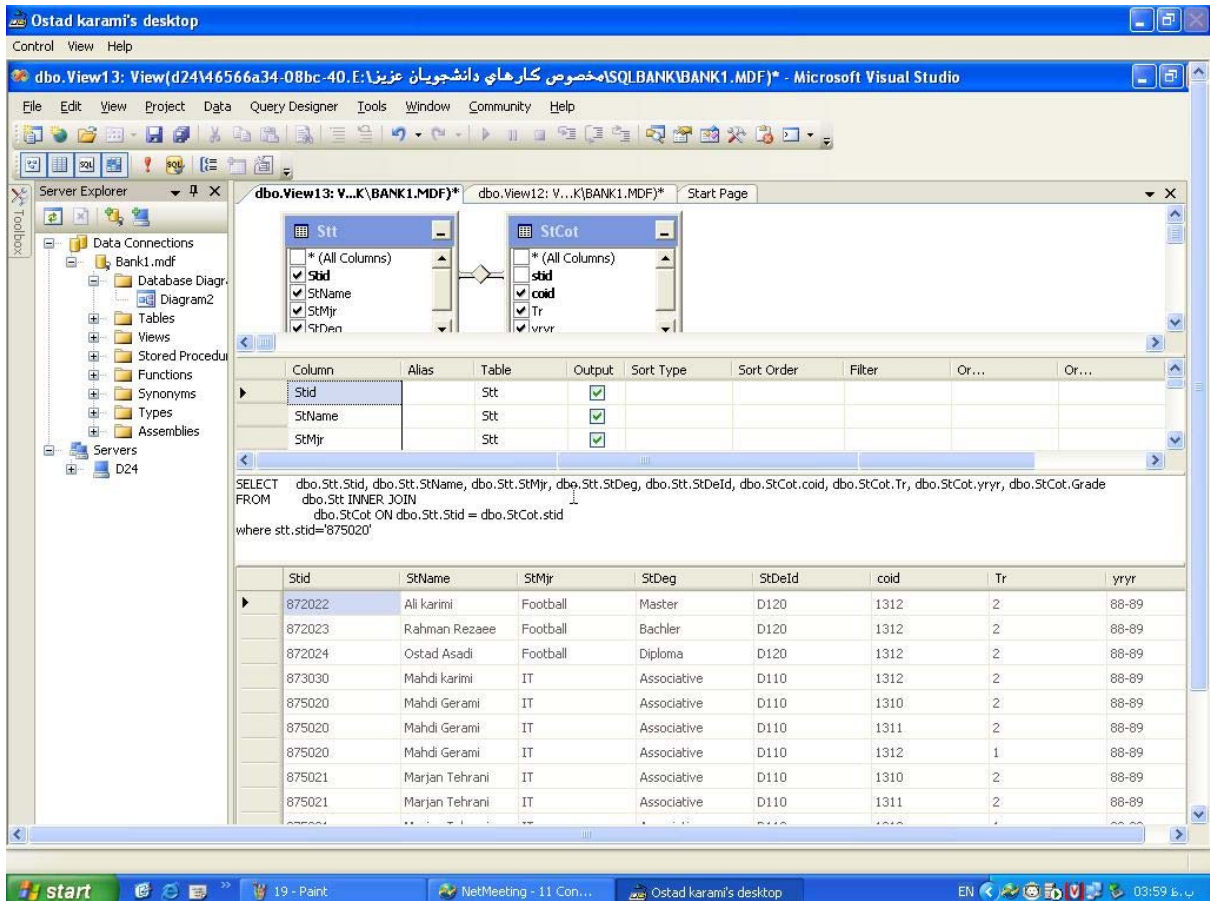
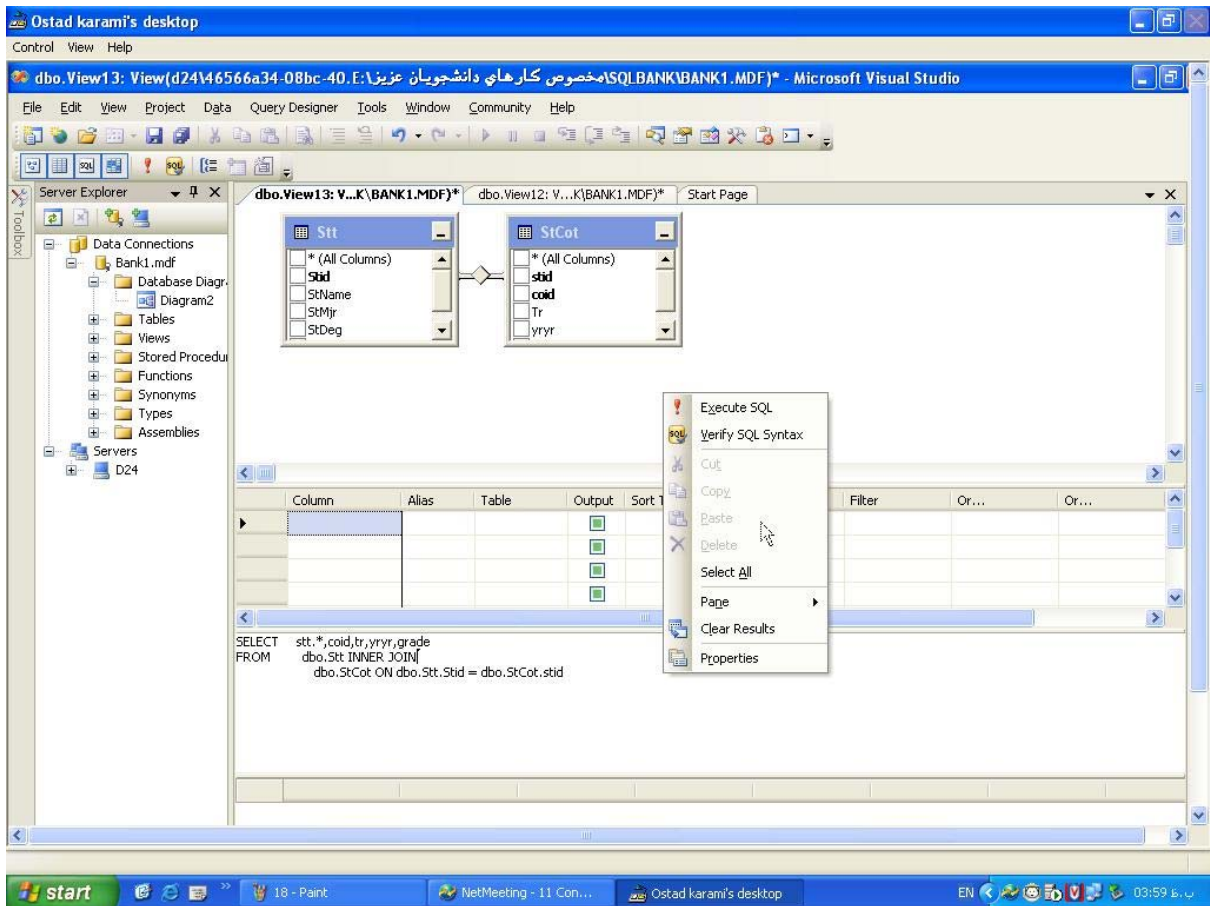
17 - Paint

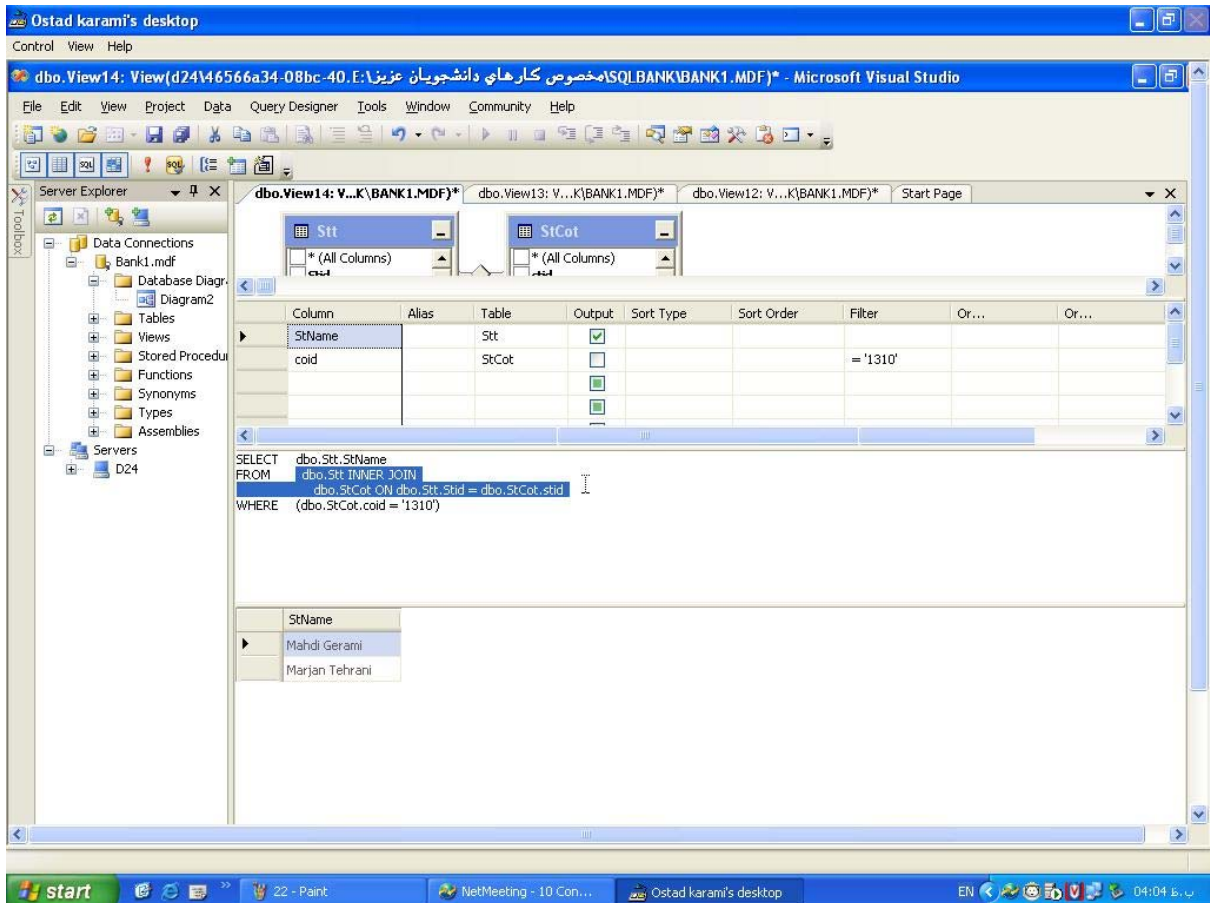
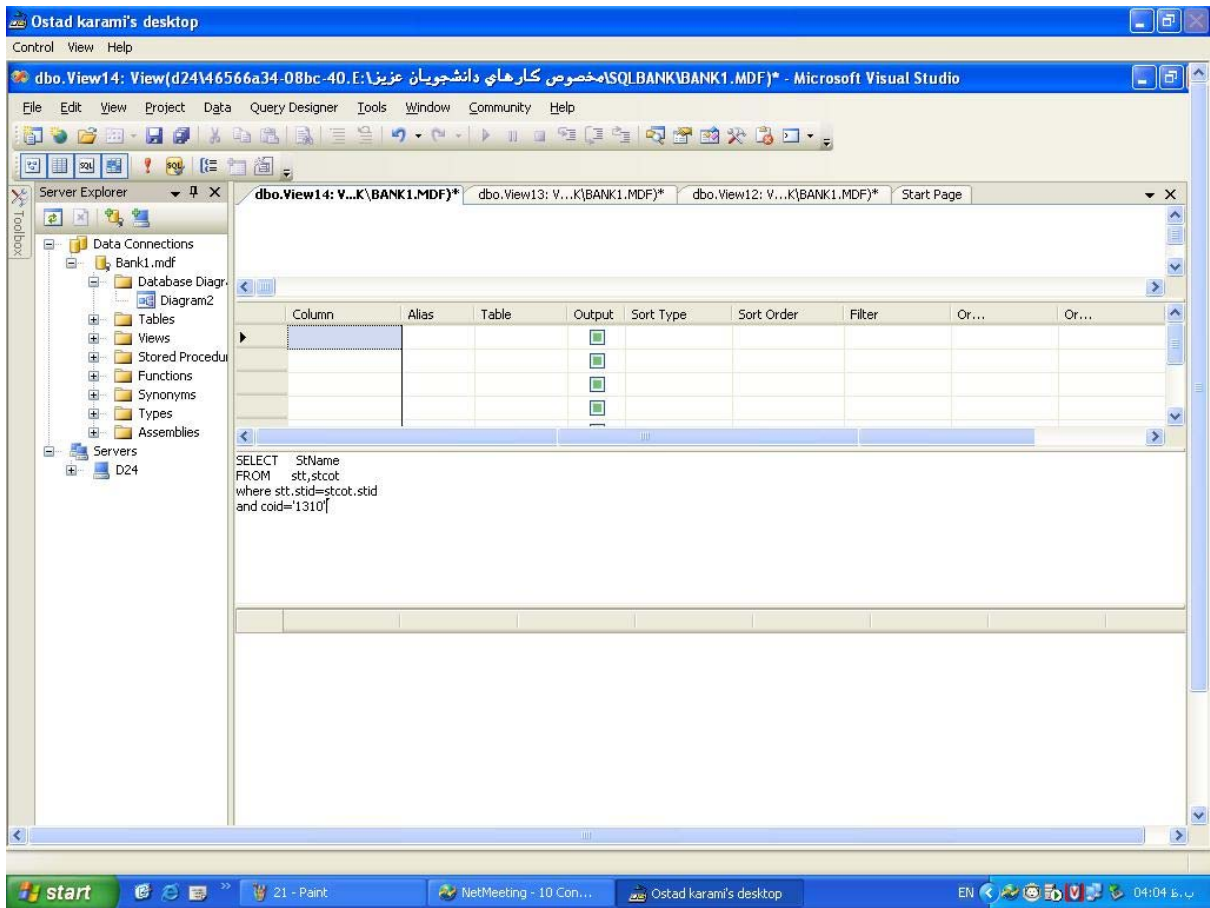
NetMeeting - 11 Con...

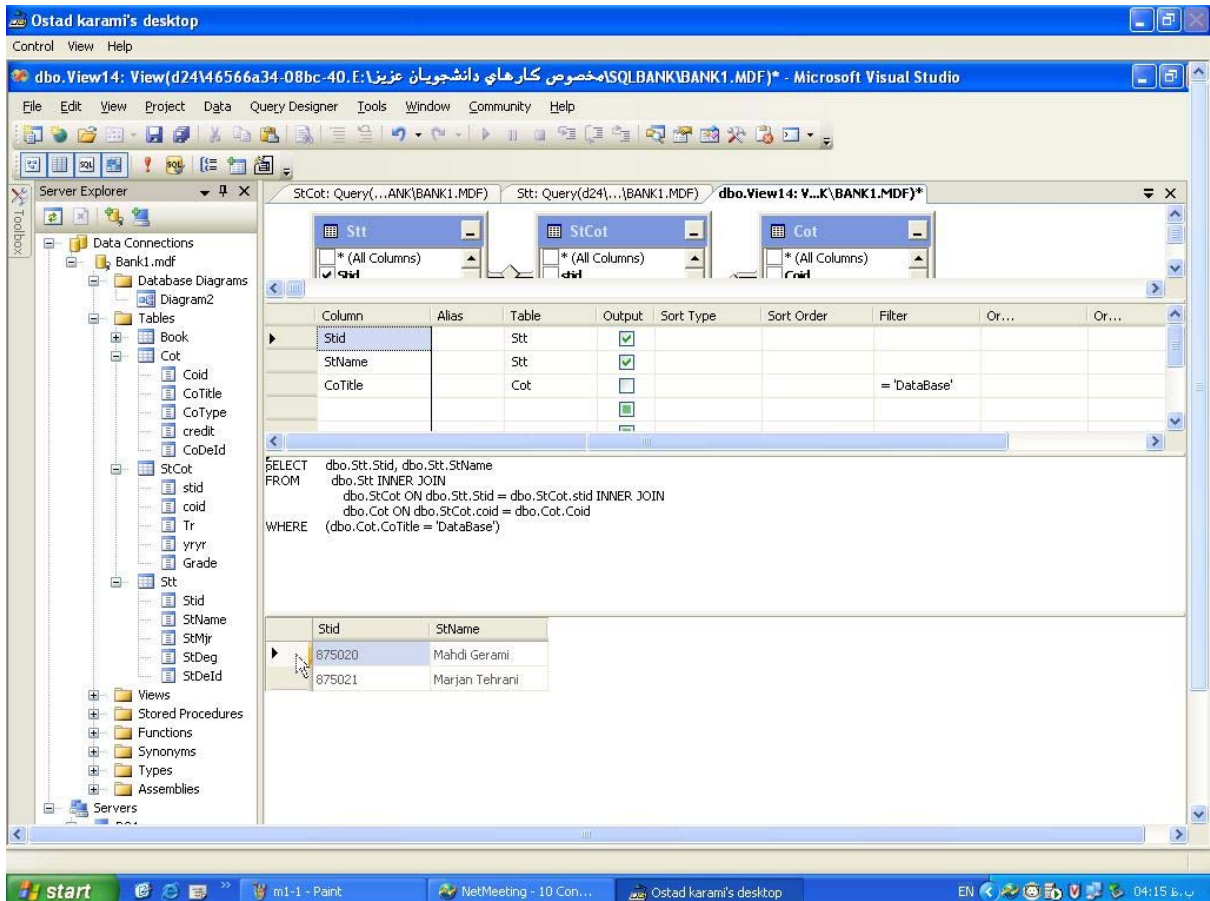
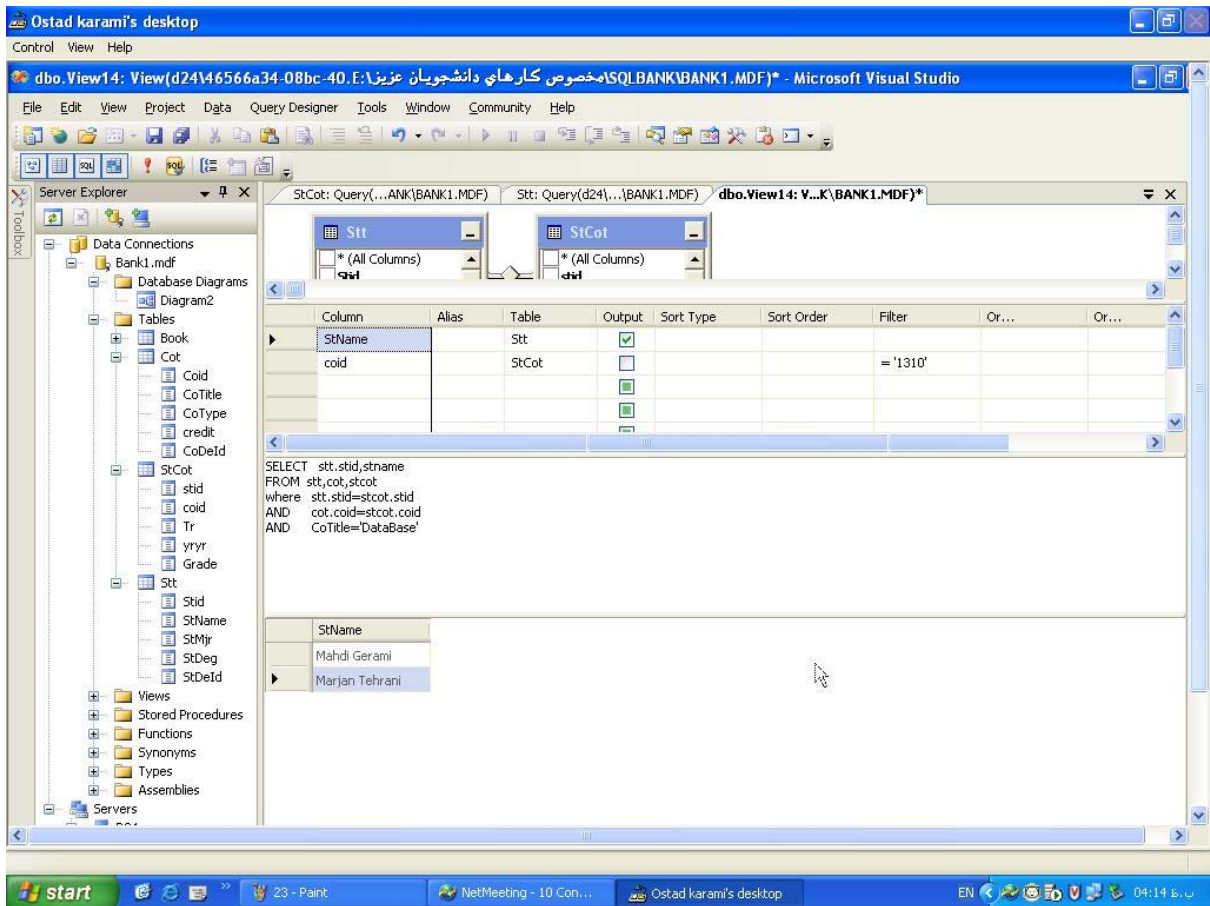
Ostad karami's desktop

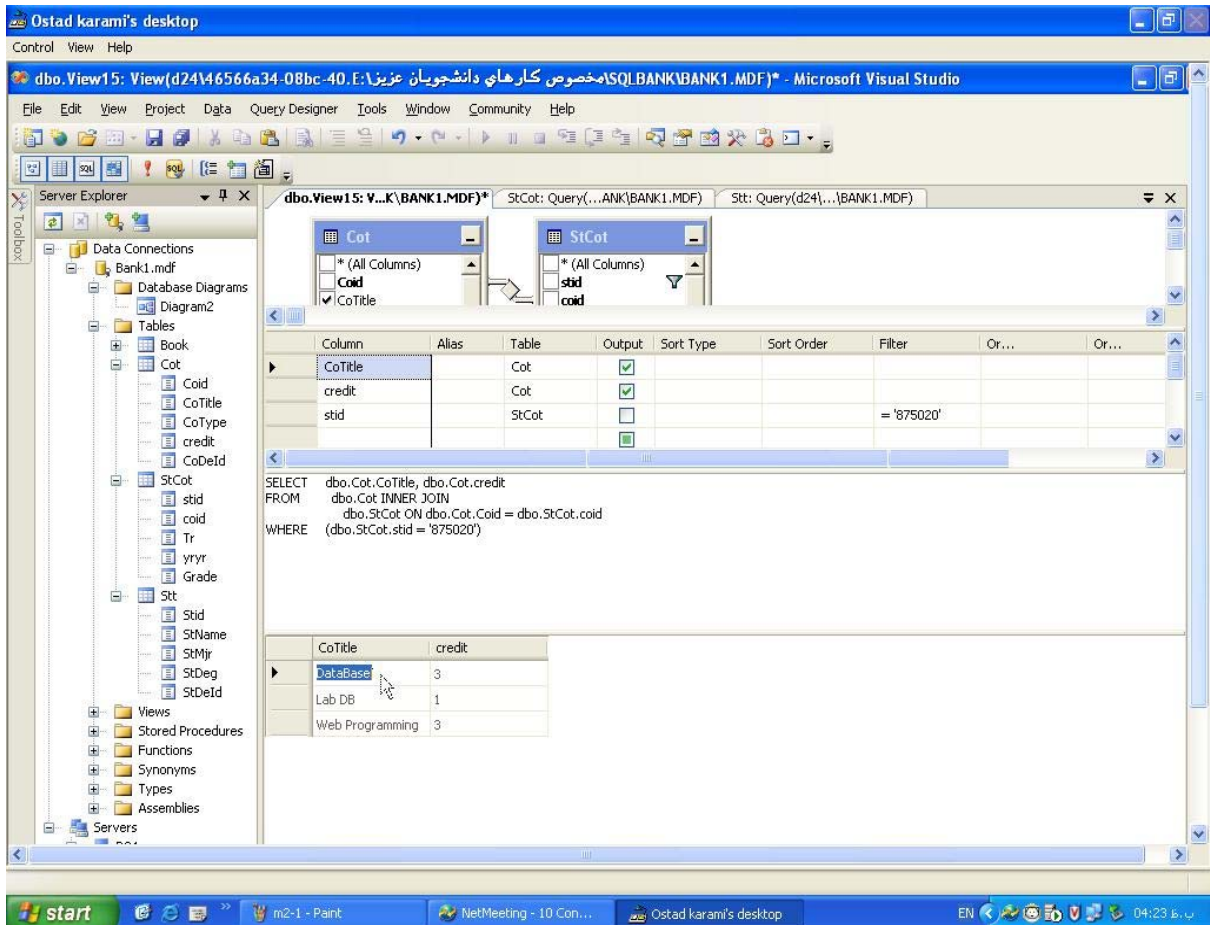
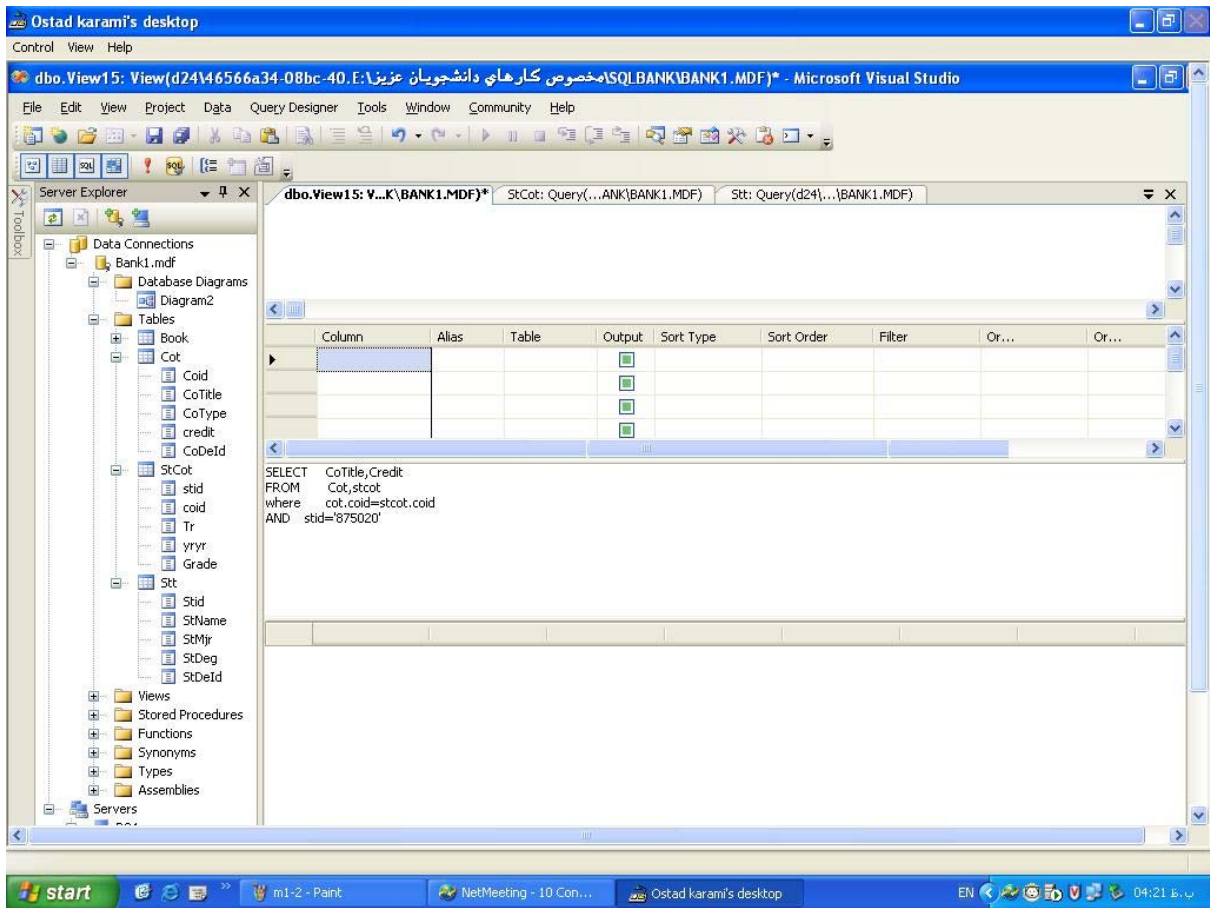
EN

03:58









dbo.View11: View(d15\ee861e6e-d138-40.F:\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

- Data Connections
  - Bank1.mdf
    - Database Diagrams
    - Tables
      - Book
      - Cot
      - StCot
      - Stt
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
  - Servers
    - D15
      - Crystal Reports Services
      - Event Logs
      - Management Classes
      - Management Events
      - Message Queues
      - Performance Counters
      - Services

dbo.View11: View(d15\ee861e6e-d138-40.F:\BANK1.MDF)\* Start Page

Stt

- \* (All Columns)
- Stid
- StName
- StMjr
- StDeg

StCot

- \* (All Columns)
- stid
- coid
- Tr
- yrjr

Cot

- \* (All Columns)
- Coid
- CoTitle
- CoType
- credit

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
Stid		Stt	<input checked="" type="checkbox"/>			

```

SELECT  dbo.Stt.Stid, dbo.Stt.StName, dbo.Stt.StMjr, dbo.Stt.StDeg, dbo.Stt.StDeId, dbo.Stt.stid AS Expr1, dbo.StCot
        dbo.StCot.Grade, dbo.Cot.Coid AS Expr2, dbo.Cot.CoTitle, dbo.Cot.CoType, dbo.Cot.credit, dbo.Cot.CoDeId
FROM      dbo.Stt INNER JOIN
        dbo.StCot ON dbo.Stt.Stid = dbo.StCot.stid INNER JOIN
        dbo.Cot ON dbo.StCot.coid = dbo.Cot.Coid
WHERE     (dbo.StCot.stid = '875020') AND (dbo.StCot.Tr = '2') AND (dbo.StCot.yrjr = '88-89')
  
```

Stid	StName	StMjr	StDeg	StDeId	Expr1	coid	Tr	yrjr	Grade
875020	Mahdi Gerami	IT	Associative	D110	875020	1310	2	88-89	18
875020	Mahdi Gerami	IT	Associative	D110	875020	1311	2	88-89	19.5

1 of 2 Cell is Read Only.

Ready

start m2-2 - Paint NetMeeting - 10 Co... Ostad karami's desk... dbo.View11: View(d... EN 04:39 B...

dbo.View11: View(d15\ee861e6e-d138-40.F:\BANK1.MDF)\* - Microsoft Visual Studio

File Edit View Project Data Query Designer Tools Window Community Help

Server Explorer

- Data Connections
  - Bank1.mdf
    - Database Diagrams
    - Tables
      - Book
      - Cot
      - StCot
      - Stt
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
  - Servers
    - D15
      - Crystal Reports Services
      - Event Logs
      - Management Classes
      - Management Events
      - Message Queues
      - Performance Counters
      - Services

dbo.View11: View(d15\ee861e6e-d138-40.F:\BANK1.MDF)\* Start Page

Stt

- \* (All Columns)
- Stid
- StName
- StMjr
- StDeg

StCot

- \* (All Columns)
- stid
- coid
- Tr
- yrjr

Cot

- \* (All Columns)
- Coid
- CoTitle
- CoType
- credit

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
Stid		Stt	<input checked="" type="checkbox"/>			

```

SELECT  dbo.Stt.Stid, dbo.Stt.StName, dbo.Stt.StMjr, dbo.Stt.StDeg, dbo.Stt.StDeId, dbo.Stt.stid AS Expr1, dbo.StCot
        dbo.StCot.Grade, dbo.Cot.Coid AS Expr2, dbo.Cot.CoTitle, dbo.Cot.CoType, dbo.Cot.credit, dbo.Cot.CoDeId
FROM      dbo.Stt INNER JOIN
        dbo.StCot ON dbo.Stt.Stid = dbo.StCot.stid INNER JOIN
        dbo.Cot ON dbo.StCot.coid = dbo.Cot.Coid
WHERE     (dbo.StCot.stid = '875020') AND (dbo.StCot.Tr = '2') AND (dbo.StCot.yrjr = '88-89')
  
```

Stid	StName	StMjr	StDeg	StDeId	Expr1	coid	Tr	yrjr	Grade
875020	Mahdi Gerami	IT	Associative	D110	875020	1310	2	88-89	18
875020	Mahdi Gerami	IT	Associative	D110	875020	1311	2	88-89	19.5

1 of 2 Cell is Read Only.

Ready

start m2-2 - Paint NetMeeting - 10 Co... Ostad karami's desk... dbo.View11: View(d... EN 04:39 B...

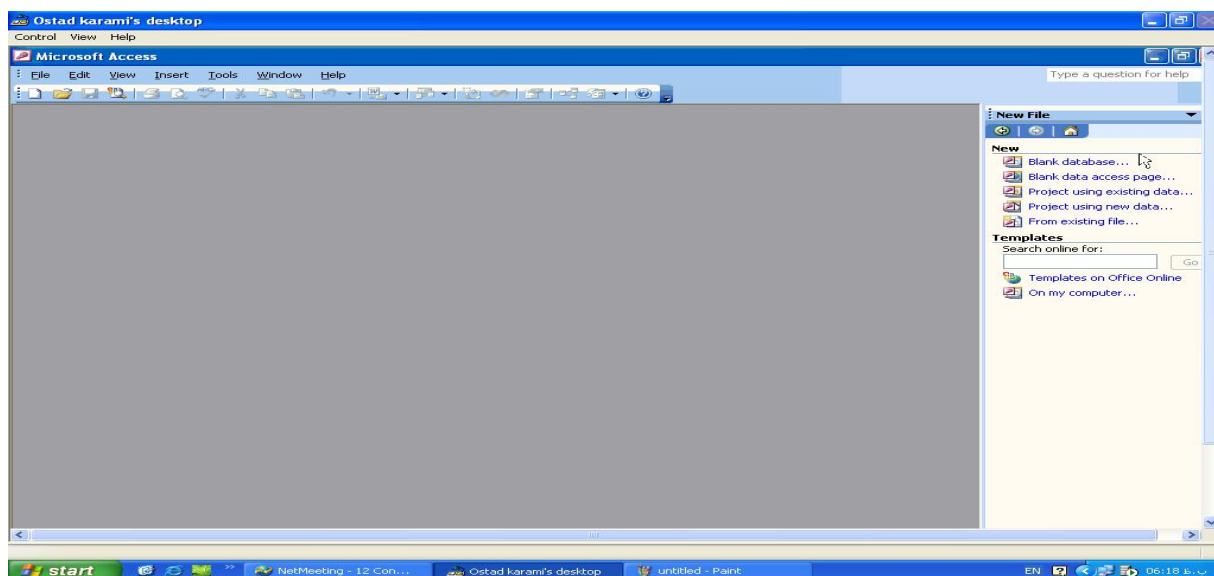


## پیوست ۳: پایگاه داده Access

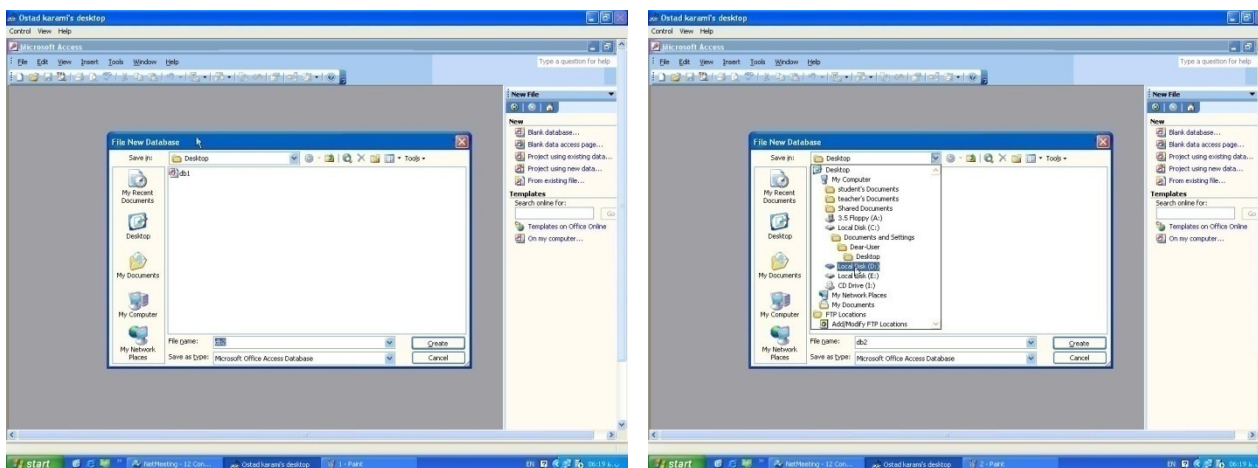
- ایجاد یک پایگاه داده (DBSM) در نرم افزار Microsoft Access ۲۰۰۳
- ایجاد جدول فردی (Fardi)

Table : Fardi			
Field name	Data Type	Field Zise	Caption
Code	Number	Integer	کدشخص
Names	Text	۳۰	نام شخص
Tel	Text	۲۰	تلفن
Address	Memo	*	آدرس

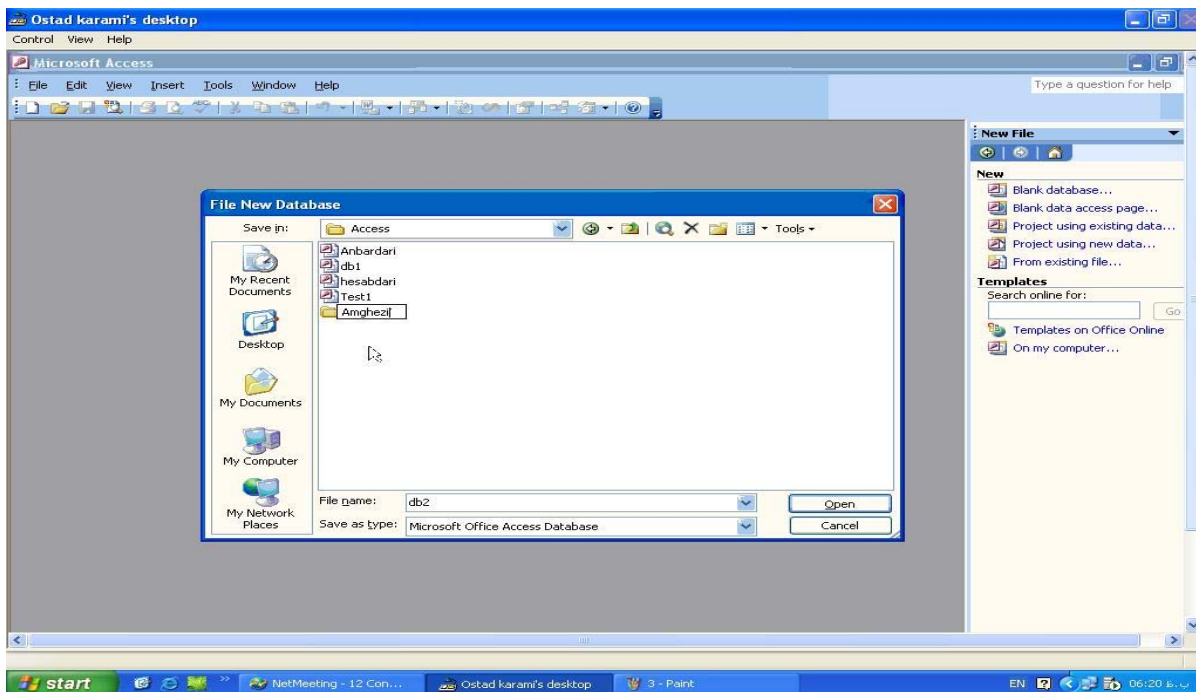
۱. ابتدا برنامه ۲۰۰۳ Access را باز نماید.
۲. گزینه Blank Database را از پنجره سمت راست انتخاب نماید.



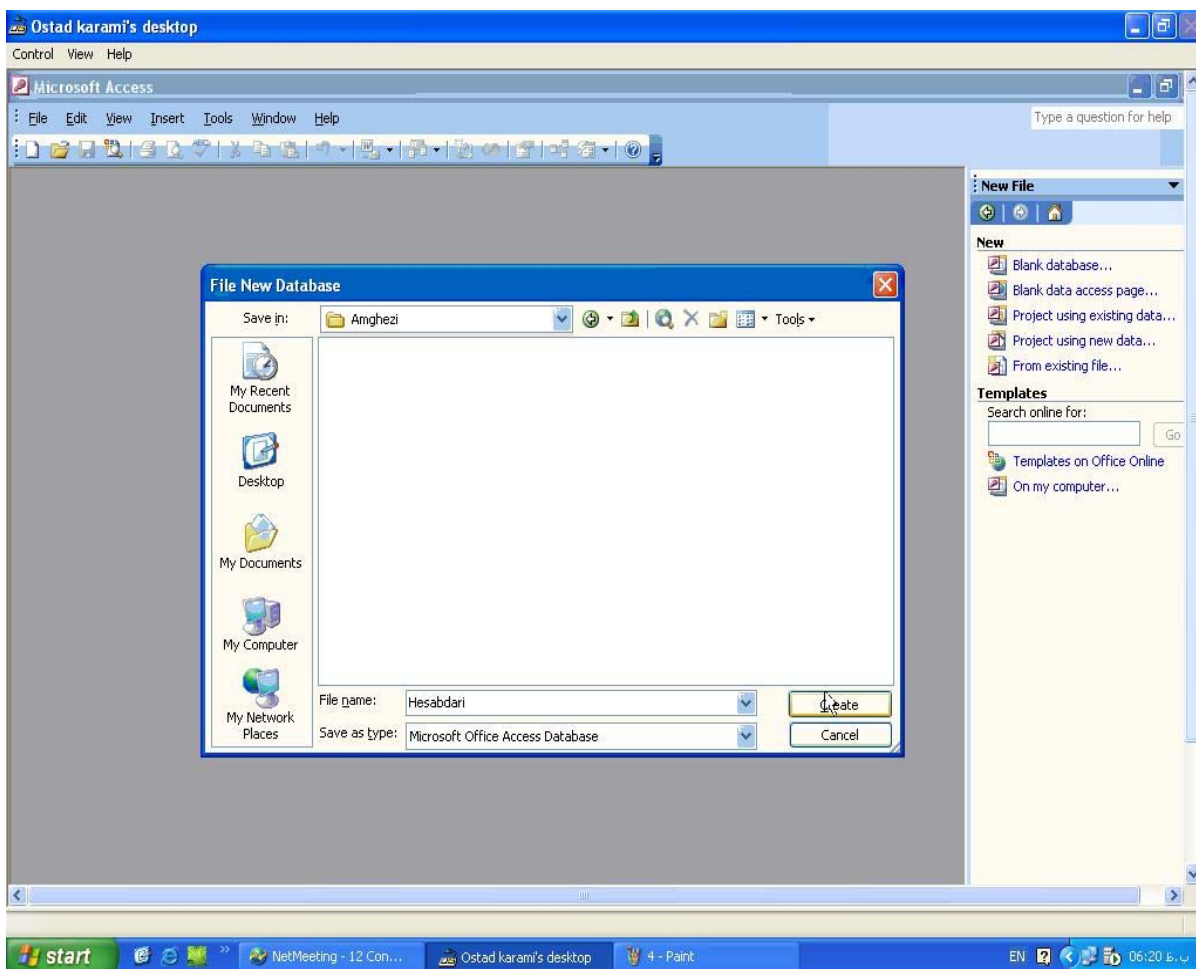
۳. در پنجره ظاهر شده مسیر مورد نظر را انتخاب نماید.



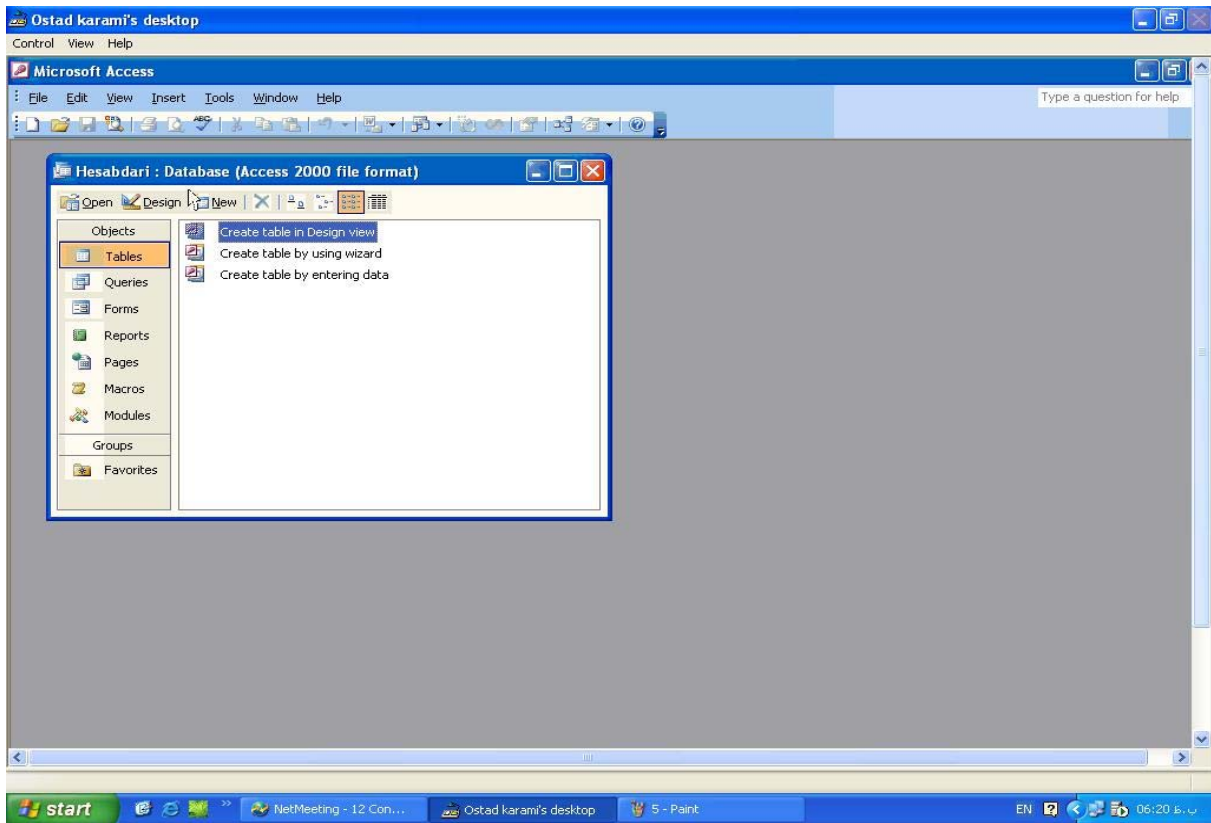
۴. در مسیر انتخابی پوشه مورد نظر را ایجاد نماید.



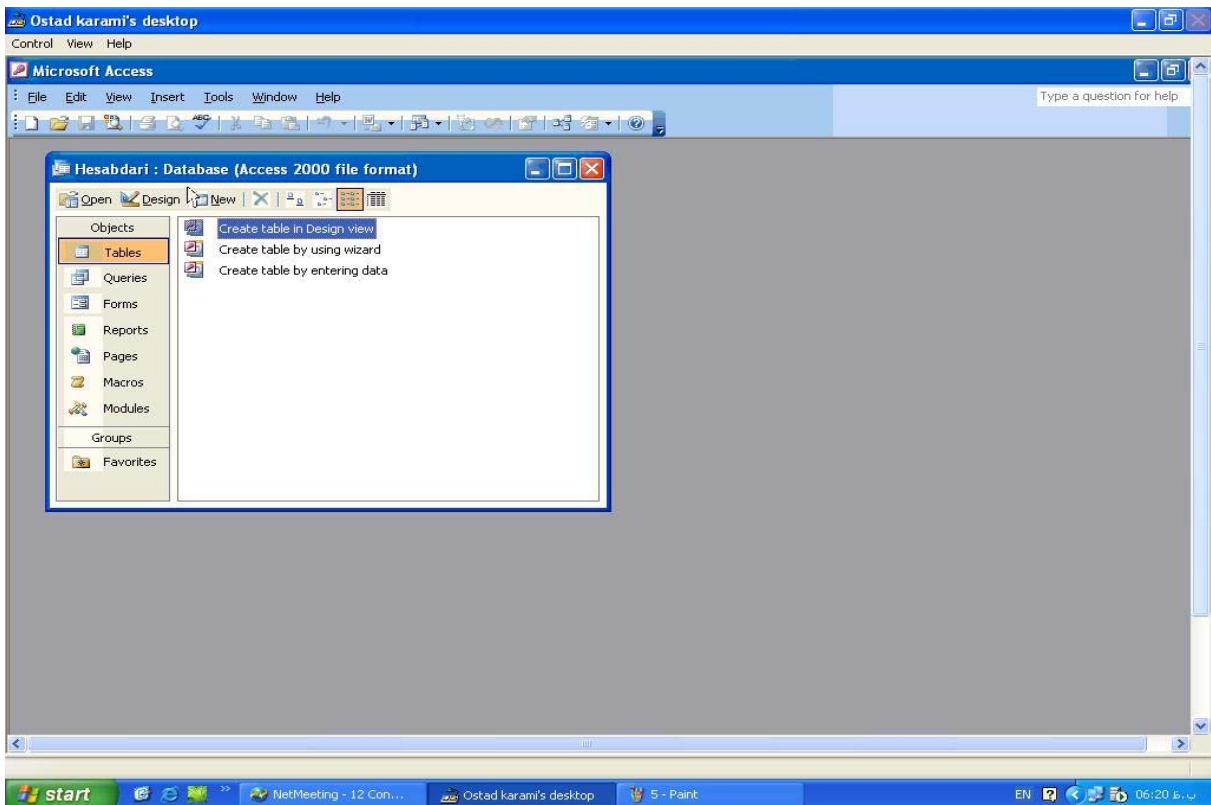
۵. و در آخر در قسمت File name نام پایگاه داده مورد نظر را تایپ نموده تا ایجاد گردد. (Hesabdari)



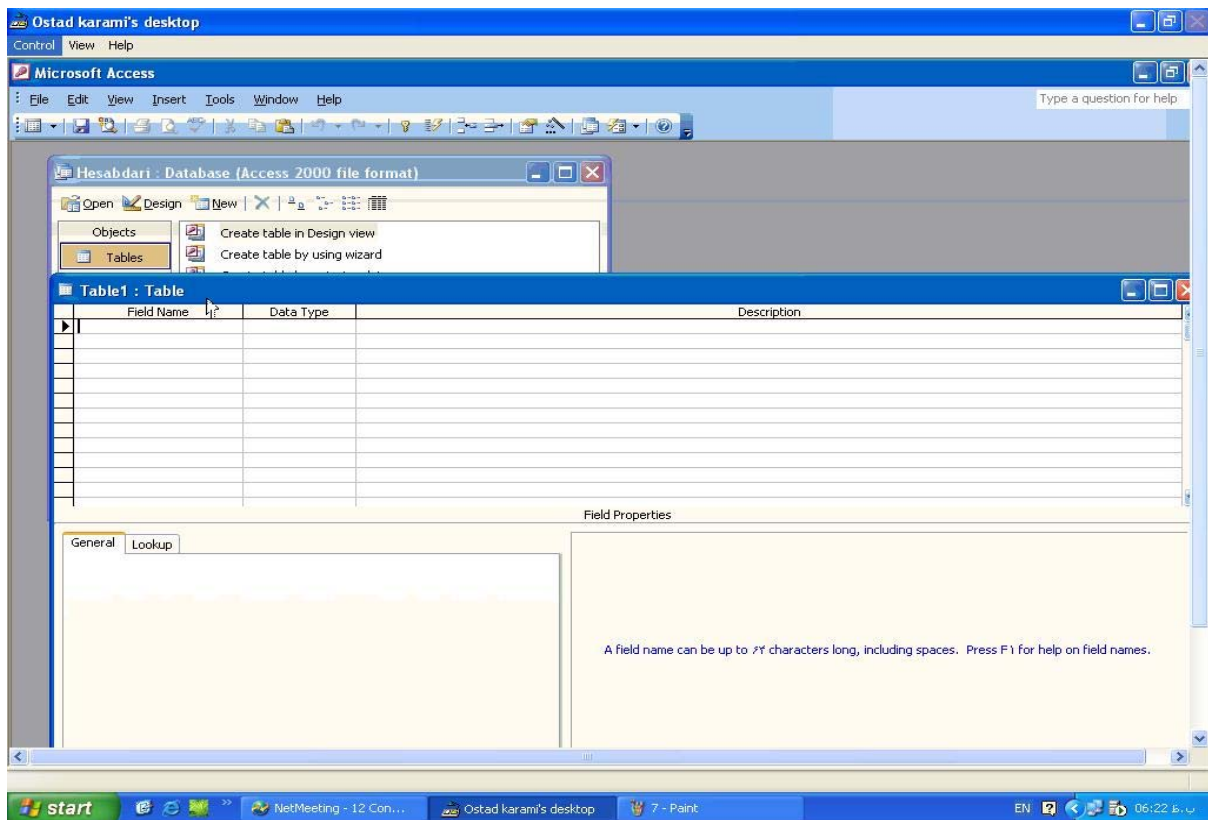
۶. بعد از ایجاد پایگاه داده وارد برنامه Access می شویم.



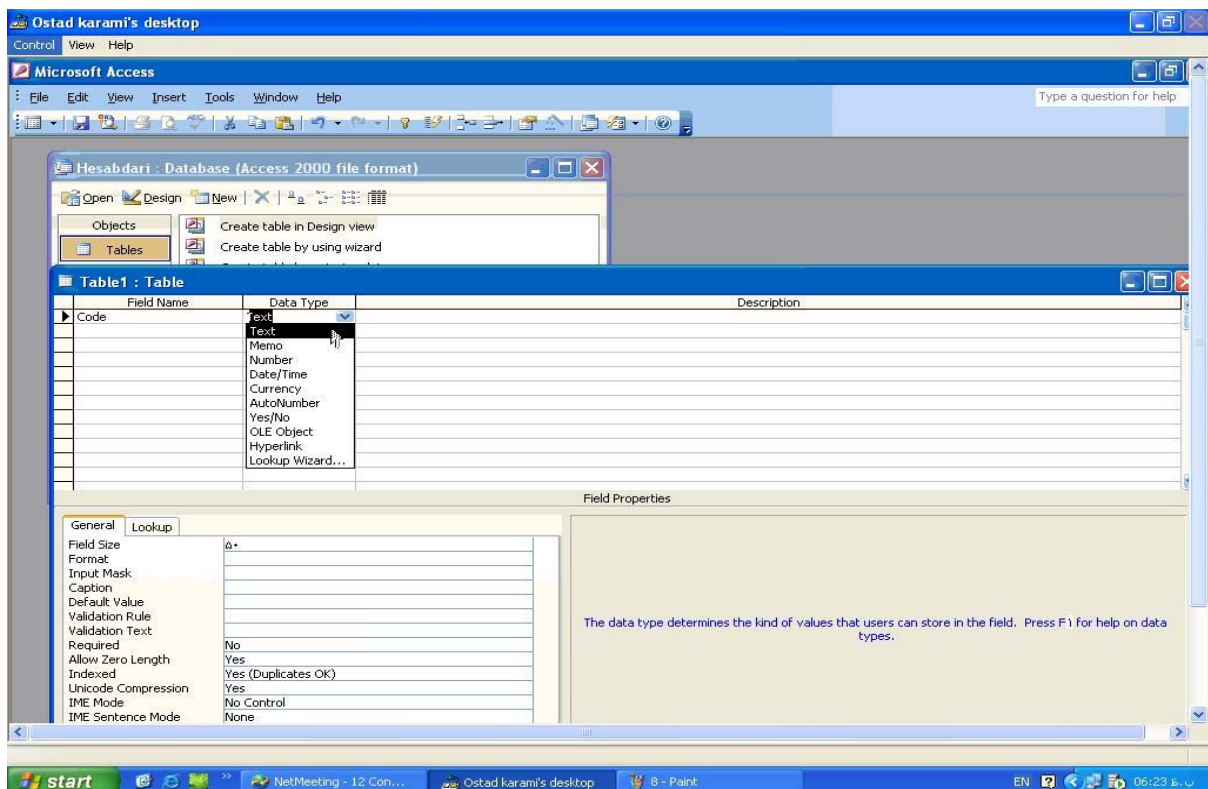
۷. جهت ایجاد یک جدول از پنجره سمت راست گزینه Table را انتخاب کرده و سپس از داخل پنجره گزینه Creat table in design view را انتخاب نماید.



۸. در پنجره ظاهر شده می توان نام Field ها و نوع (Type) آنها را درج نمود تا جدول اطلاعاتی ایجاد گردد.



۹. همانطور که می بینید پس از درج فیلد مورد نظر از قسمت باز شو (Combobox) می توان نوع فیلد را انتخاب نمود.



## تعریف انواع داده ها:

هر داده ای نوع خاصی دارد. هر فیلد جدول، داده هایی از یک نوع خاص را می تواند در خود جای دهد. مثلاً برای فیلدی با اسم "نام کارگر" باید داده هایی از نوع متن تعریف شود تا تمامی رکوردهای ثبت شده در این فیلد دارای یک نوع داده و آن هم از نوع متن باشند. لذا هرگاه جدولی طراحی می کنیم باید نوع داده هر فیلد را از فهرست پایین جهنده ستون **Data Type** انتخاب کنیم. (برای نمایش فهرست ستون **Data Type** کافی است بر روی فلش موجود در این ستون کلیک کنیم تا فهرستی از نوع تمامی داده هایی که می شود در اکسس ثبت کرد برای ما نشان داده شود)

توجه: هنگامی که موردی را از فهرست بازشونده **Data Type** انتخاب می کنیم اکسس فهرستی از اطلاعات را در بخش زیرین پنجره نمایش می دهد که به آنها (**Field Properties** یا خصوصیات فیلد) گفته می شود. مثلاً وقتی ما فیلدی از نوع (**Text** متن) را از ستون بازشونده انتخاب می کنیم. در زیر همین پنجره خصوصیات فیلد مربوط به متن برای ما نشان داده می شود. این خصوصیات عنوان درس بعدی ما می باشد که به طور مفصل شرح داده های خواهد شد که در این قسمت ما تنظیمات جالب توجه ای داریم.

حال در زیر به تشریح تمامی انواع داده ها می پردازیم:

### داده از نوع (**Text** متن) :

از داده نوع **Text** در فیلدهایی که حروف، واژه ها، اسامی و عبارات را نگه می دارد استفاده می شود. لازم نیست که محتوای فیلد متن فقط متون الفبایی باشد بلکه می تواند اعداد و علائم خاص نیز باشد. ولی توجه کنید که برای اعدادی که در محاسبات از آن ها استفاده می کنید، از فیلد نوع متن استفاده نکنید. فیلد متن برای داده های مناسب است که می خواهید آن ها را نگه دارید و نه آن هایی که قصد انجام محاسبه بر روی آن ها را داشته و یا به عنوان تاریخ و زمان بکار ببرید. مثلاً داده های با عنوان "نام" باید در این نوع فیلد ذخیره شوند.

توجه: از فیلد متنی برای داده های با طول حداکثر ۲۵۵ کاراکتر استفاده کنید. اگر طول داده بیش از این مقدار است، از داده نوع **Memo** استفاده کنید.

هم چنین دقت کنید که برای کنترل طول فیلد متنی، از خصوصیات فیلد استفاده کنید (در درس بعدی شرح داده خواهد شد) مثلاً می توانید از اکسس بخواهید تا در هر رکورد فیلد متنی، تا ۲۵۵ حرف را نگه دارد.

آدرس و شماره تامین اجتماعی مثال های کاملی از داده های متنی هستند که از اعداد و علائم خاص (مانند تیره و ویرگول) تشکیل می شوند.

### داده از نوع (**Memo** یادداشت):

از داده نوع **Memo** برای فیلدهایی استفاده می شود که تا ۶۴۰۰۰ کاراکتر (شامل حروف، اعداد و کارکترهای خاص) طول دارند. تعیین طول فیلدهای **Memo** ضروری نیست چون اکسس فقط به همان مقدار که لازم است برای آنها جا تعیین می کند. در واقع جدول هایی که فیلد **Memo** دارند، رکوردهایی با طول متغیر دارند. چون طول فیلد **Memo** در هر رکورد، با رکوردهای دیگر فرق دارد. مثال برای این نوع فیلد، فیلدی به نام "توضیحات" می تواند باشد.

توجه کنید که اگر بخواهید متون شکل بندی شده ای مانند متون واژه پردازی شده در **Word** را در اکسس نگه داری کنید باید از داده های نوع **OLE** استفاده کنید نه داده های نوع **Memo**.

## داده از نوع (Number عدد):

هر داده ای از جدول را که بخواهید در محاسبات به کار ببرید، از نوع Number تعریف کنید. داده از نوع Number برای تمامی ارقام، بجز مقادیر پولی و ارقام با دقت زیاد که در محاسبات علمی به کار می‌رود مناسب است. (توجه کنید که از داده نوع Currency ارز) برای ارقام پولی و دقت بالا استفاده می‌شود. اکسس در ارقام با دقت بالا زیاد دقیق نیست مگر این که از داده نوع Currency استفاده کنید. مثال برای داده های نوع Number می‌تواند مثلاً شماره کدپستی یا شماره کارگری باشد.

## داده از نوع (Data/Time زمان/تاریخ):

داده های از نوع تاریخ و زمان را در فیلدهایی که نوع آنها Data/Time است نگه دارید. از این نوع داده در فیلدهایی مانند تاریخ استخدام و زمان ملاقات ها استفاده می‌شود. وقتی فیلدهای تاریخ و زمان را از نوع Data/Time تعیین می‌کنید، اکسس به شما امکان مرتب کردن آن ها را بر اساس تاریخ و زمان می‌دهد. مثلاً می‌توانید گزارشی از تمامی کارمندانی که تاریخ شروع به کار آن ها اول مهرماه ۱۳۷۷ بوده است و یا فهرستی از تمامی حساب هایی که بیش از ۶۰ روز به سررسید آن ها مانده را چاپ کنید.

## فیلد داده از نوع (Currency ارز):

از این نوع داده ها برای نگه داری مقادیر پولی یا ارزی استفاده می‌شود. داده از نوع Currency, کاربر را مطمئن می‌سازد که مقادیر پولی/ارزی که در محاسبات به کار می‌روند به درستی گرد شده اند. (ولی یک مشکل کوچک در اکسس ۲۰۰۰ در این نوع فیلد برای ثبت نتایج به ریال وجود دارد) اکسس تمامی مقادیر پولی/ارزی را با ۱۵ رقم در سمت چپ و چهار رقم در سمت راست نقطه اعشار (ممیز کسری) نگه می‌دارد، لذا می‌توانید اعداد با دقت بالا را در فیلدهای پولی نگه دارید. از مثال های بارز این نوع فیلد مقدار حقوق کارمند را می‌توان ذکر کرد.

## فیلد داده از نوع (Auto Number عدد خودکار):

وقتی که فیلدی را از نوع داده Auto number تعریف می‌کنید، با هر رکوردی که به جدول اضافه شود اکسس به طور خودکار عددی را در این فیلد نگه می‌دارد. سه نوع ارقامی که اکسس در فیلد عدد خودکار نگه می‌دارد، عبارتند از:

-----  
اعداد متوالی: هنگامی که رکورد جدی را به جدول اضافه می‌کنید، اکسس رقمی متوالی را تخصیص می‌دهد که از ۱ شروع می‌شود. مقادیر متوالی Auto Number معمولاً برای کلید اصلی جدول به کار برده می‌شوند. اعداد تصادفی: اکسس در رکورد جدید به این فیلد، عددی منحصر به فرد و تصادفی را اختصاص می‌دهد. اعداد شناسائی انعکاسی: اکسس مقدار منحصر به فردی را به این فیلد تخصیص می‌دهد که پایگاه داده های تکرار شونده را با یک دیگر هماهنگ می‌کند.

## فیلد داده از نوع (Yes/No بلی/خیر):

فیلدی که از نوع Yes/No تعریف می شود به ازای Yes عدد ۱- و به ازای No عدد ۰ را دارا می شود. با استفاده از بخش Filed Properties در پنجره مذکور می توانید بسته به دلیل استفاده از فیلد، یکی از سه حالت Yes/No, True/False (درست یا غلط) یا (On/Off روشن و خاموش) را به کار ببرید. از موارد استفاده از این نوع داده می توان به متاهل بودن اشاره کرد که می تواند یا بله یا خیر را دربرگیرد.

## فیلد داده از نوع شی

### OLE:

OLE مخفف (Object Linking and Embedding اتصال و نشانیدن شی) است. اگر برنامه دیگری تحت ویندوز از OLE پشتیبانی کند یعنی بتواند شی قابل انتقال به برنامه دیگری را ایجاد کند آنگاه فیلد شی OLE در اکسس نیز قادر خواهد بود تا آن داده را در خود نگه دارد. فیلدهای OLE برای نگه داری اشیایی از قبیل عکس، تصاویر ویدیویی و صدا مناسب است. (همان طور که ملاحظه می کنید پایگاه داده اکسس علاوه بر داده های عددی و متنی می تواند انواع متنوع دیگری از داده ها را نیز نگه دارد).

وقتی که فیلدی از نوع OLE را در جدول قرار می دهید، می توانید علاوه بر مثلاً اطلاعات مربوط به اقلام موجودی، تصویر آن را نیز به شکل یک شی OLE در درون فیلد OLE قرار دهد. به این ترتیب هنگامی که رکورد را نمایش می دهید، اکسس داده های رکورد را همراه با تصویر نمایش می دهد.

برای نگه داری فیلدهای از نوع OLE در پایگاه داده اکسس به نرم افزارهای کمکی از قبیل برنامه های گرافیکی و یا صفحه گسترده نیاز دارید.

## داده از نوع (Hyperlink ابرمتن):

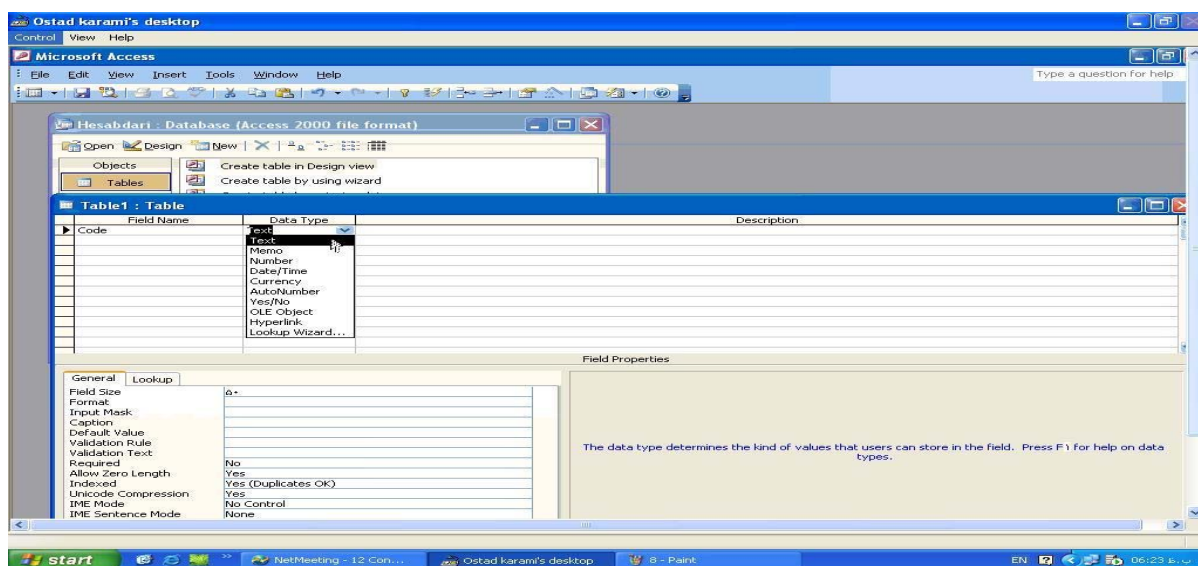
این نوع فیلد برای نگه داری ابرمتنها به کار می روند. اگر تا حالا تجربه کار کردن با برنامه های طراحی صفحات وب یا کارکردن با کدهای HTML را داشته باشید منظور از ابرمتنها را به خوبی می دانید. ابرمتنها در واقع لینک های اینترنتی هستند که ما را به صفحات مختلف در اینترنت هدایت می کنند مثلاً ما اگر لینکی به صورت <http://www.yahoo.com> داشته باشیم و بر روی آن کلیک کنیم ما را به صفحه یاهو خواهد برد و در واقع این نوع فیلد از انواع فیلدهایی است که به ندرت مجبور به استفاده از آن می شویم. عملکرد آن ها درست مانند فیلو از نوع متن می باشد با این تفاوت که در این جا هر متنی را که برای نگه داری تایپ کنیم در واقع آن لینکی از به یک صفحه اینترنتی.

از موارد استفاده از این فیلد می توان به نگه داری آدرسهای اینترنتی مرتبط با شرکت اشاره کرد.

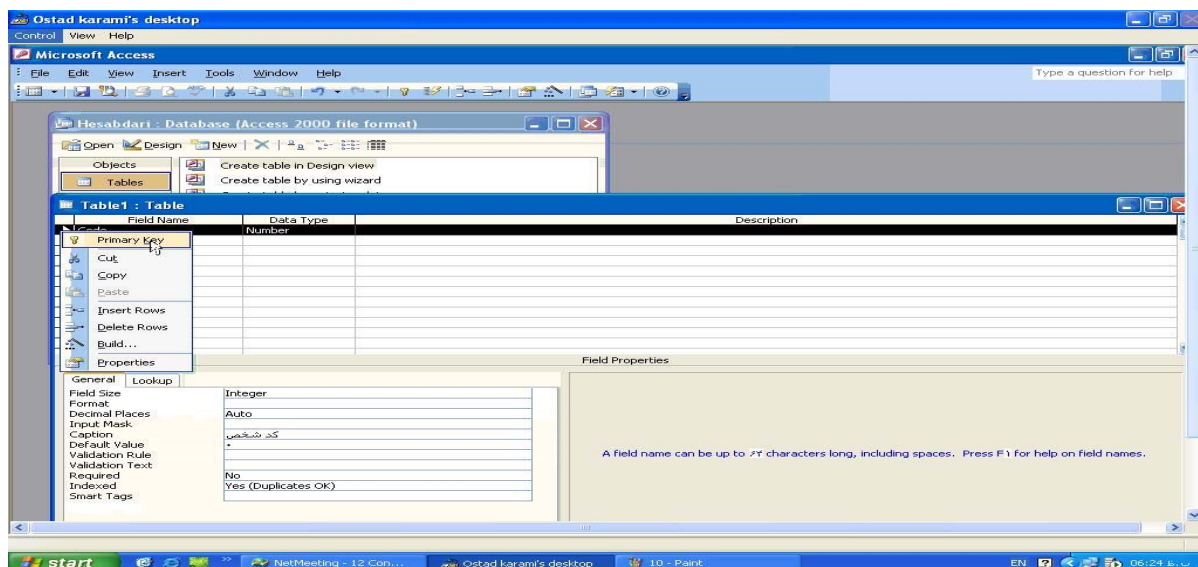
## داده از نوع (Lookup Wizard) جادوگر جستجوگر):

هنگامی که فیلدی از نوع Lookup Wizard ایجاد می کنید، اکسس با به کار گرفتن جادوگری به آن امکان می دهد تا به جای یک مقدار، فهرستی از مقادیر را نگه دارد. این فیلد می تواند فهرستی از مقادیر را از یک جدول یا بازجست و یا فهرستی از مقادیر ثابتی که در هنگام ایجاد فیلد تعریف کرده اید، (برای فهرست هایی که مقادیر ثابتی دارند) باشد. فرض کنید که شرکتی به پنج کشور کالا می فروشد. هنگام طراحی پایگاه داده می توانید با به کار انداختن Lookup Wizard، فهرست ثابتی از اسامی کشورها بسازید که در هنگام ورود داده ها به جدول، ظاهر شود. هنگامی که کاربر بخواهد کشوری را انتخاب کند به جای تایپ نام کشور برای هر رکورد فقط باید از فهرست ثابت انتخاب کند.

۱۰. همچنین جهت تغییر خواص (Properties) فیلدها می توان از قسمت پایین (General) پنجره جدول استفاده نمود.

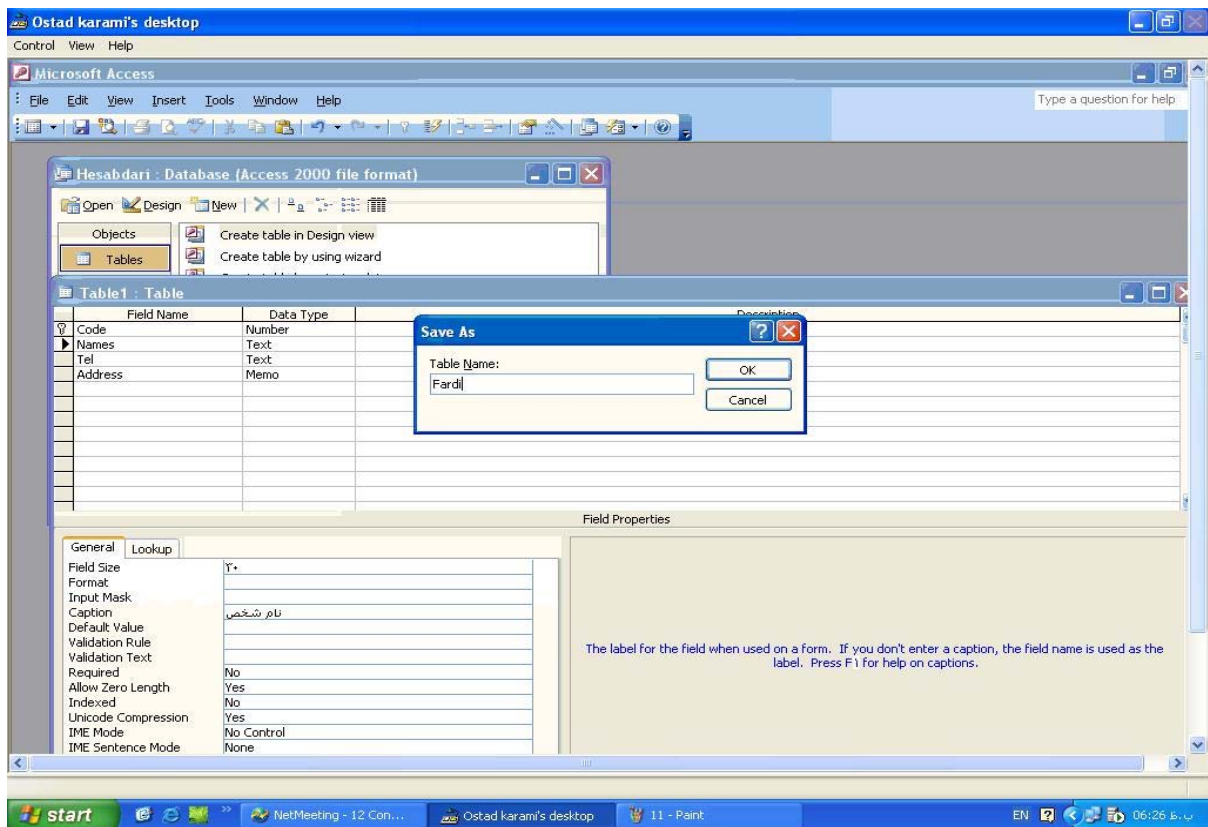


۱۱. جهت کلید نمود یکی از فیلدها می توان روی فیلد مورد نظر کلیک سمت راست نموده و از پنجره ظاهر شده گزینه (Primary Key) را انتخاب نماید.

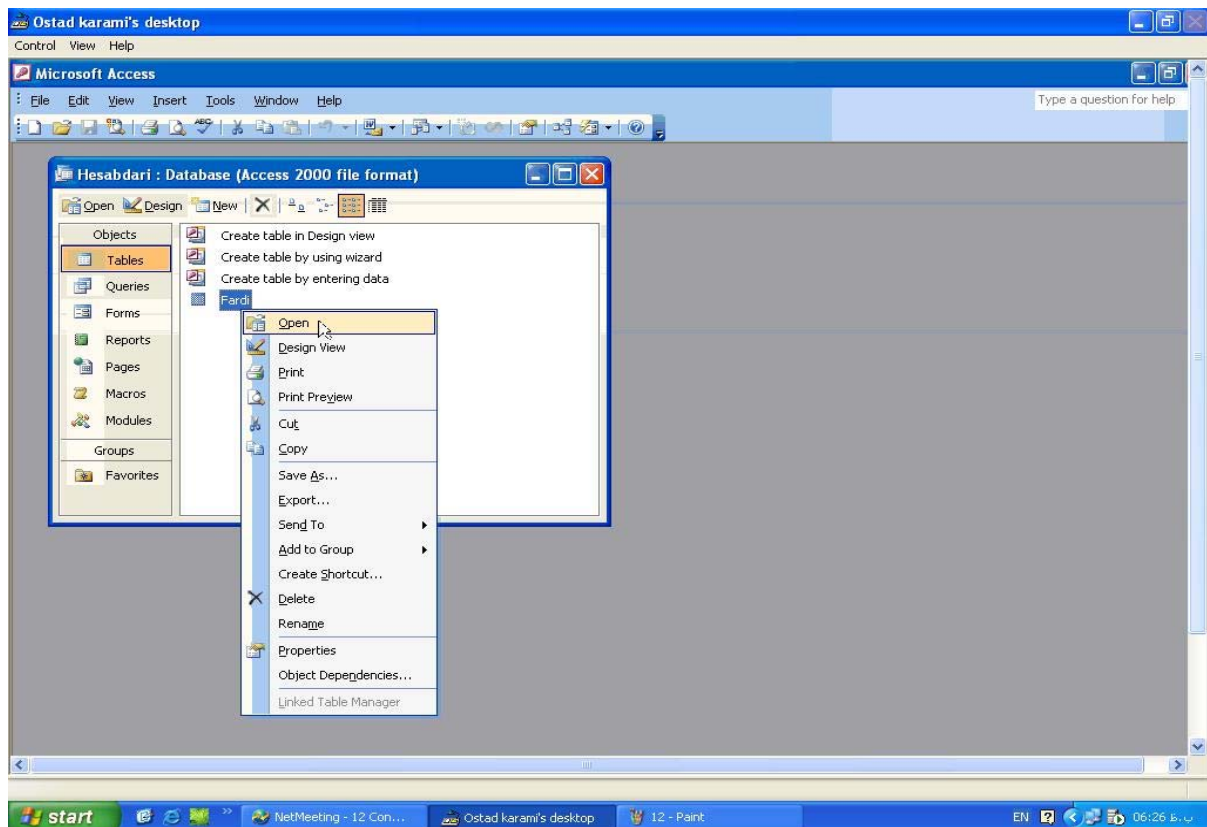




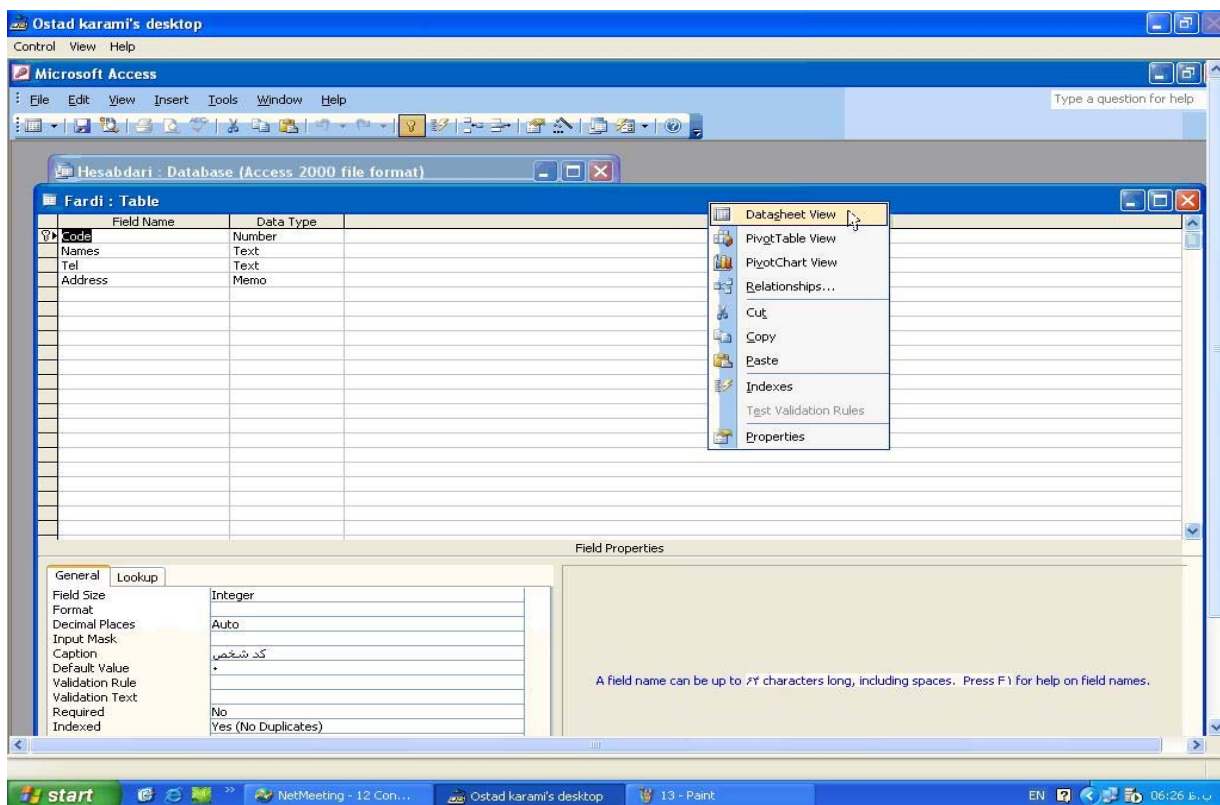
۱۲. در آخر جهت بستن پنجره جدول از شما نام جدول را درخواست می نماید. (نام جدول مورد نظر fardi)



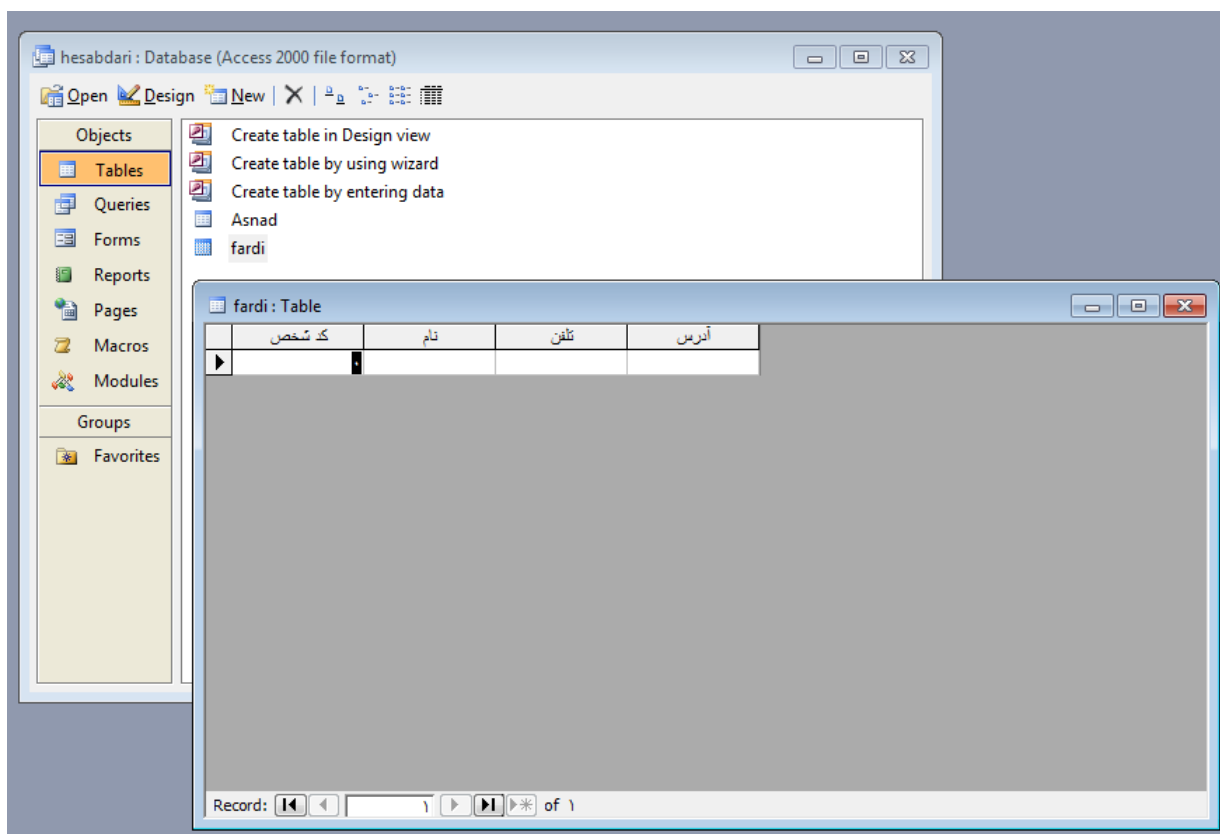
۱۳. جهت باز کردن جدول ذخیره شده روی جدول fardi کلیک راست نمود و گزینه Open را انتخاب نماید.



۱۴. در پنجره باز شده جدول ، روی نوار بالای جدول کلیک راست نموده و گزینه Datasheet View را انتخاب نمایید.



۱۵. در این قسمت شما می توانید اطلاعات فردی خود را وارد نمایید.

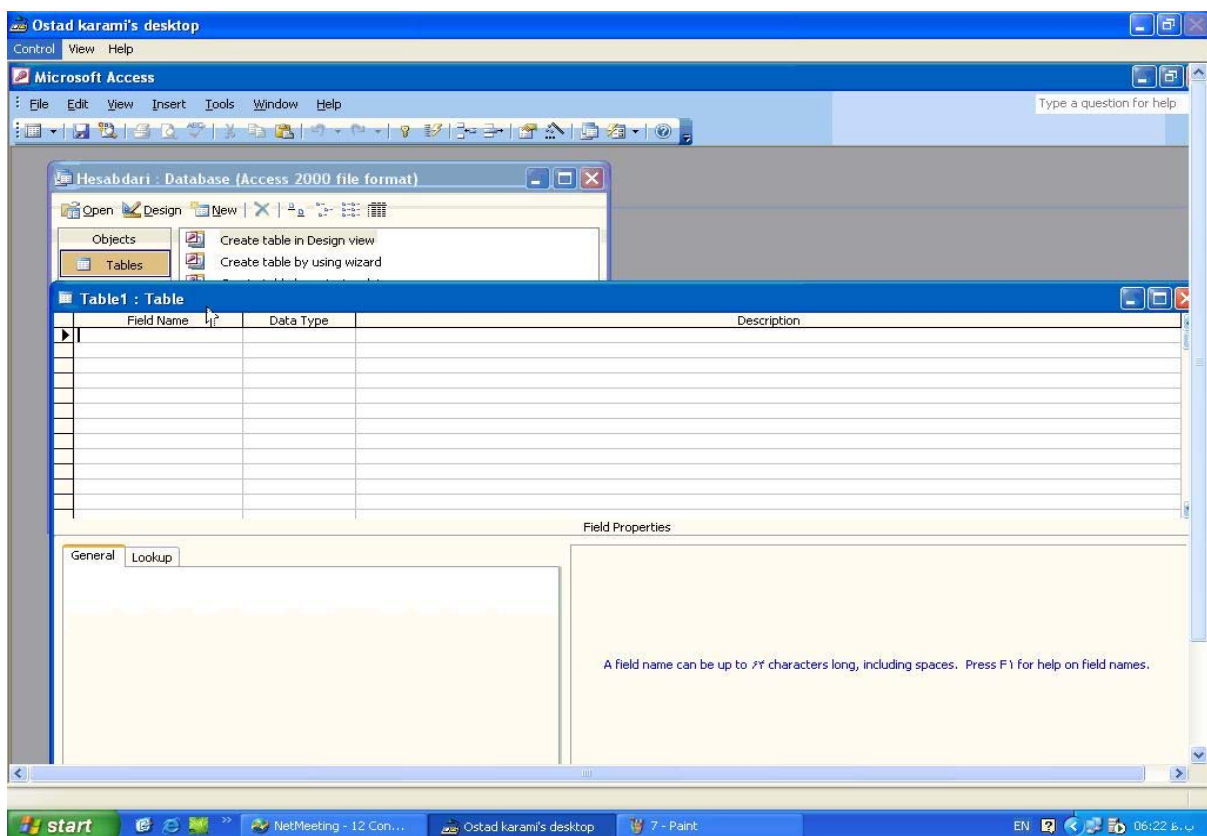


• ایجاد جدول اسناد (Asnad)

Table : Asnad			
Field name	Data Type	Field Zise	Caption
Row	Auto Number	*	شماره سند
Dates	Date	۶	تاریخ
Names	Lookup Wizard	*	نام شخص
Sharh	Text	۸۰	شرح
Bedehikar	Currency	*	بدهکار
Bestankar	Currency	*	بستنکار

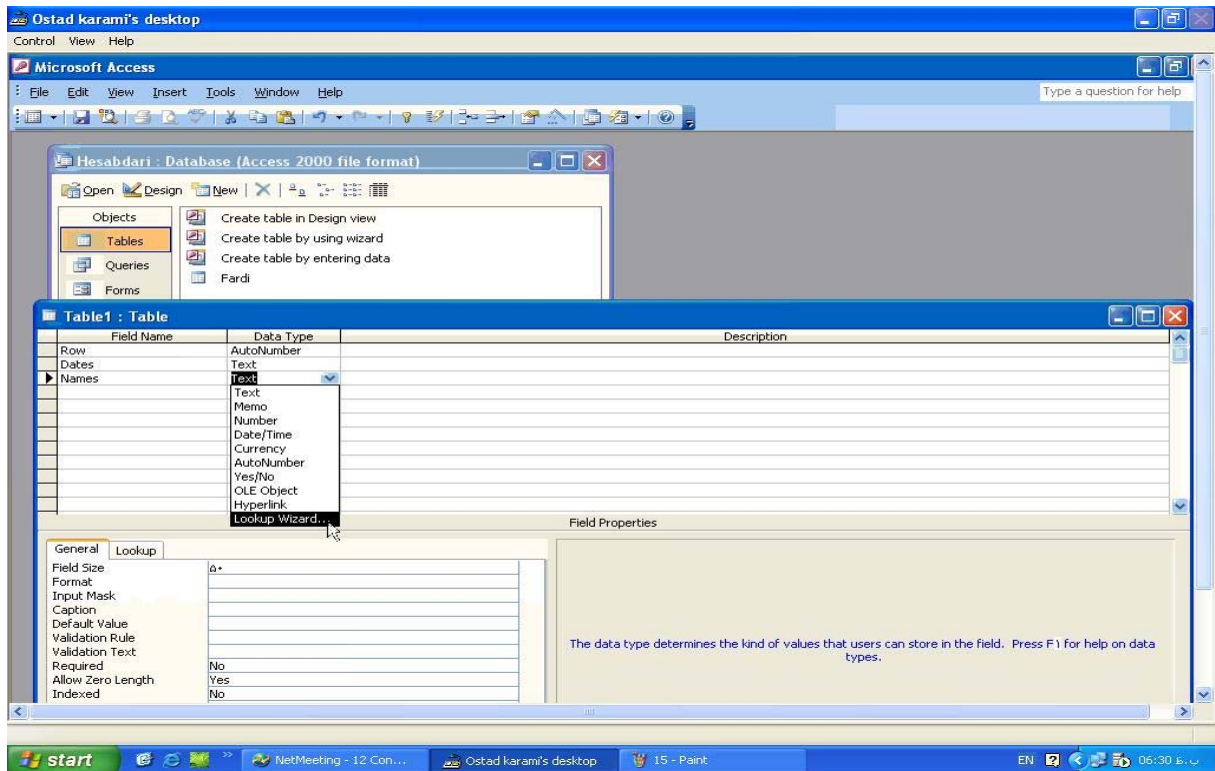
۱. جهت ایجاد یک جدول از پنجره سمت راست گزینه Table را انتخاب کرده و سپس از داخل پنجره گزینه Creat table in design view را انتخاب نماید.

۲. طبق جدول فردی فیلدها و نوع آنها را وارد نماید.



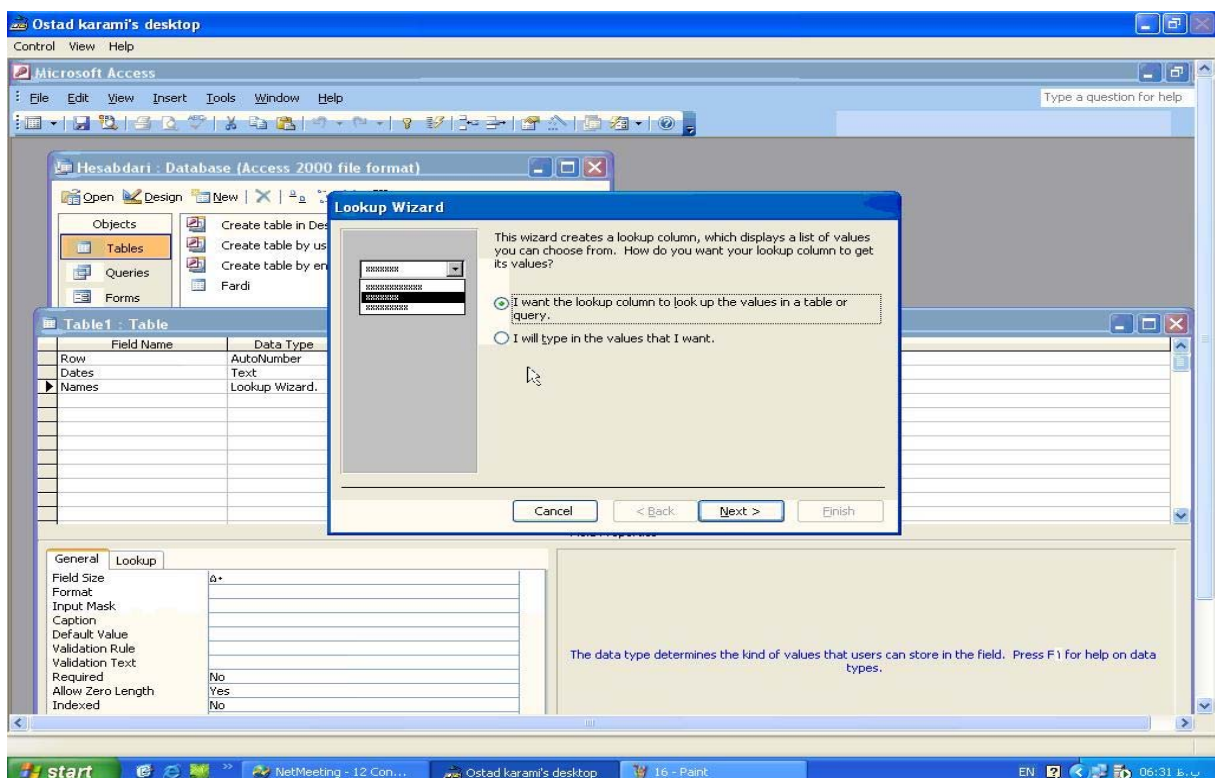
• نحوه ایجاد یک Lookup Wizard

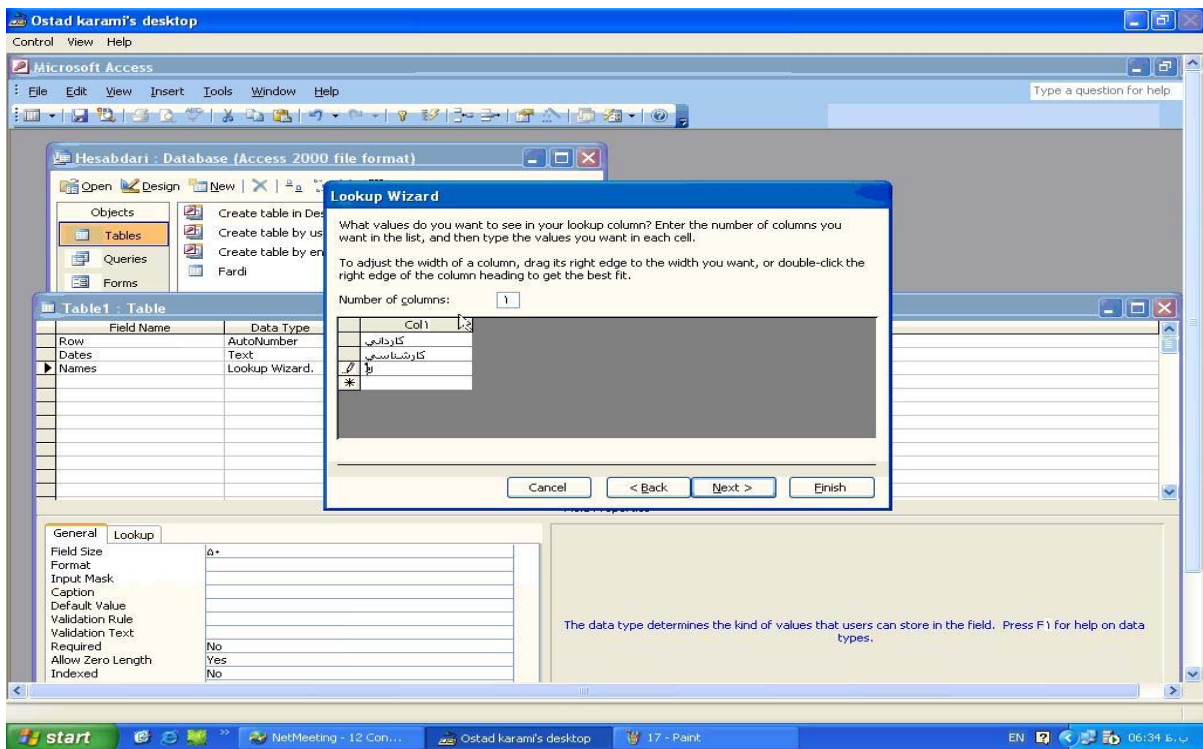
۱. نام فیلد مورد نظر را تایپ و از انواع فیلد گزینه Lookup Wizard را انتخاب نماید.



۲. در پنجره Lookup wizard به دو روش می توان آن را تعریف نمود.

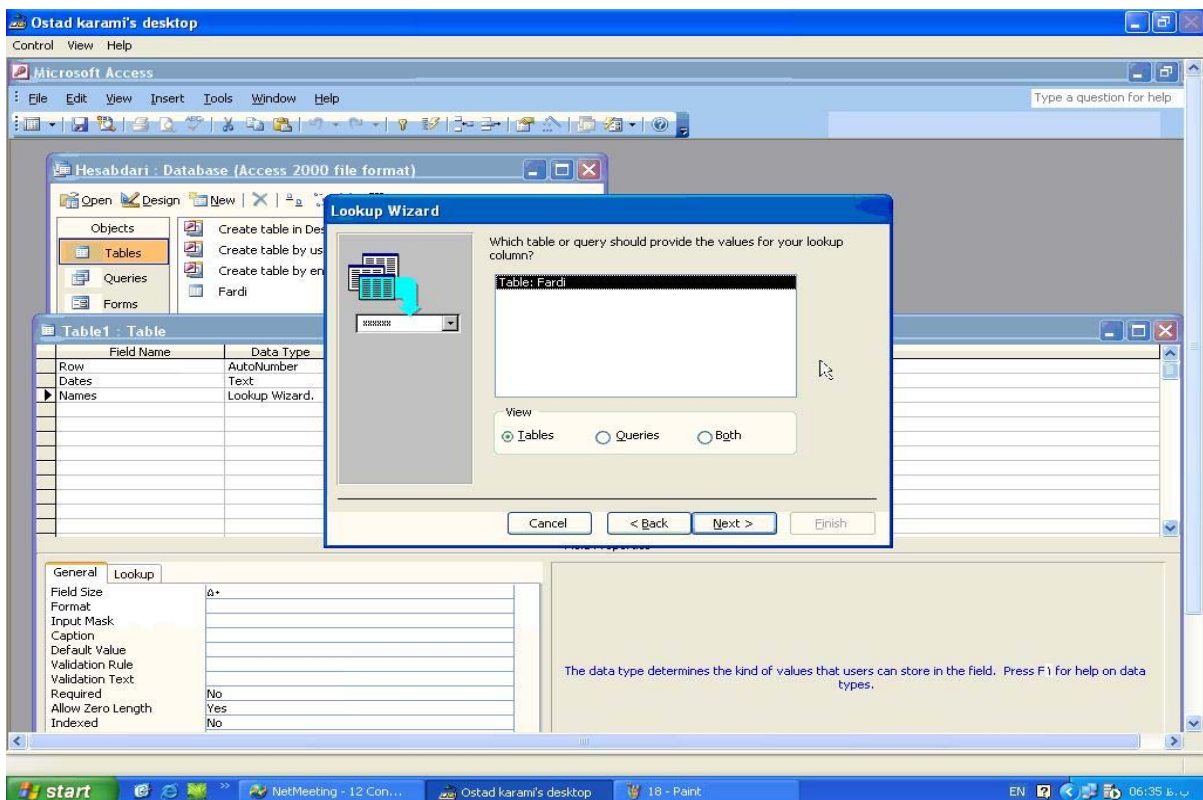
- روش اول : مقادیر Lookup را از یک جدولی که قبلاً ایجاد شده استخراج نمود.
- روش دوم : مقادیر Lookup را خود در یک جدول درج نماید.



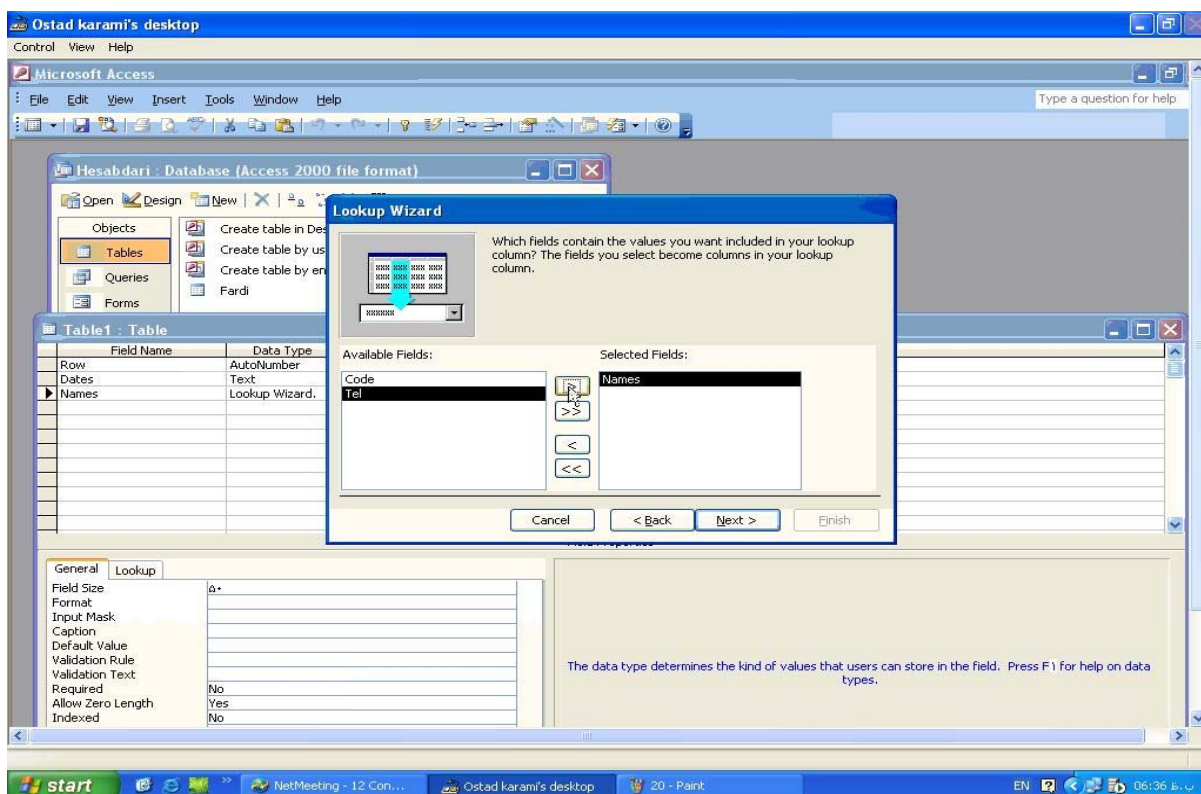
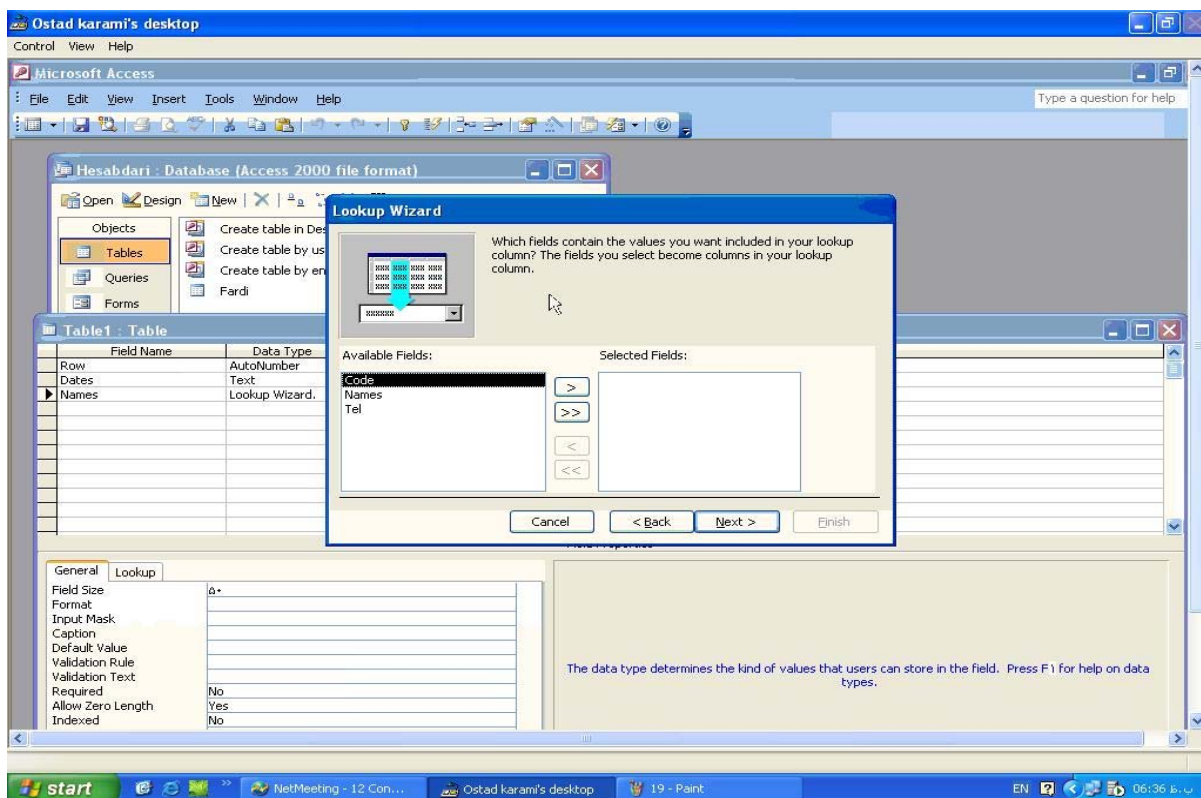


\* در پنجره Lookup با استفاده از گزینه دوم طبق شکل فوق می توان مقادیر را خود درج و در آخر به نام یک جدول ذخیره نمود.

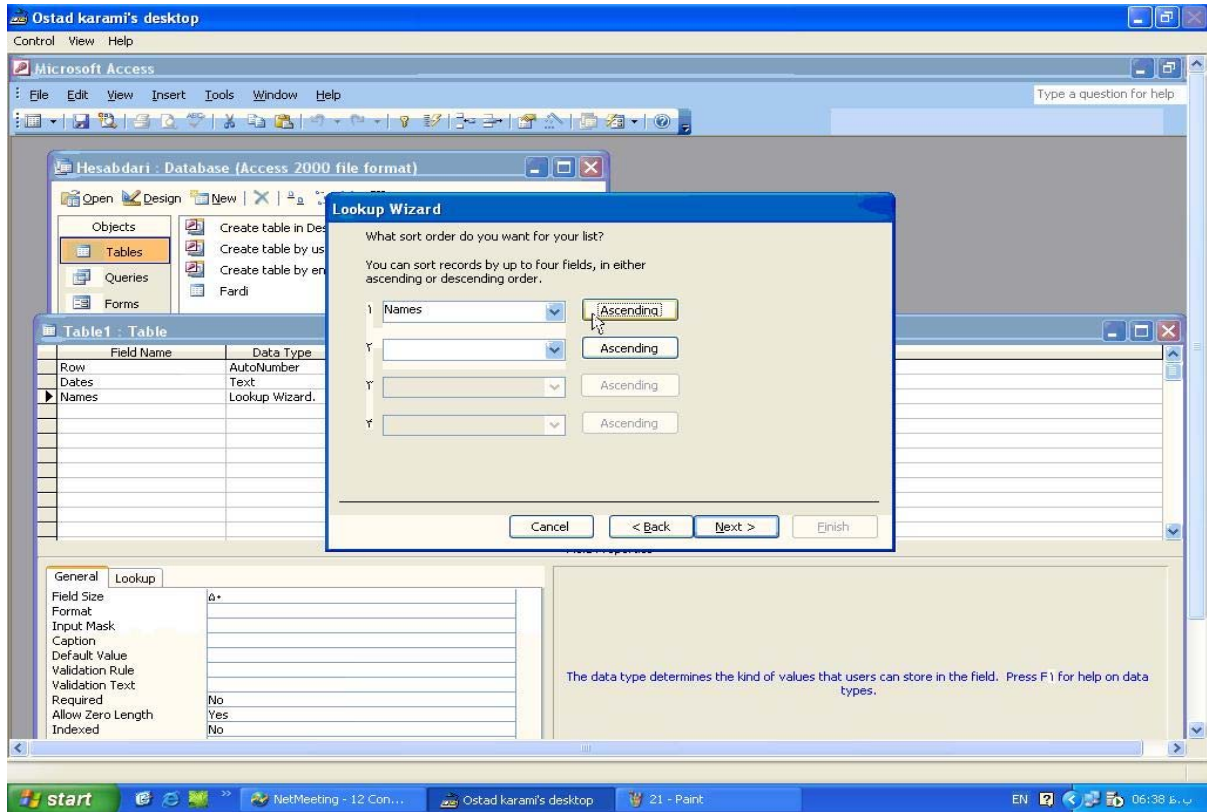
۳. در پنجره Lookup جهت استخراج مقادیر از یک جدول گزینه اول را انتخاب و Next نماید تا پنجره زیر ظاهر شود از جدول موجود جدول Fardi را انتخاب و Next نماید.



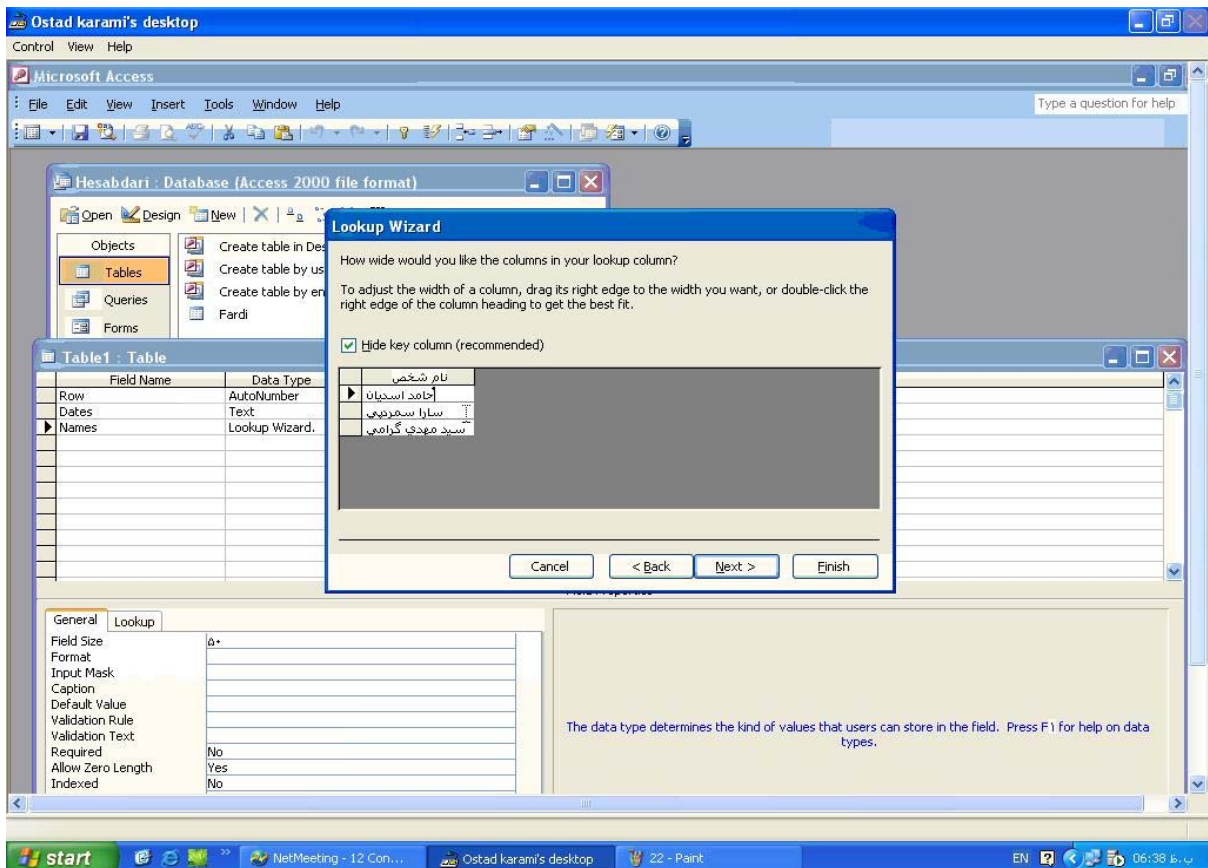
۴. از فیلدهای جدول Fardi مورد نظر را انتخاب و با استفاده از نشانه ( > ) به فیلد انتخابی انتقال دهید. و Next نماید.



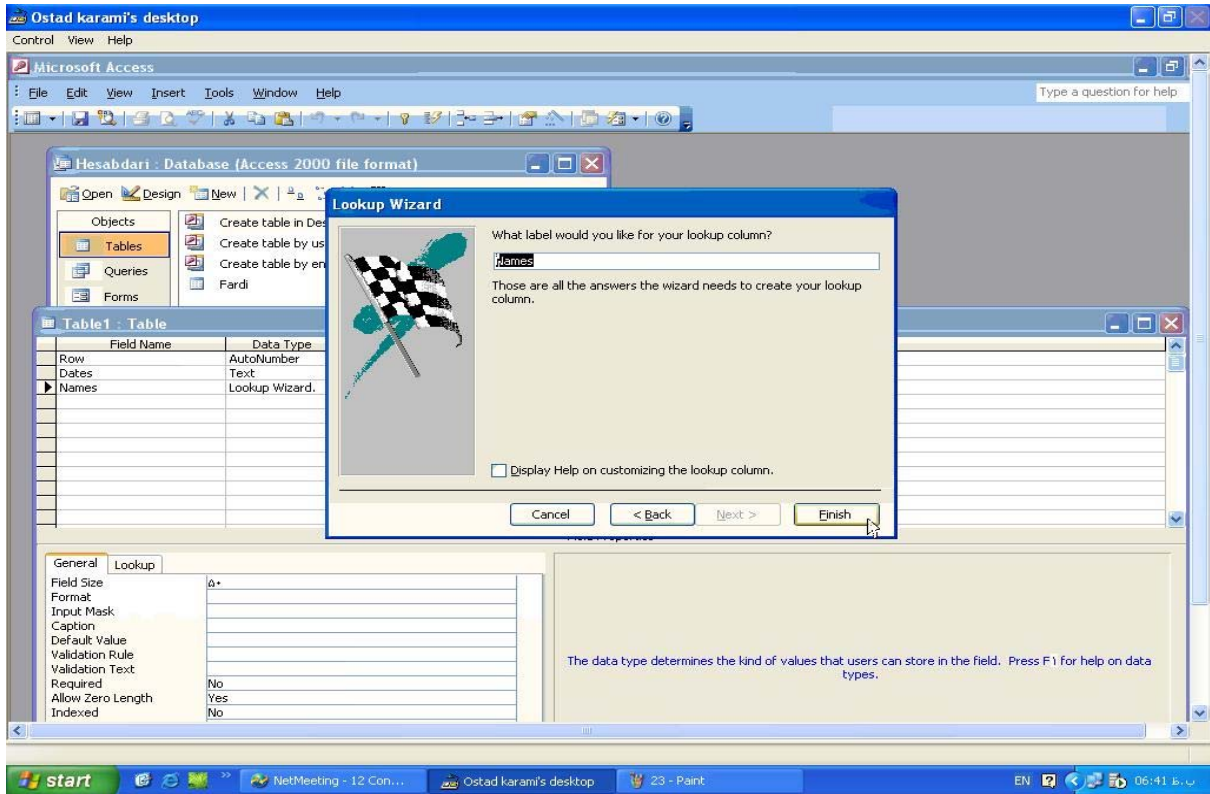
۵. در این پنجره بر حسب صعودی یا نزولی می توان فیلد را مرتب نمود.



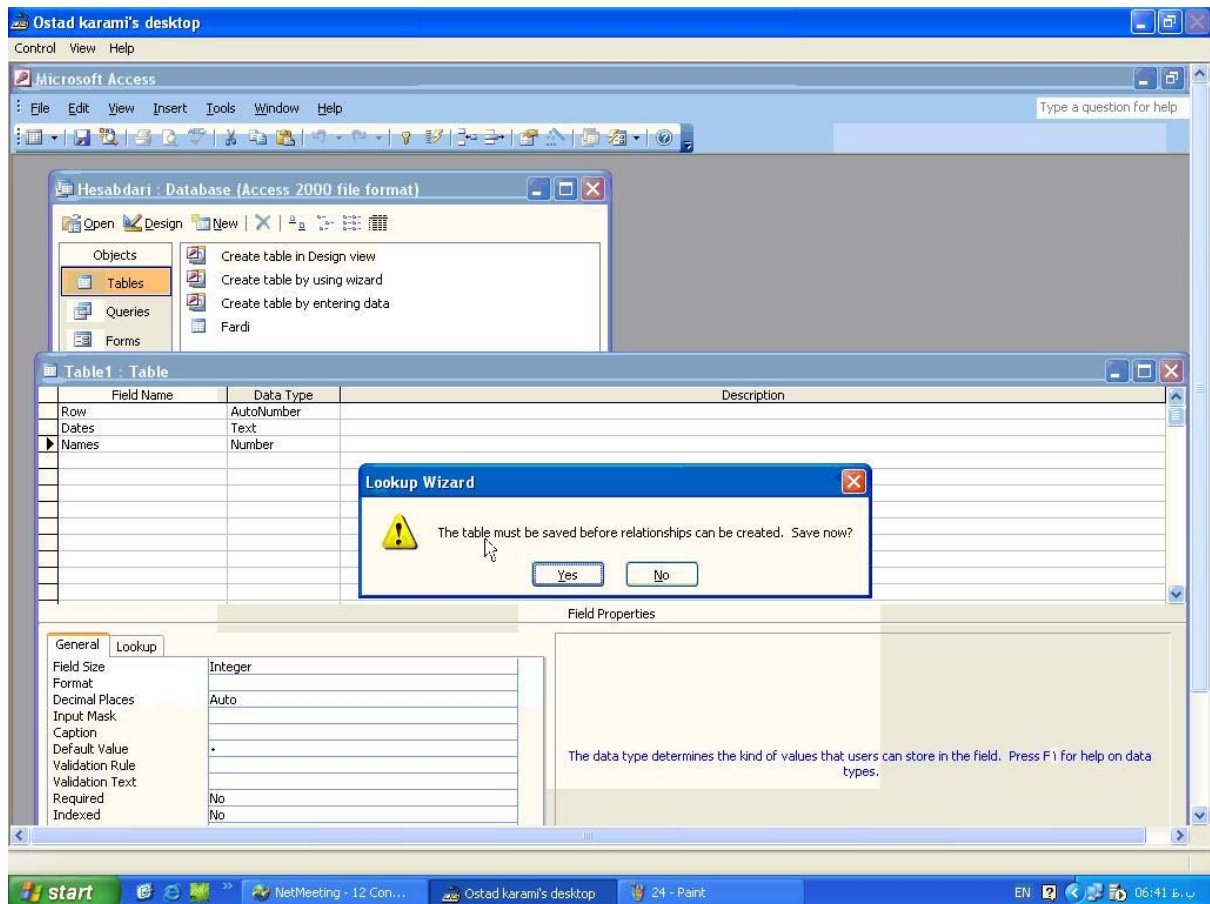
۶. در این پنجره مقادیر فیلد انتخابی را مشاهده نمود ، Next نماید.



۷. در انتها برای Lookup یک نام تایپ و Finish نماید.

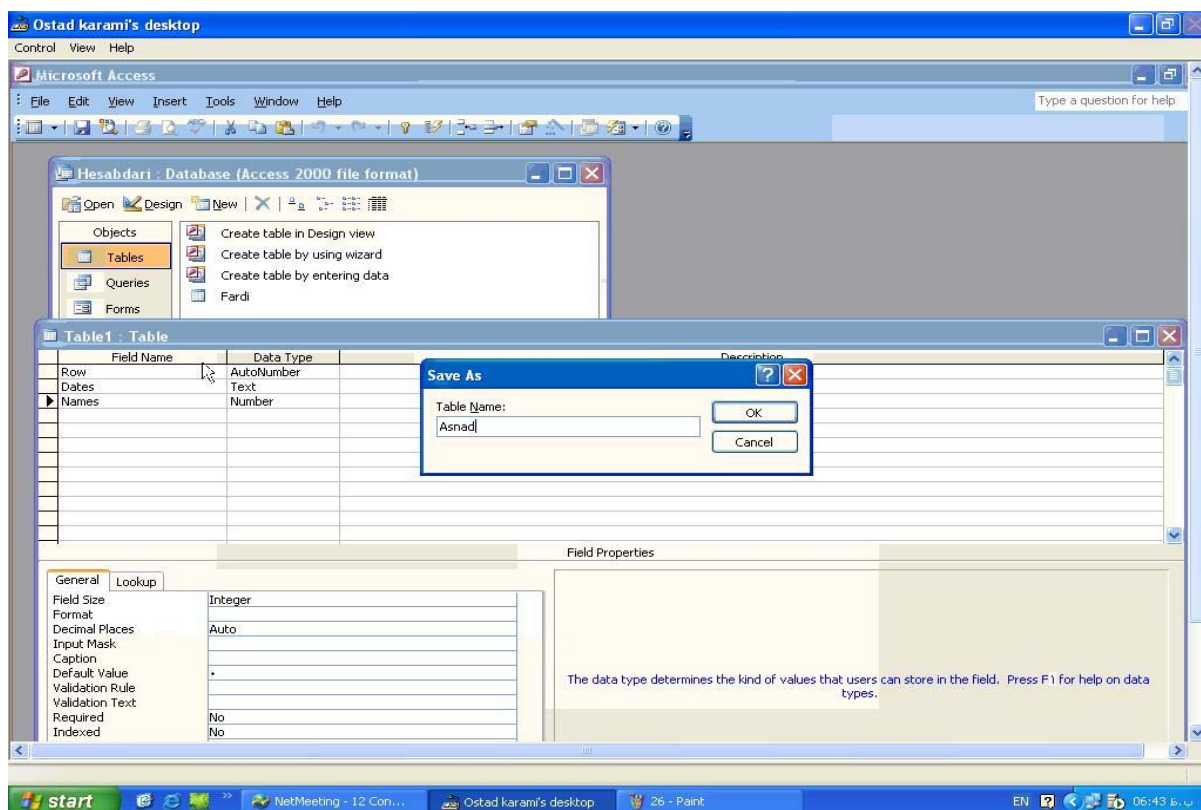


۸. در این پنجره کلید Yes را انتخاب نماید.

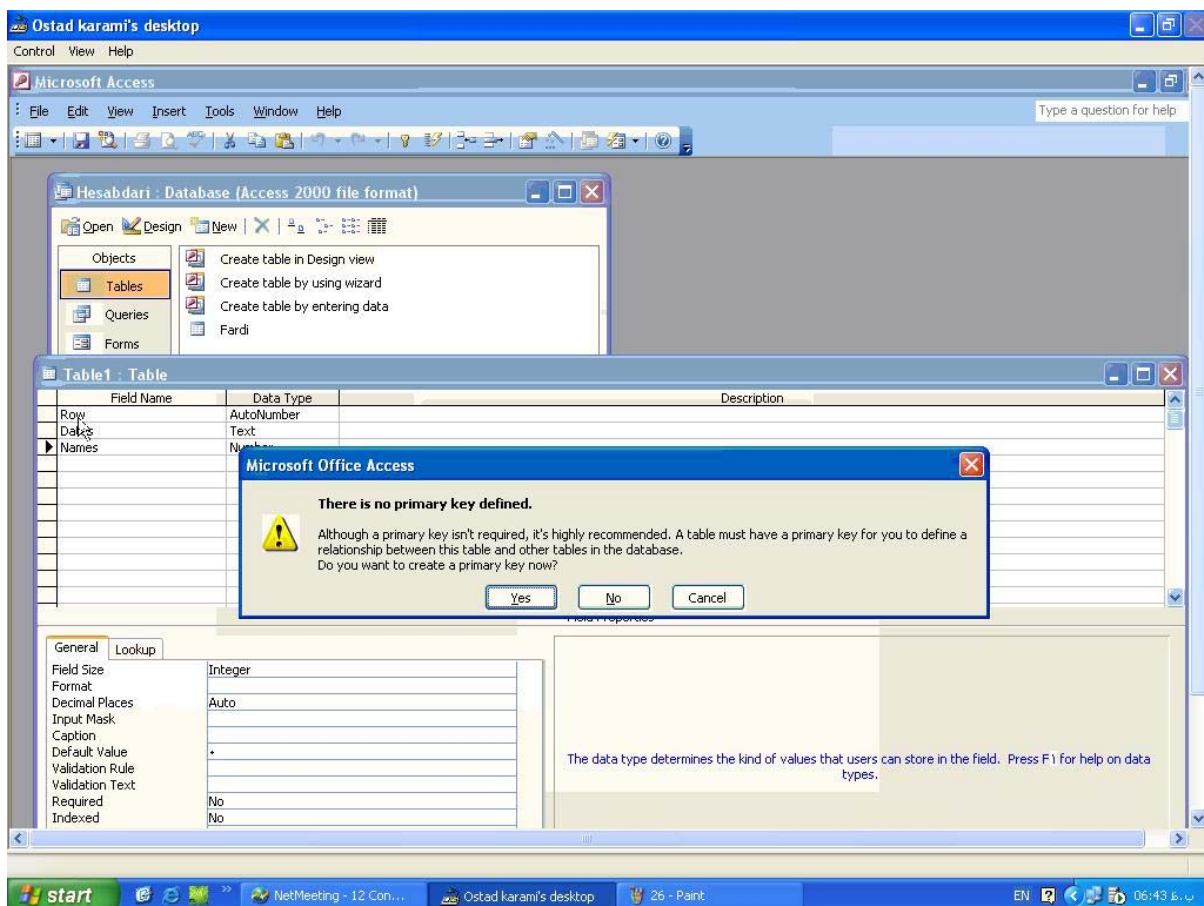




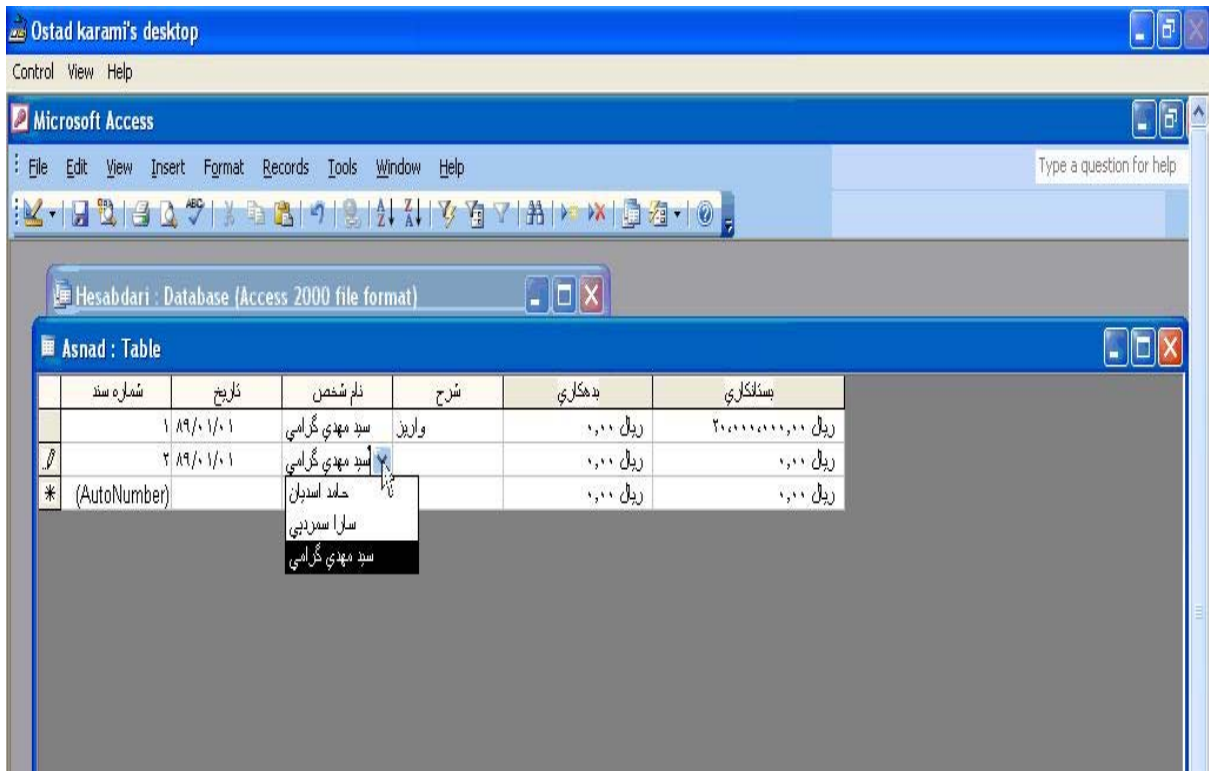
۹. جدول مذکور را به نام Asnad ذخیره نماید.



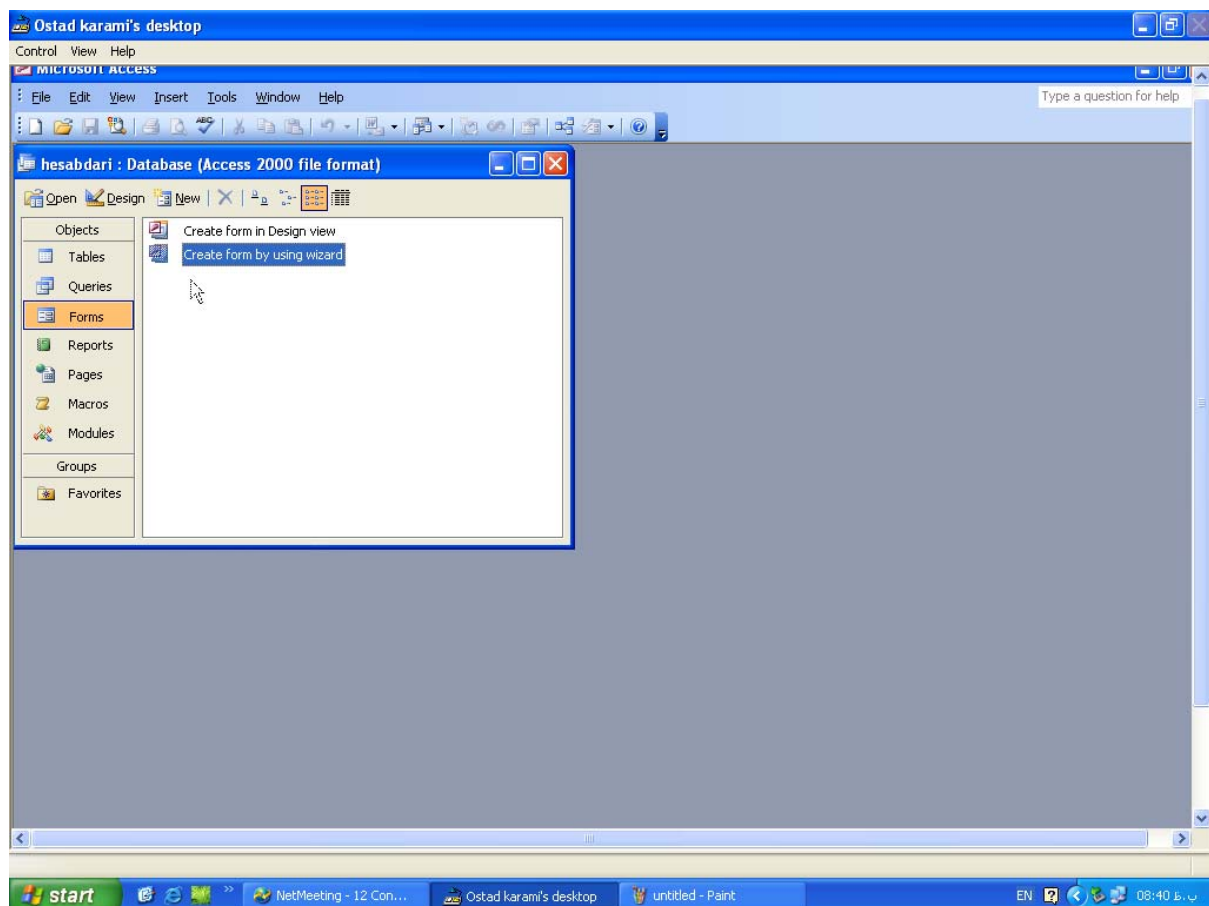
۱۰. در صورت مشخص نشدن فیلد کلیدی سیستم پیغام داده و خود اتوماتیک فیلد اول را کلید می کند.



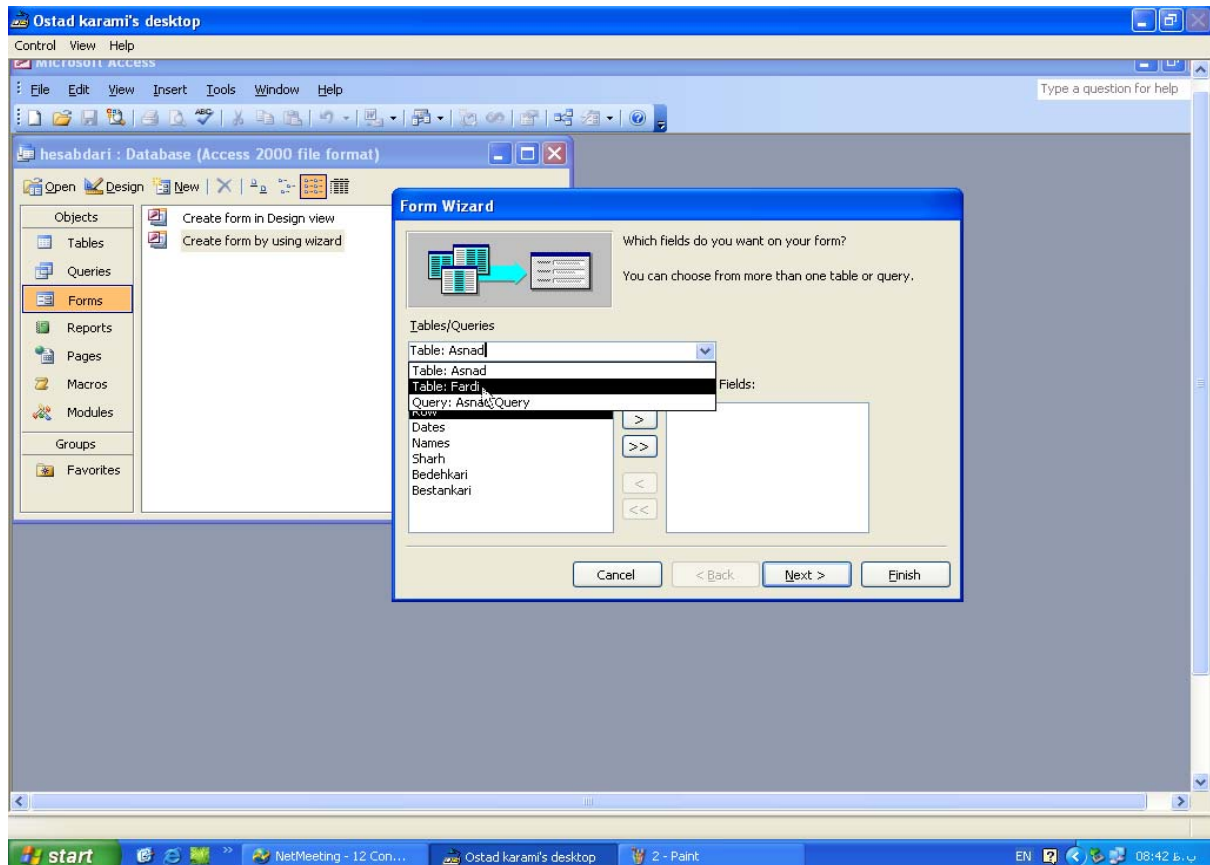
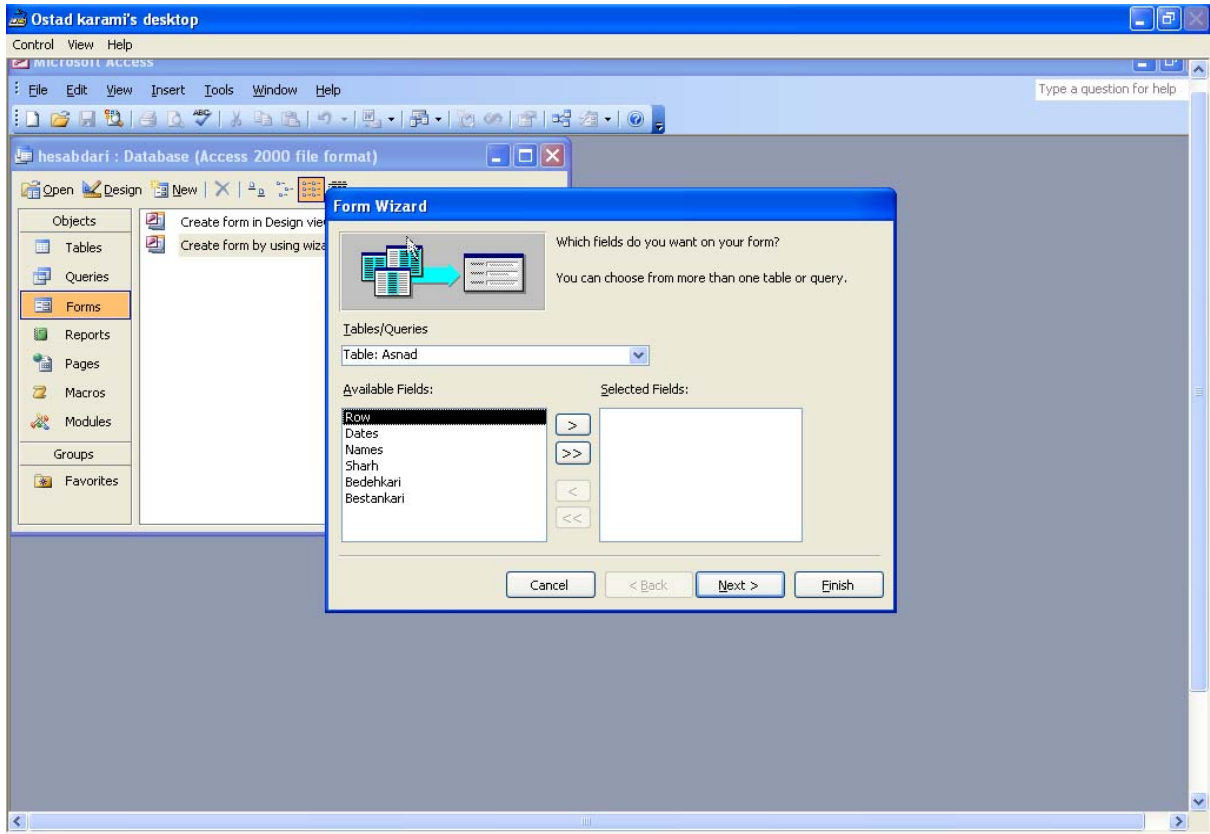
۱۱. اطلاعات جدول اسناد را در جدول آن درج نماید.

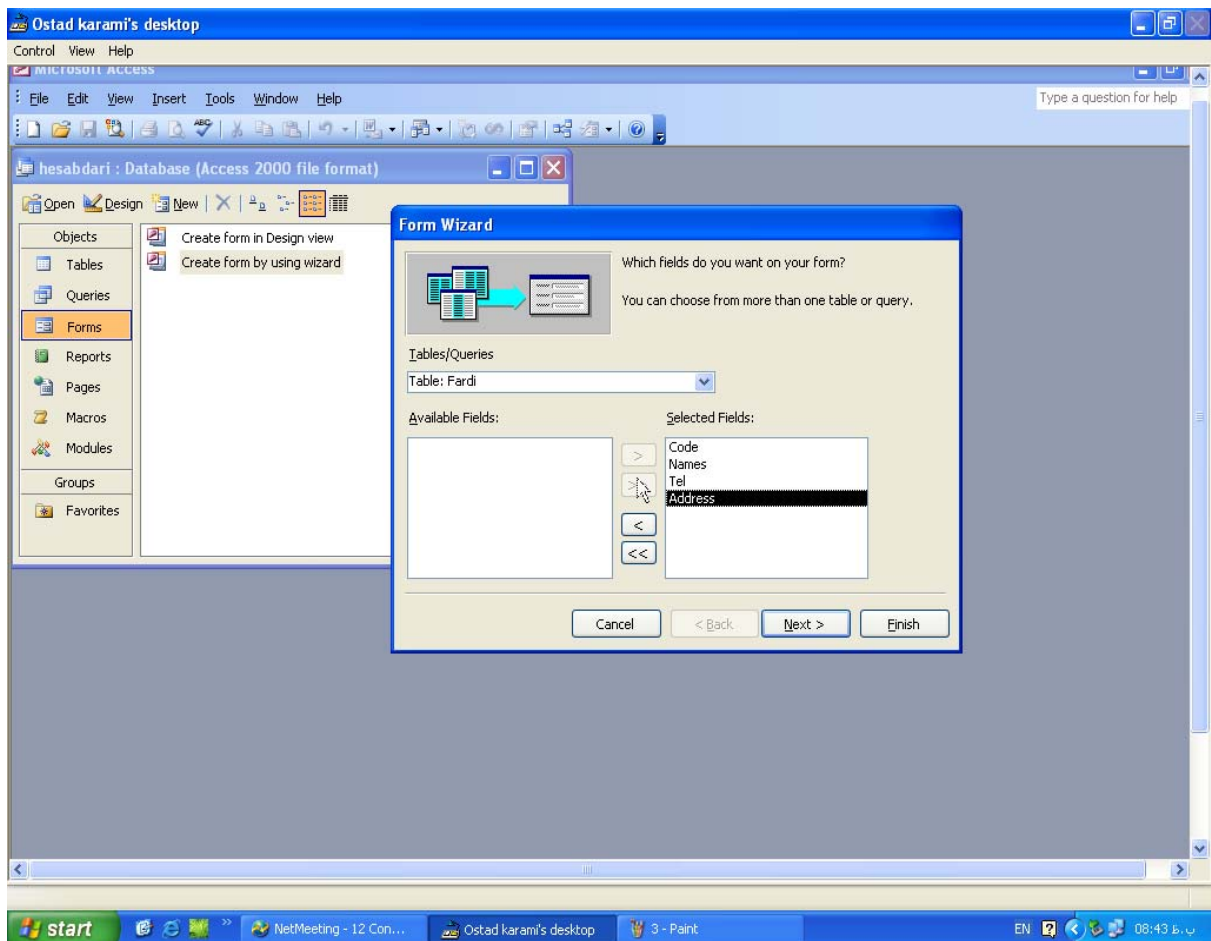


برای طراحی فرم ابتدا یک فرم جدید خام ایجاد می کنیم :

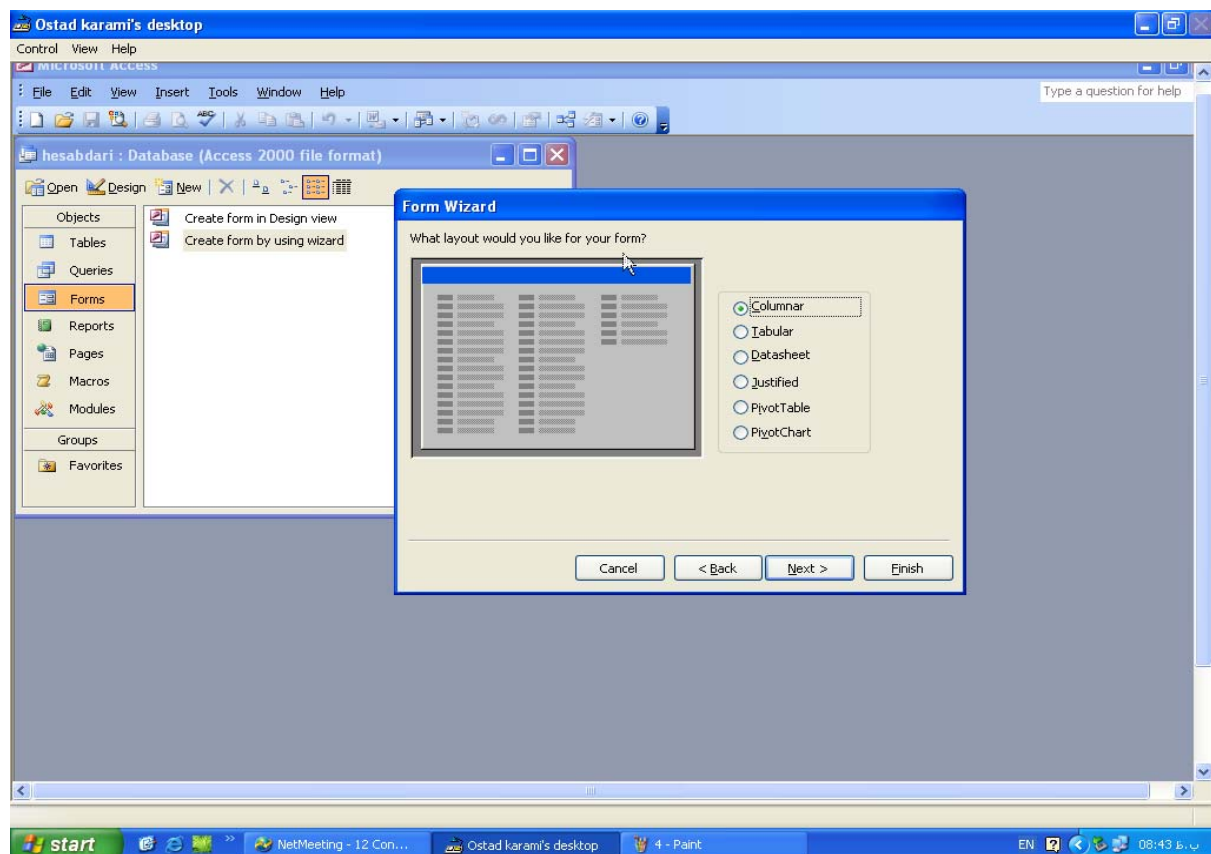


در این مرحله Table مورد نظر را انتخاب و فیلدهای لازم را select می کنیم

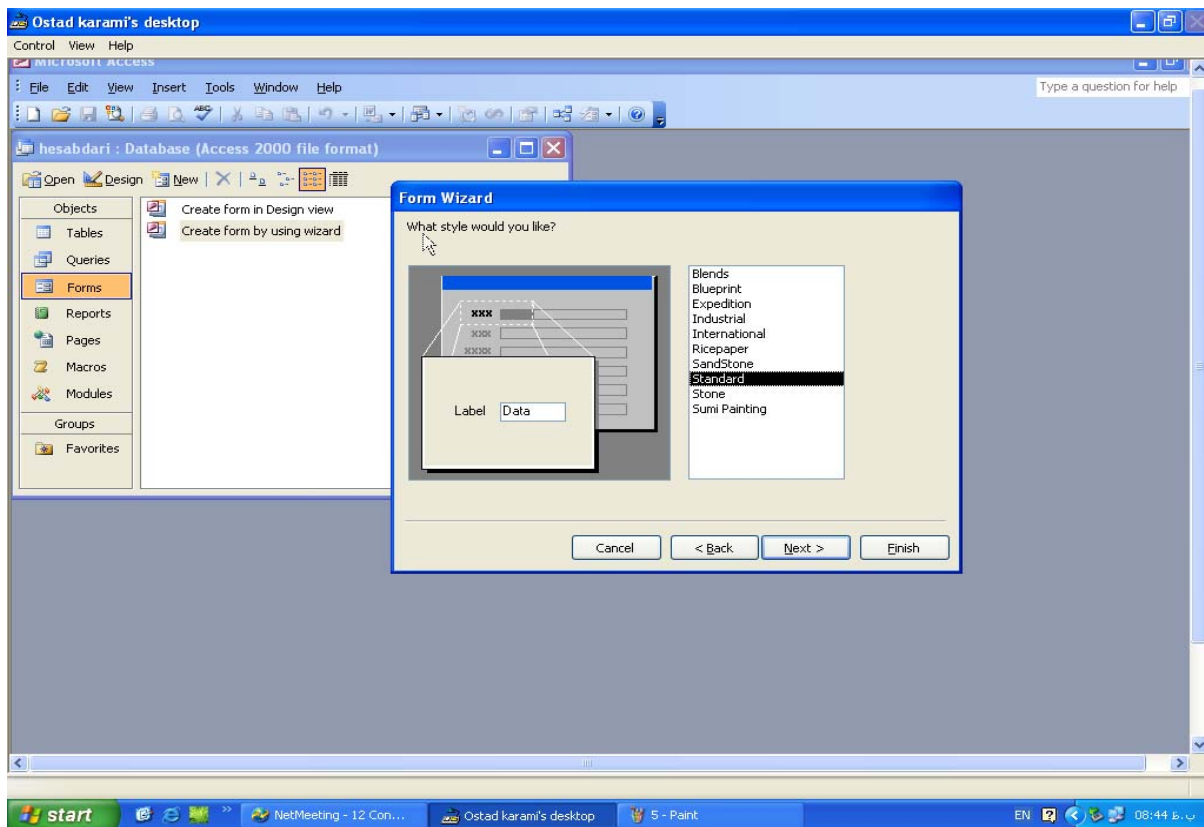




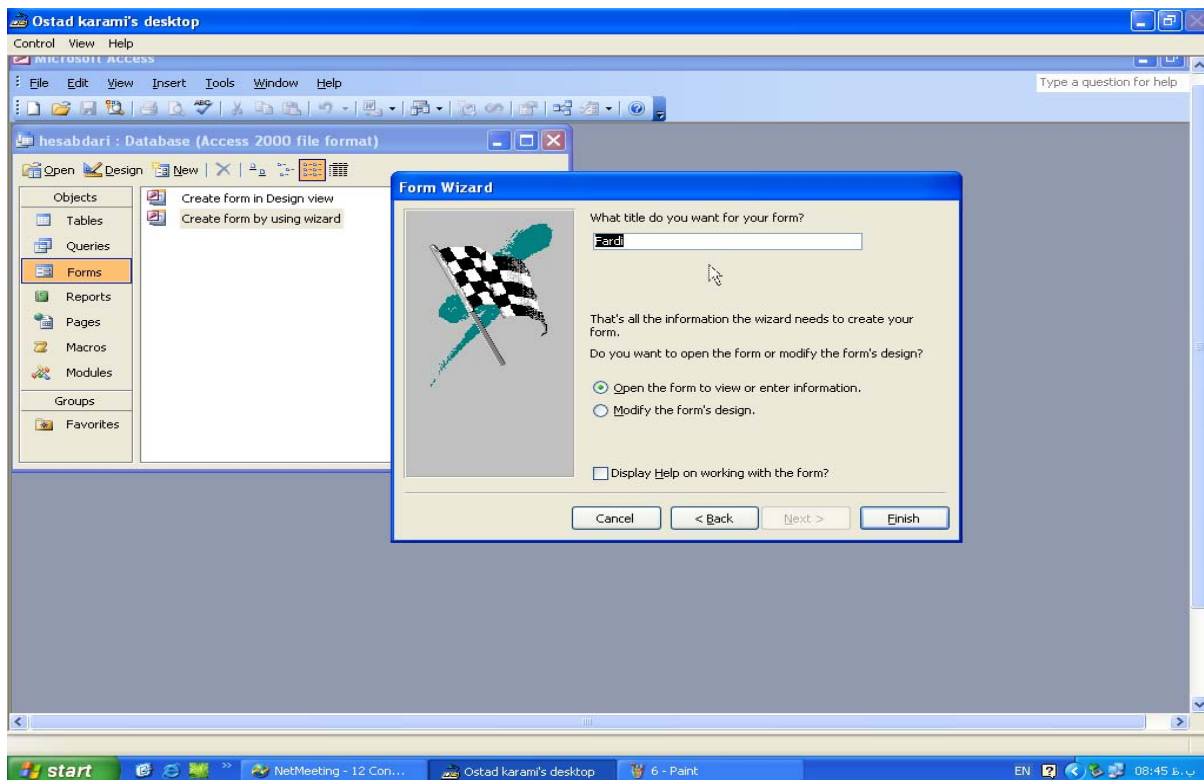
نوع چیدمان پیش فرض را از میان مدل های معرفی شده کلیک می کنیم ،



یکی از اشکال و فرمهای پیش فرض را انتخاب می کنیم ،

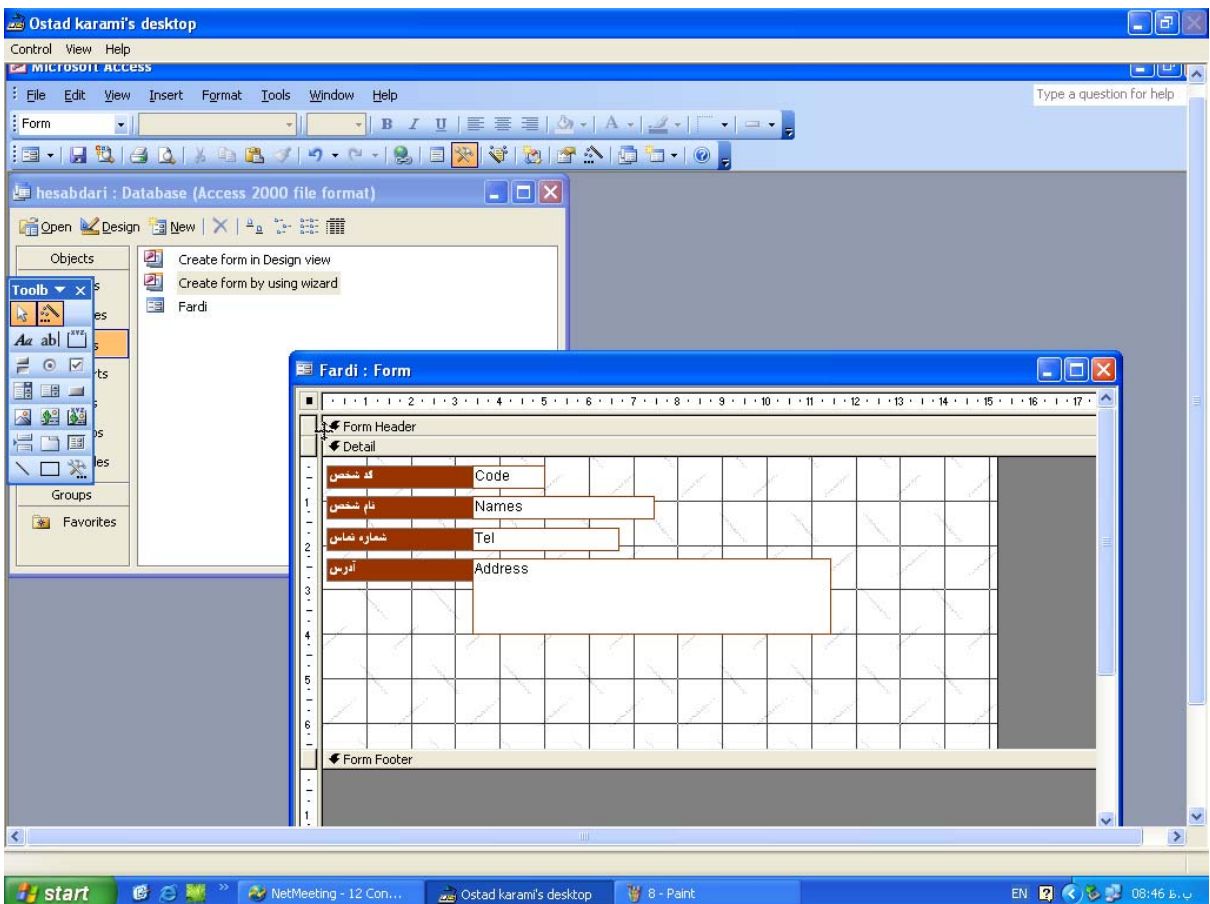
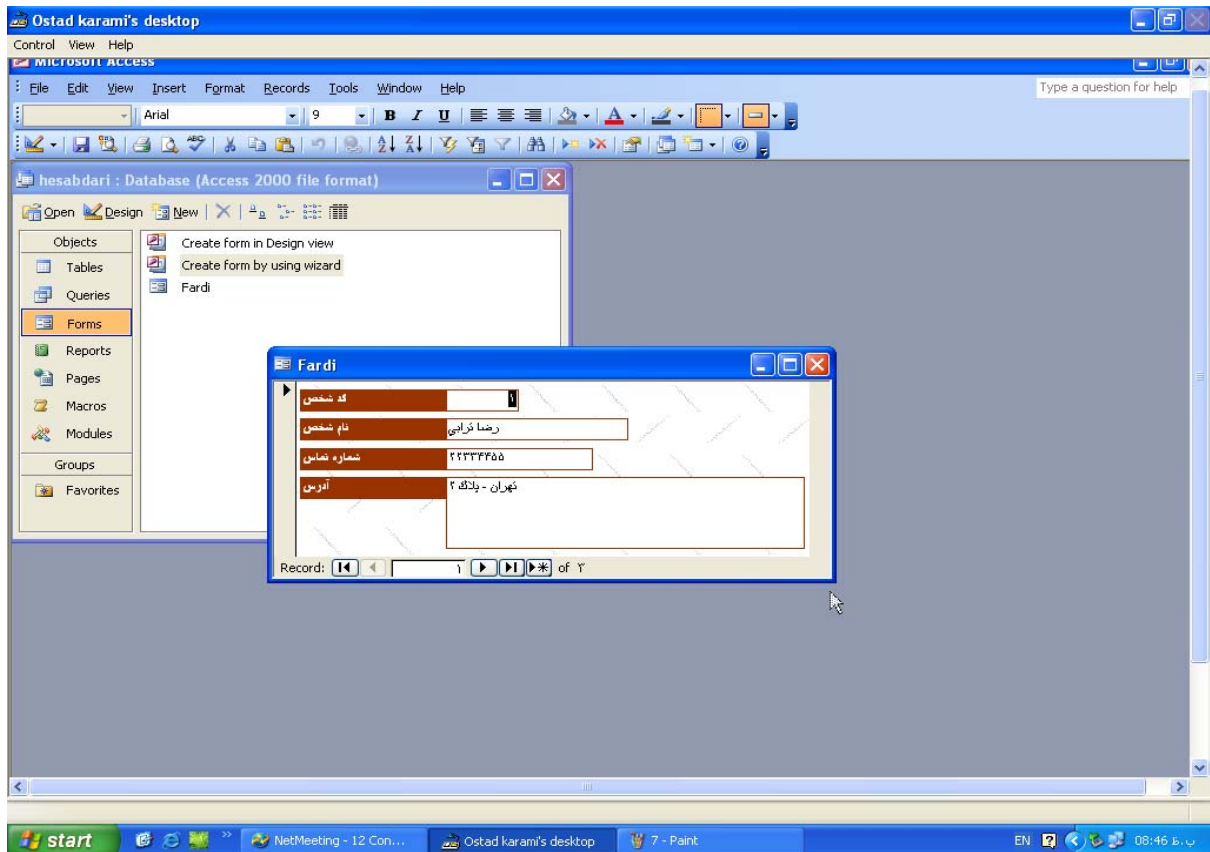


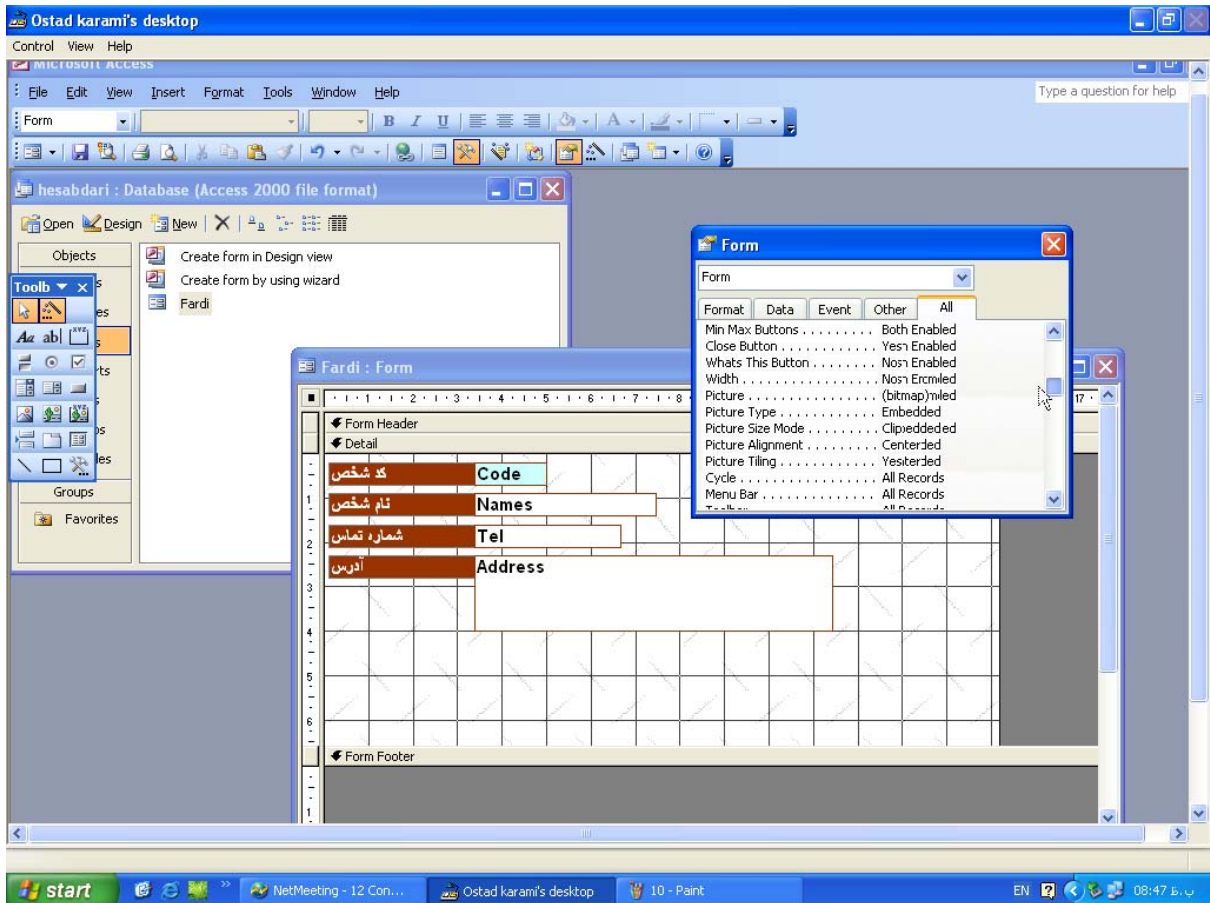
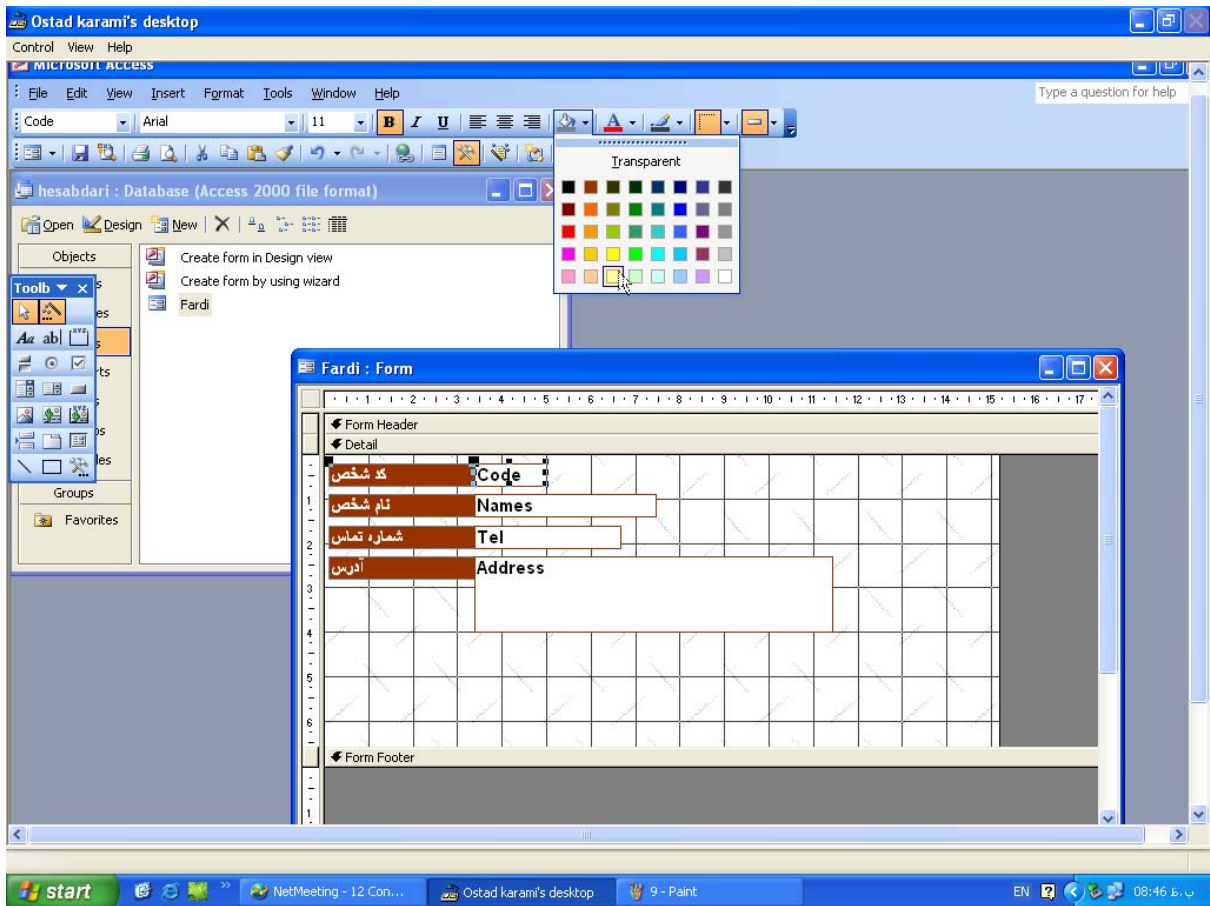
نام فرم را هر چه که لازم است می گذاریم

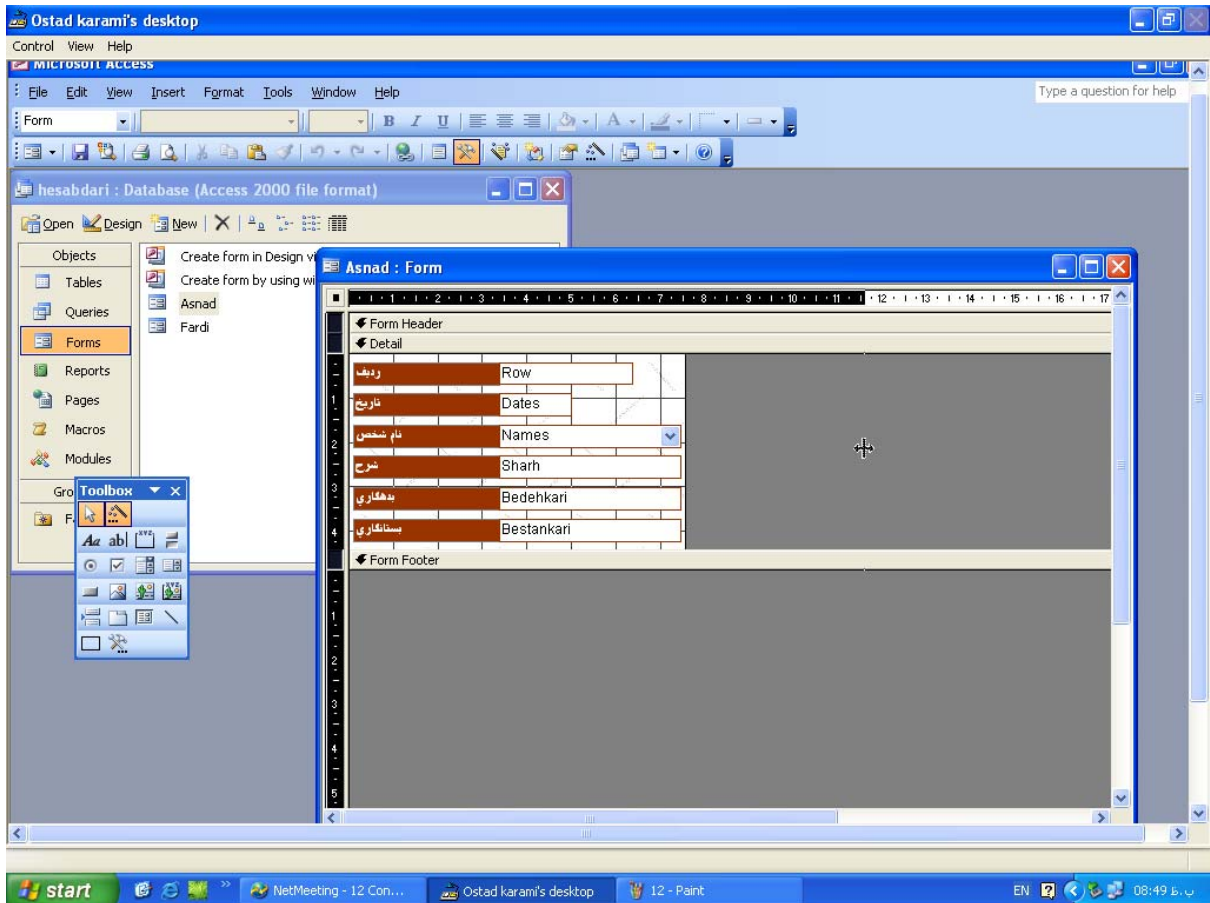
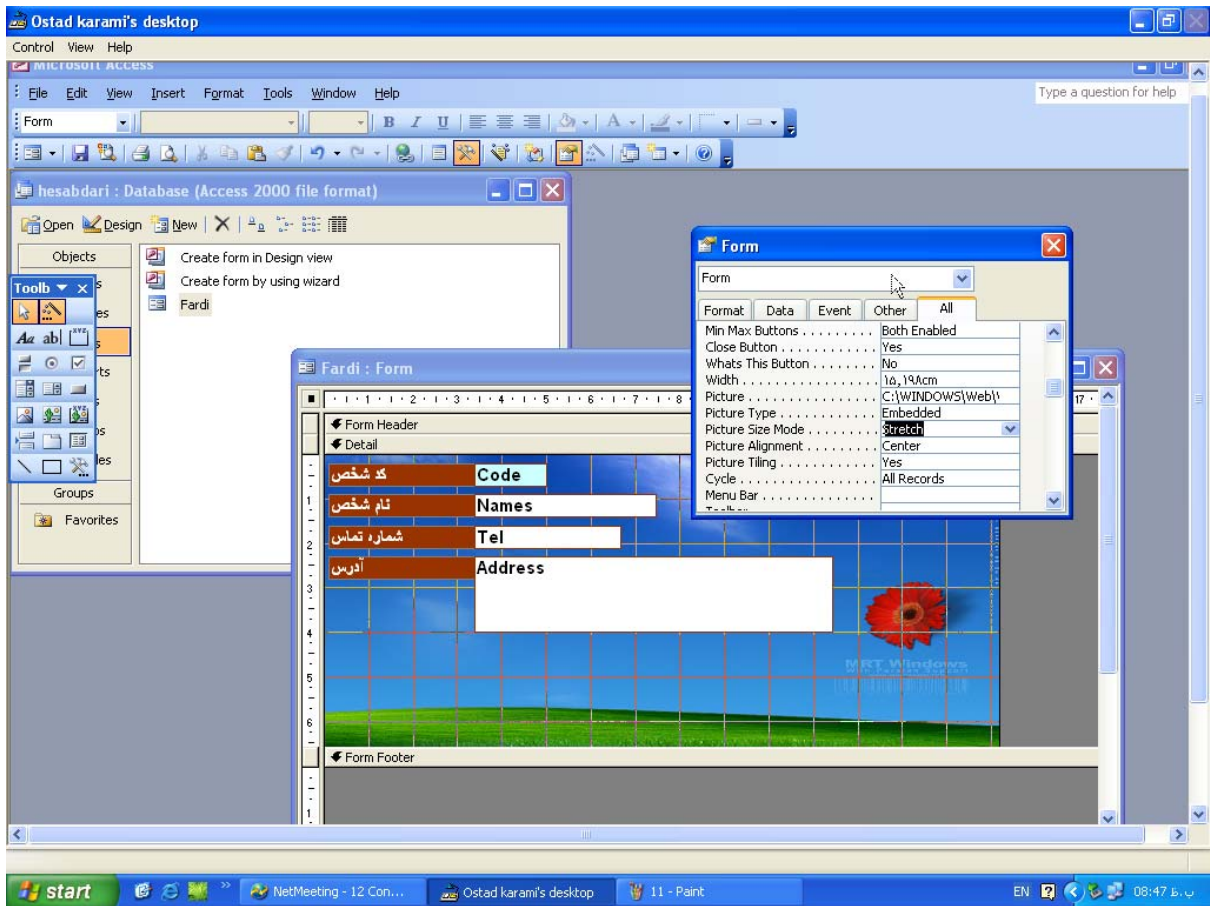


با کلیک بر روی کلید finish طراحی ابتدایی فرم موردنظرمان به پایان می رسد.

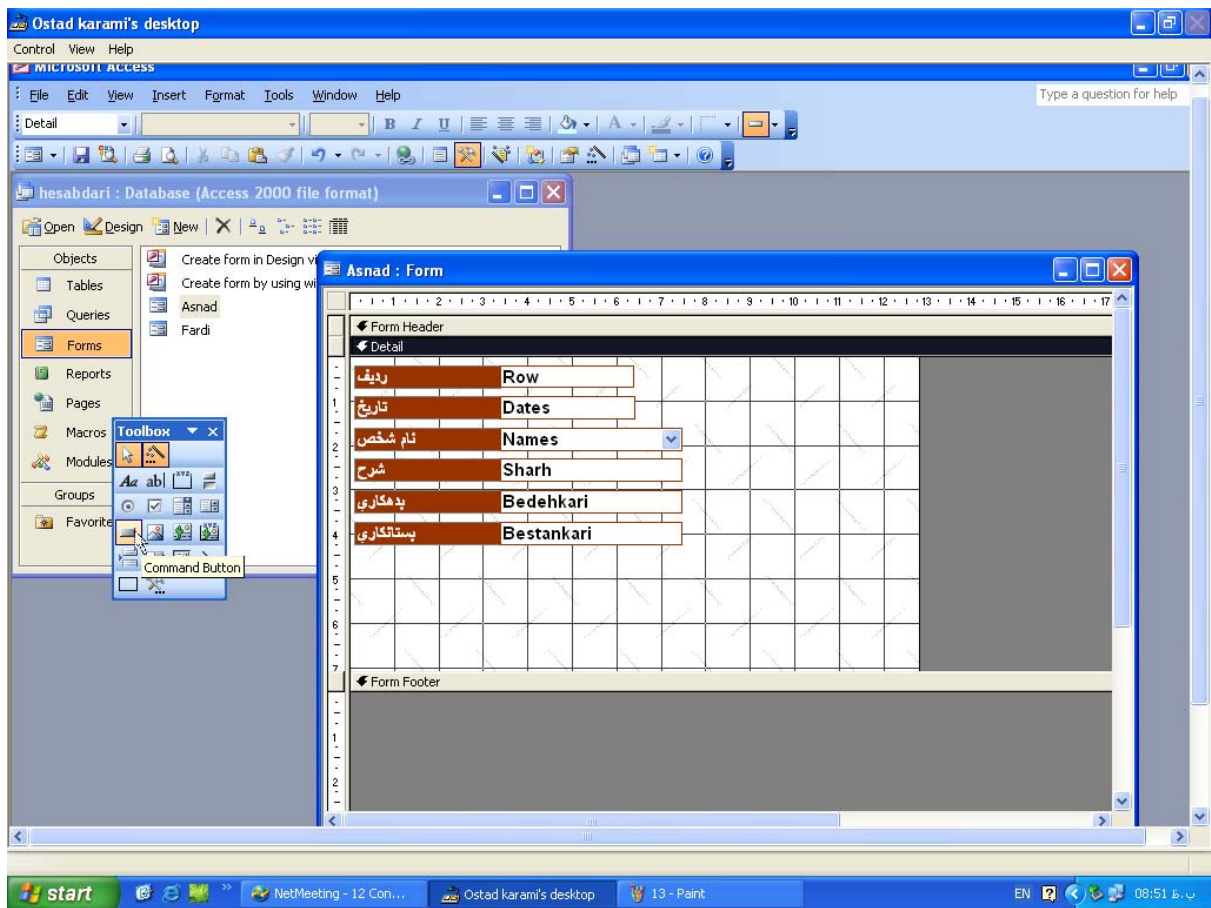
موارد دلخواه و خاص هریک از فرم ها را می توانیم در حالت design پیاده سازی و طراحی کنیم .



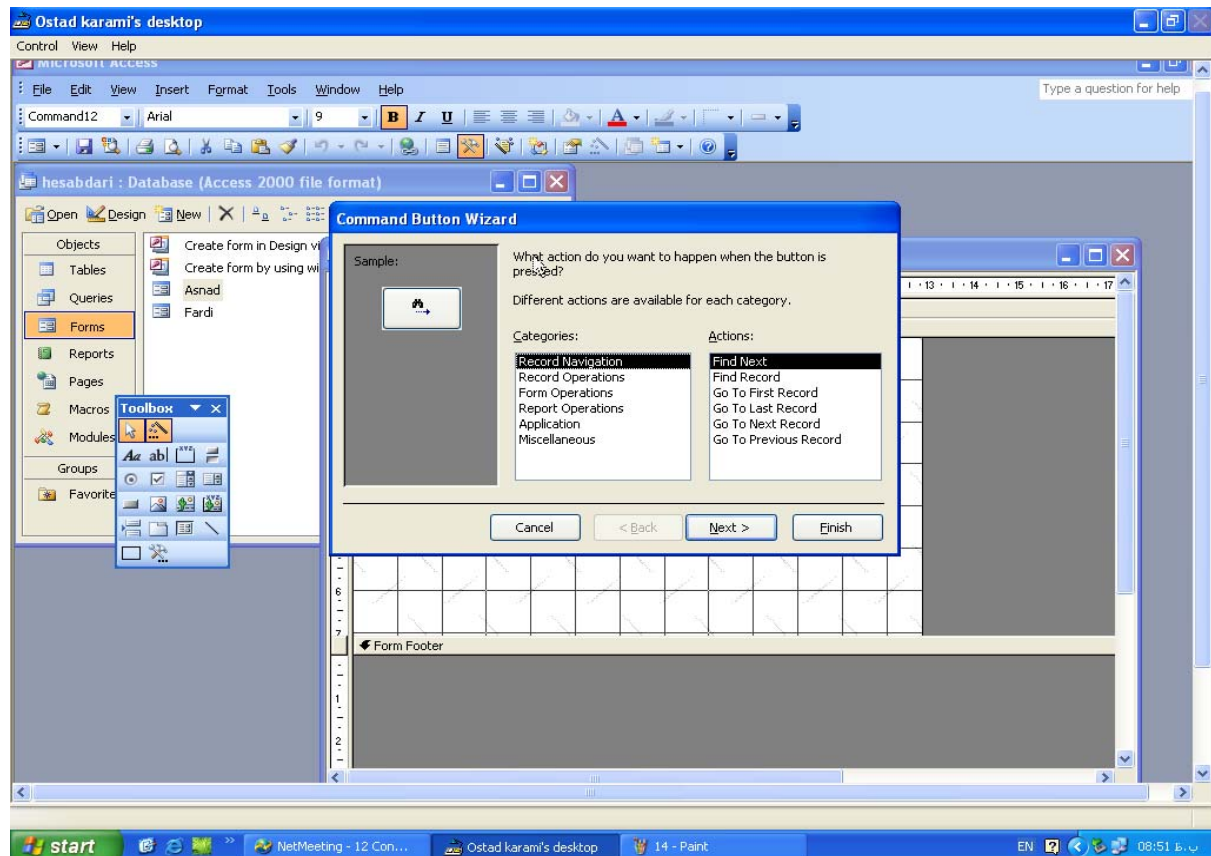




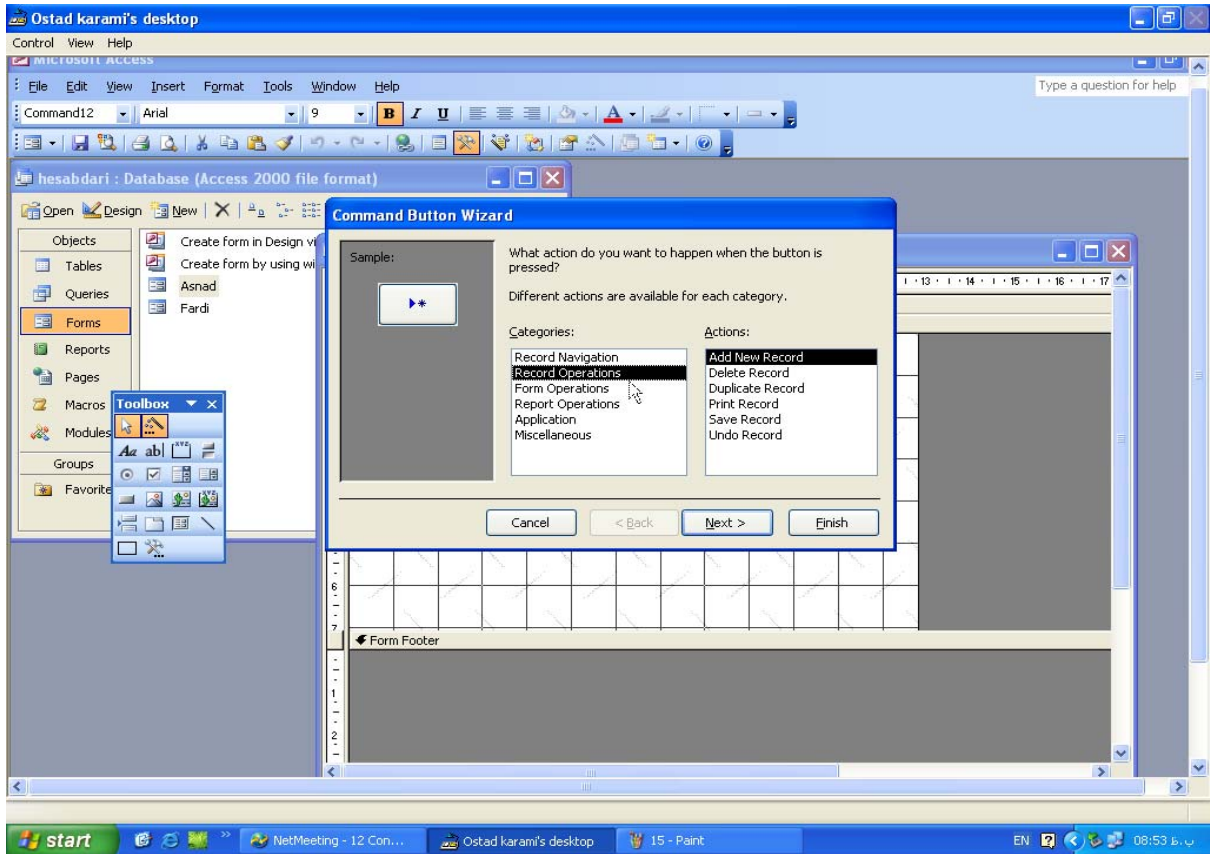




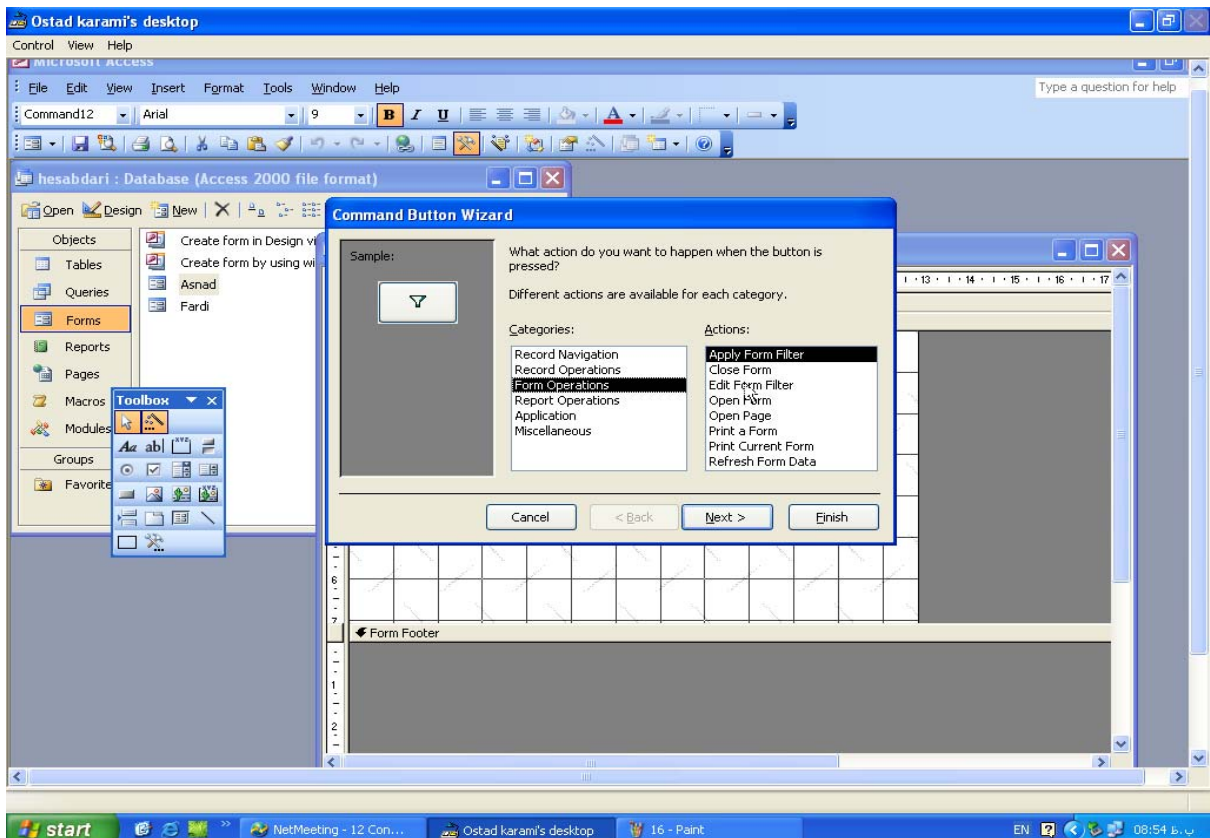
اضافه کردن کلیدهای اجرایی در هر یک از فرم ها با استفاده از دکمه های command می باشد.



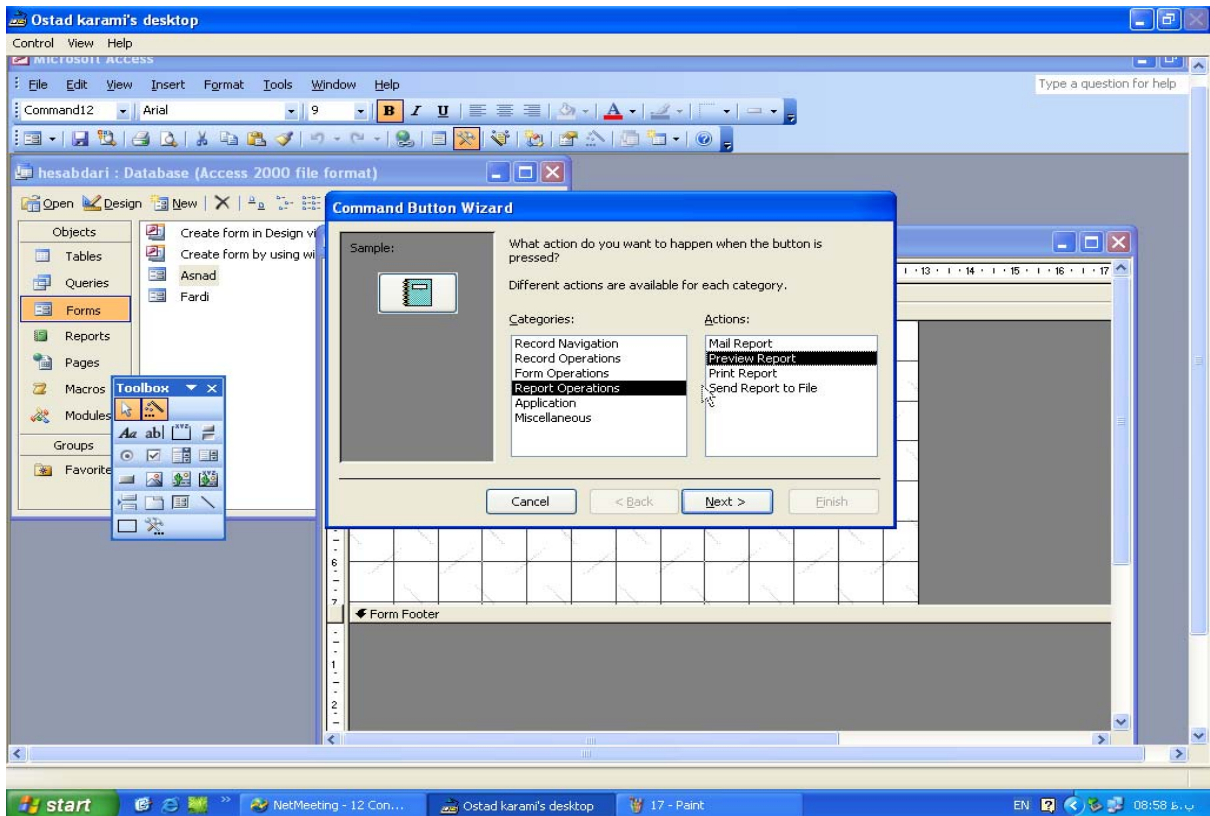
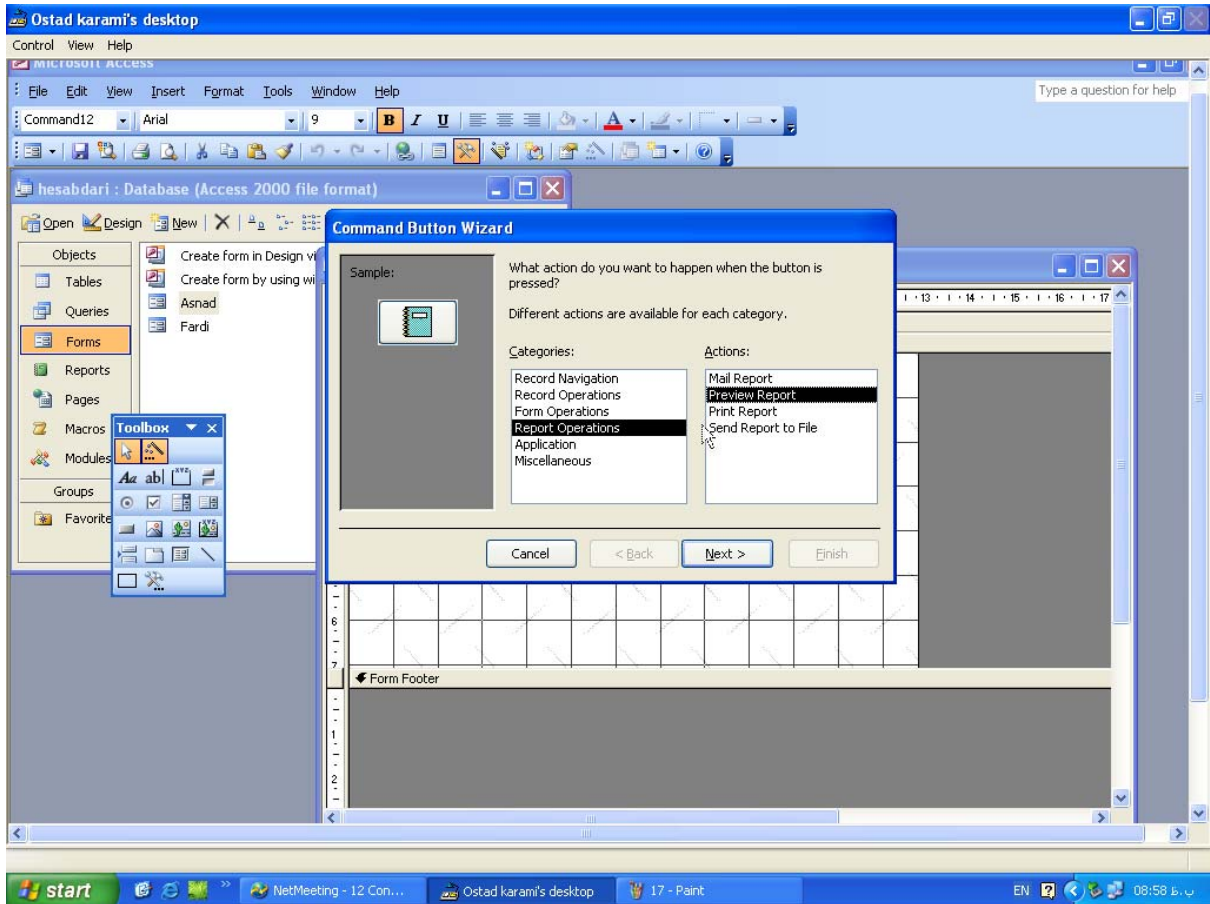
مثلاً" برای ایجاد کلیدر کورد جدید add new record را انتخاب می کنیم .



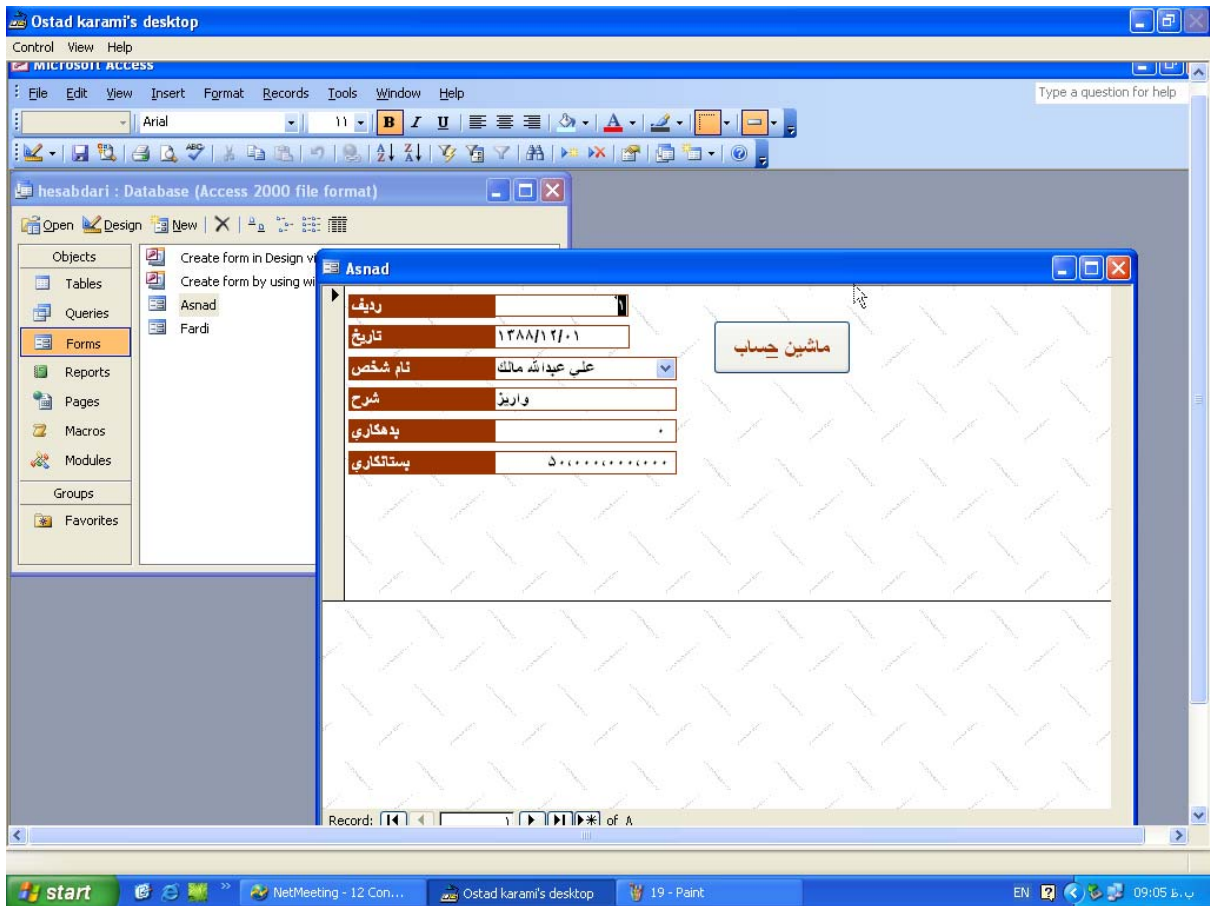
مثلاً" برای بازکردن یک فرم روی open form کلیک می کنیم.



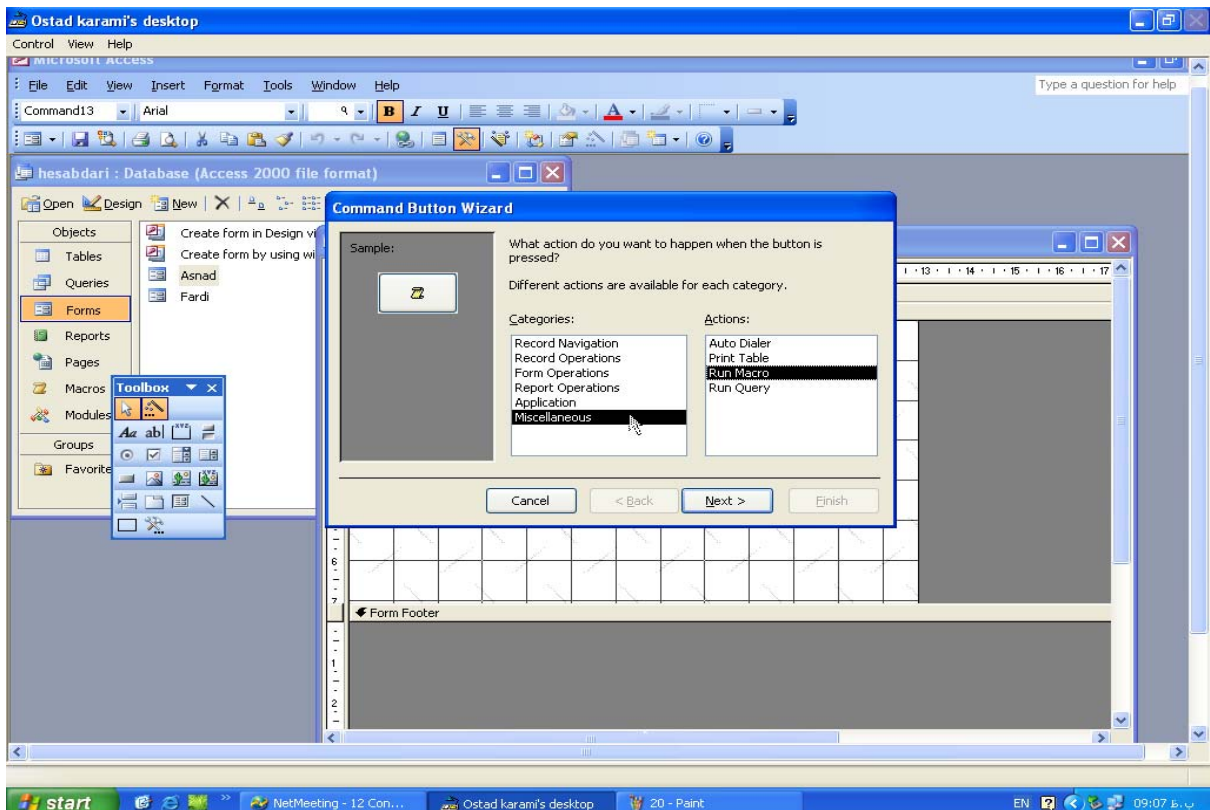
برای نشان دادن پیش نمایش preview report انتخاب می کنیم

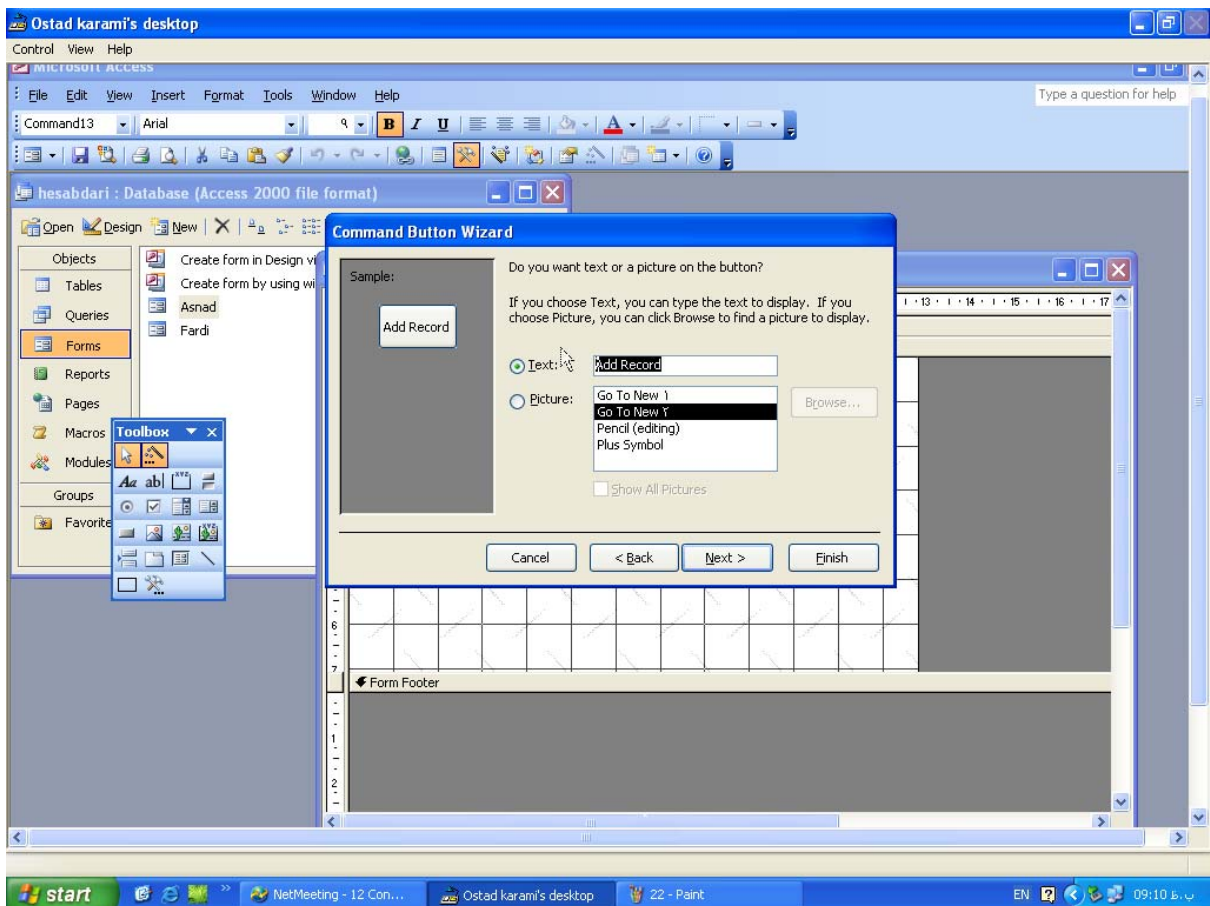
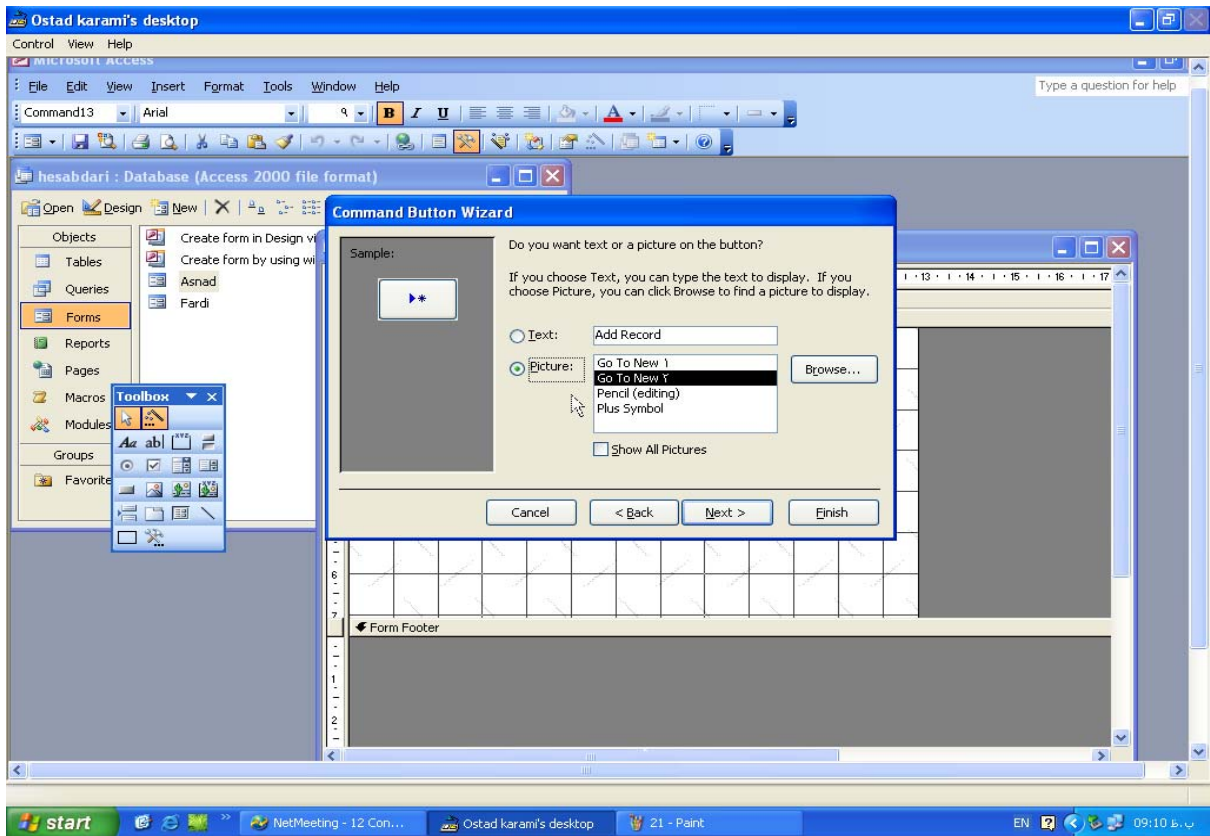


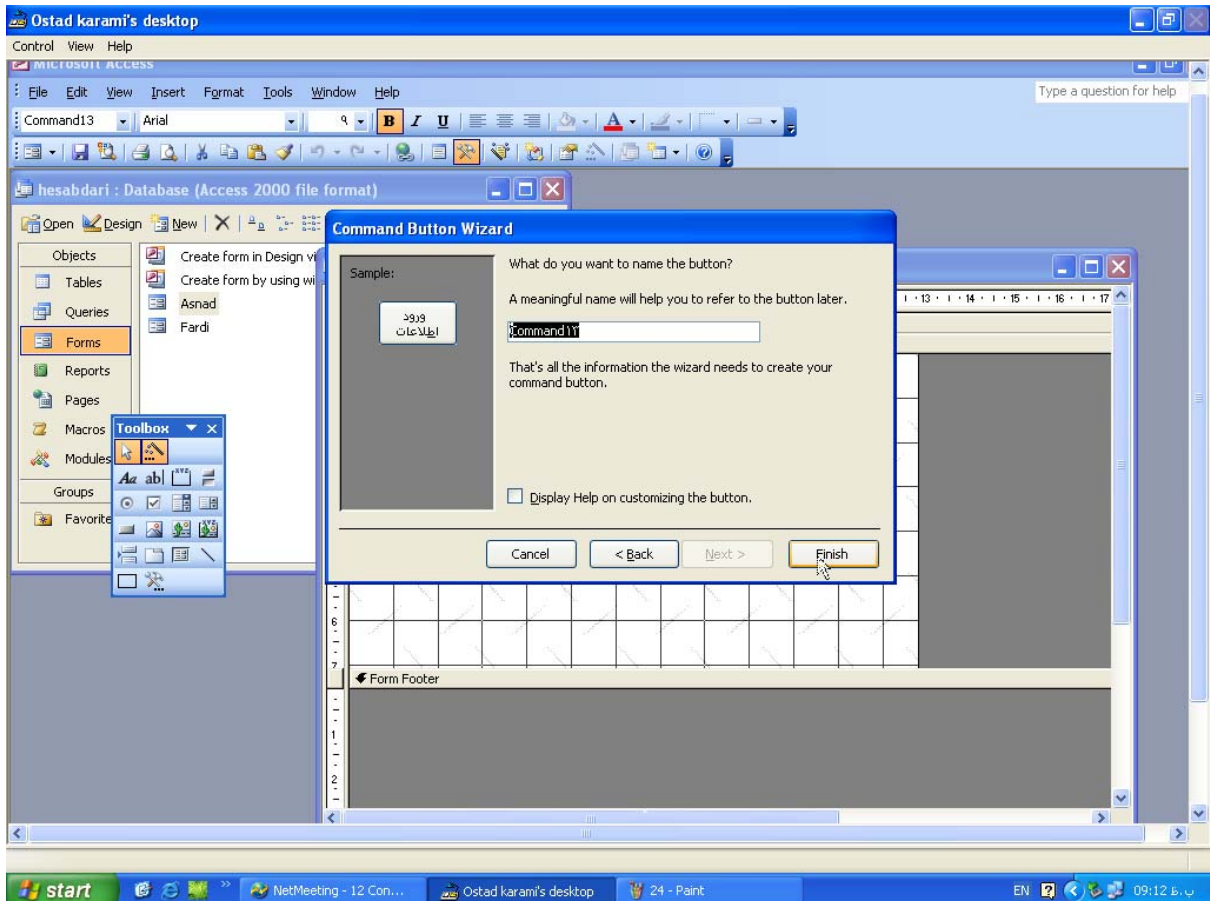
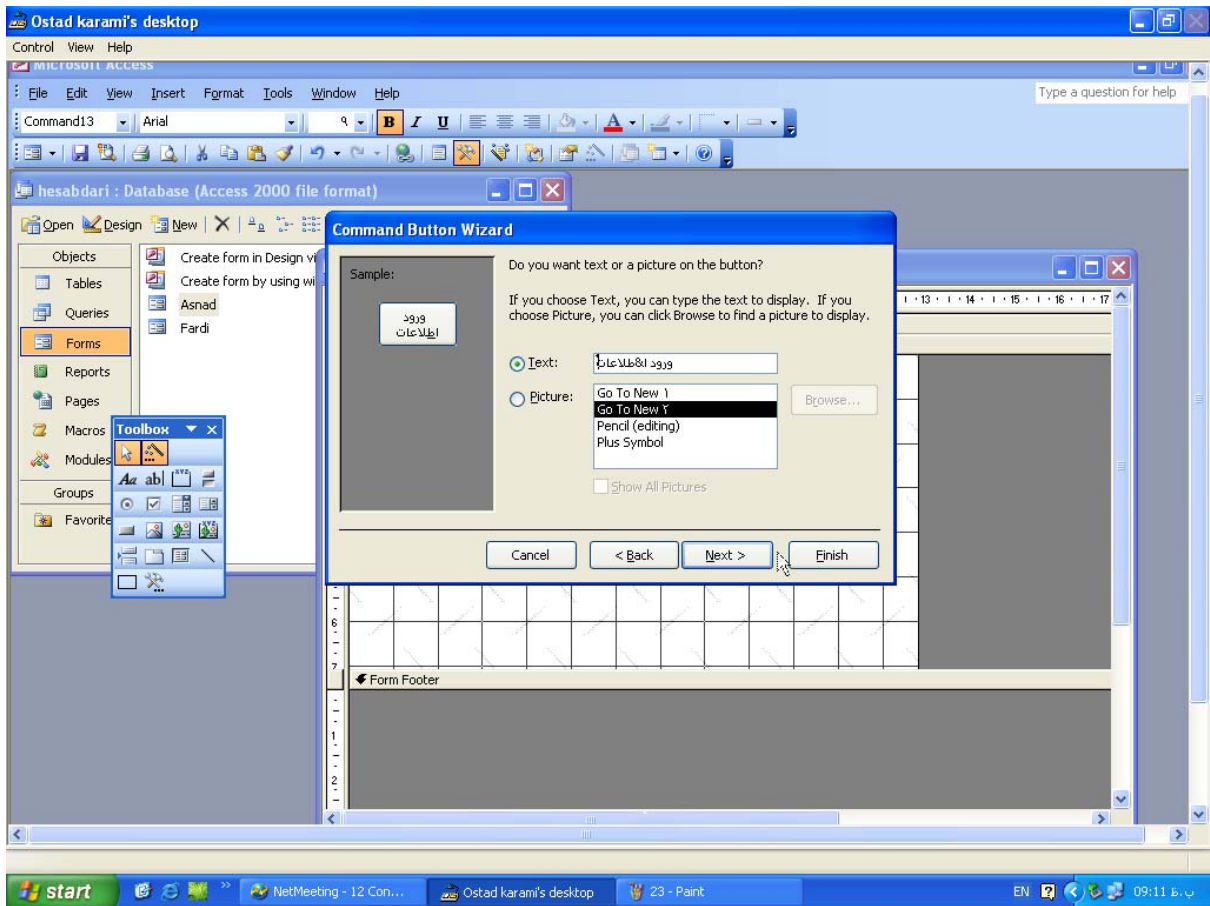
و برای اجرای نرم افزارهای کاربردی در قسمت application هریک از exeهایی را که می خواهیم انتخاب می کنیم .

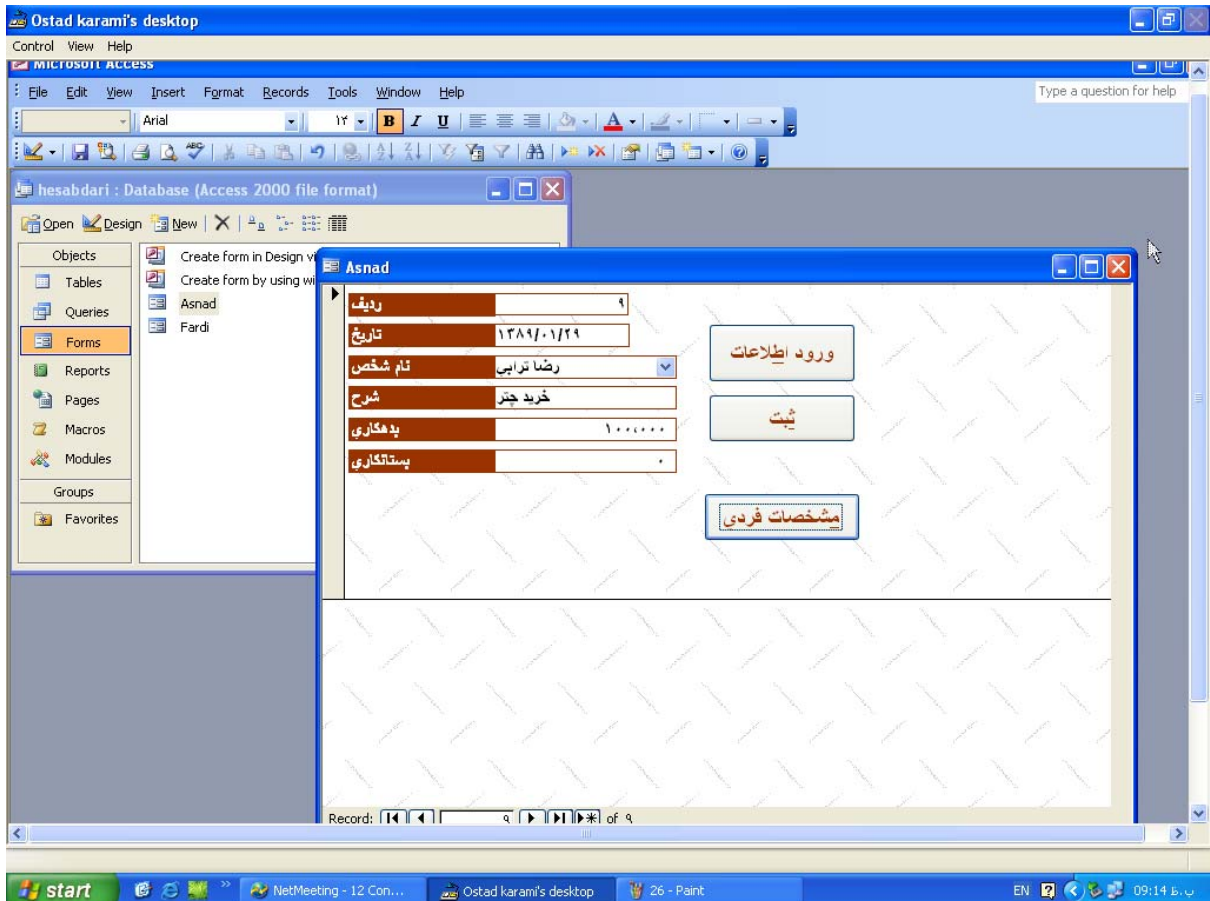
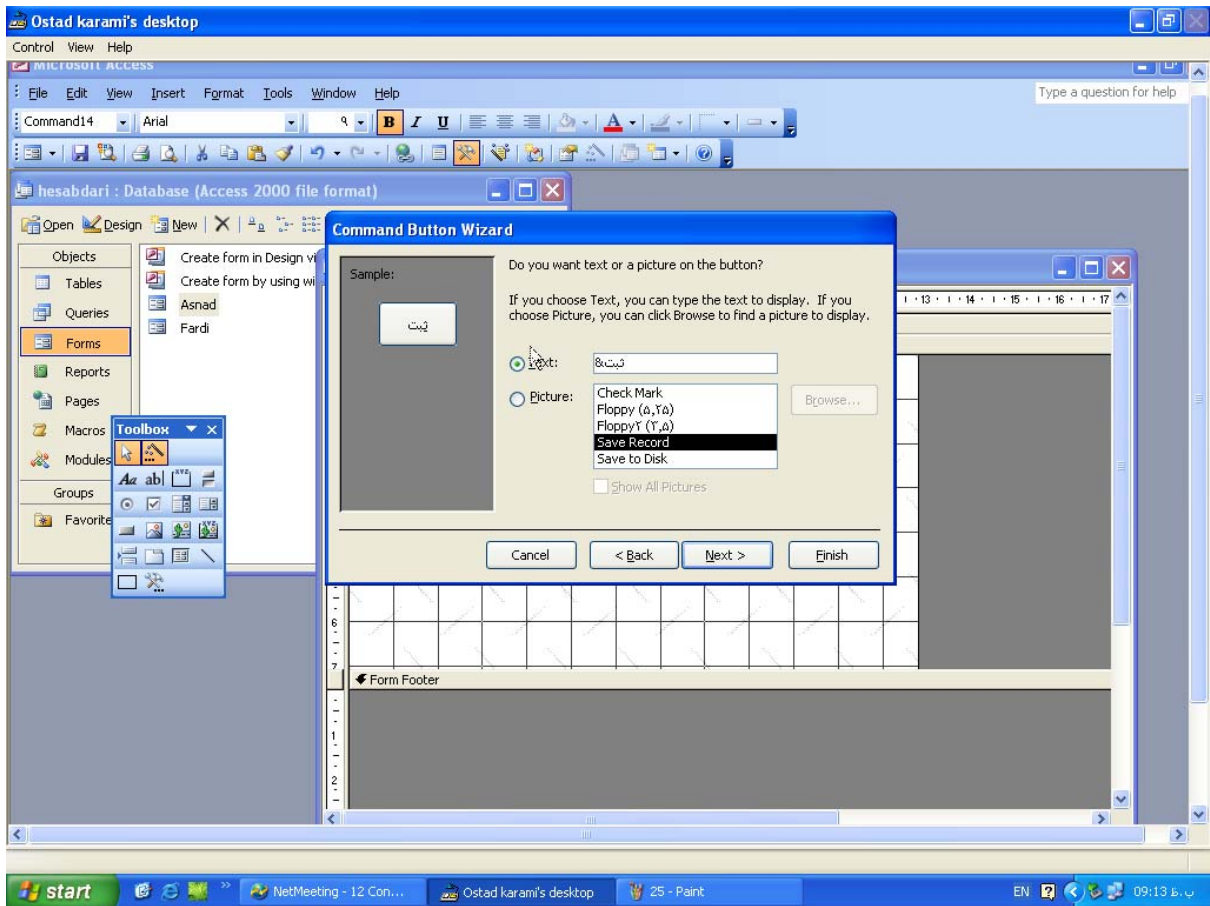


برای اجرای ماکروهم به ترتیب زیرعمل می کنیم ،

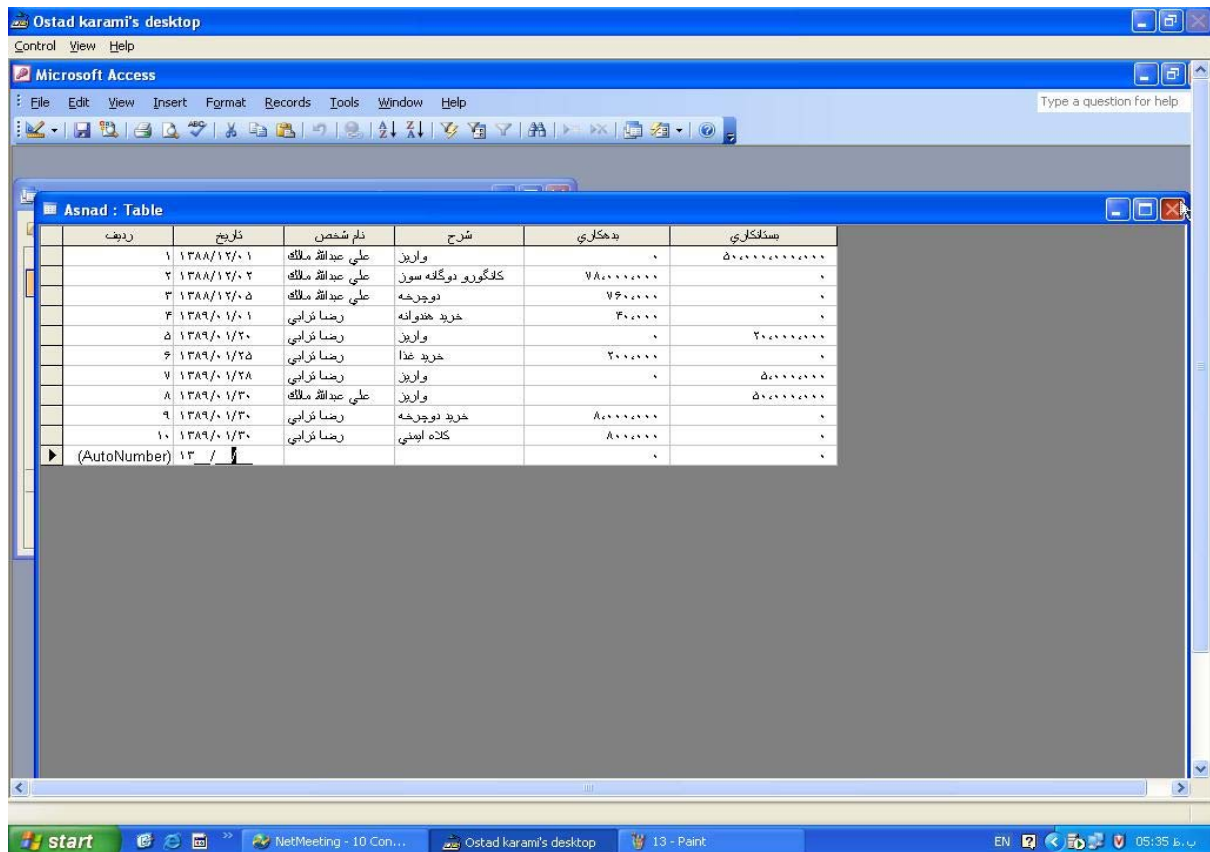
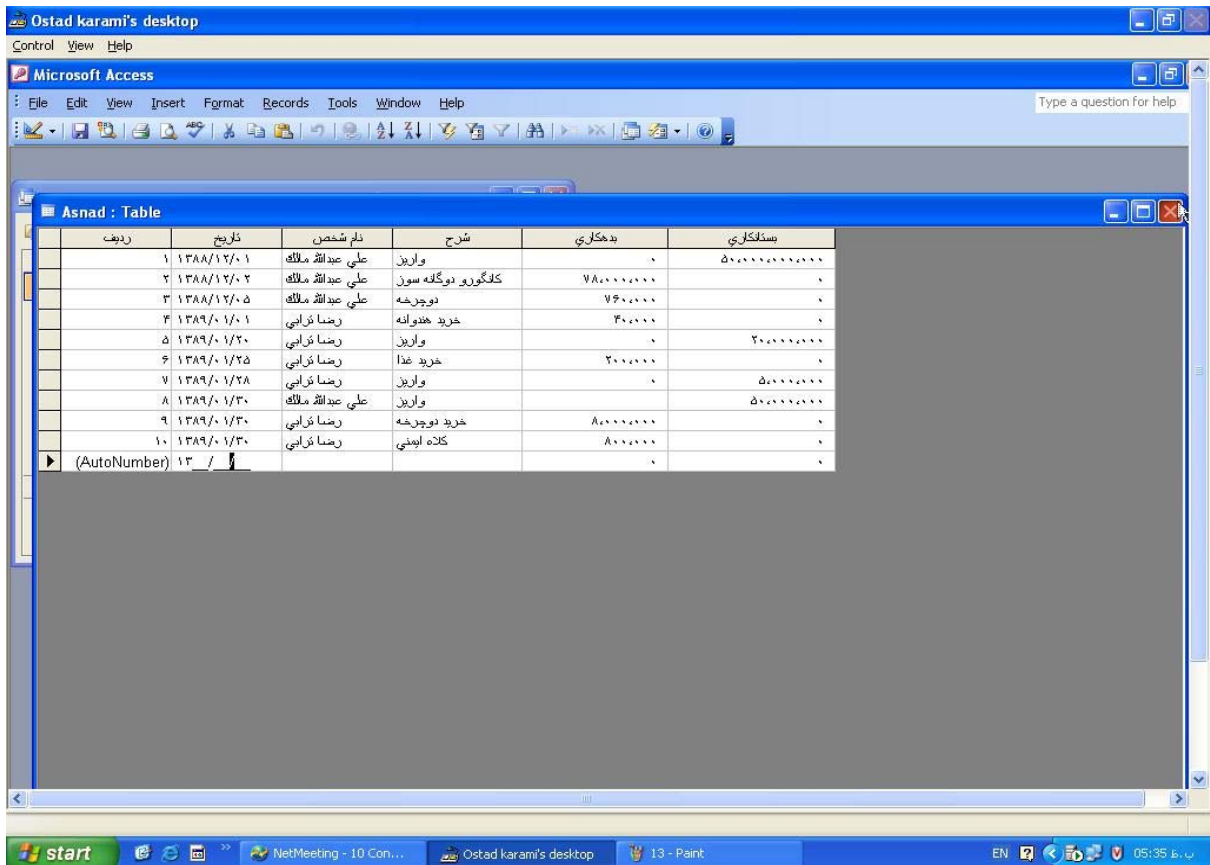




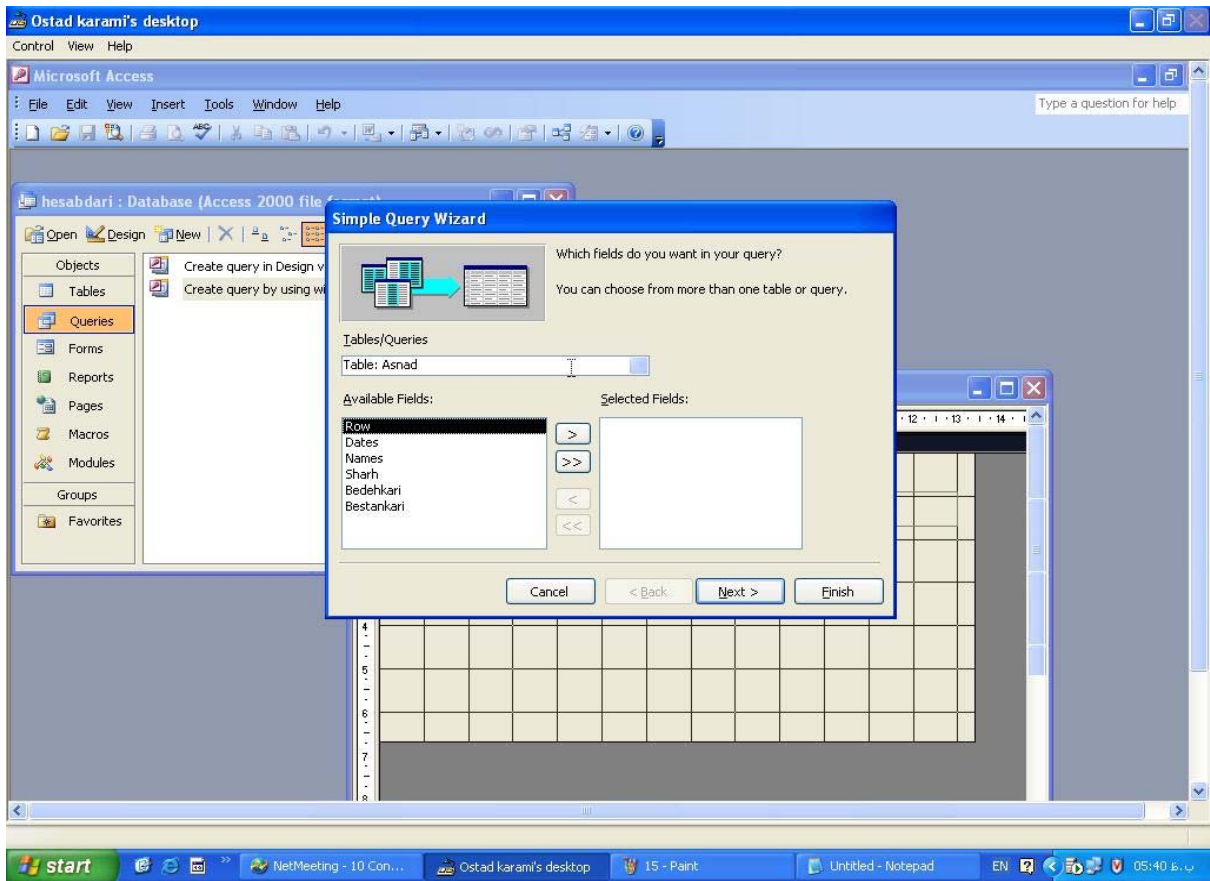




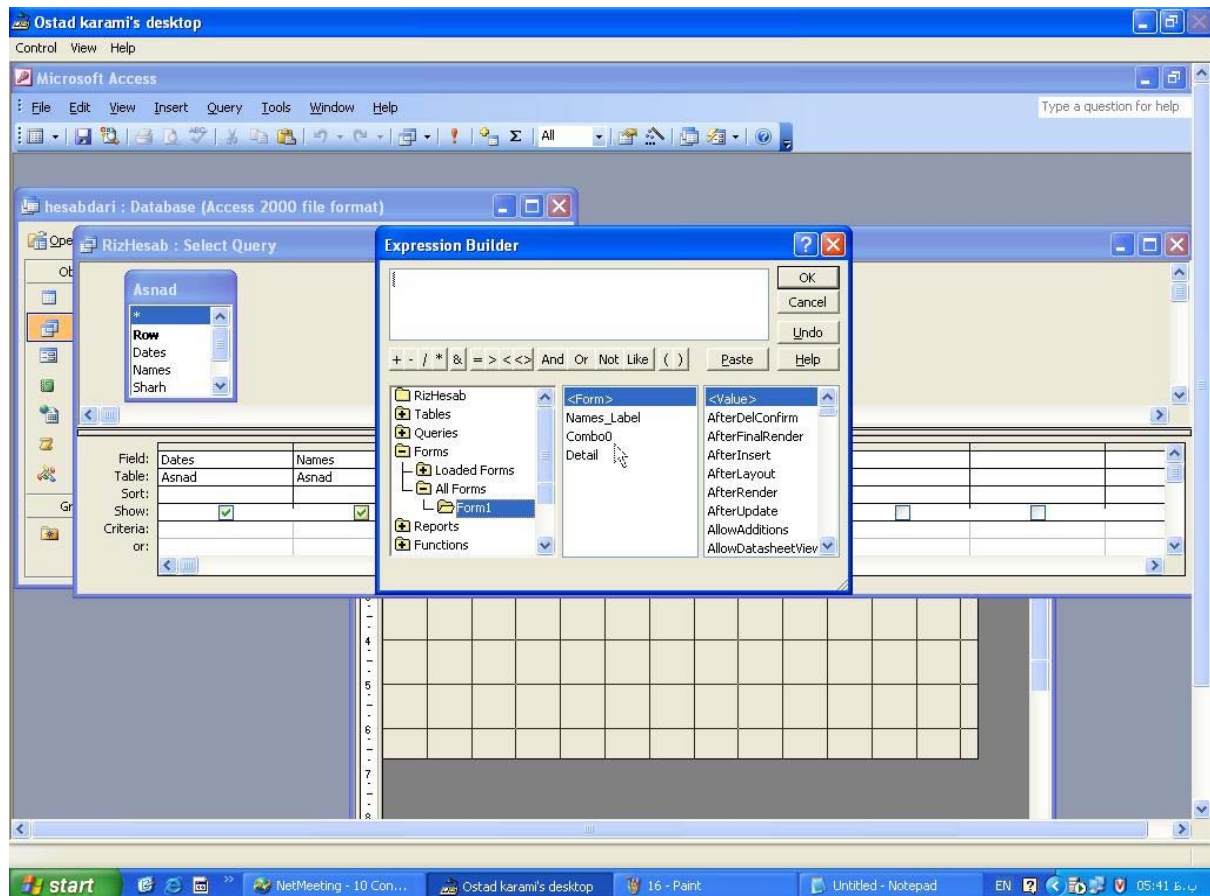
برای گرفت گزارش حرفه ای ابتدا برای گرفت گزارش حرفه ای query مورد نظر رامی سازیم .

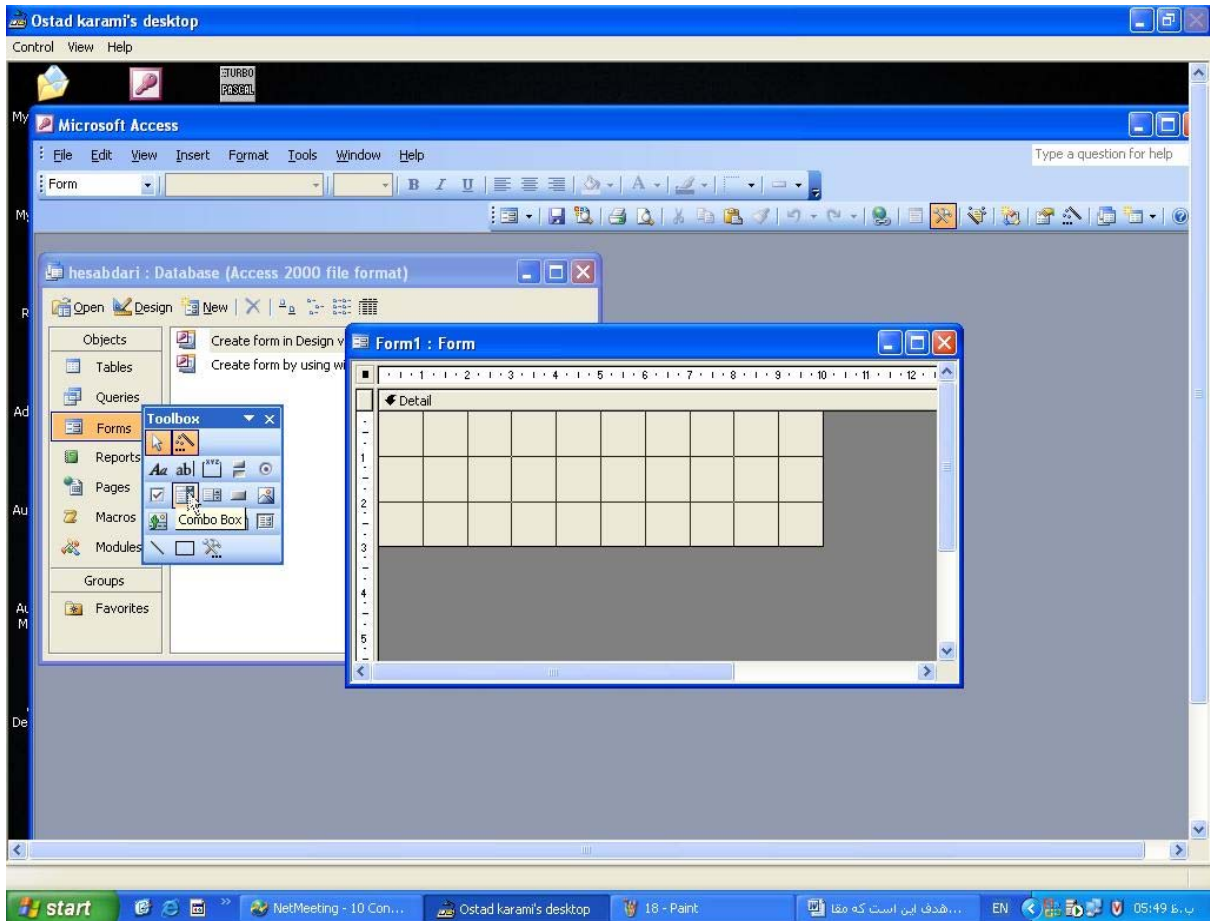
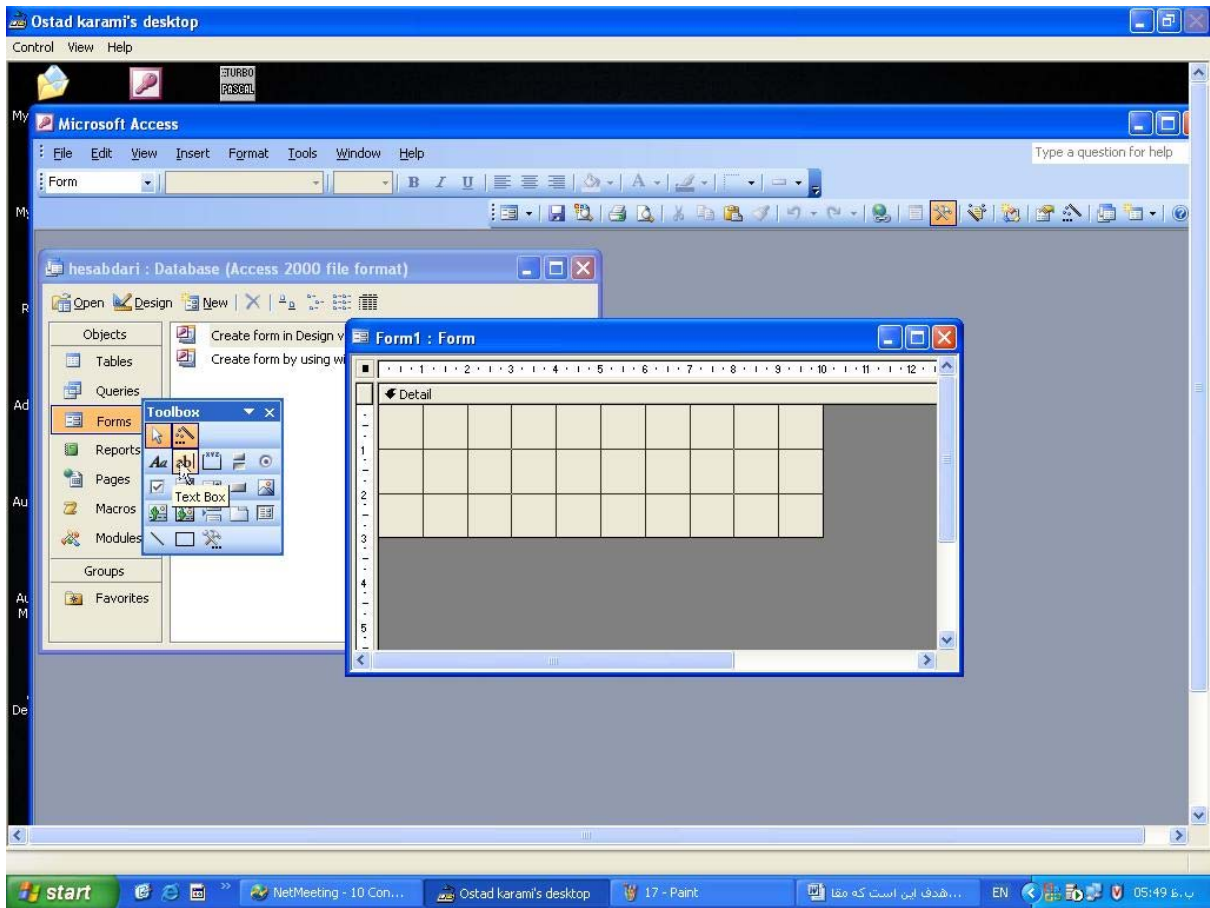


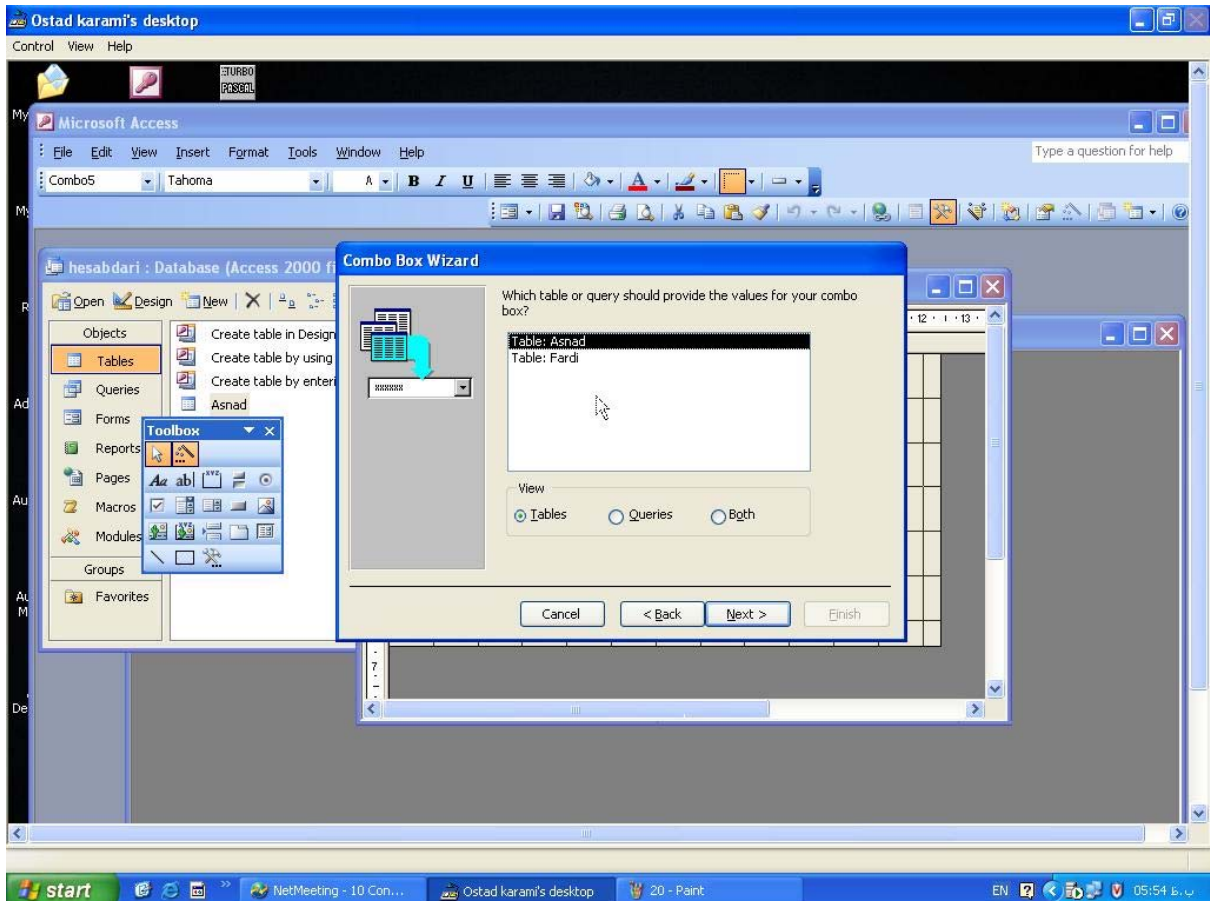
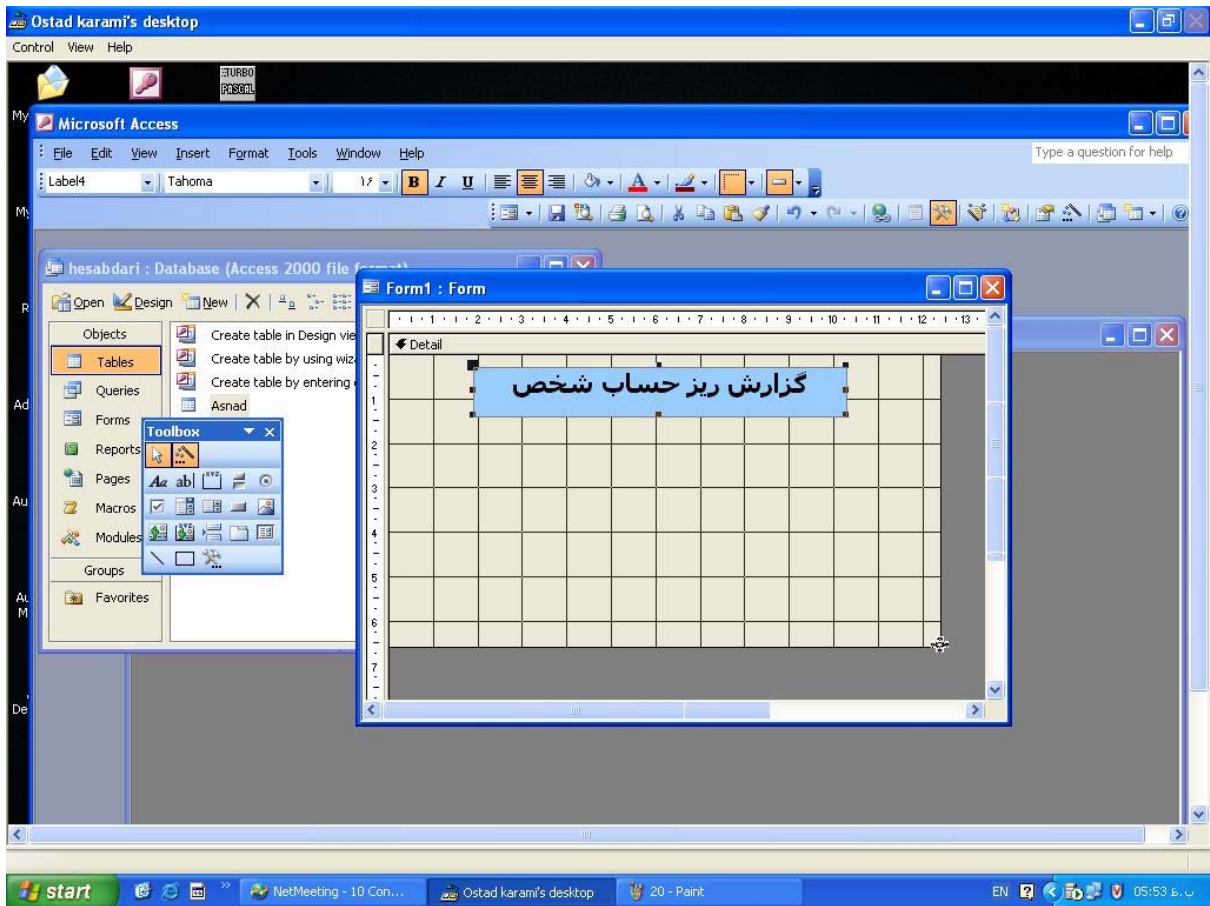


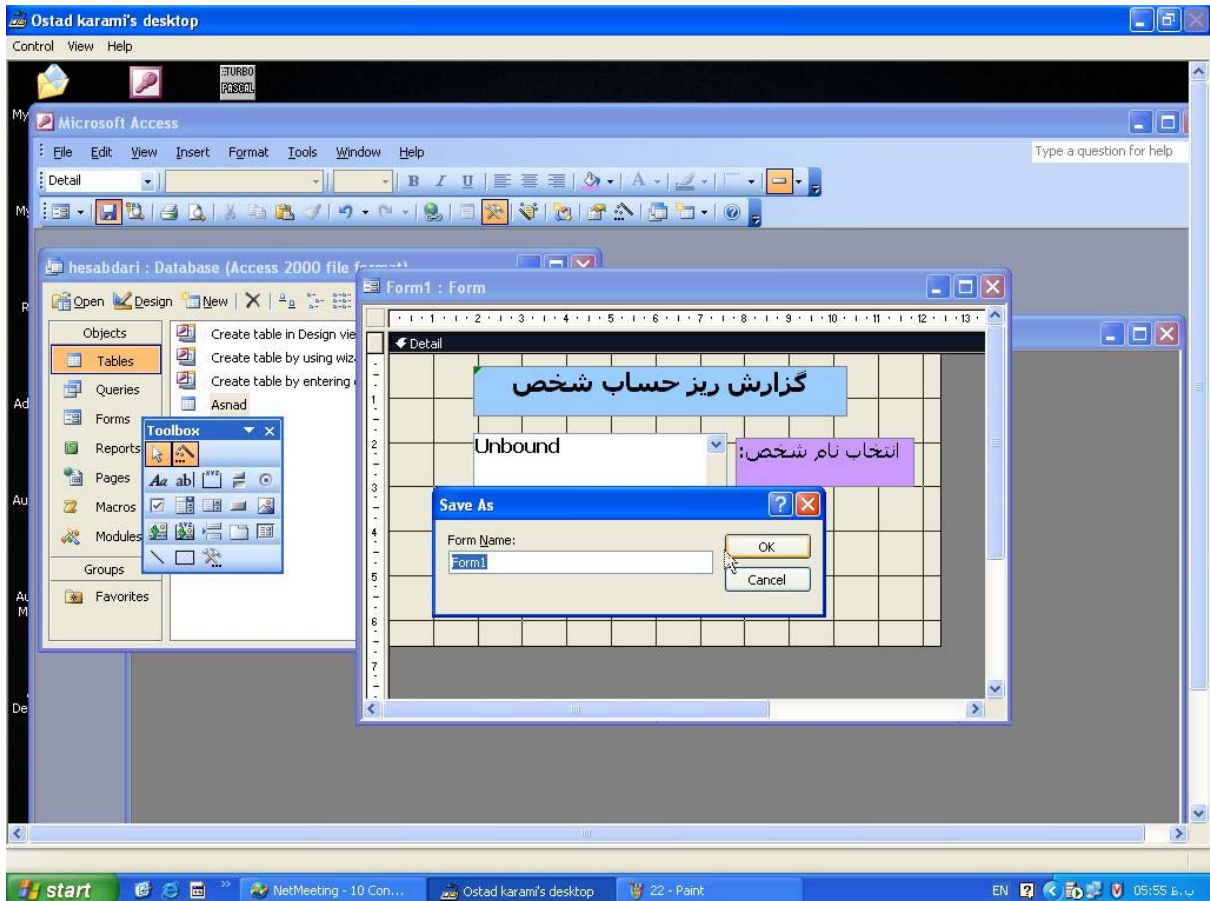
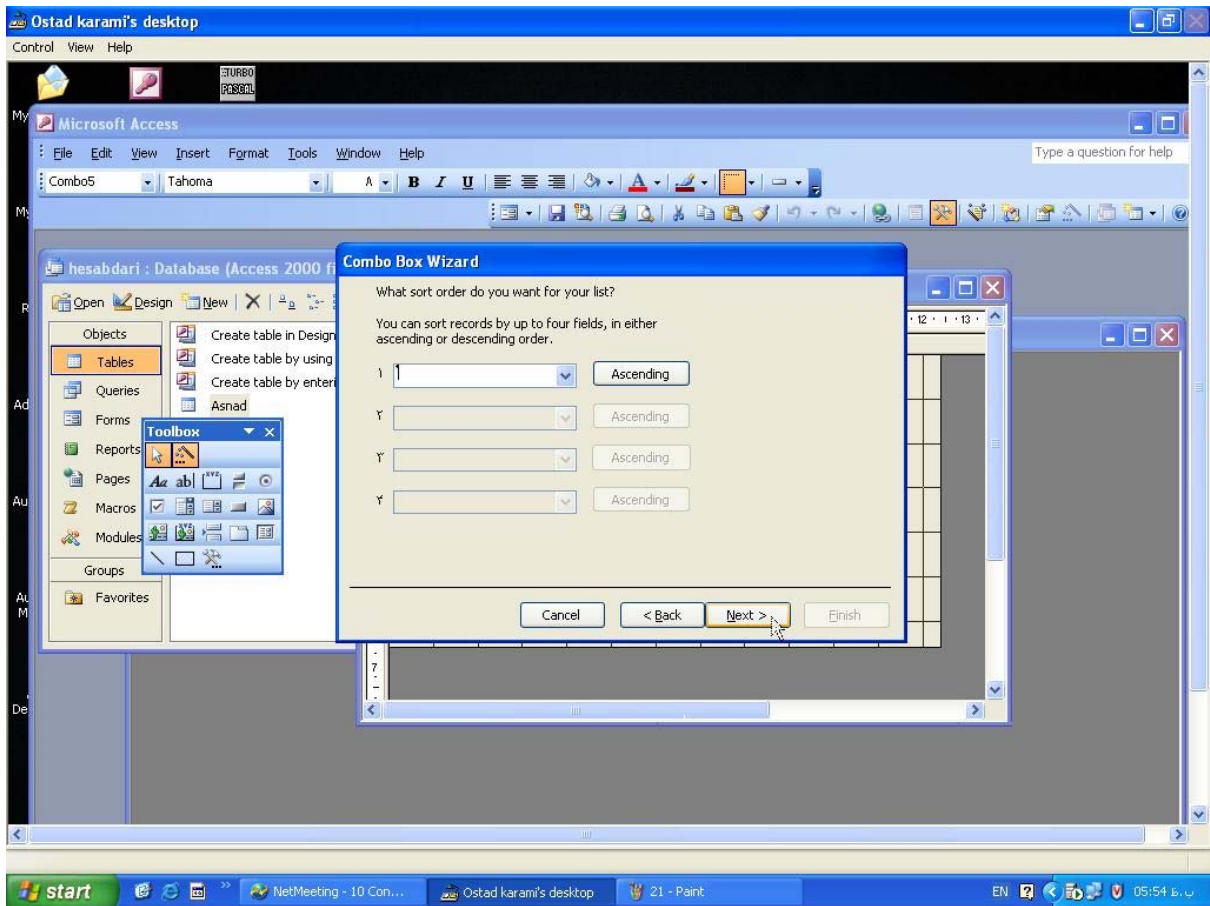


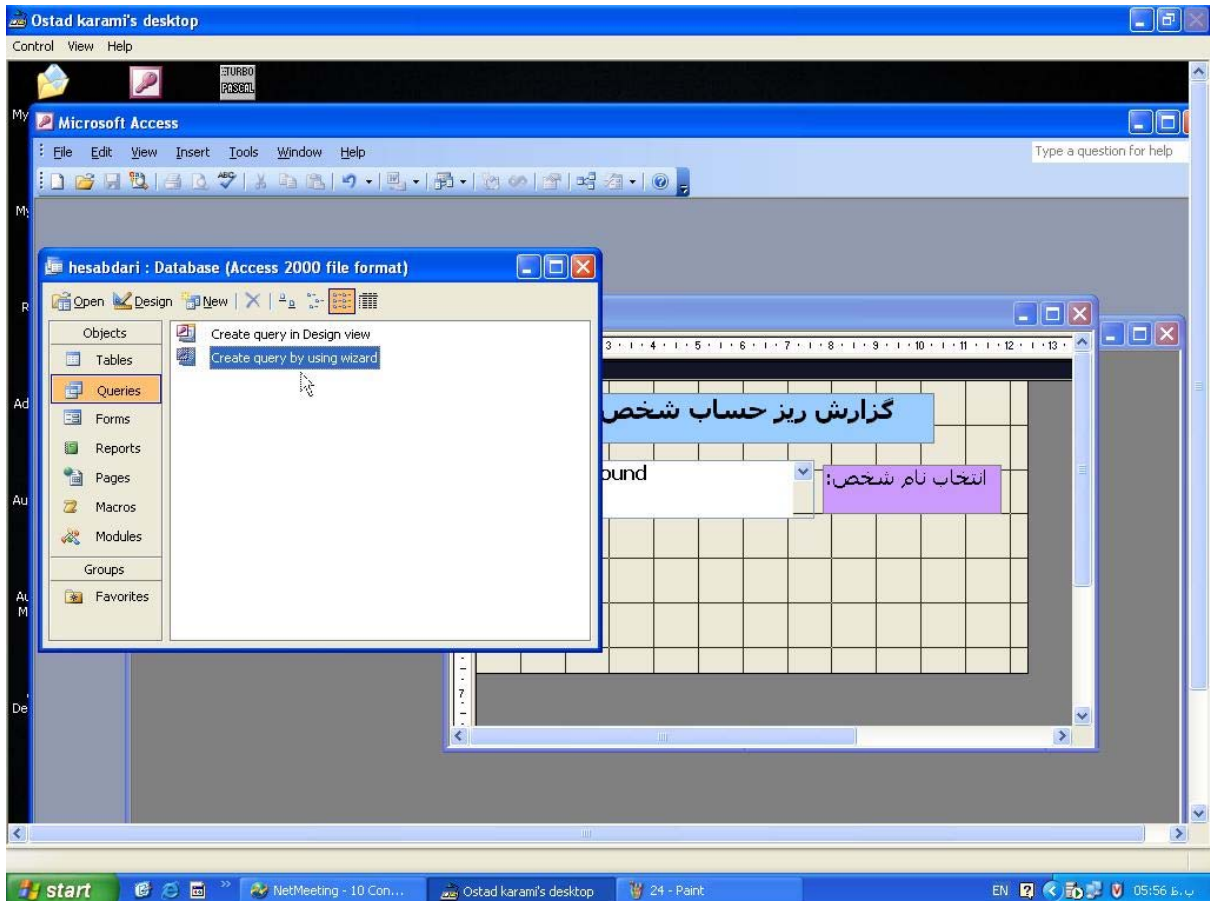
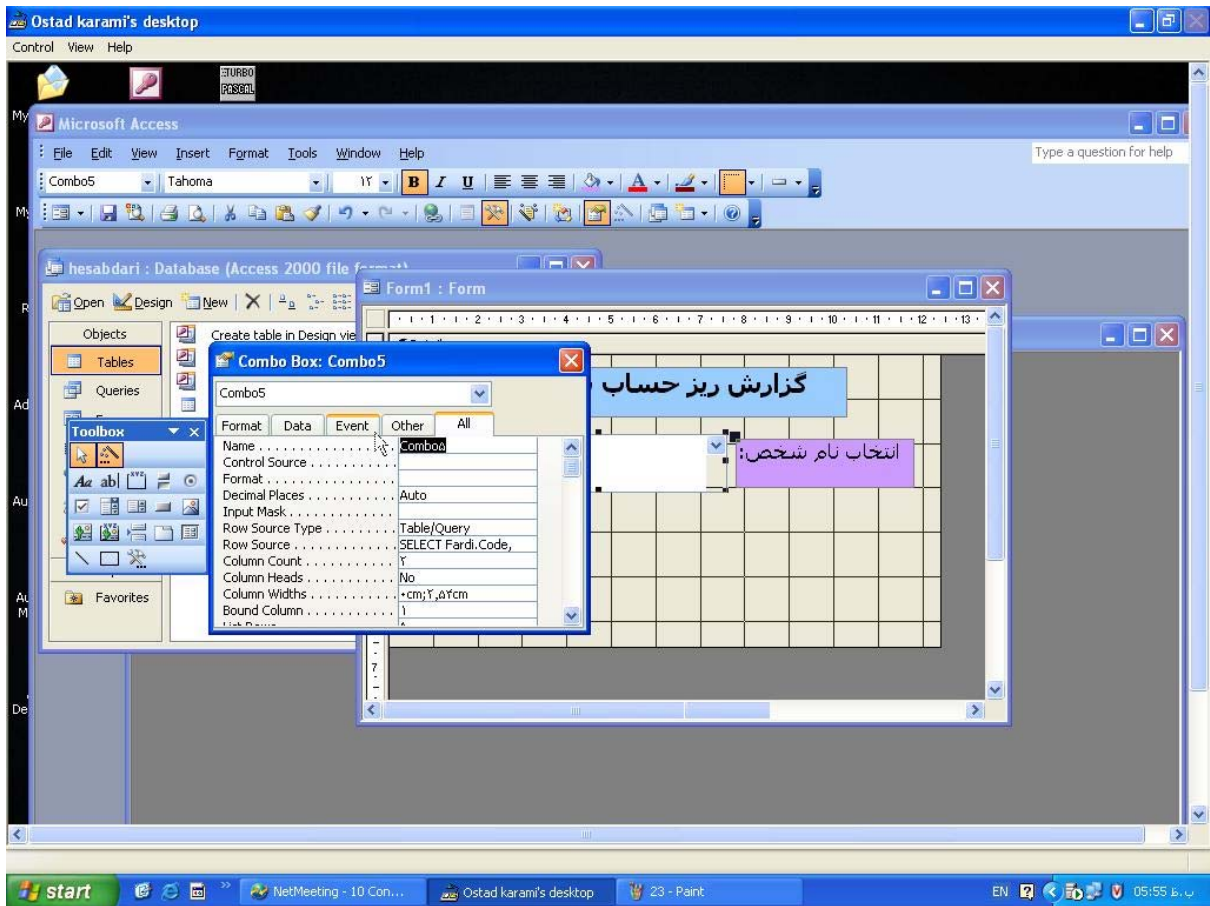
سپس شرایط مورد نظر را اعمال می کنیم

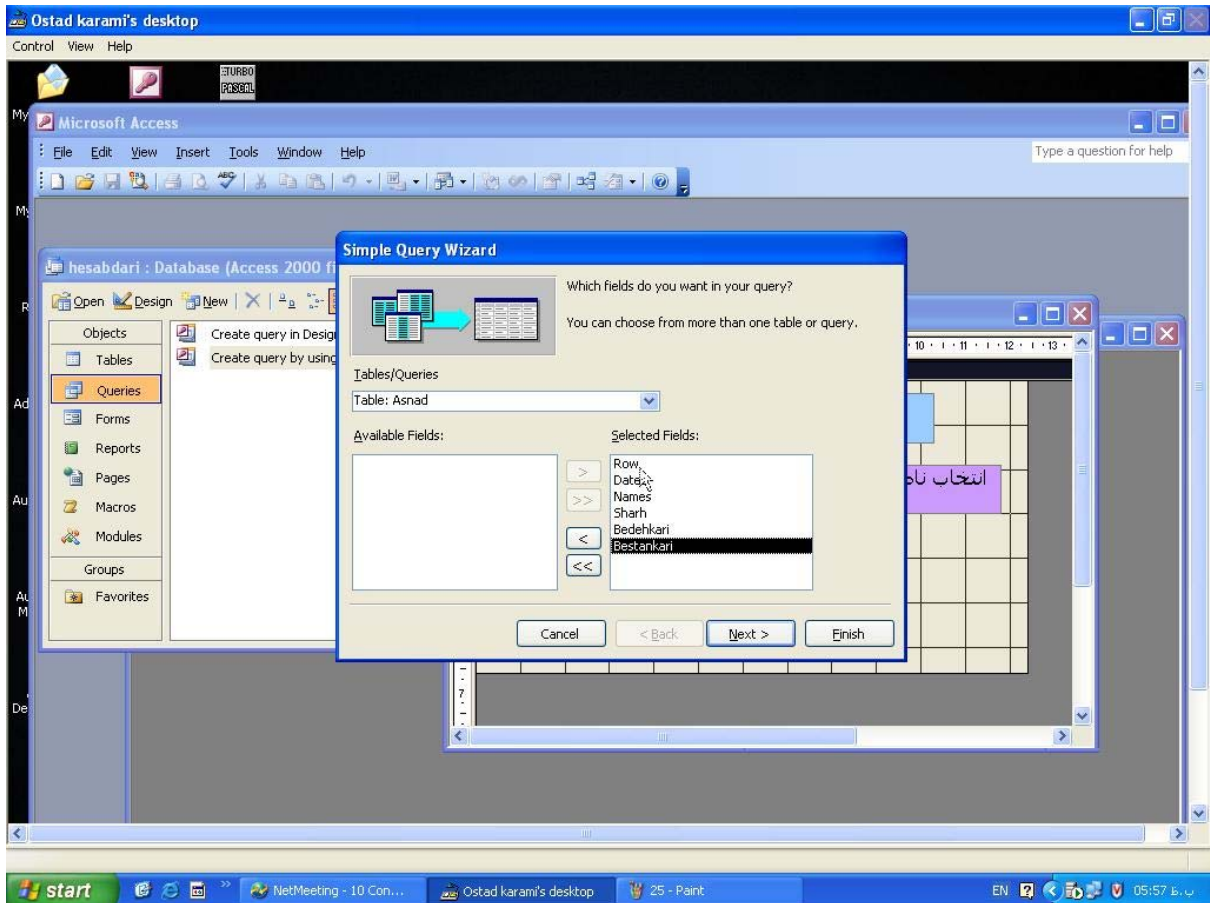
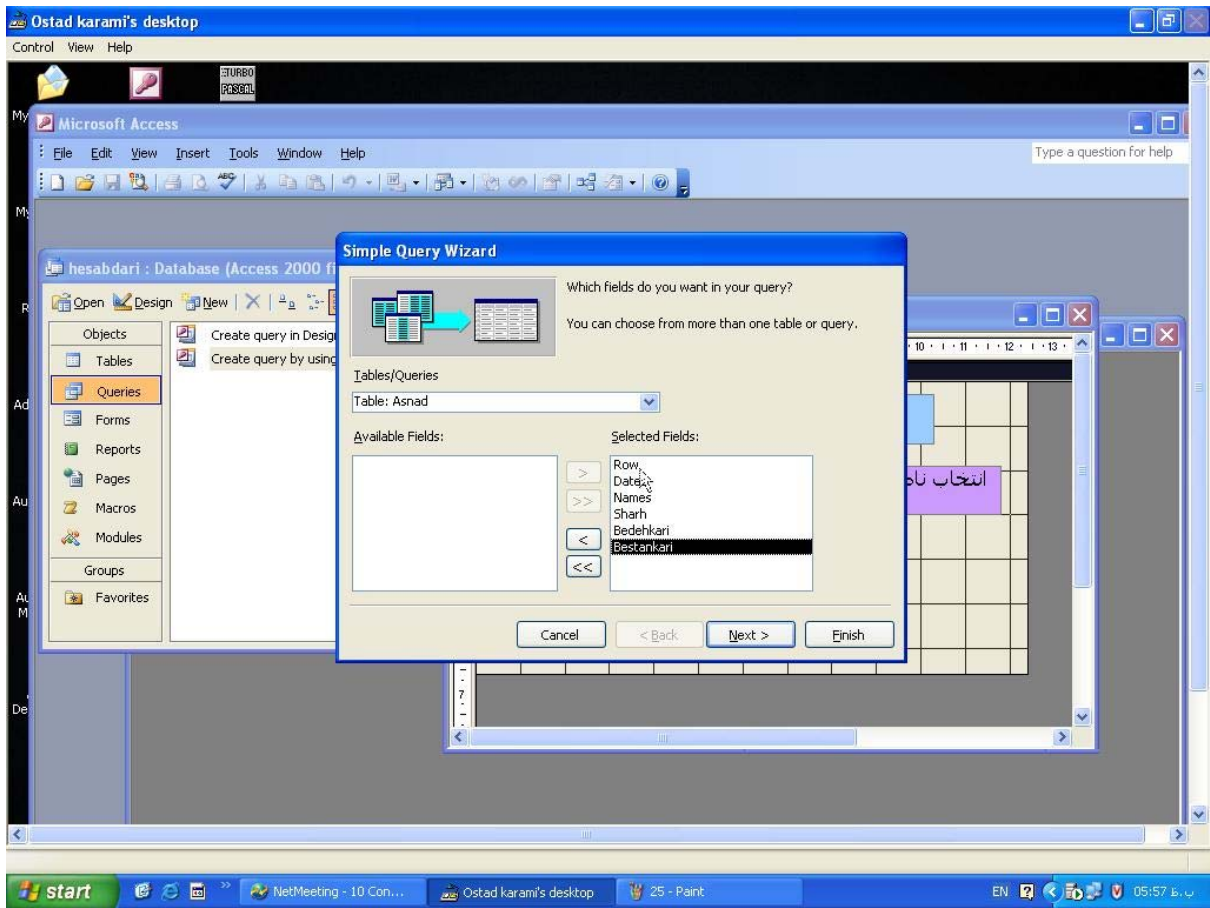


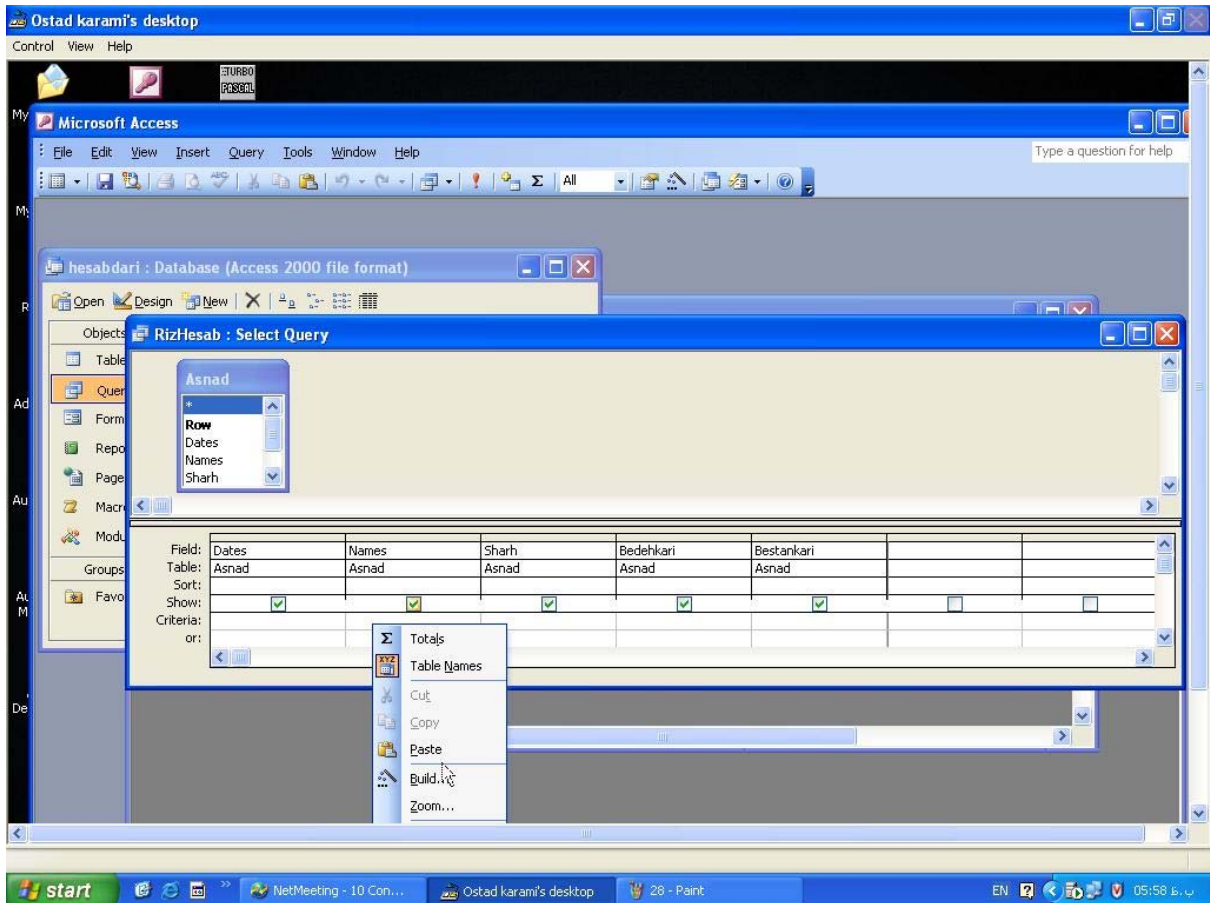
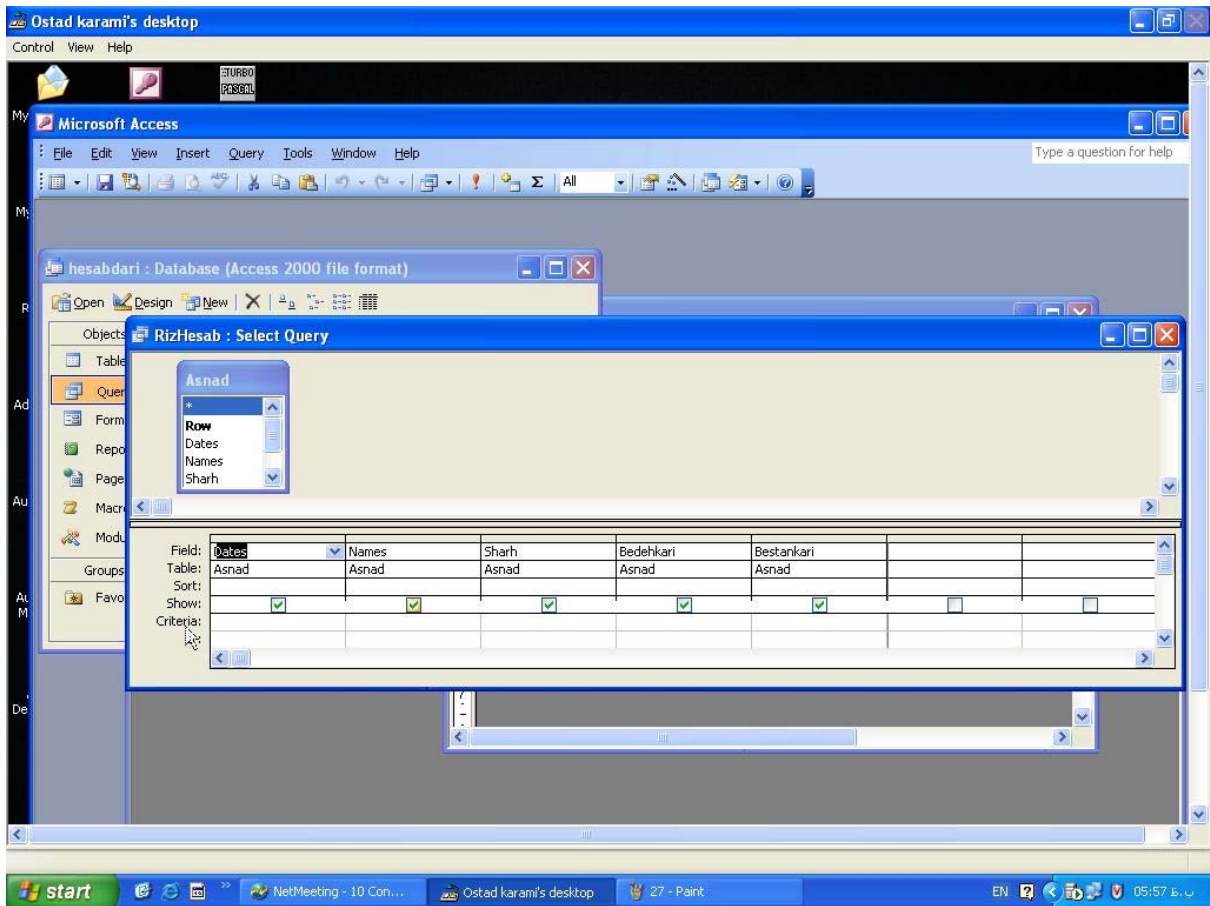


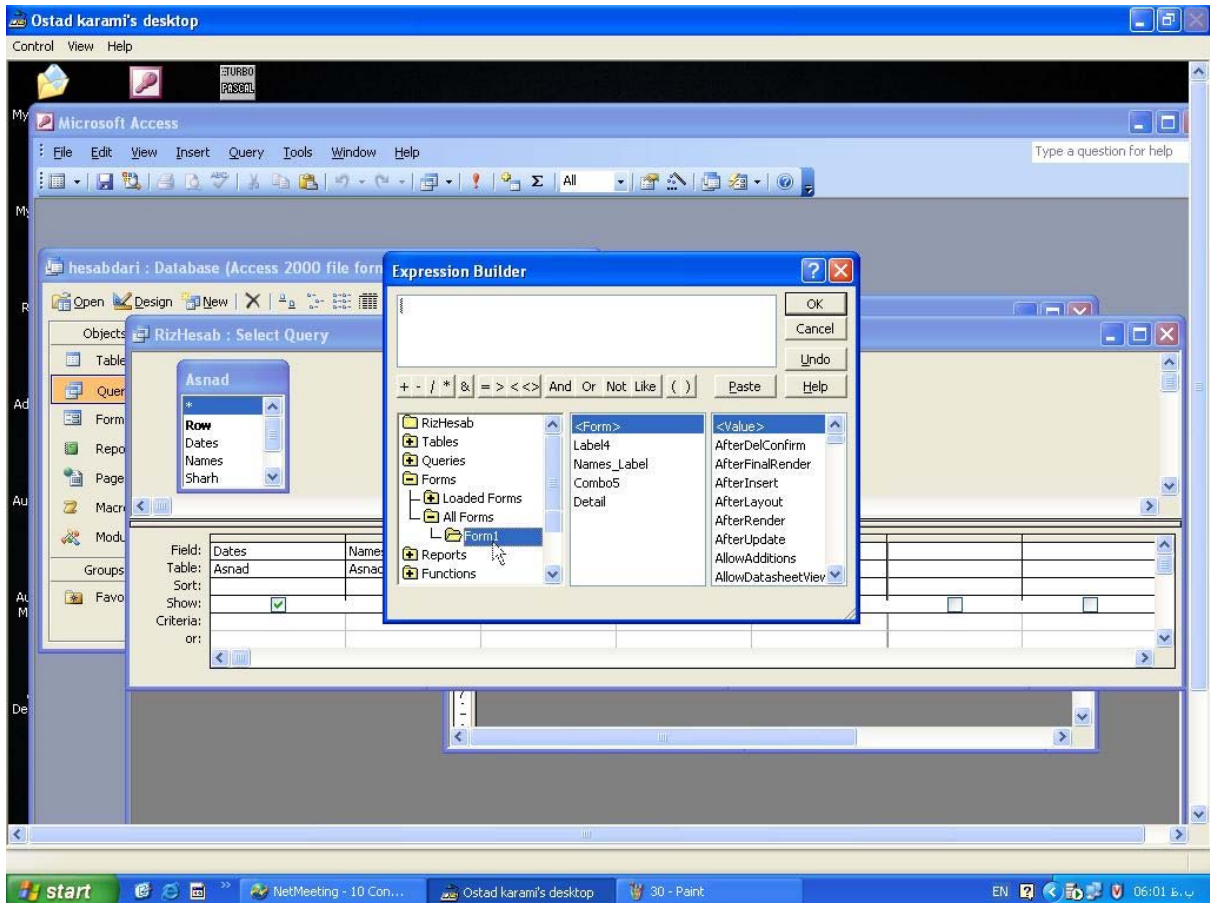
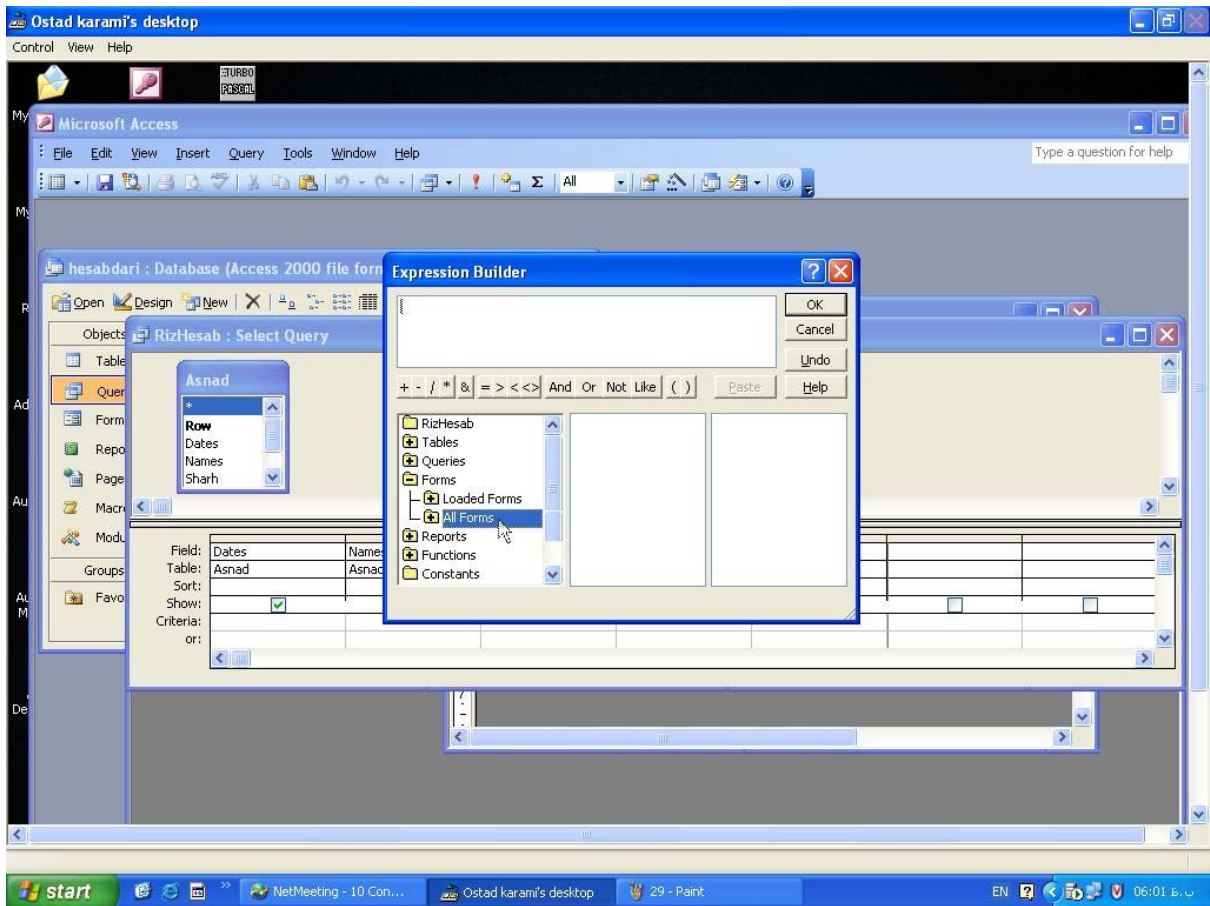




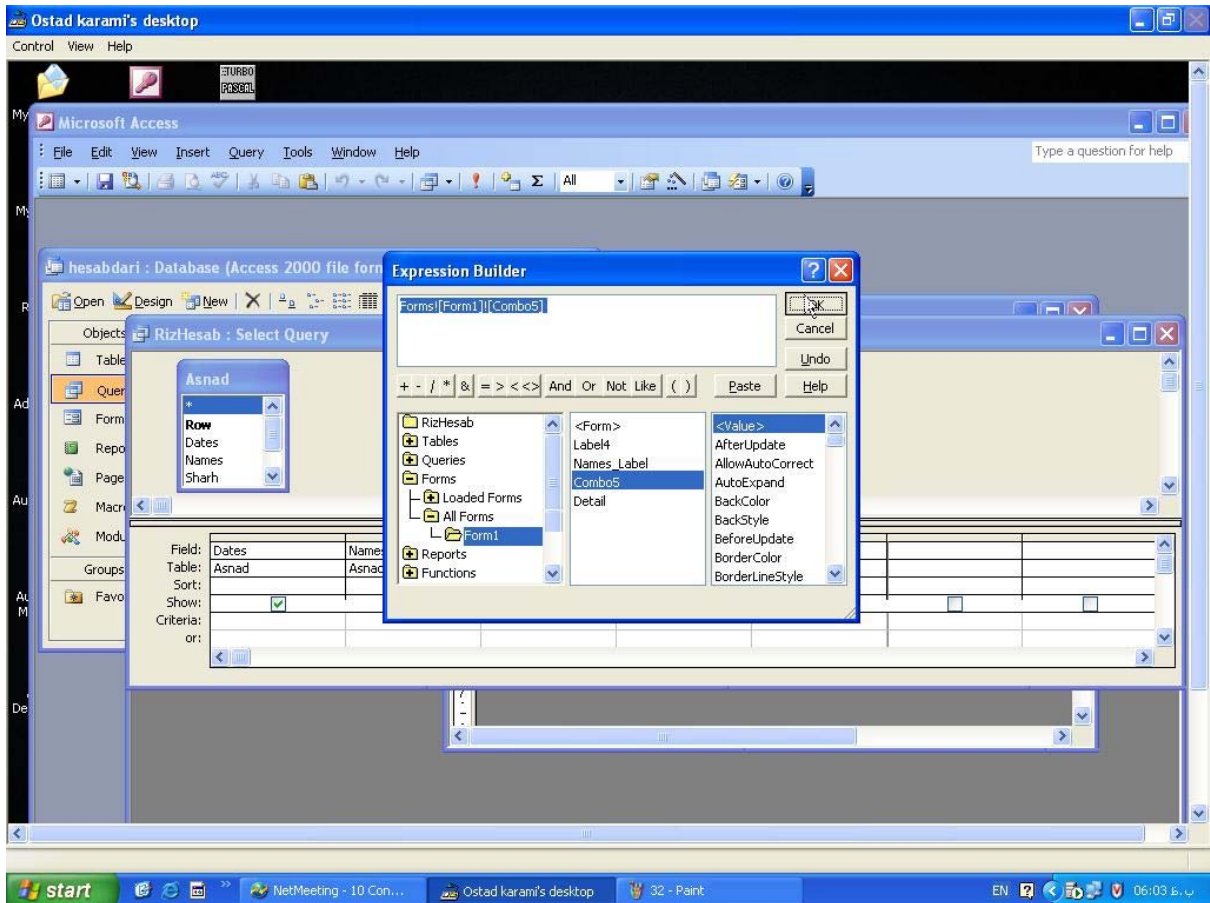
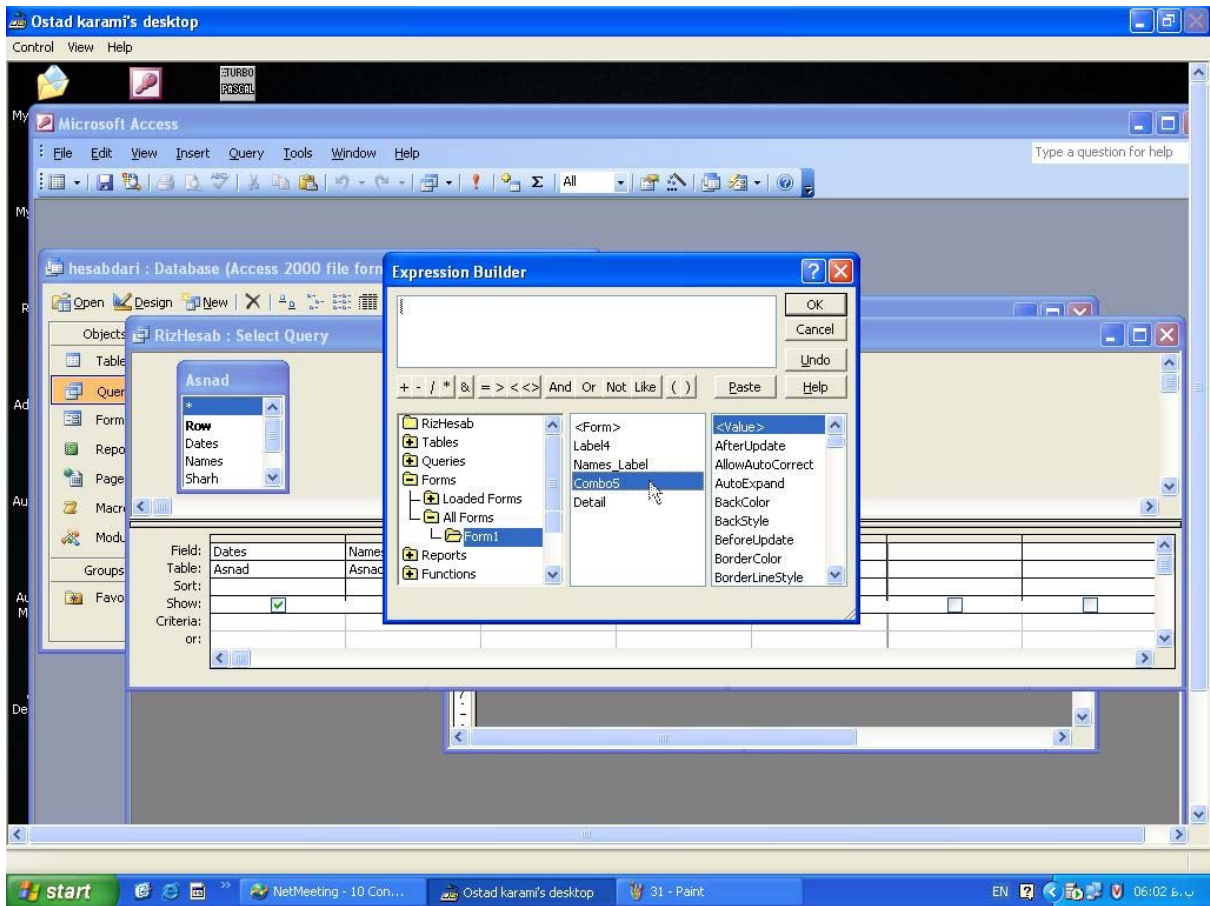


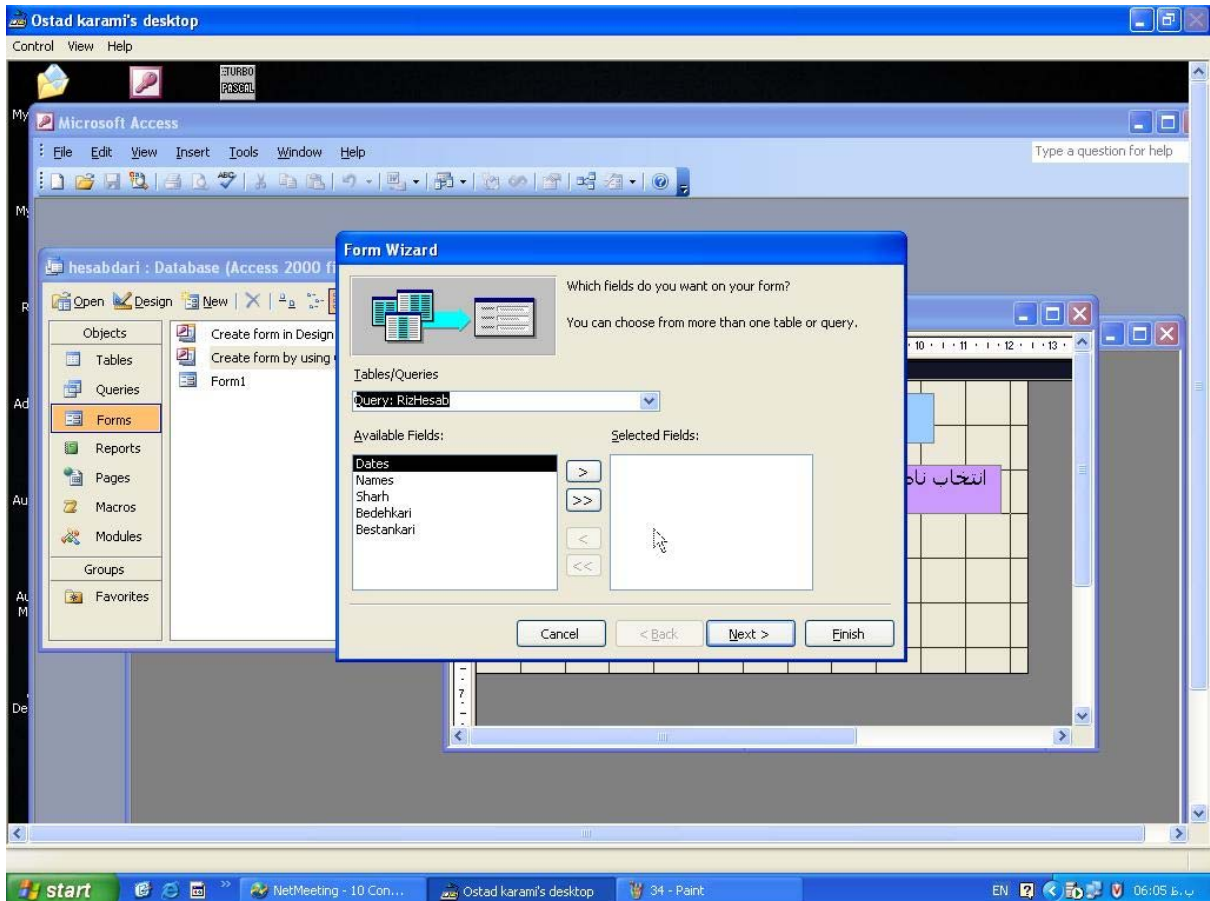
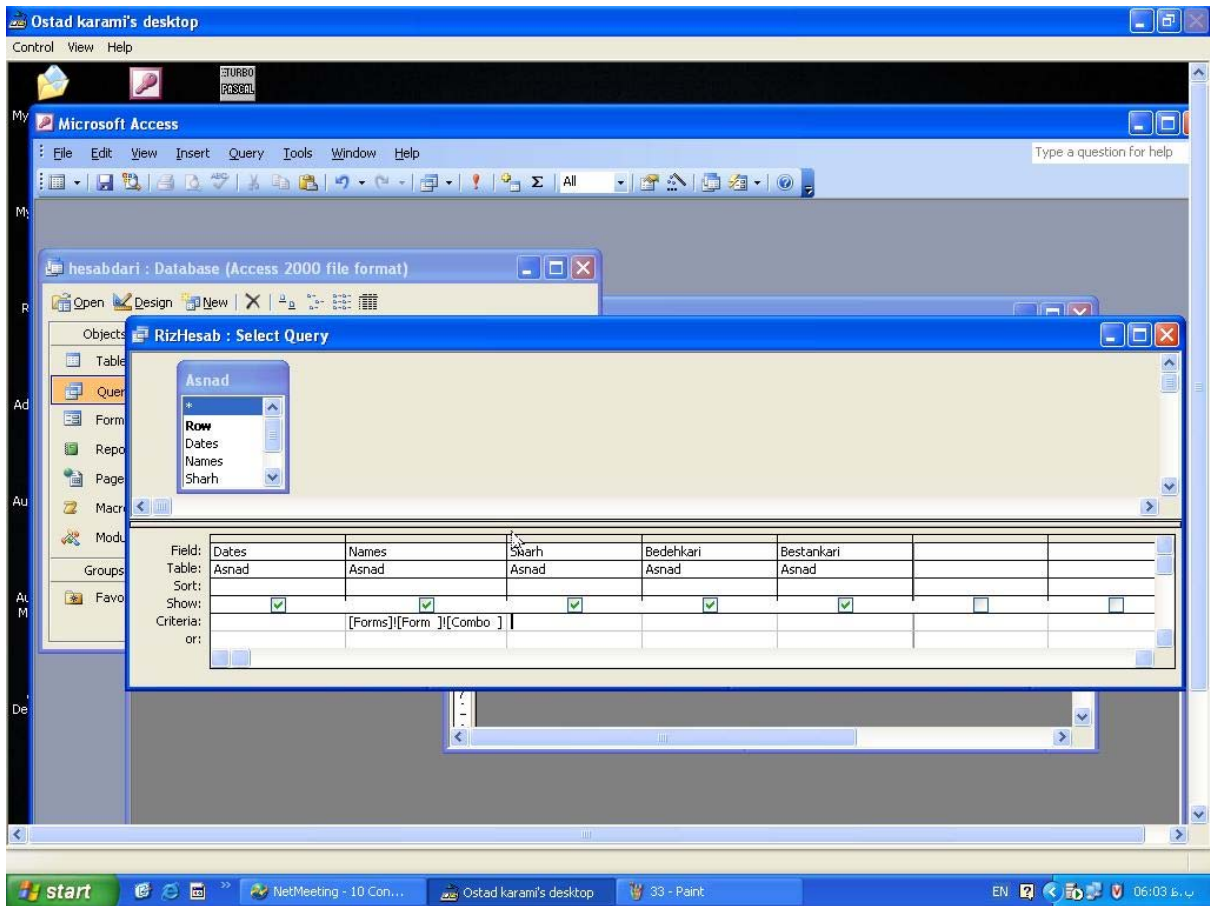


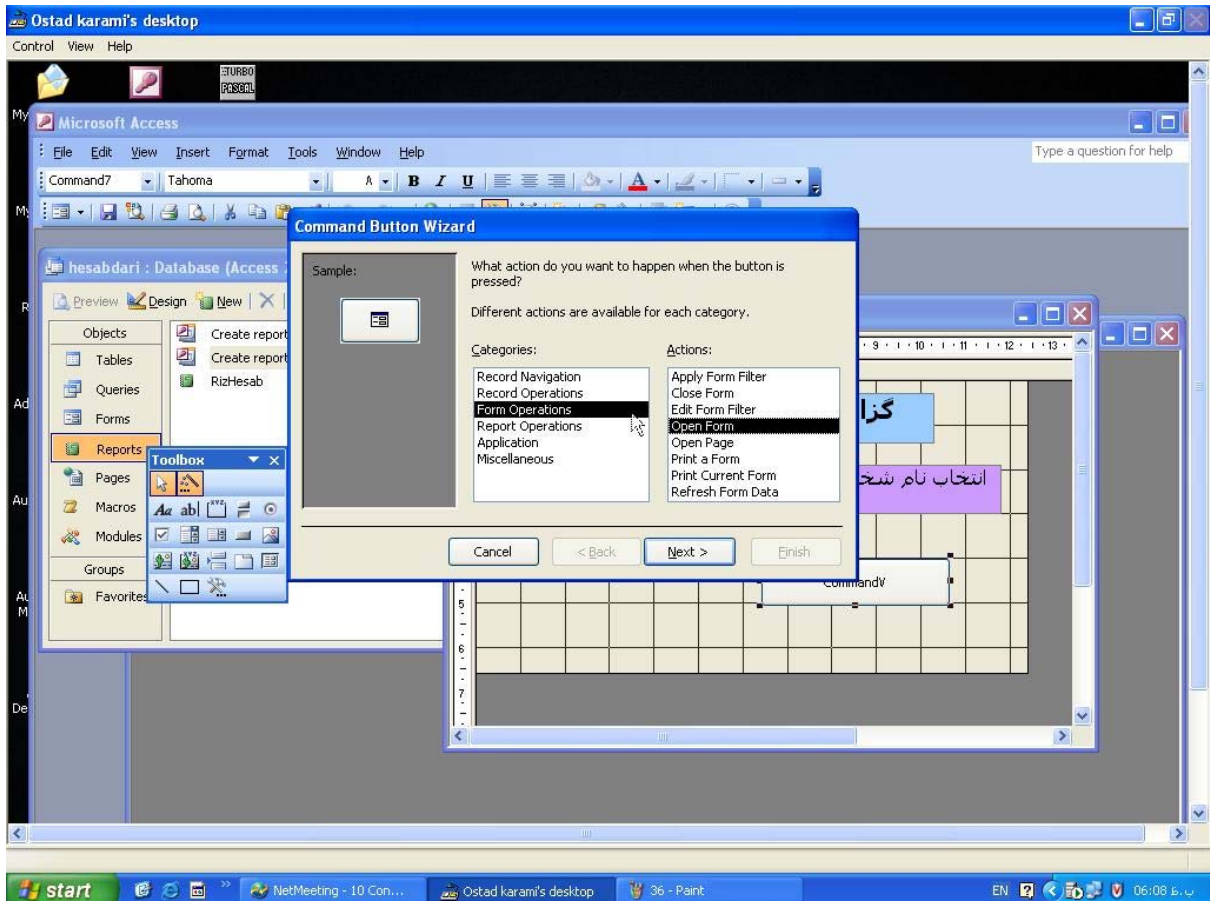
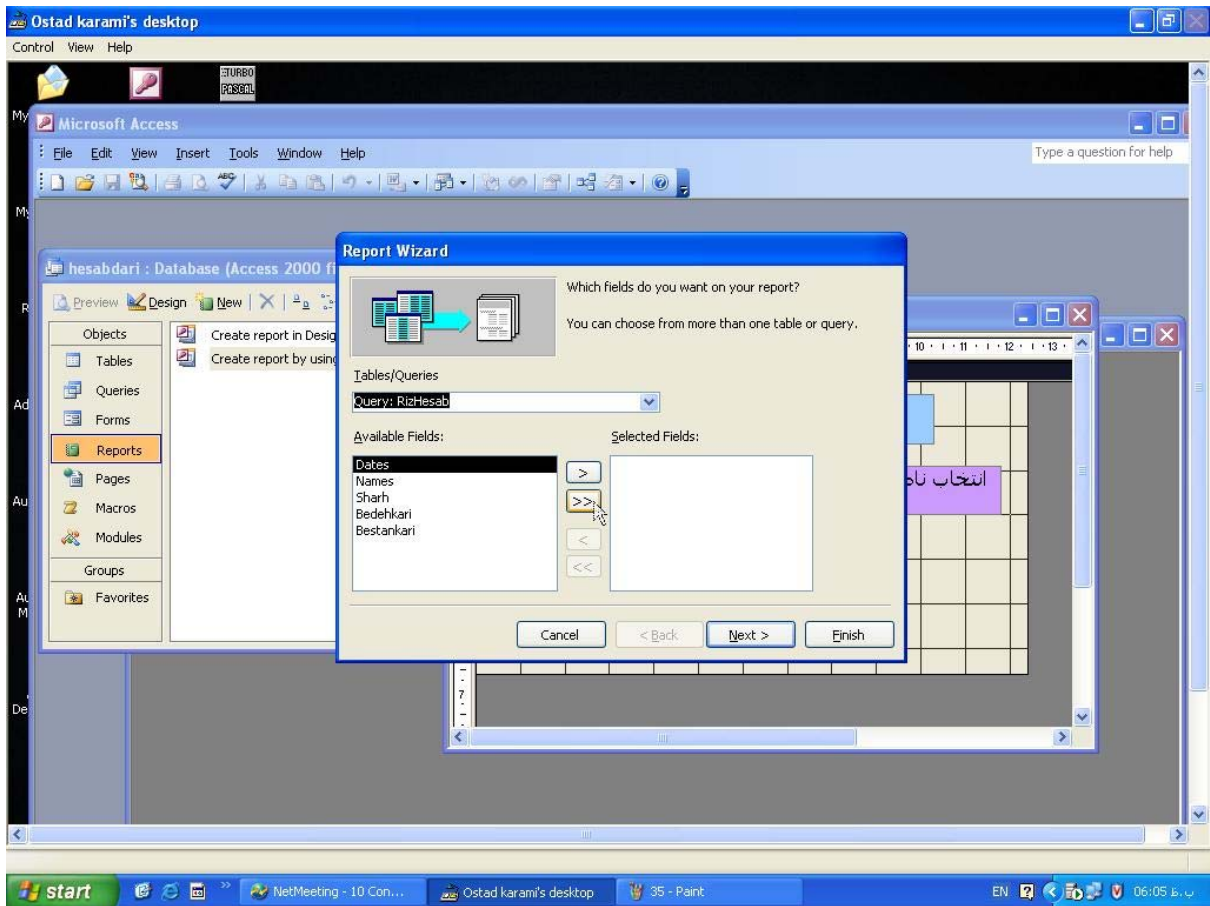


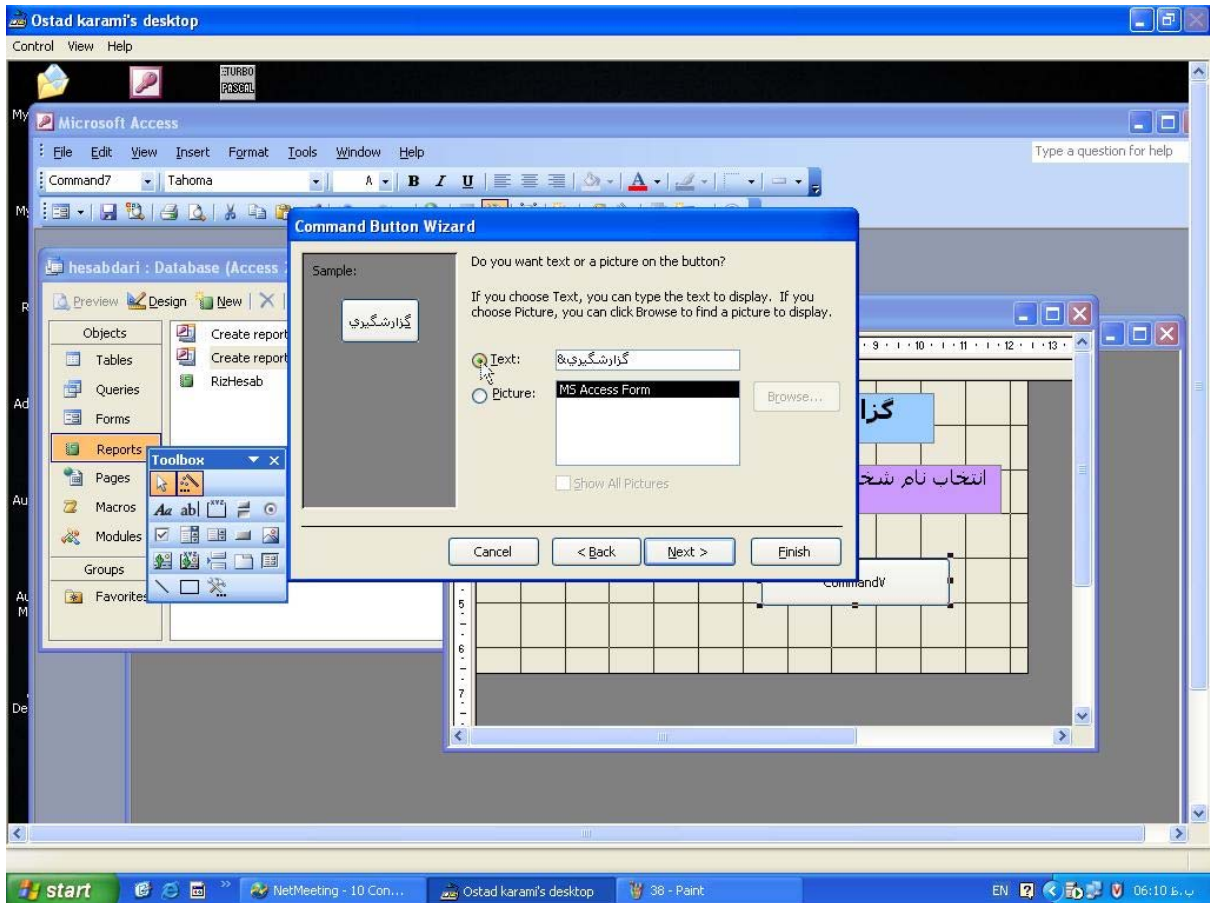
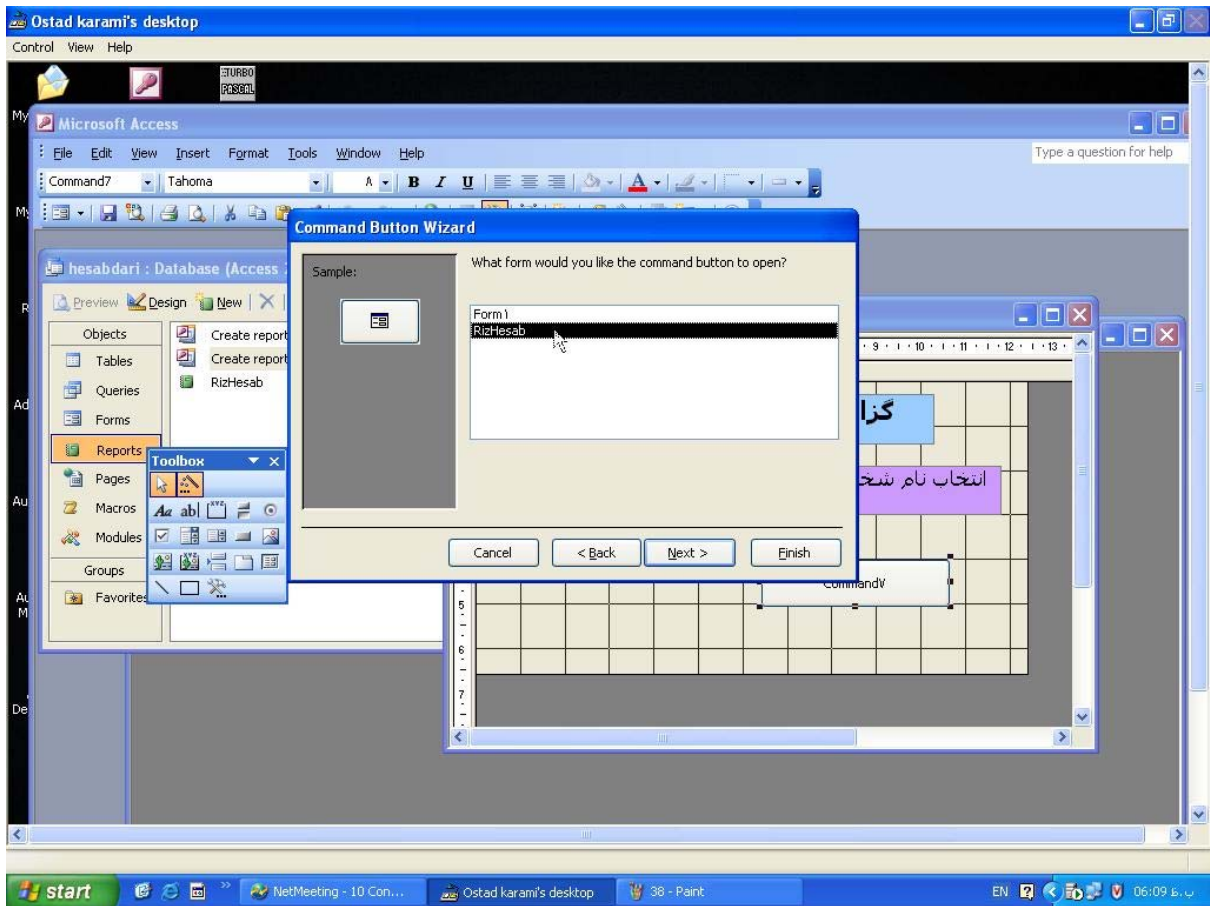


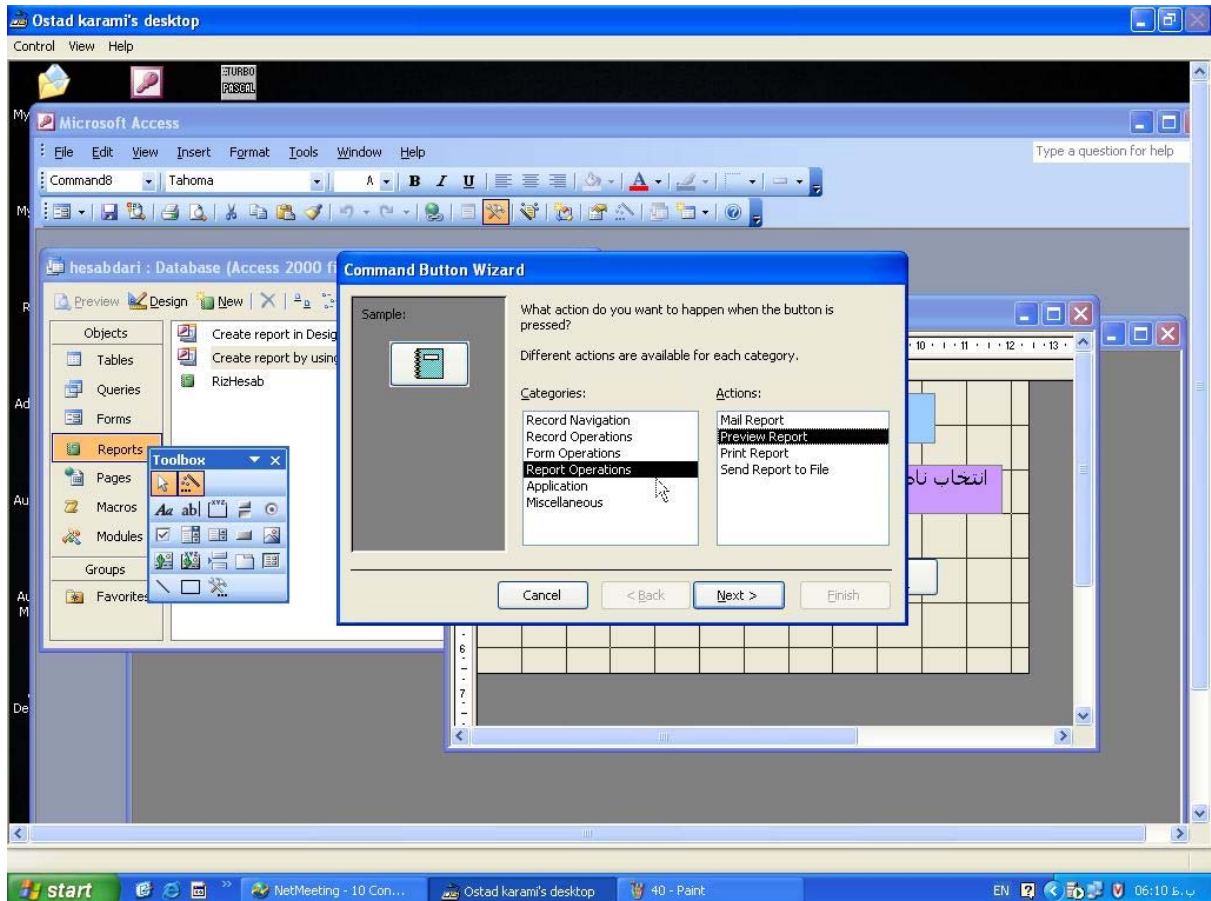
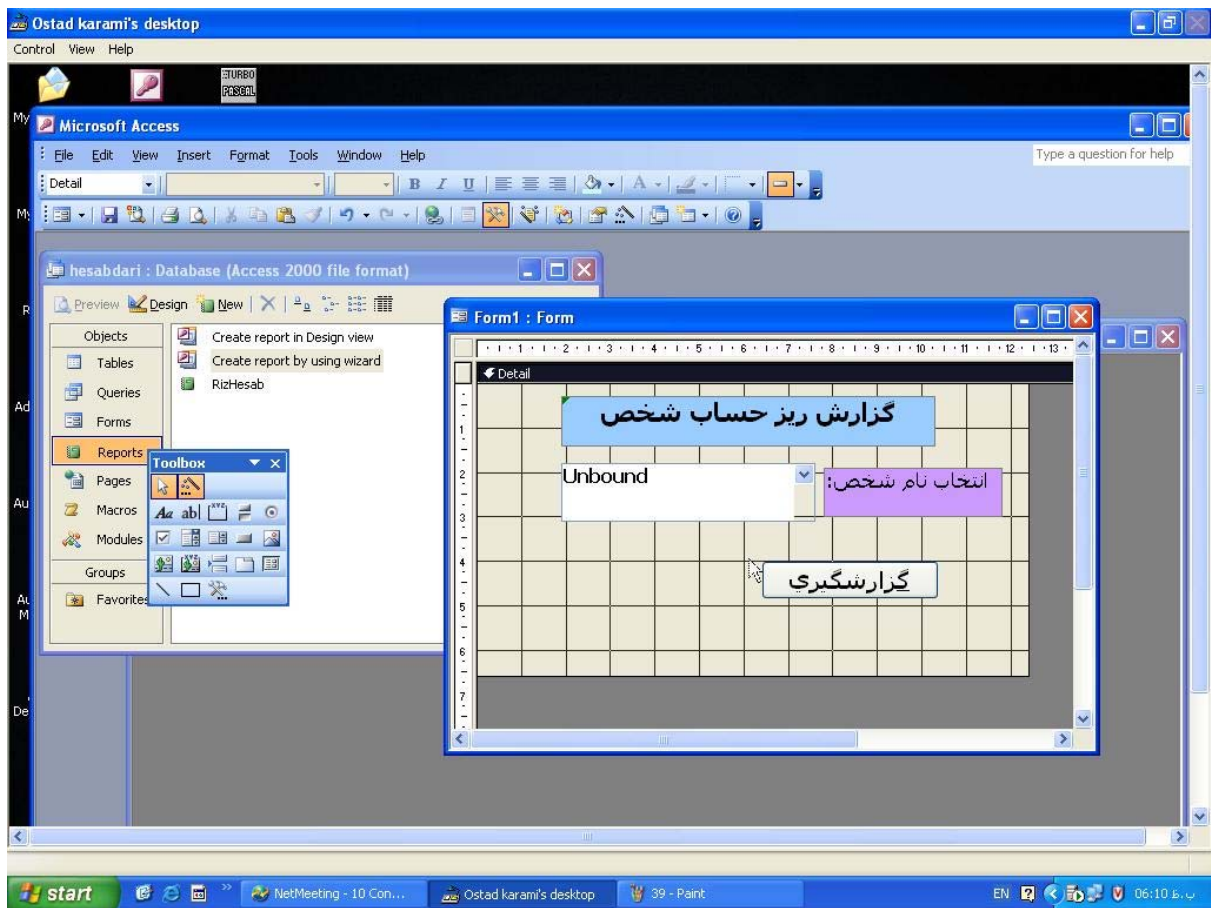


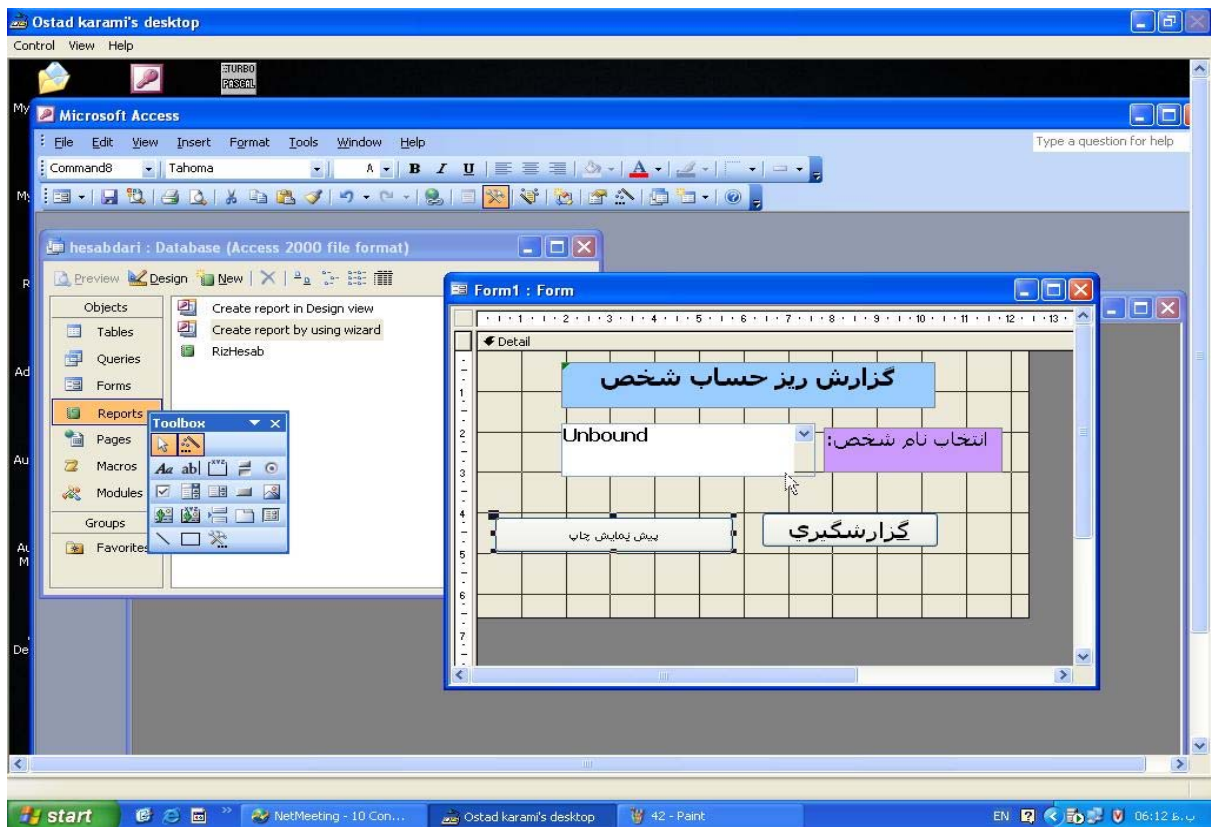
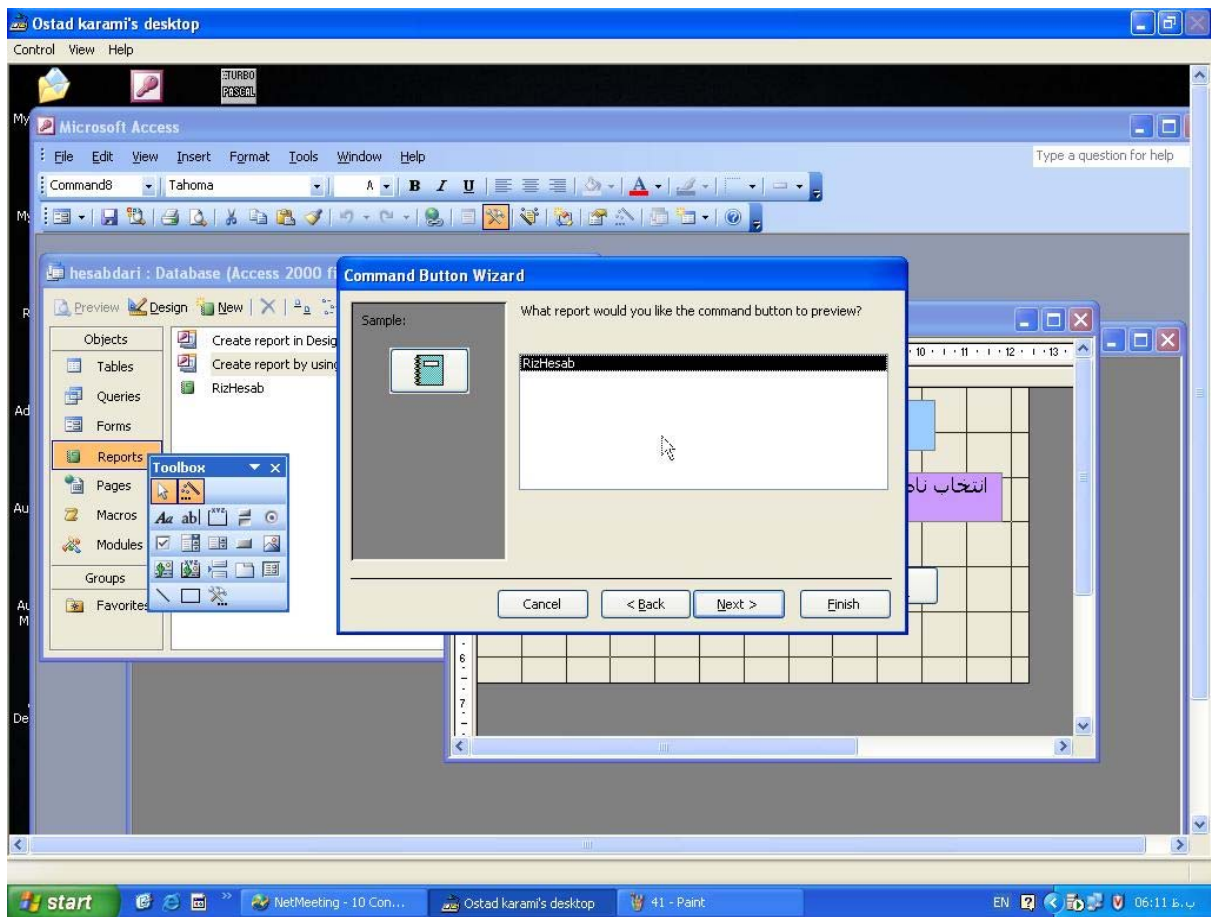






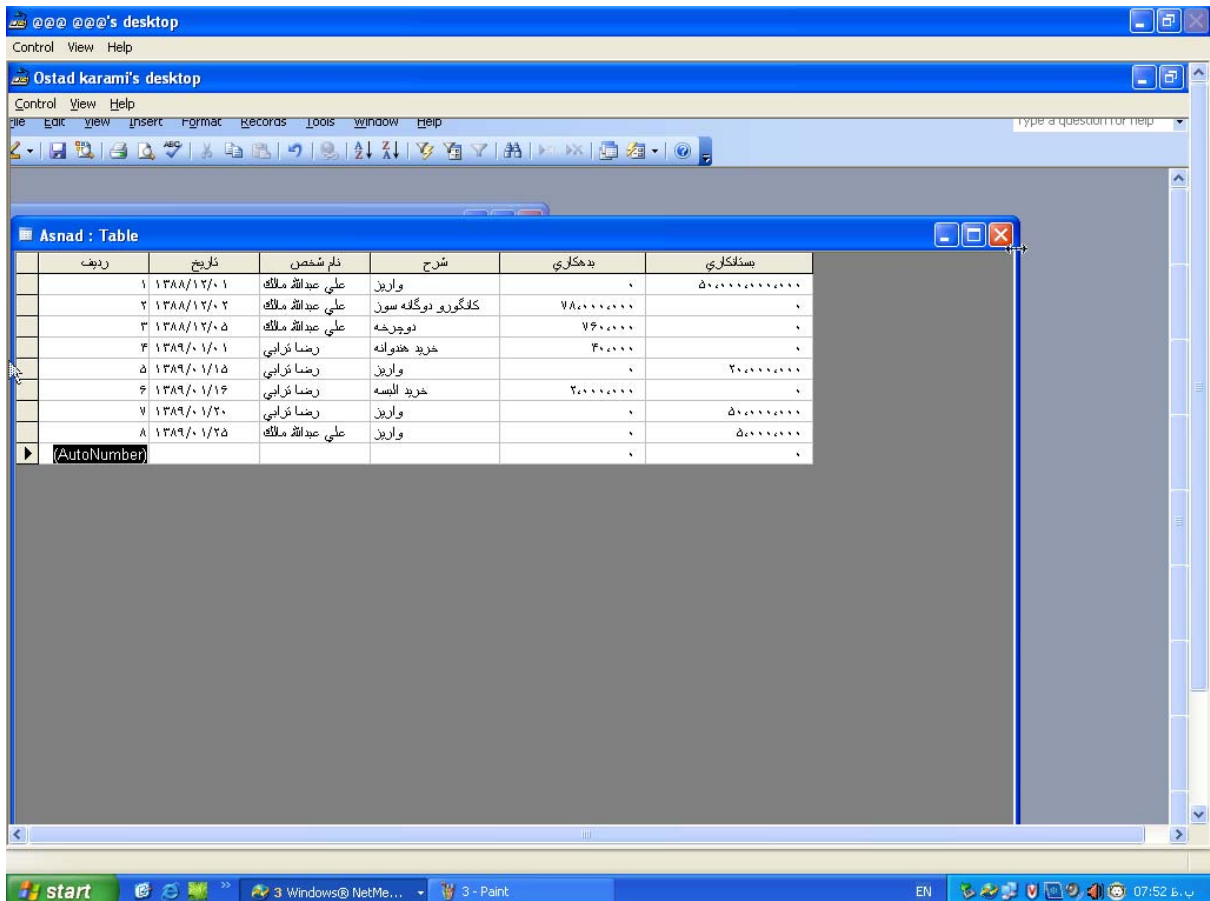
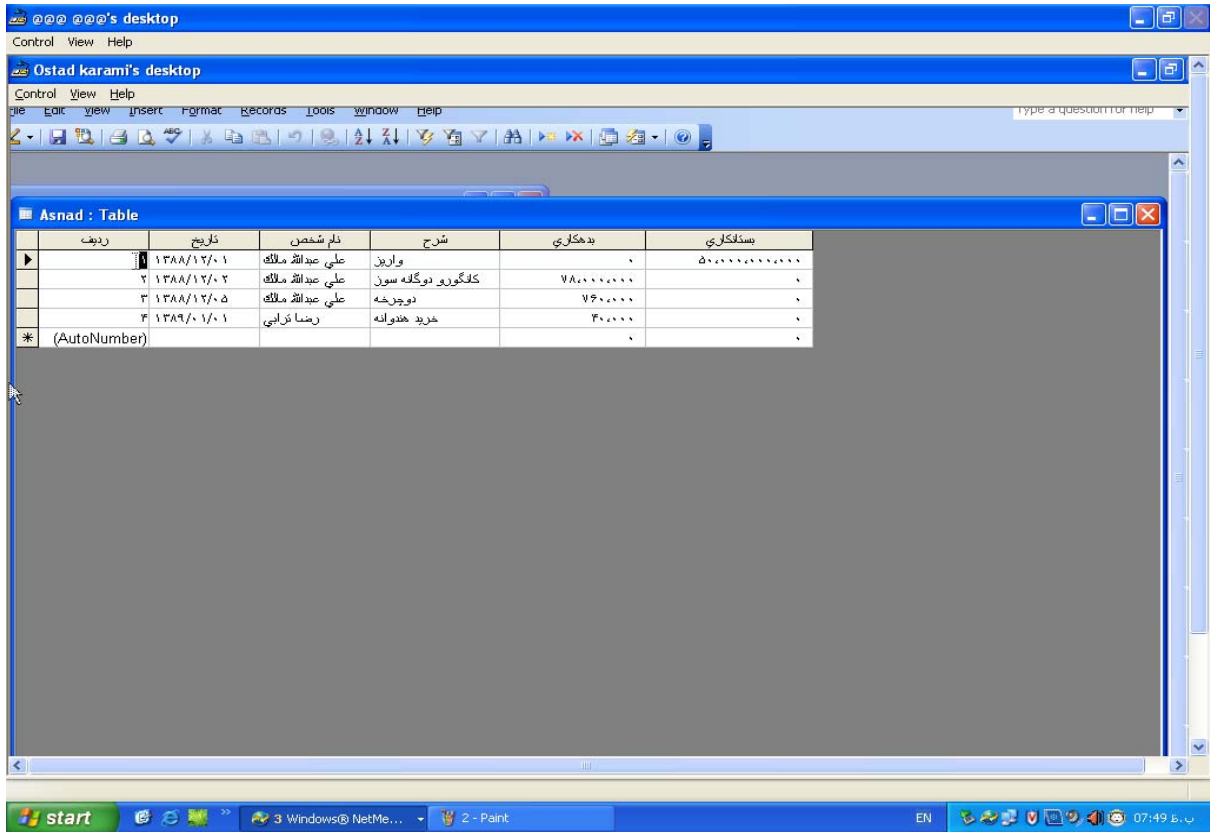


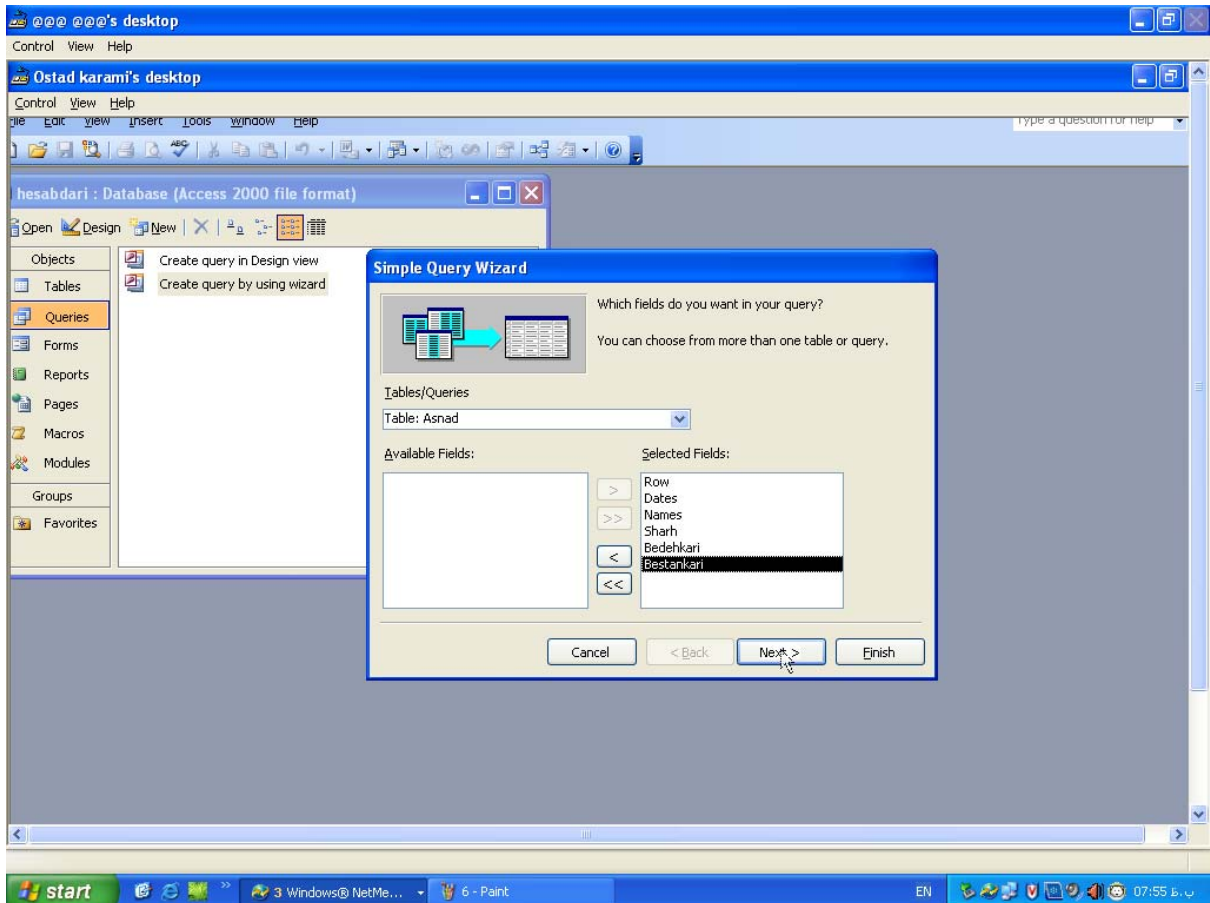
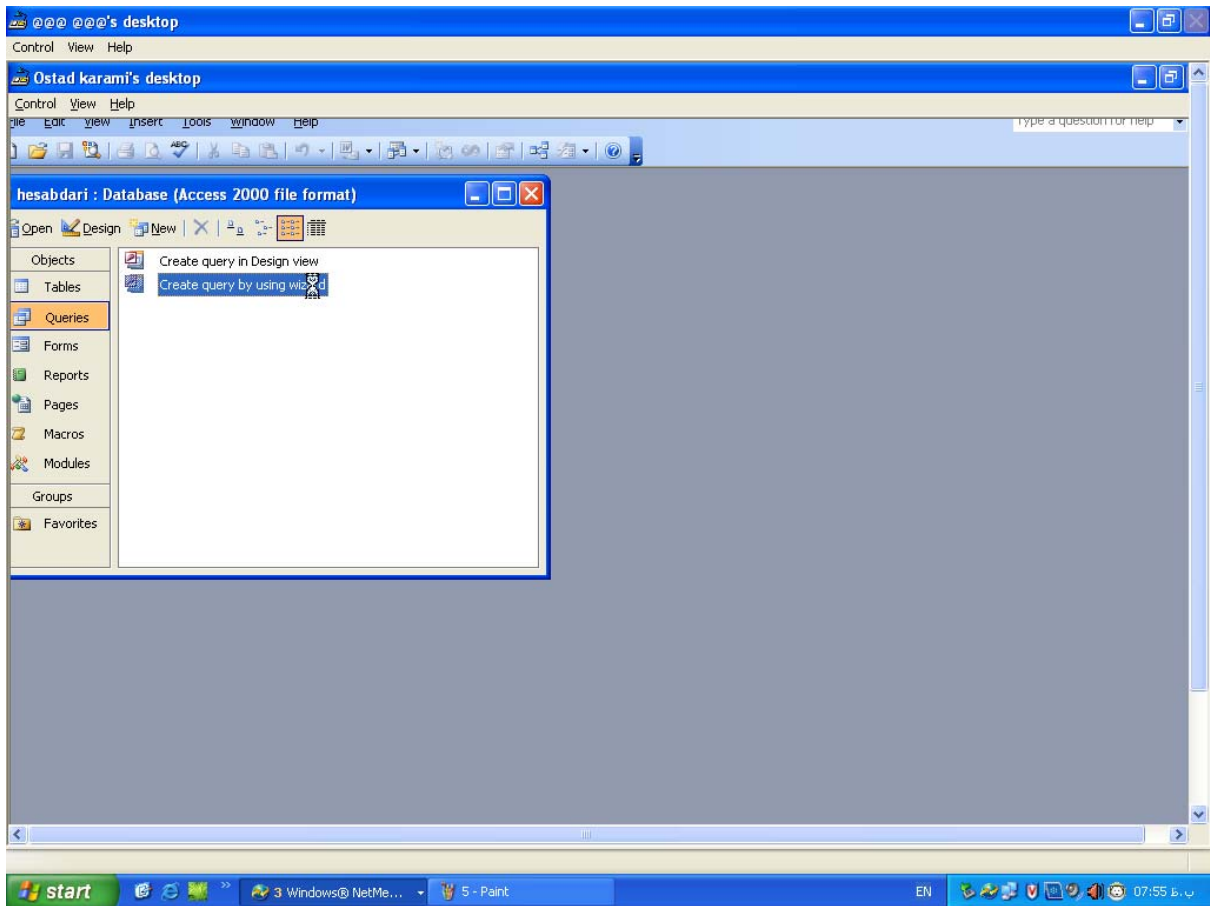




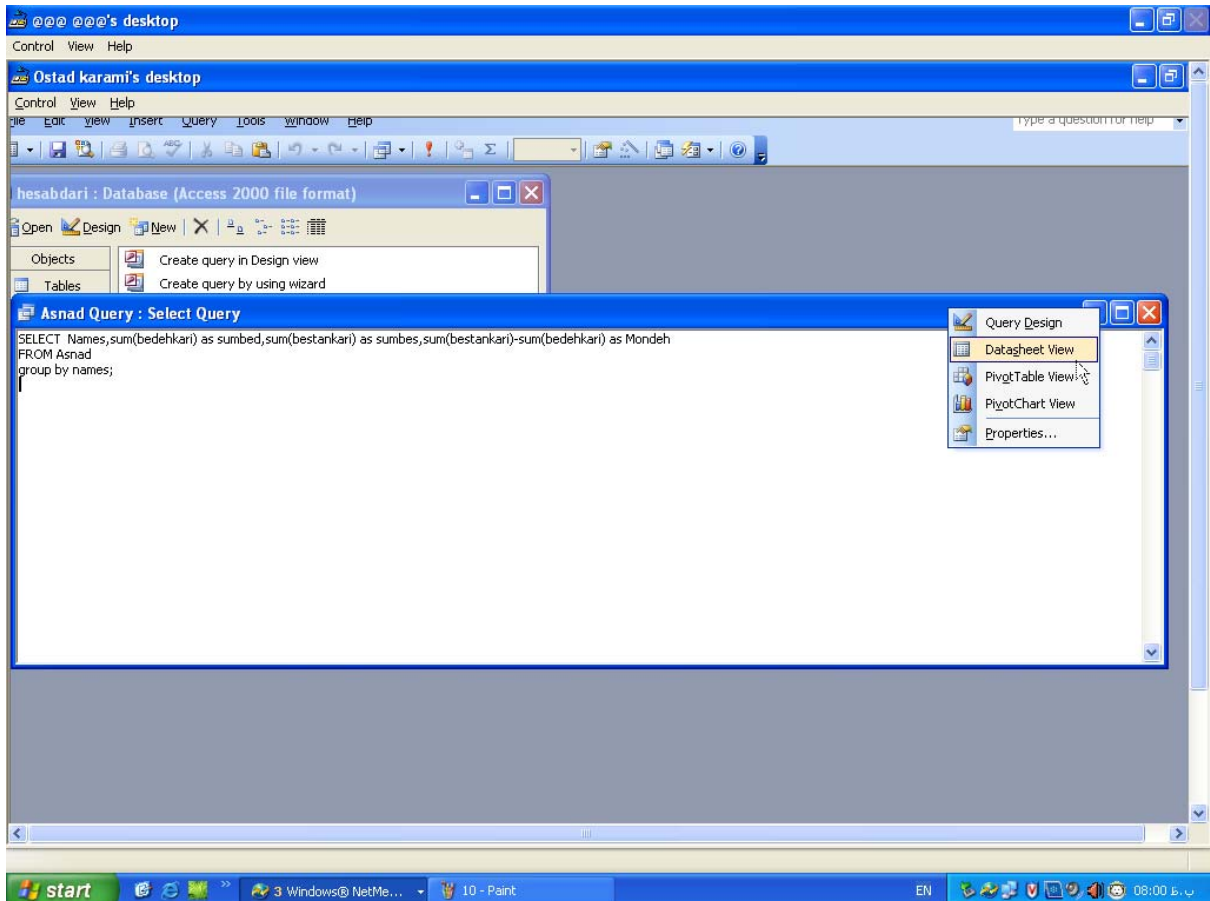
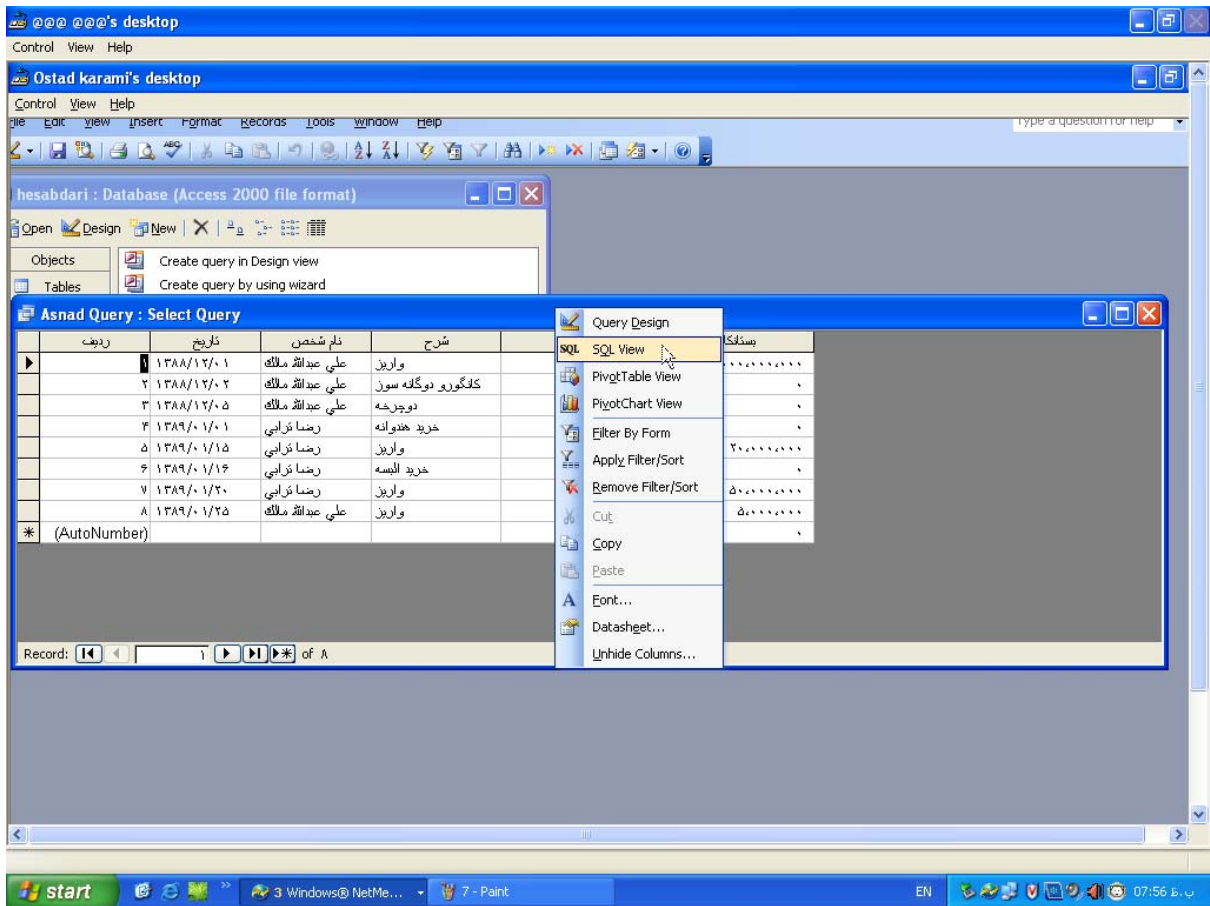
کلید گزارش گیری فرم query را بازمی کند و کیدپیش نمایش preview report از گزارشی است که از query گرفته است .

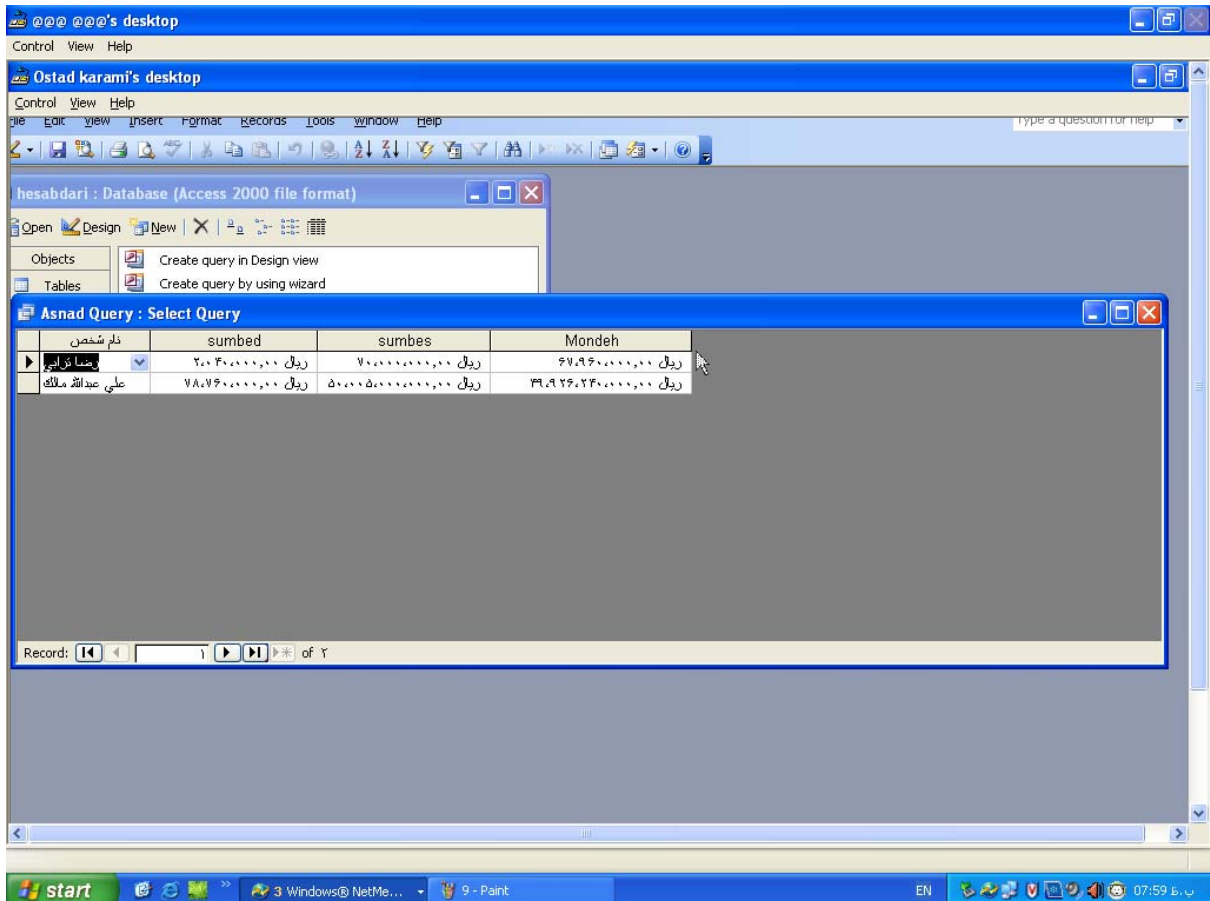
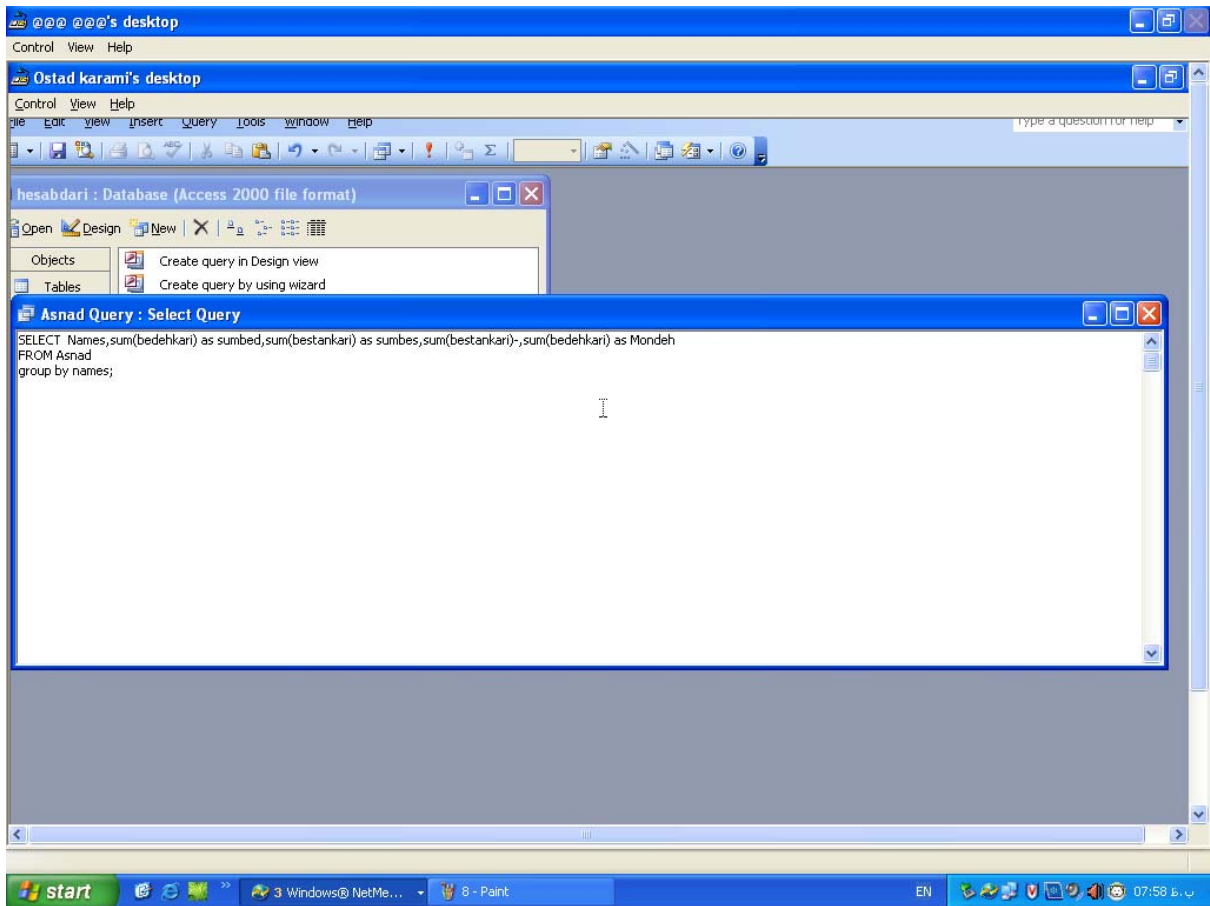
نحوه محاسبه مانده خطی :

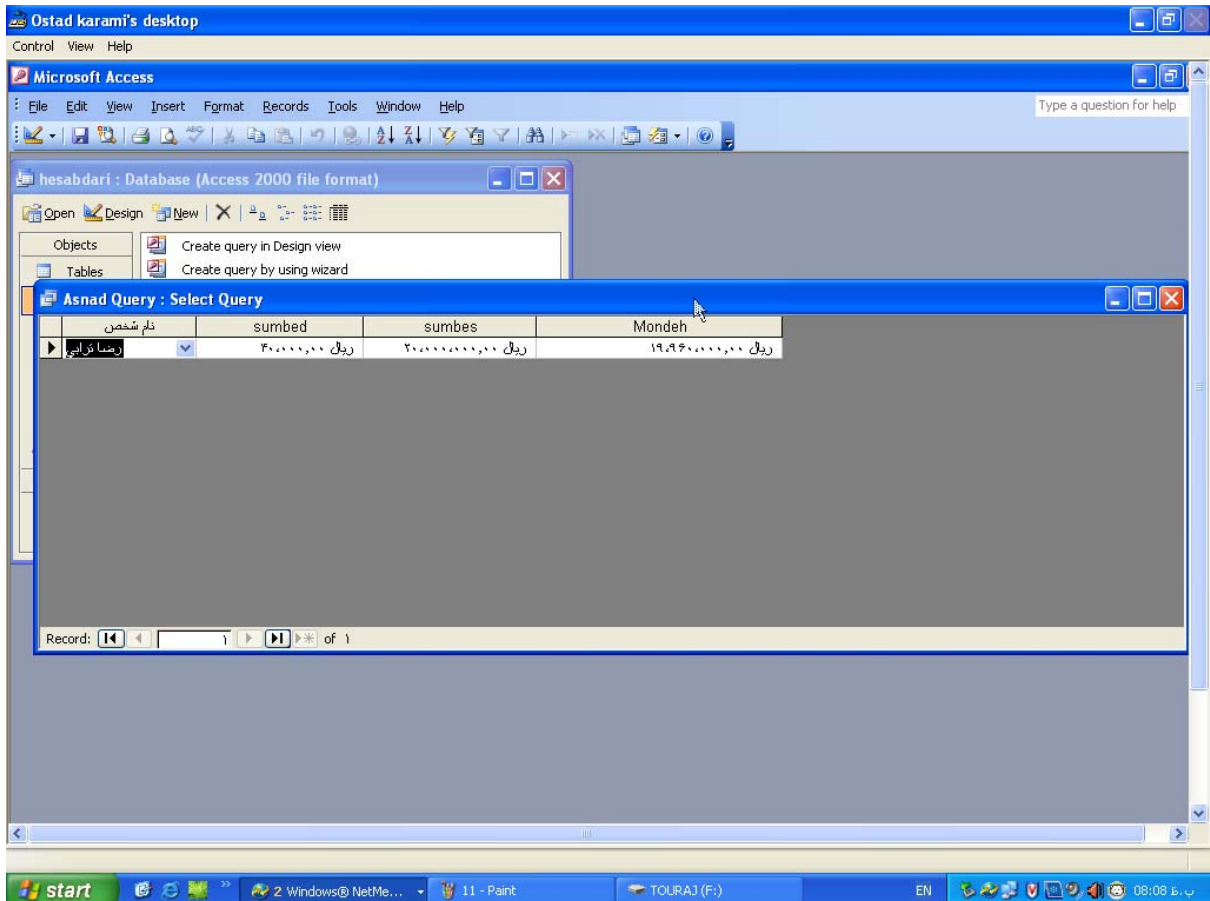
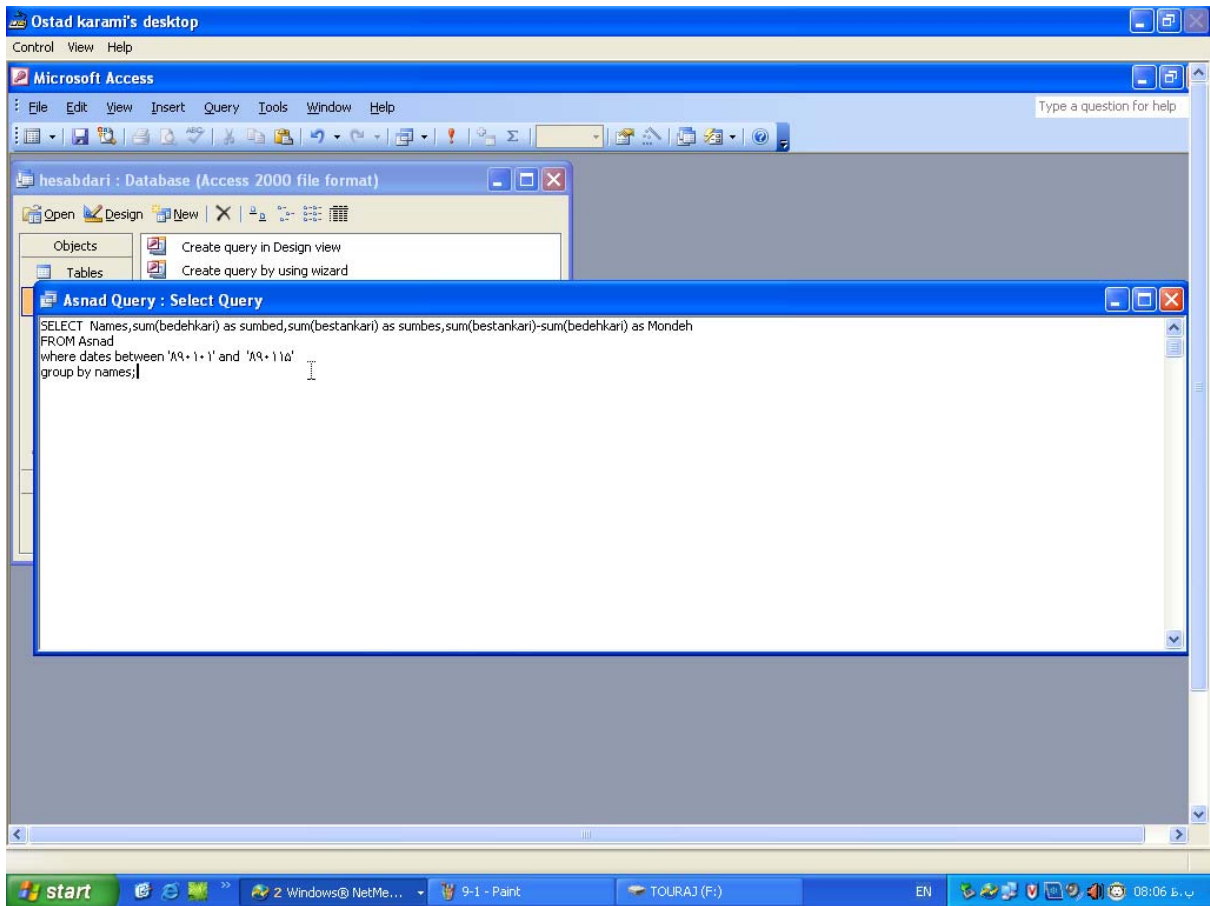






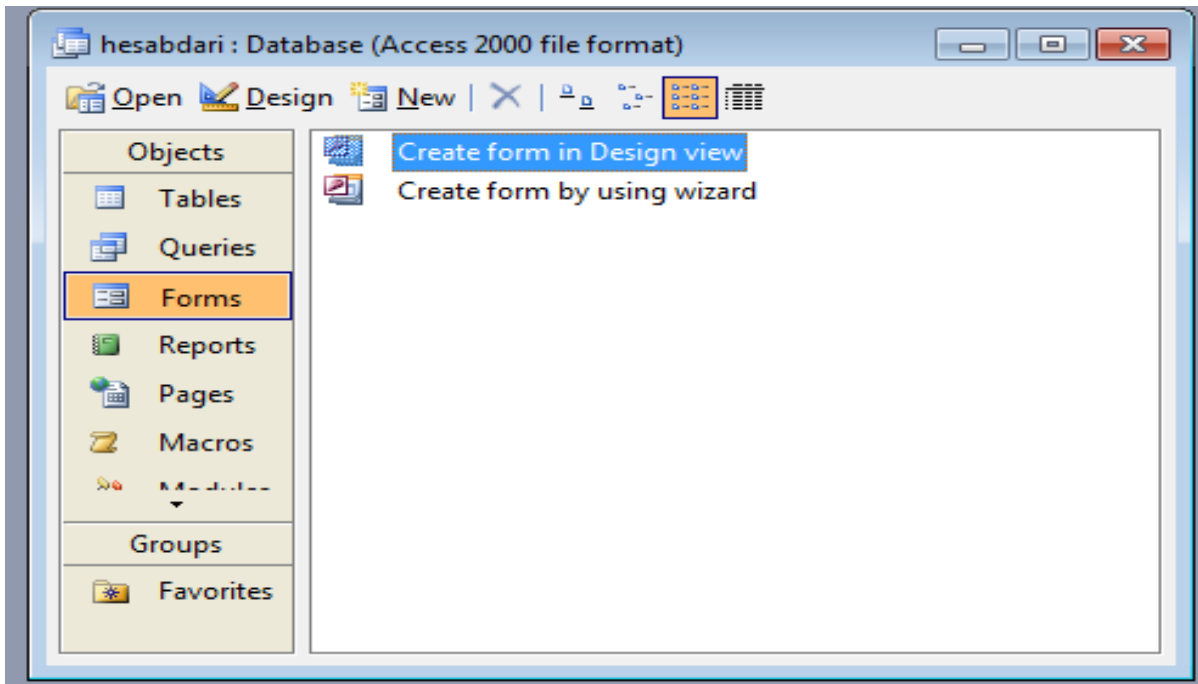




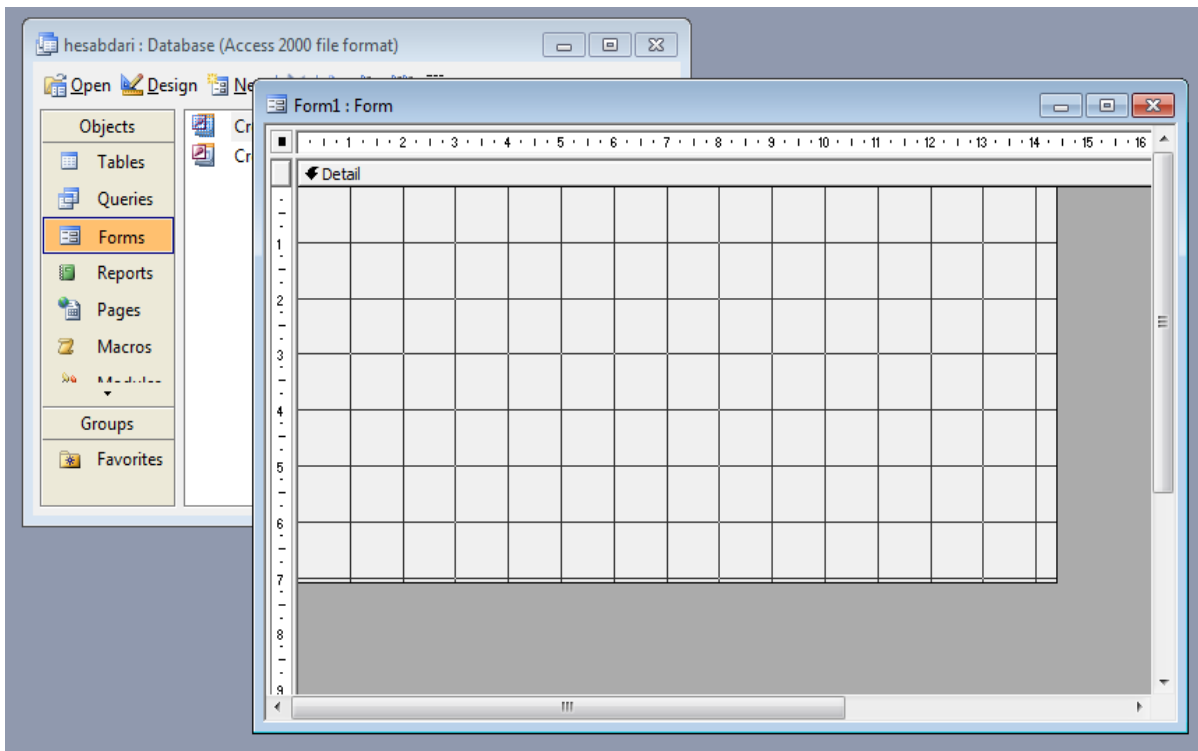


با استفاده از فایل hesabdari اقدام به ایجاد یک فرم گزارش می نمائیم.

۱. با توجه به موجود بودن جدول Asnad و Fardi نسبت به ایجاد فرم مذکور طبق مراحل ذیل عمل نماید.
۲. طبق شکل ذیل گزینه create form in design view انتخاب کرده.

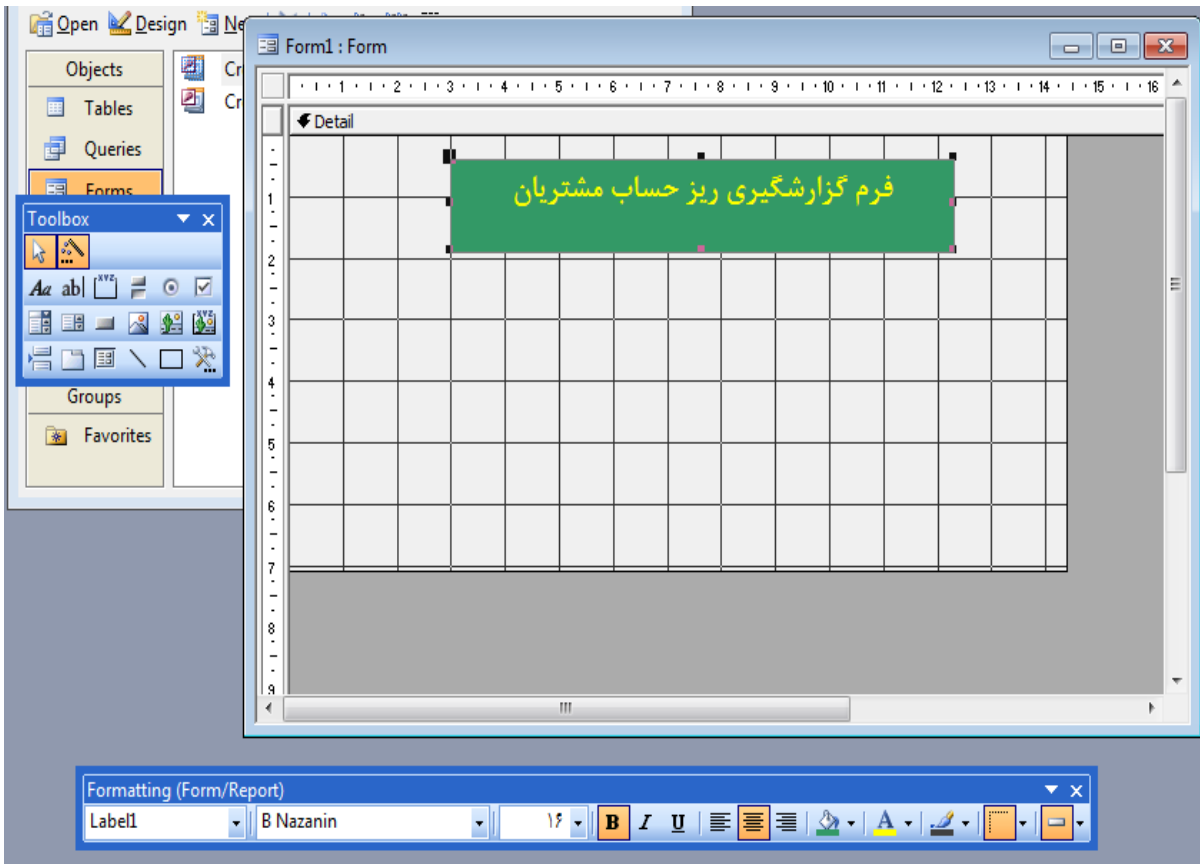
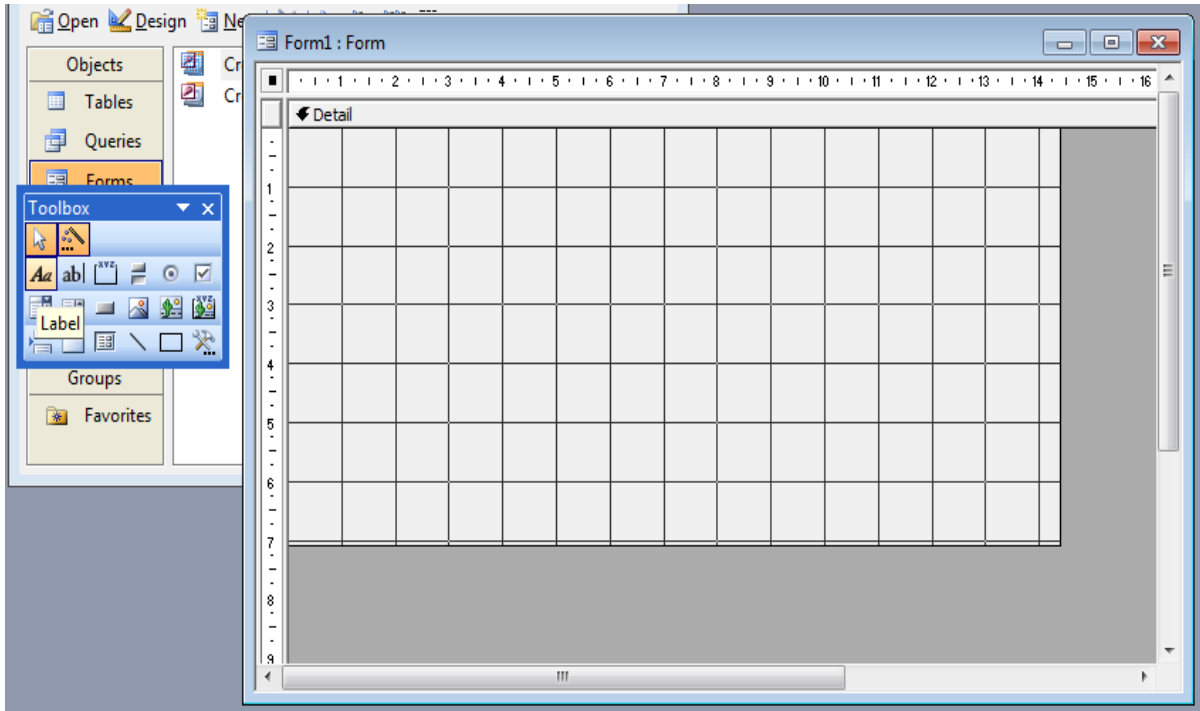


۳. پنجره طراحی فرم ظاهر می شود.



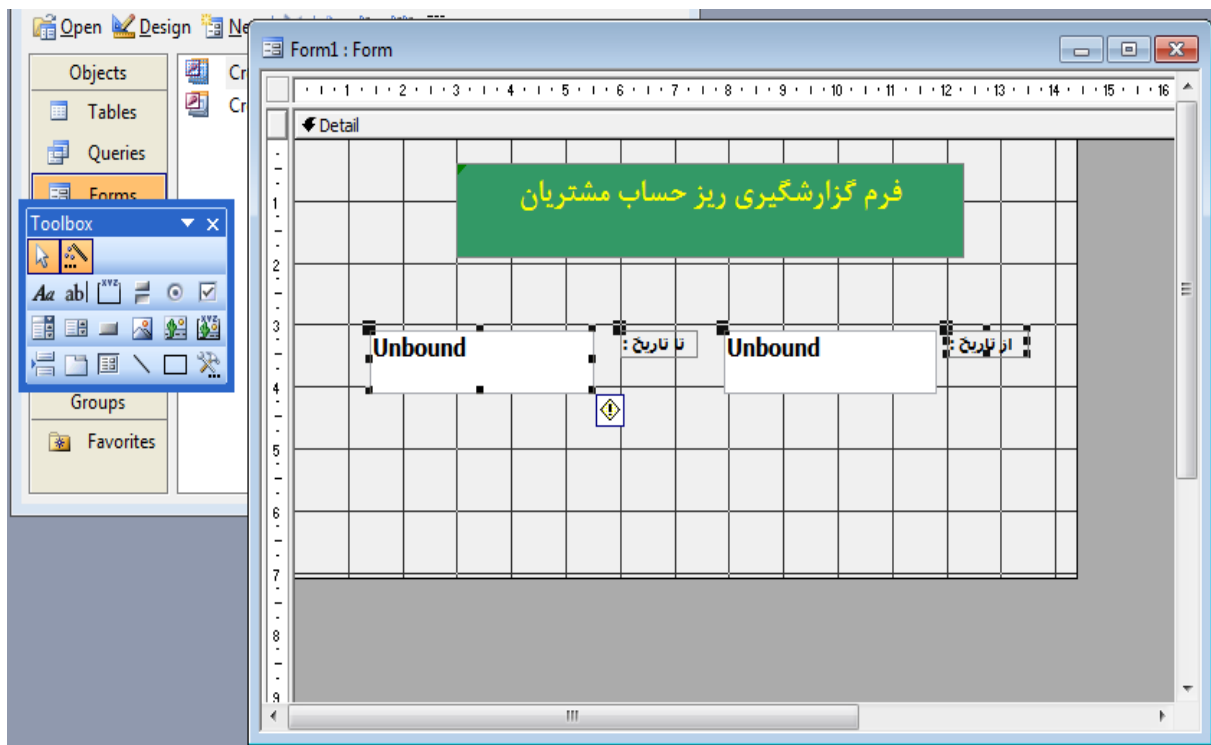
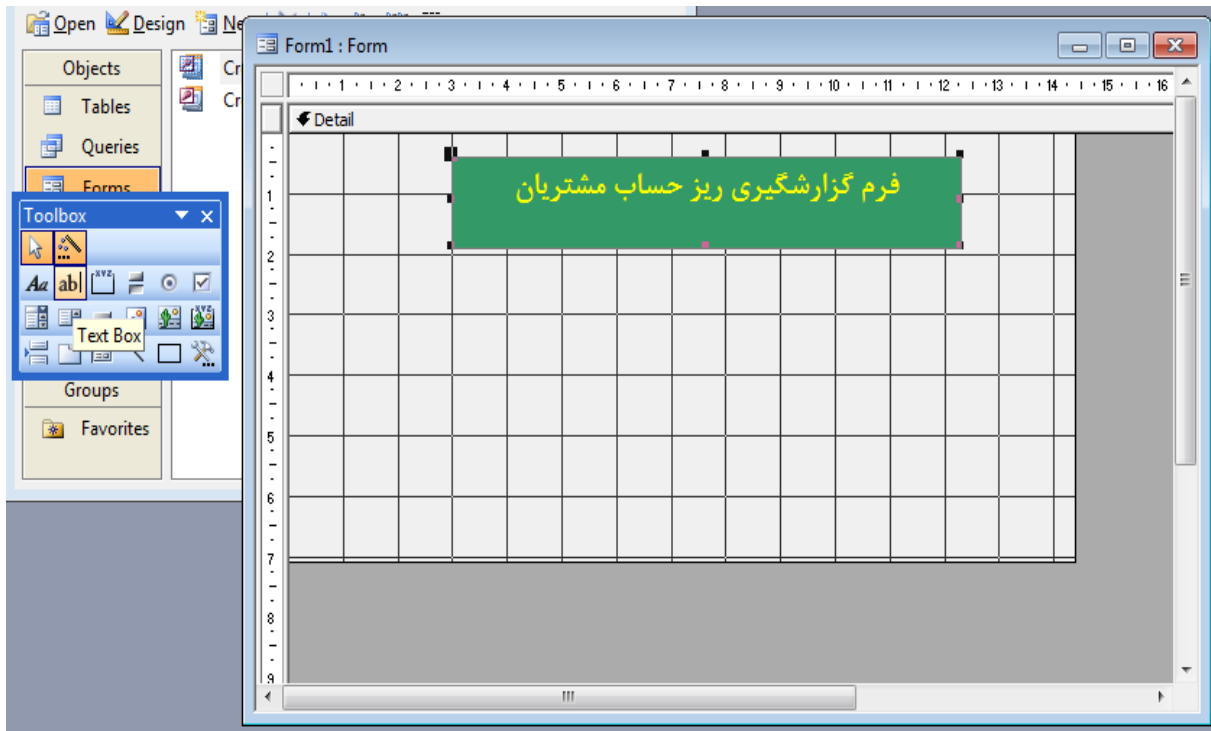
نکته : با استفاده از Toolbox می توان object های مورد نیاز را روی پنجره طراحی ایجاد نمود.

۴. ابتدا عنوان فرم را با استفاده از Object Label طراحی می کنیم.



نکته: با استفاده از object های رنگ زمینه و رنگ نوشتار می توان عنوان را زیباتر نمود.

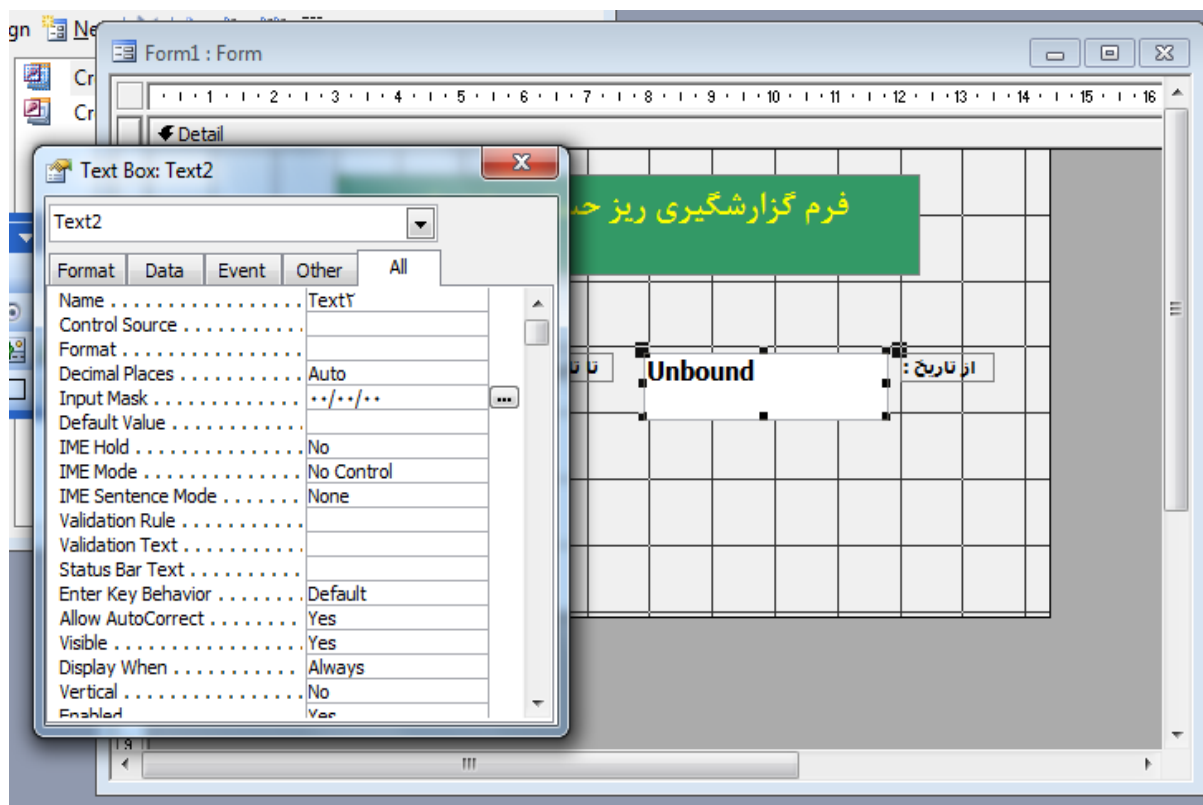
۵. جهت درج تاریخ در فرم گزارشگیری از Object text box استفاده می کنیم.



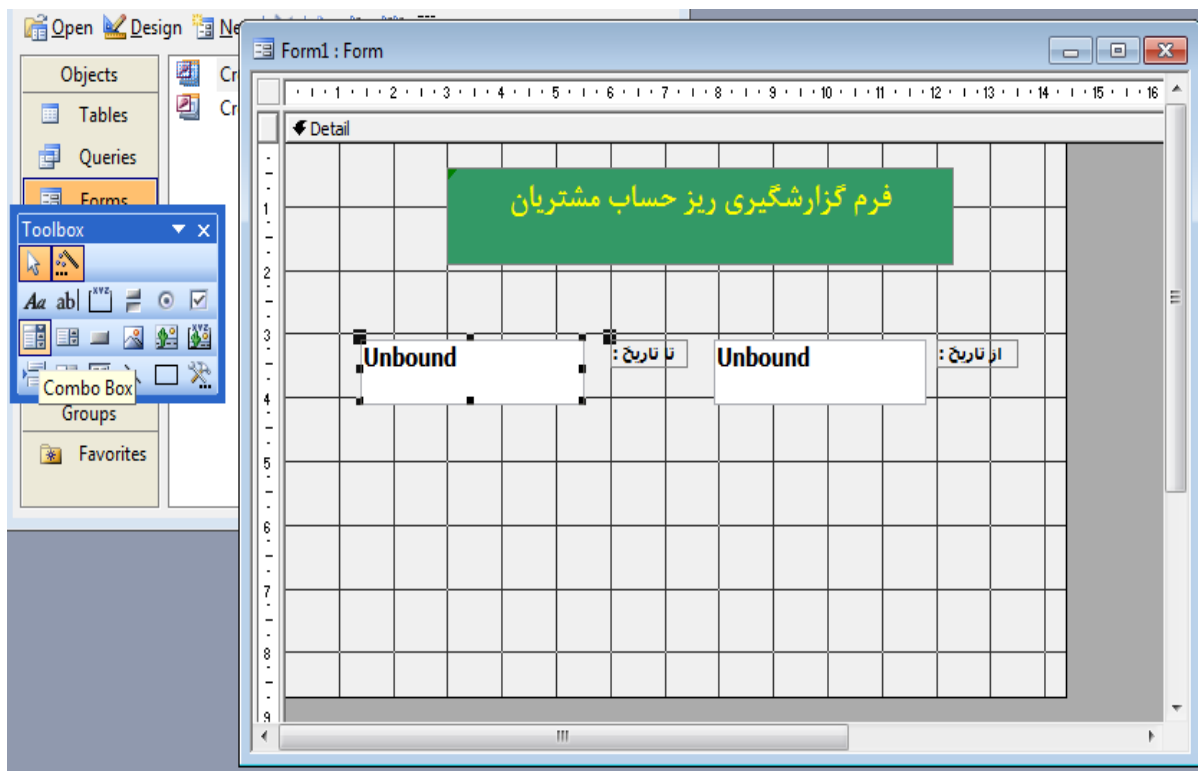
نکته ۱: با استفاده از گره های بزرگتر در گوشه سمت چپ box ها می توان هر کدام از آنها را جابجا نمود.

نکته ۲: با استفاده از خواص (Properties) ، Object ها می توان فرمت درج مقادیر در آنها را تغییر داد. ( مثل : در تاریخ به شکل - ۰۰/۰۰/۰۰ در قسمت Input Mask )

۶. جهت تغییر دادن فرمت مقادیر Object ابتدا روی Object کلیک سمت راست نموده و گزینه آخر Properties را انتخاب نماید و در قسمت Input Mask فرمت مورد نظر را درج نماید.

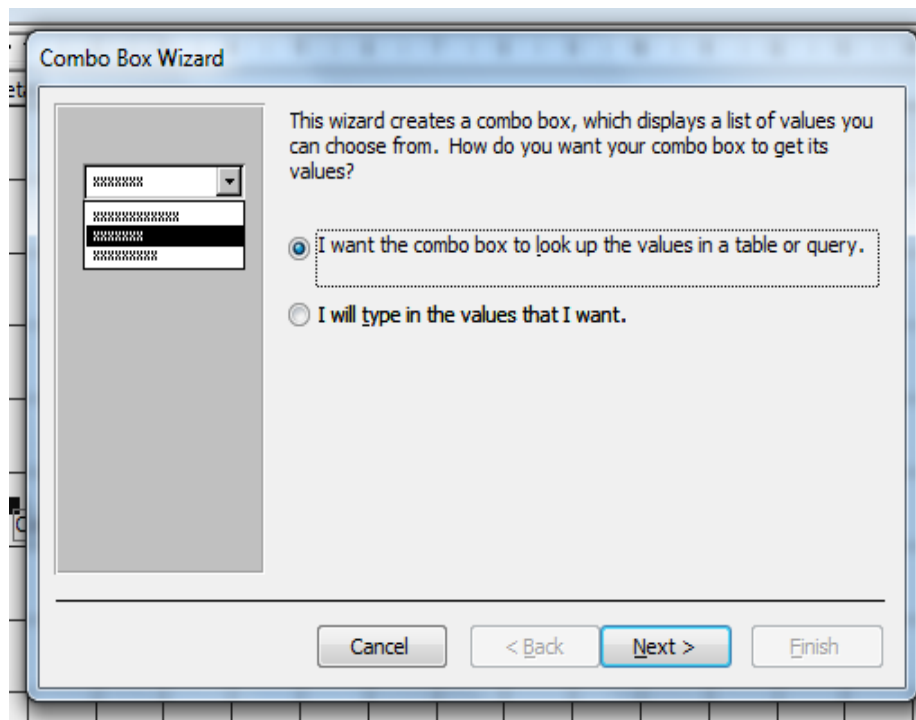


۷. چون تعداد افراد در گزارش زیبا و مختلف می باشند می توان برای انتخاب هر یک از Object combo box استفاده نمود

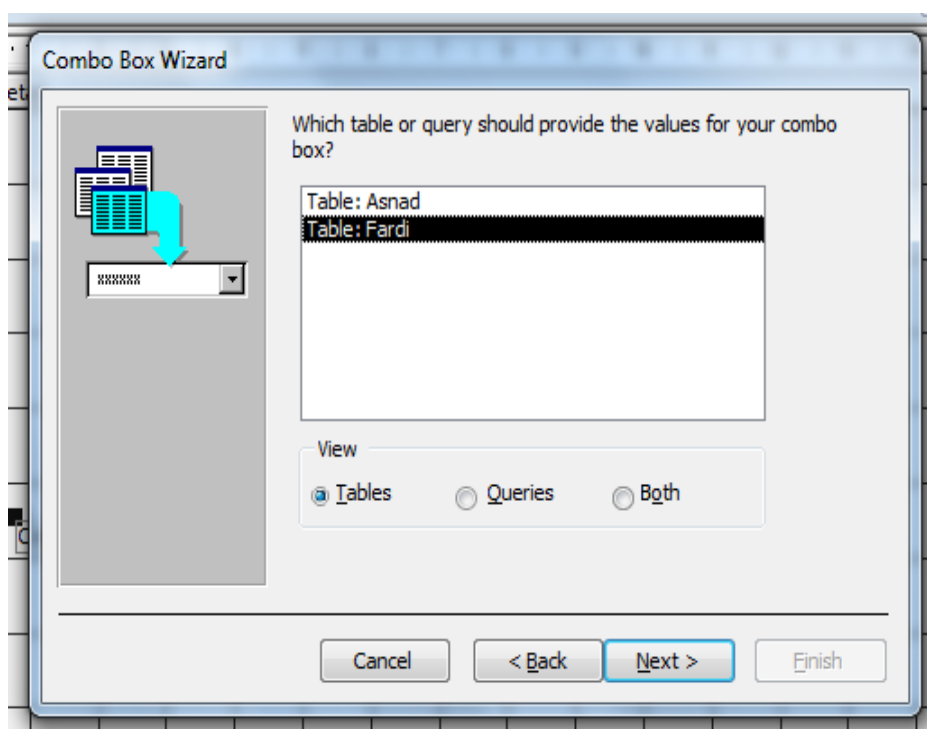


۸. بعد از درج این object در قسمت طراحی ذیل که از شما می خواهد نحوه درج مقادیر در combo را انتخاب نماید.

- انتخاب مقادیر از جداول و Qurey های موجود ( در این برنامه این گزینه را انتخاب می نمائیم سپس Next می کنیم.)
- درج مقادیر توسط خود برنامه نویس

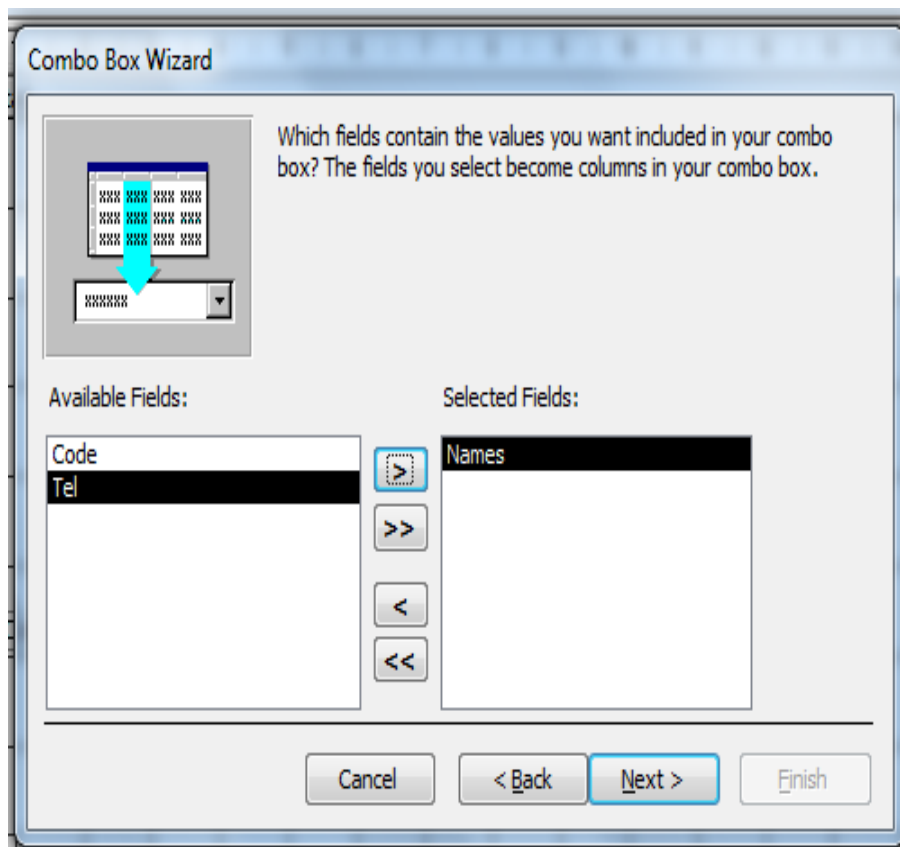


۹. در پنجره بعدی جدول مورد نظر را انتخاب و Next نماید.

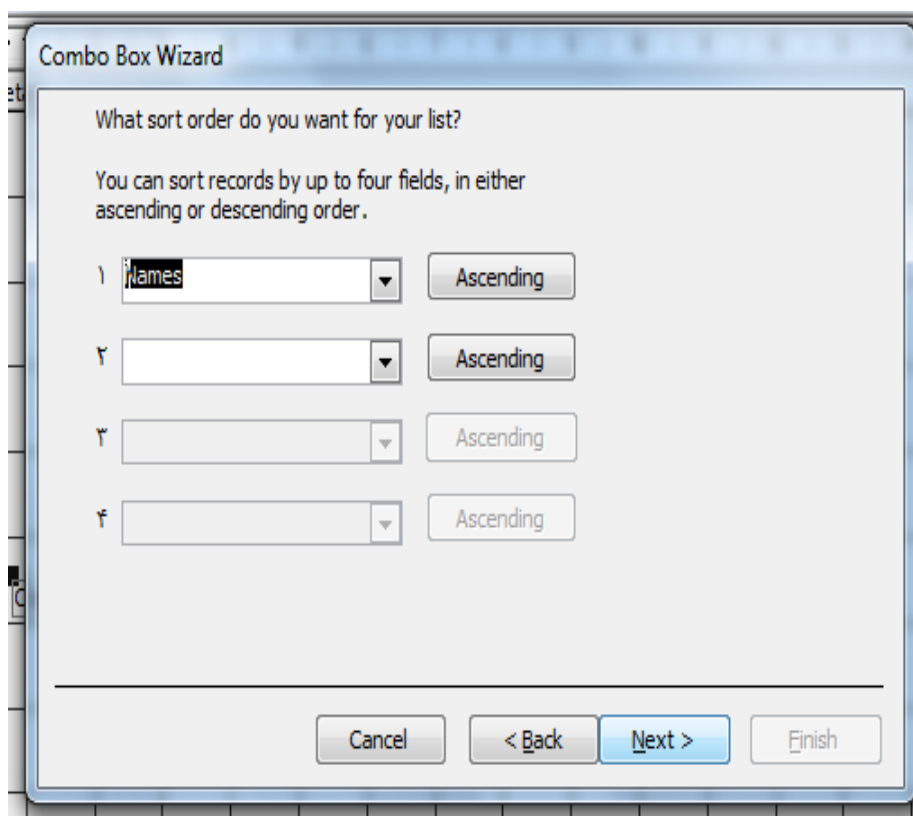




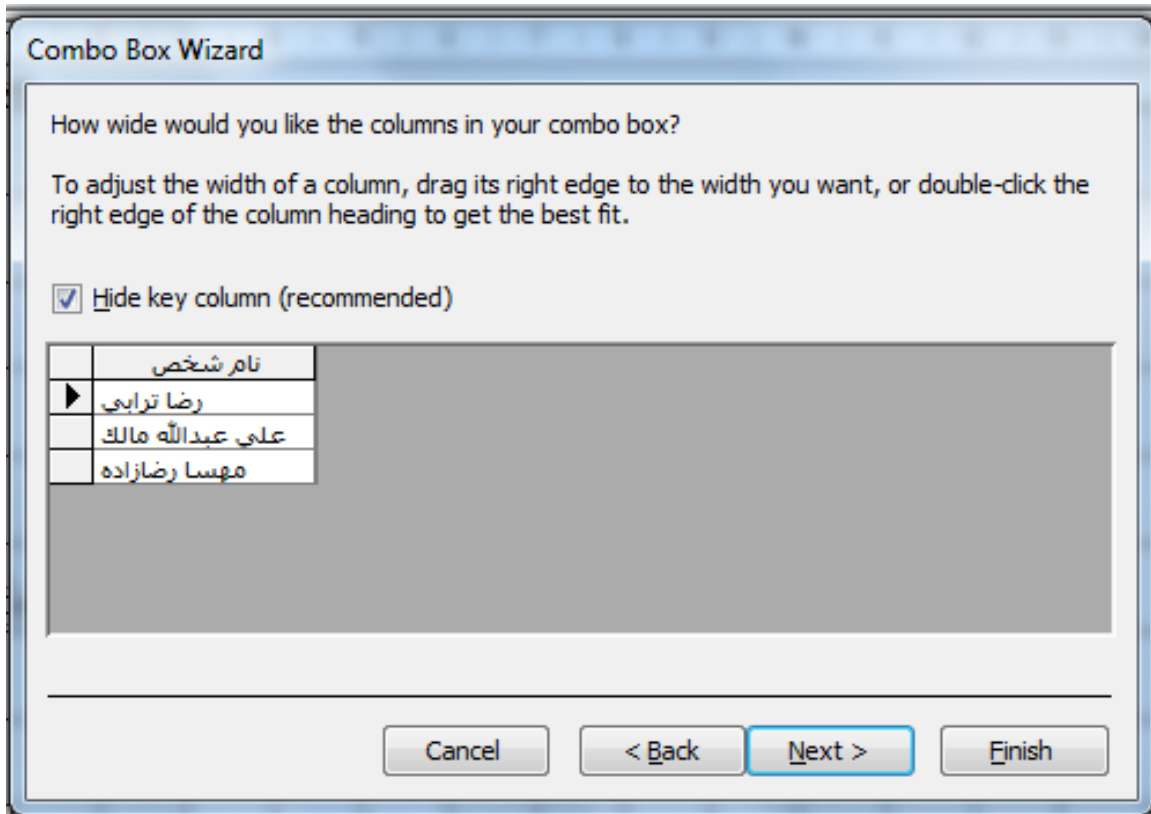
۱۰. در پنجره بعدی فیلد مورد نظر را انتخاب و Next نماید.



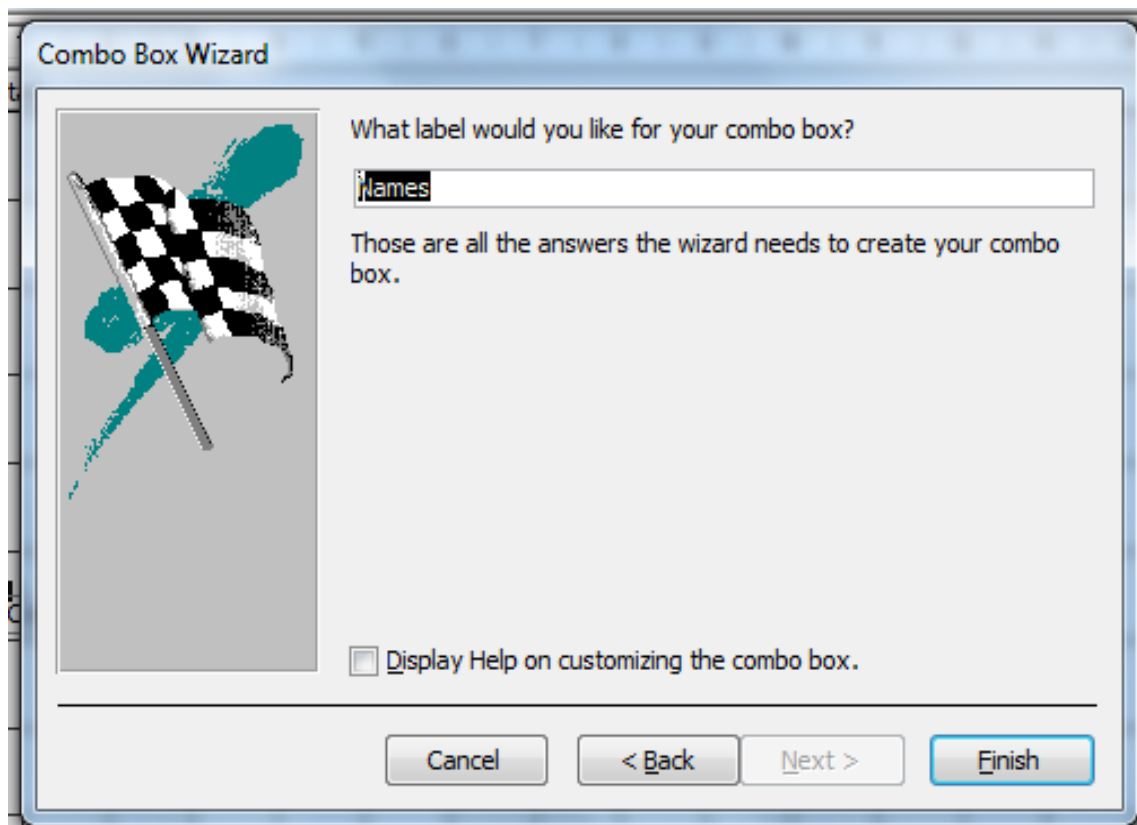
۱۱. در پنجره بعدی بر حسب صعودی و نزولی بودن فیلد انتخاب، مقادیر sort می شود و سپس Next نماید.



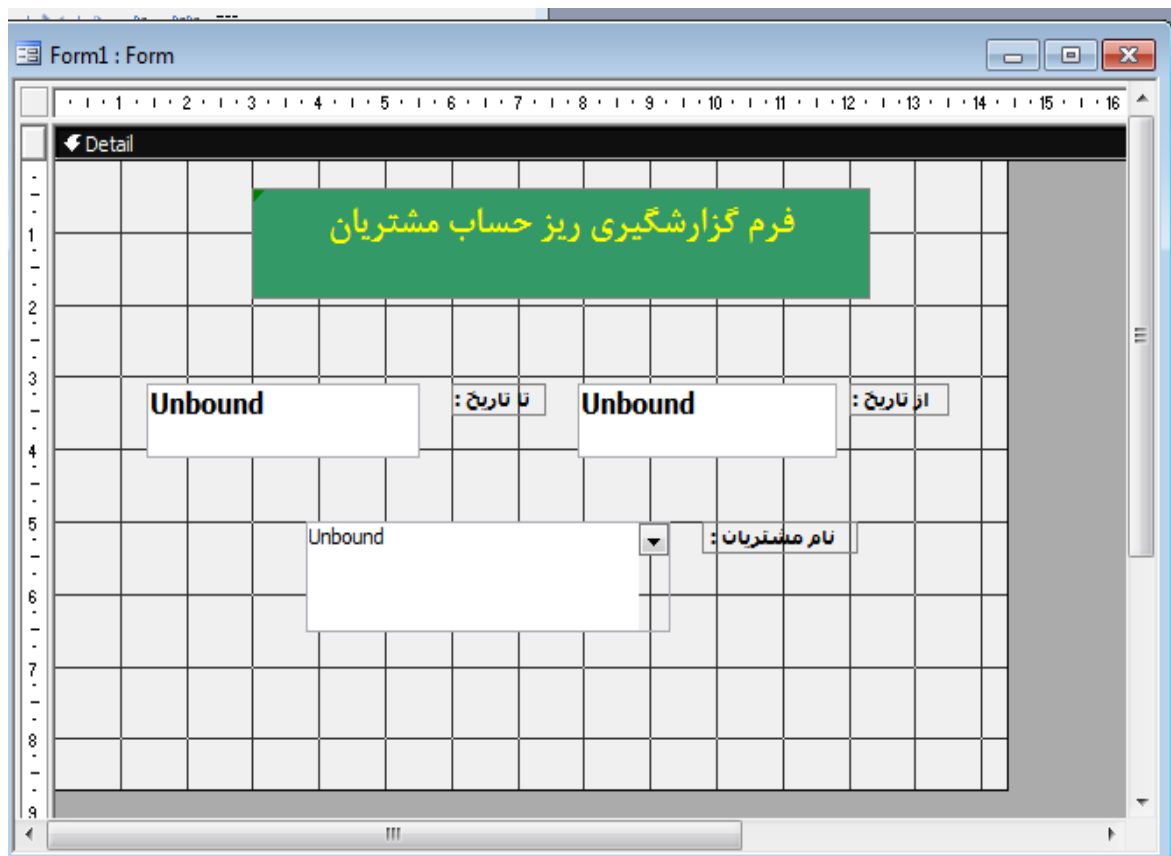
۱۲. در این پنجره مقادیر انتخابی رویت می گردد و پس از تایید Next نماید.



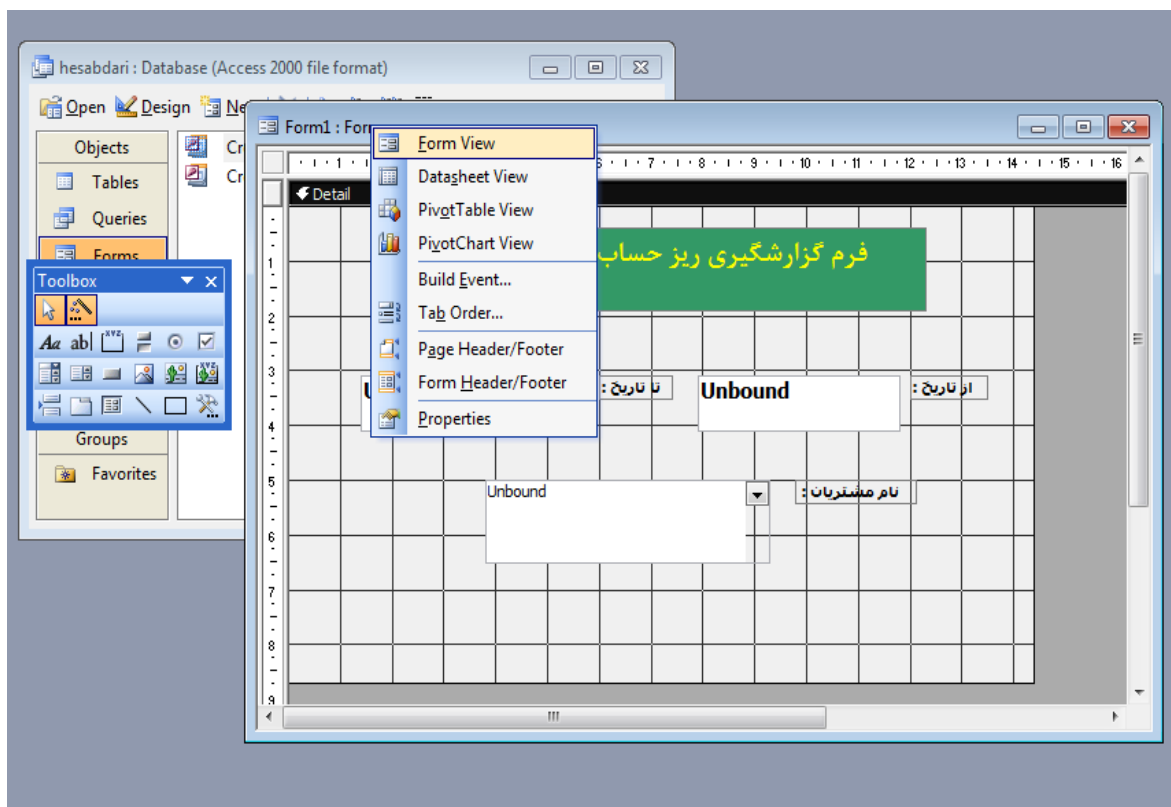
۱۳. در پنجره آخر نام combo را داده و Finish می کنیم. ( معمولاً نام از پیش تعریف شده سیستم را تغییر نمی دهیم.)



۱۴. فرم گزارشگیری به شکل ذیل طراحی می گردد.



۱۵. جهت دیدن فرم از دید کاربر روی نوار بالای پنجره کلیک سمت راست نموده و گزینه form view را انتخاب نماید.



۱۶. فرم طراحی تا این مرحله به شکل ذیل می باشد.

Form1 : Form

فرم گزارشگیری ریز حساب مشتریان

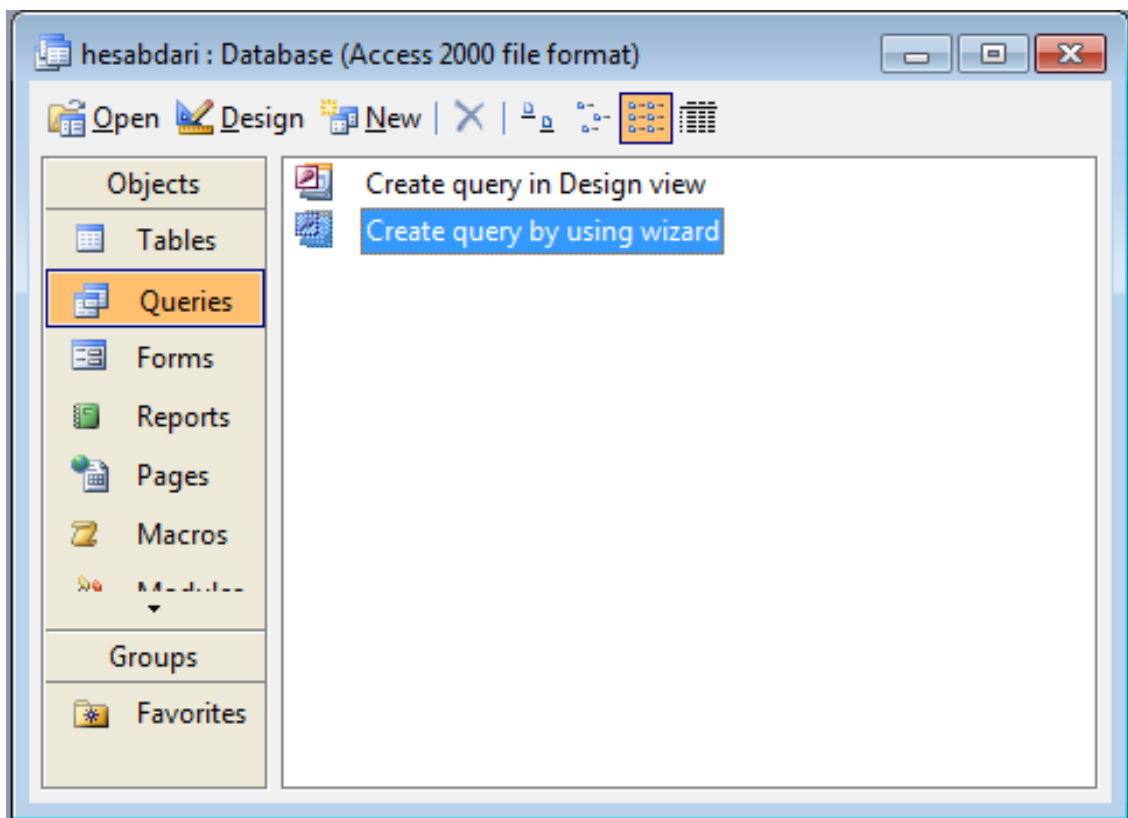
از تاریخ : تا تاریخ :

نام مشتریان :

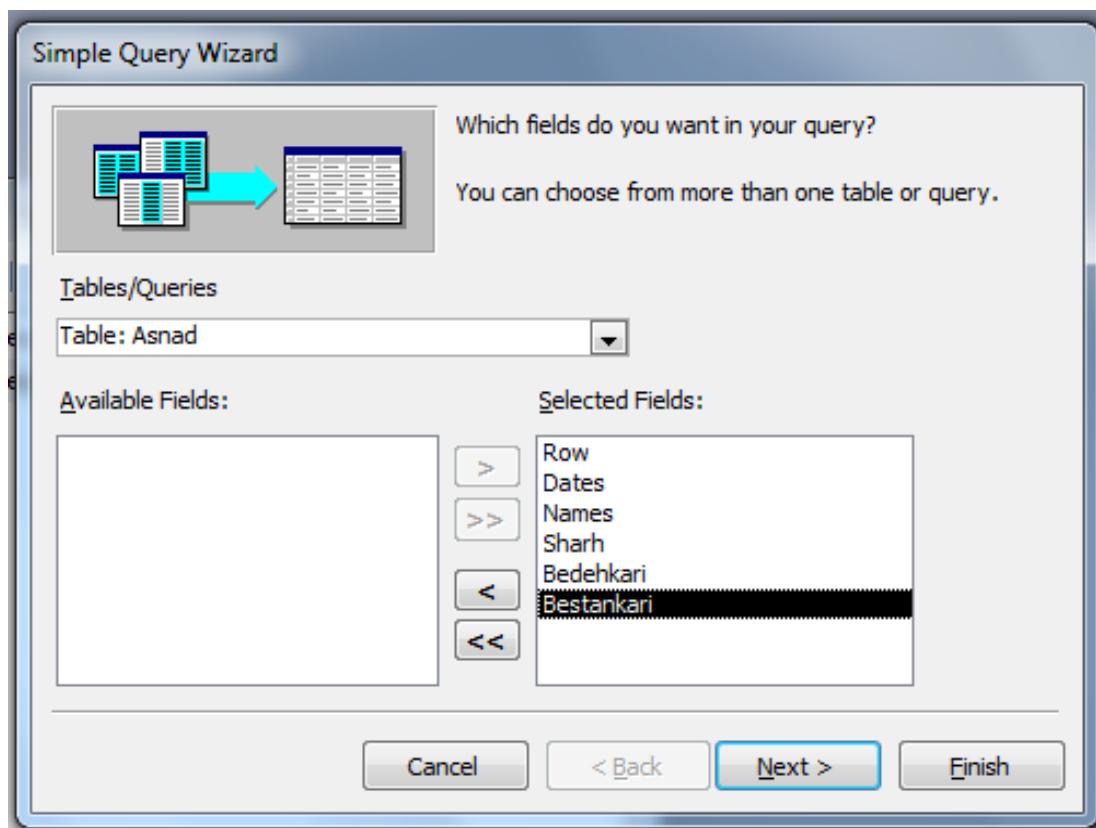
Record: 1 of 1

۱۷. فرم فوق را به نام Form ذخیر نمایید.

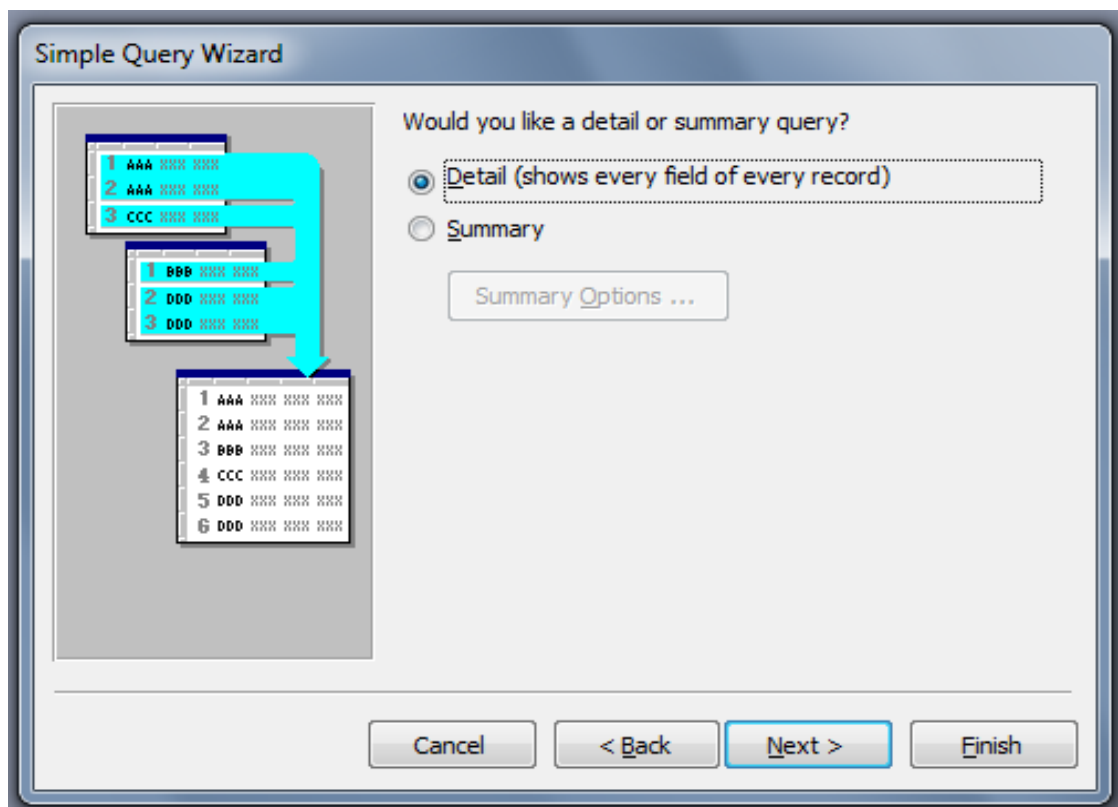
۱۸. در این مرحله جهت ایجاد Query از جدول Asnad وارد قسمت Queries می شویم و روی گزینه دوم دوبار کلیک می کنیم.

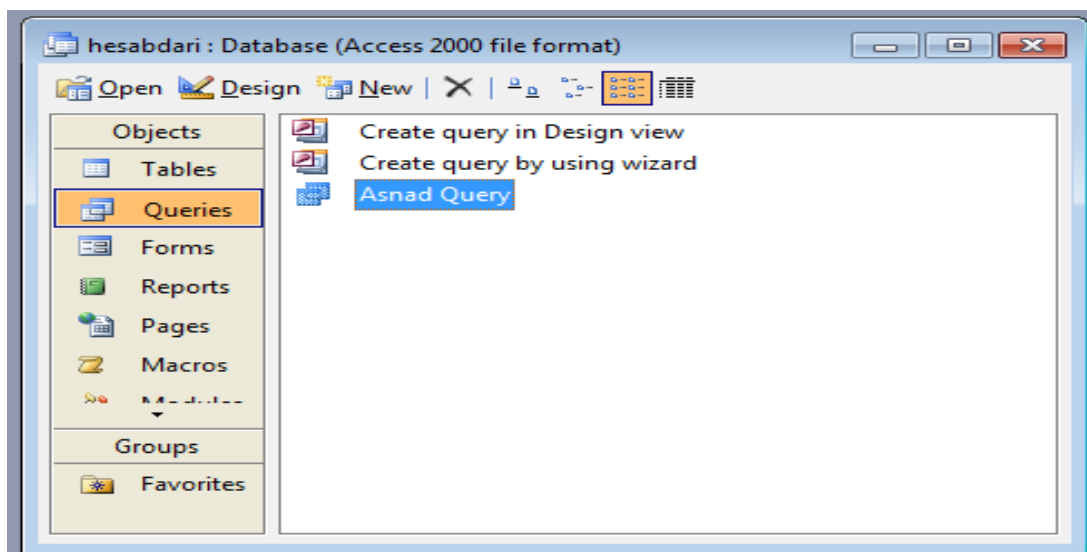
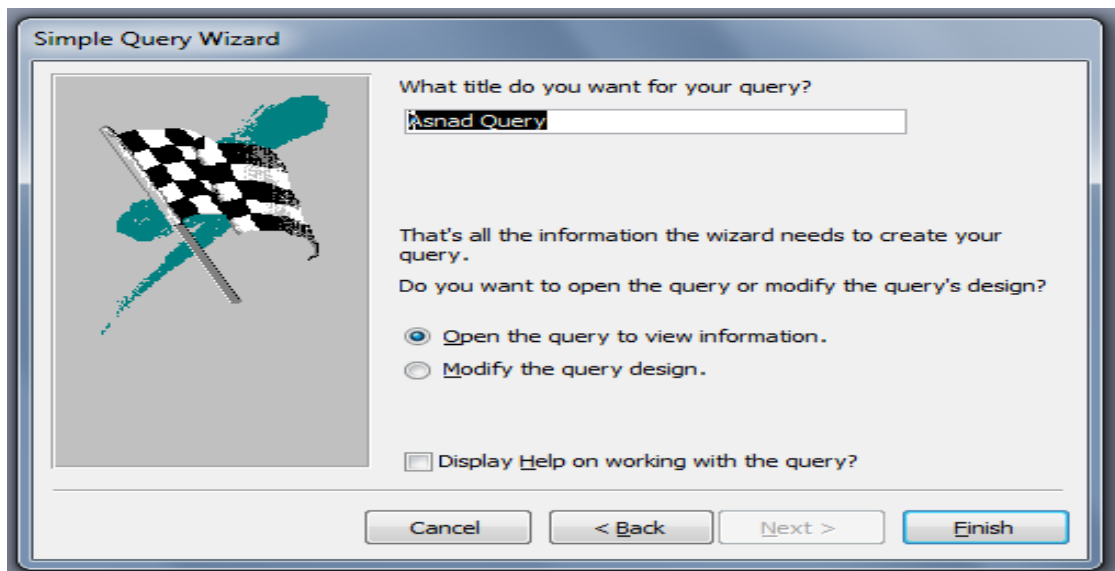


۱۹. پنجره Wizard ظاهر شده و پس از انتخاب جدول مورد نظر (Asnad) تمامی فیلدها را انتخاب و Next می‌کنیم.

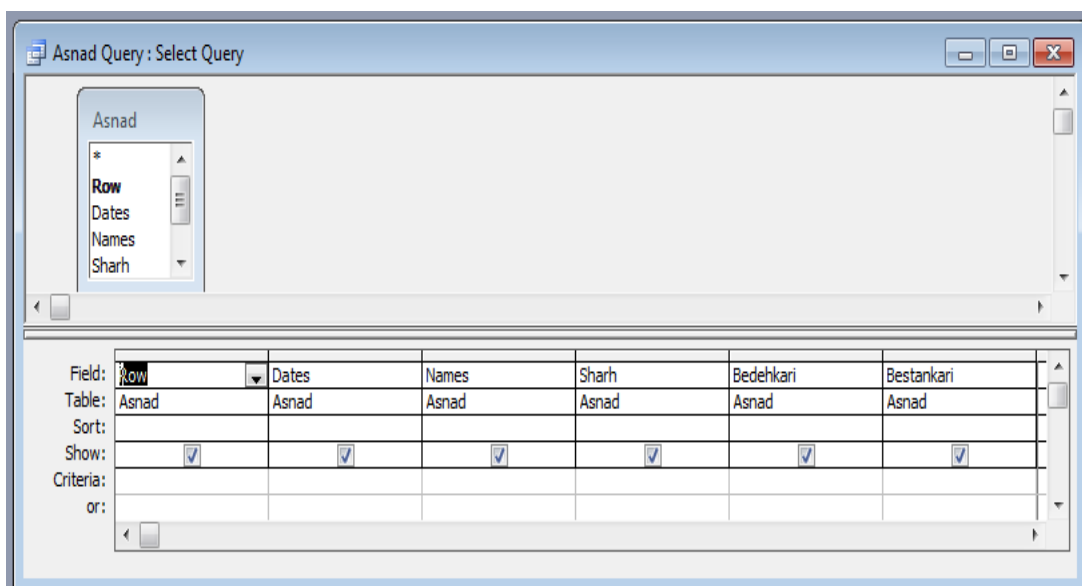


۲۰. پنجره بعدی را Next و در آخر Finish نماید.

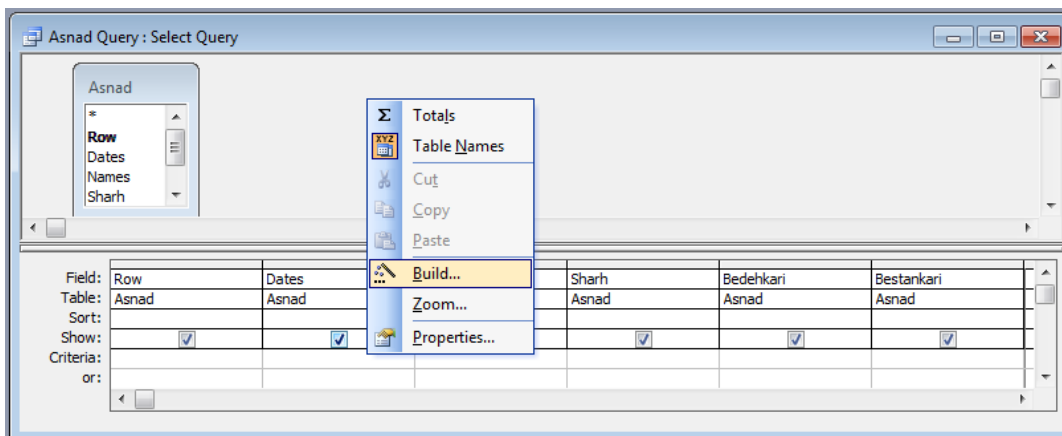




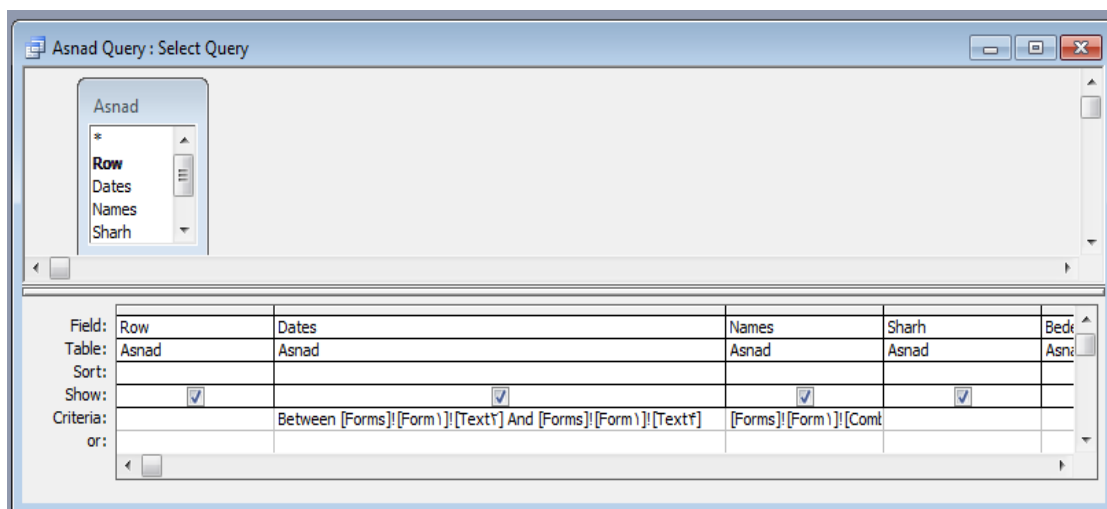
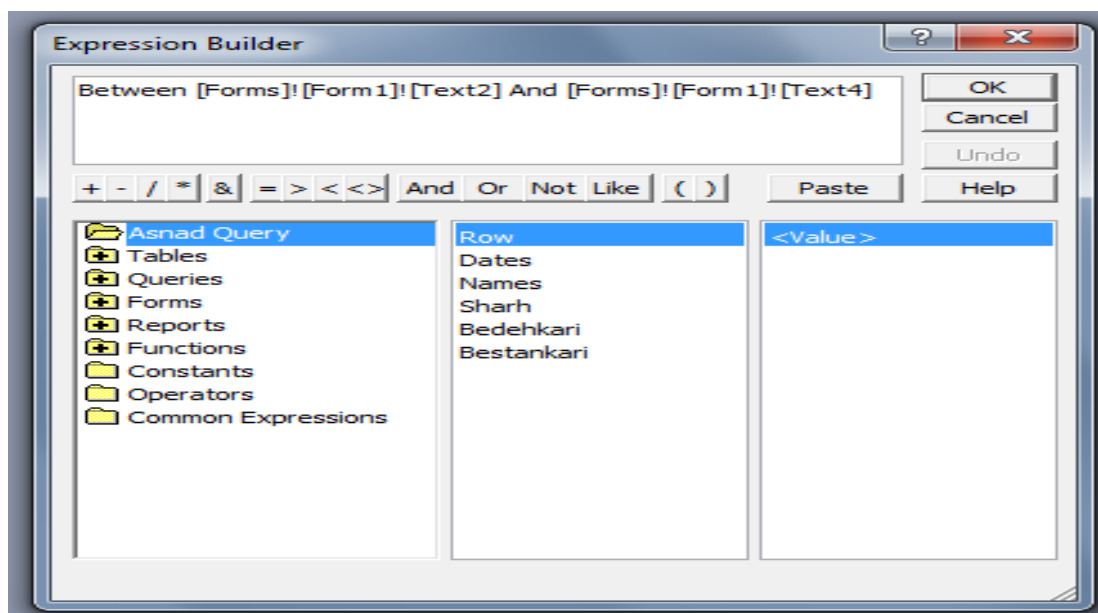
۲۱. جهت کنترل فیلدهای تاریخ و combo box در فرم (Form می بایست دستوراتی در گزینه Criteria فرم Asnad Query درج نمود. لذا روی Asnad Query کلیک راست نموده و گزینه Design را انتخاب نماید.



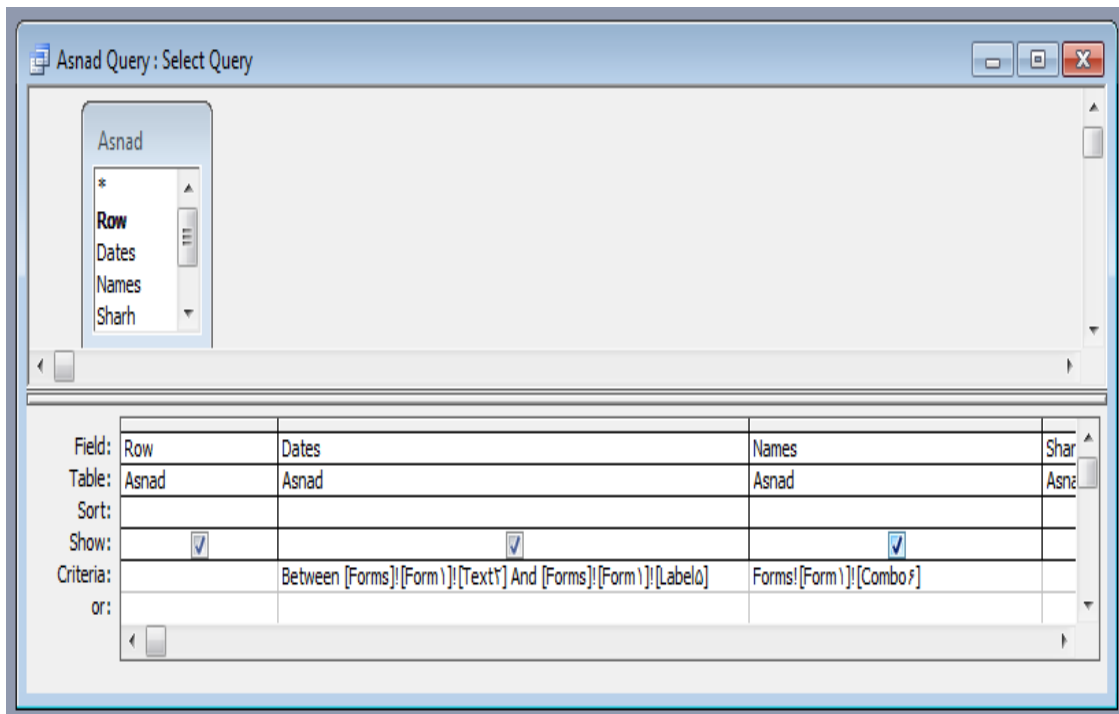
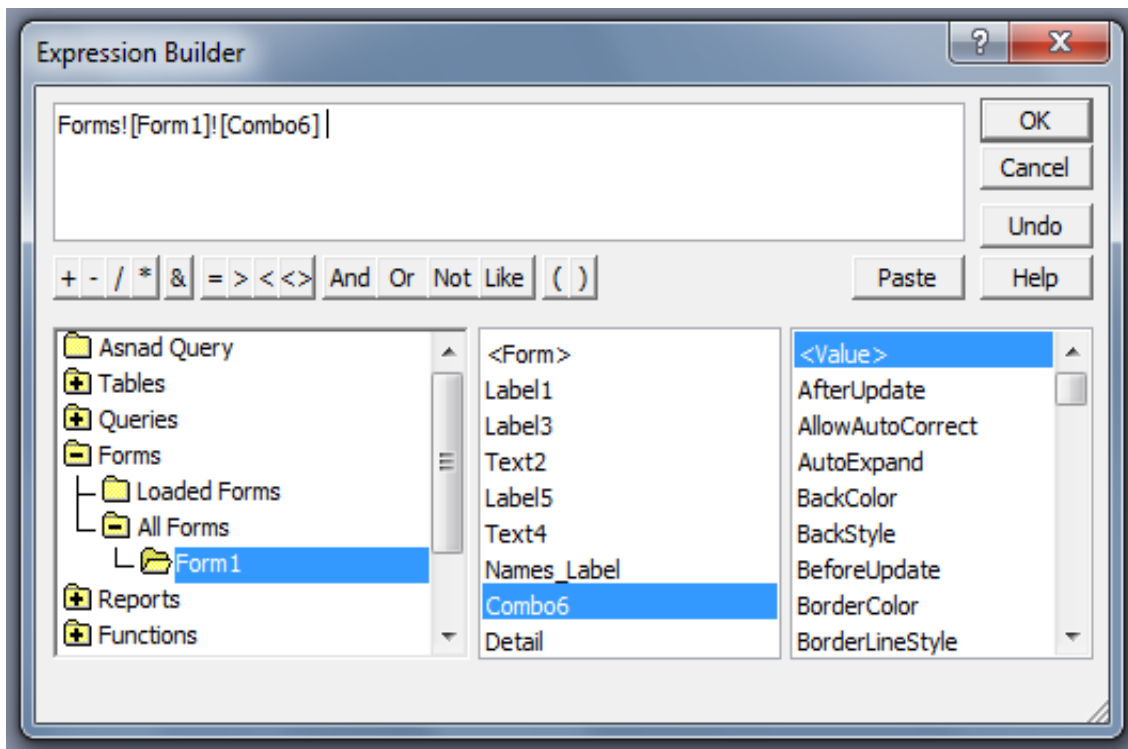
۲۲. روی خانه Criteria Dates کلیک سمت راست کرده و گزینه Build... را انتخاب نماید.



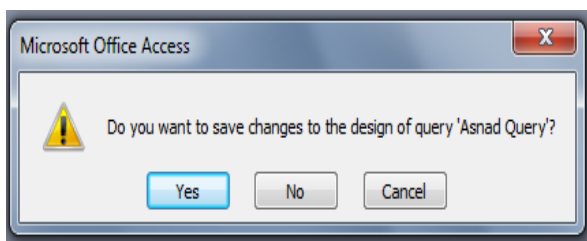
۲۳. در پنجره Expression Builder فرمول شکل زیر را درج و سپس ok نماید.



۲۴. حال روی خانه Criteria Names کلیک سمت راست نموده و دستور ذیل را در آن درج نماید.

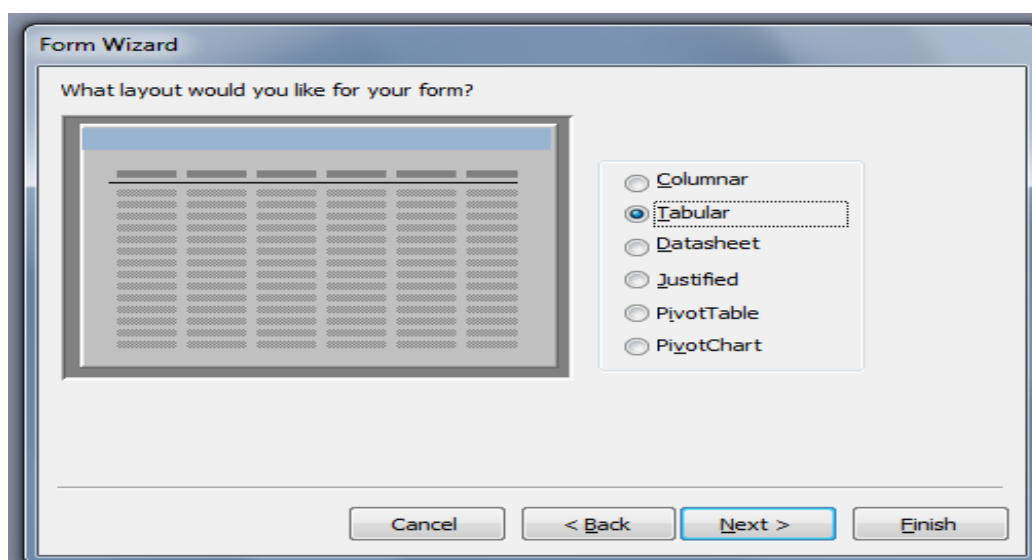
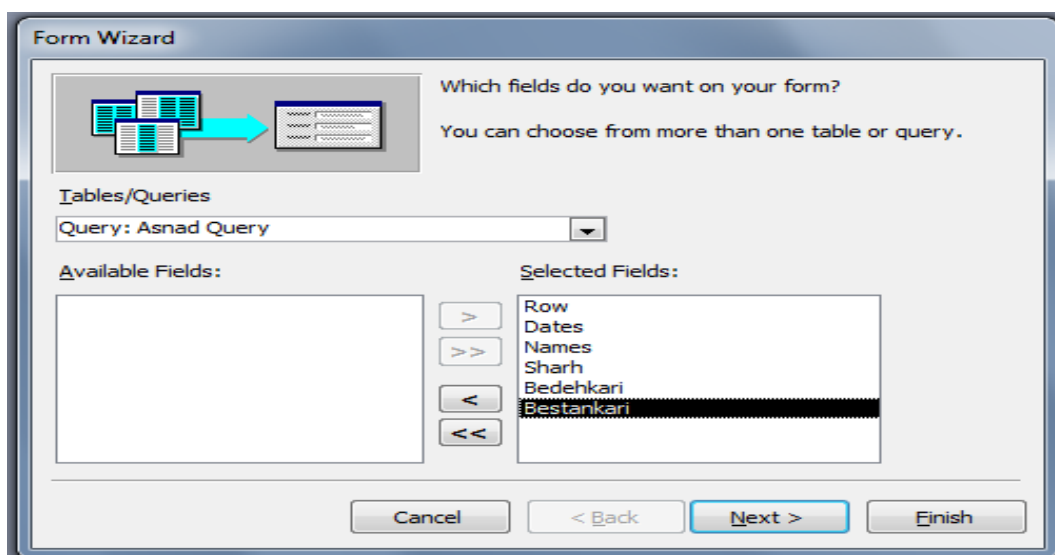
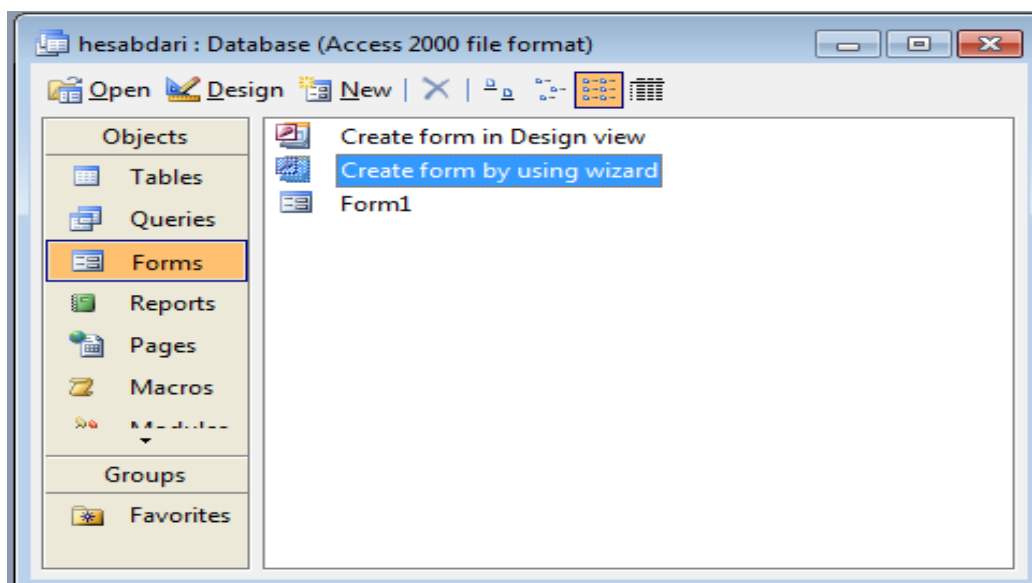


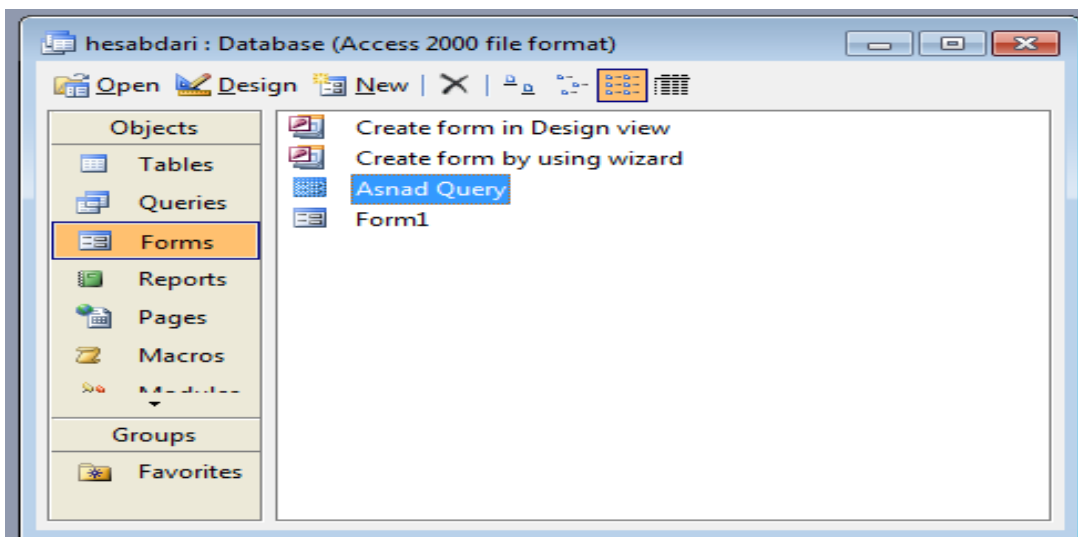
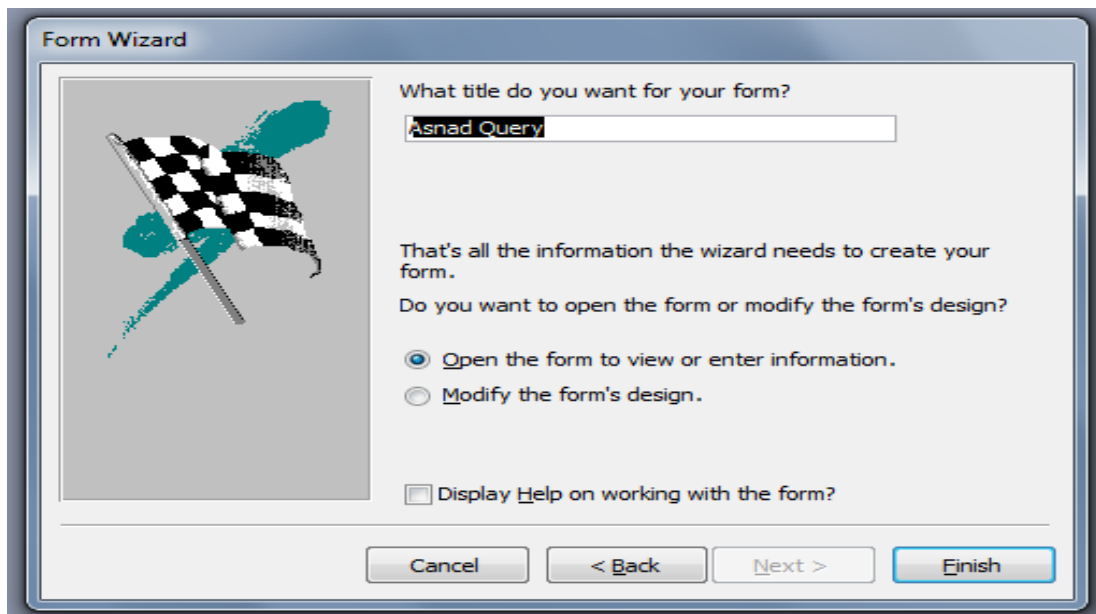
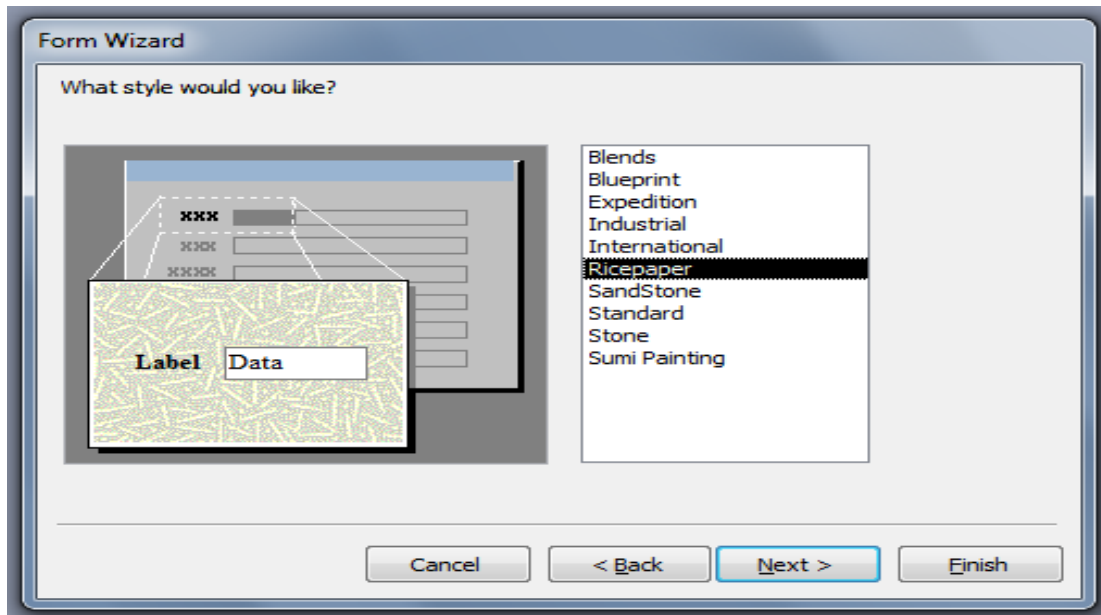
۲۵. هنگام خروج از این پنجره آن را ذخیره نماید.



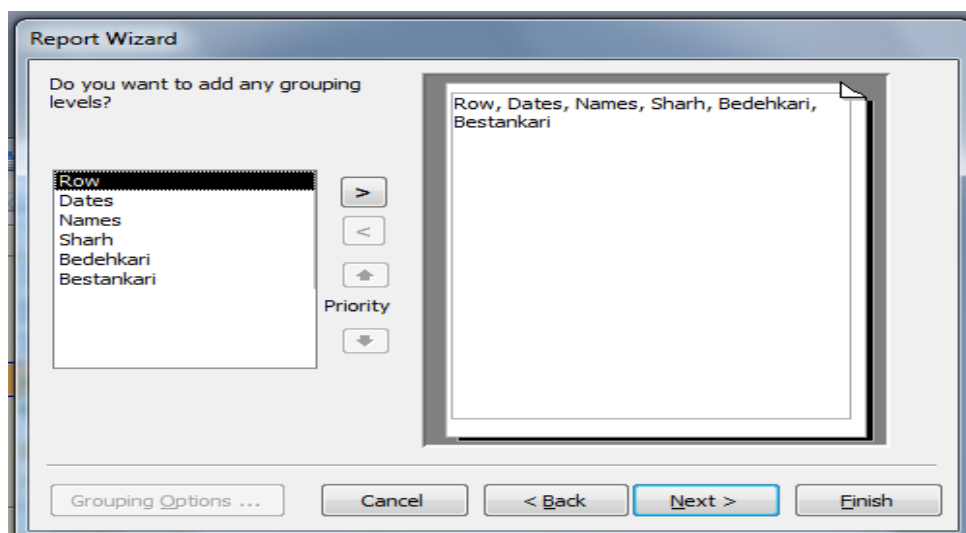
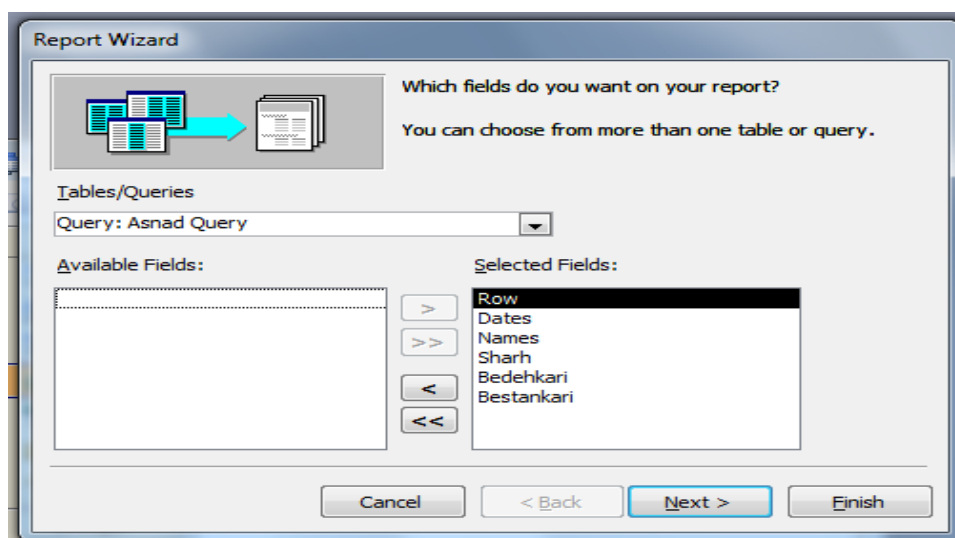
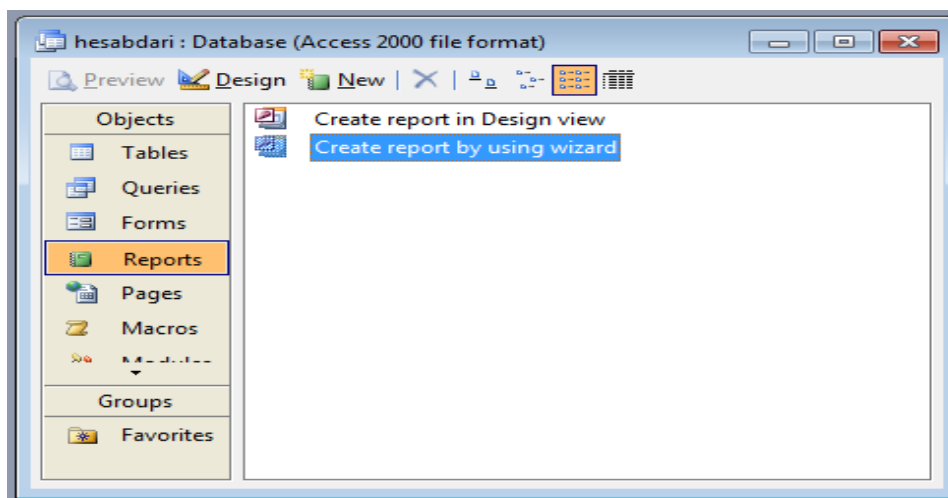


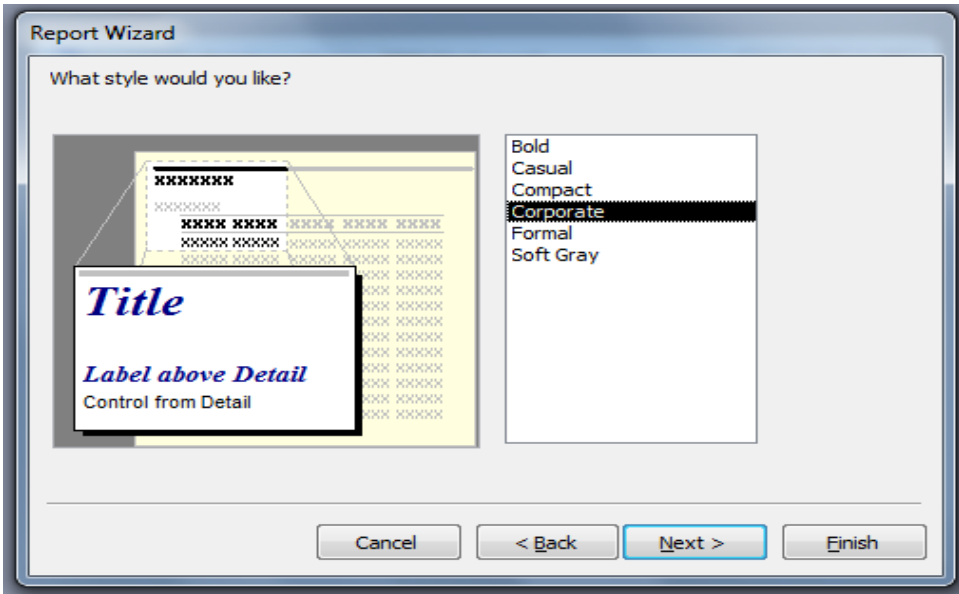
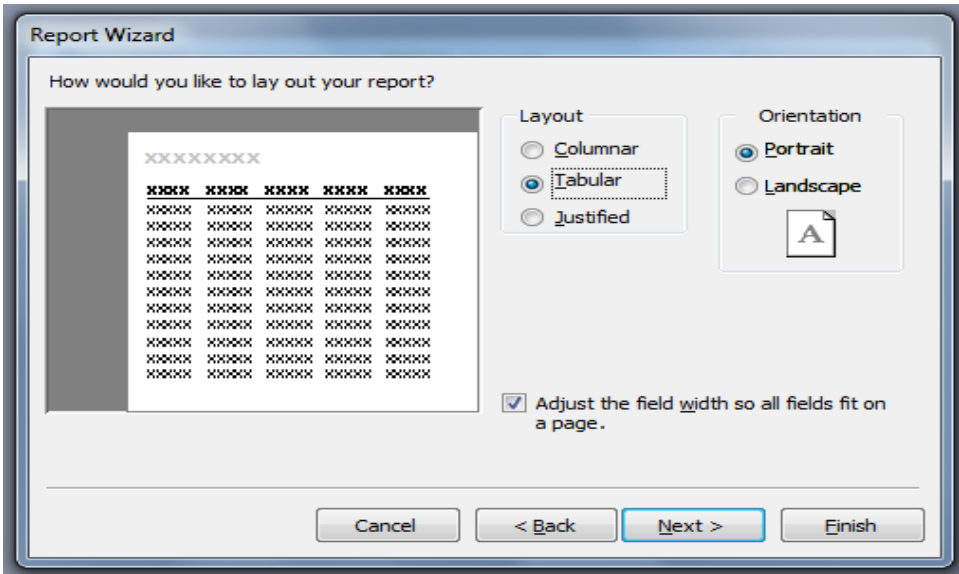
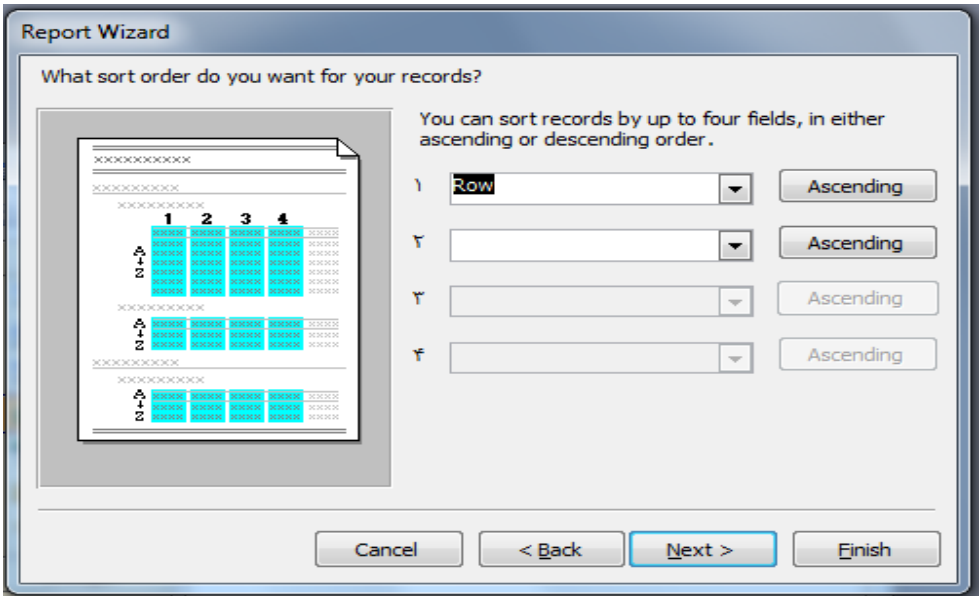
۲۶. جهت طراحی کلید گزارشگیری در فرم (Form)، اول بایستی یک فرم از Asnad Query ایجاد نمایم.



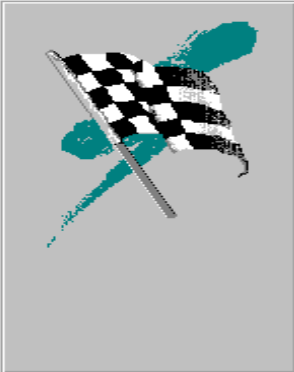


۲۷. جهت طراحی کلید پیش نمایش در فرم (Form) بایستی یک Report از Asnad Query ایجاد نمود.





**Report Wizard**



What title do you want for your report?

That's all the information the wizard needs to create your report.  
 Do you want to preview the report or modify the report's design?  
 Preview the report.  
 Modify the report's design.

Display Help on working with the report?

Cancel < Back Next > Finish

۲۸. جهت مرتب سازی فرم گزارشگیری و پیش نمایش می توان آن را تغییر داد.

Asnad Query : Form

Form Header

سند گزارش از ریز حساب مشتریان

ردیف	تاریخ	نام شخص	شرح	بدهکاری	بستانکاری
Row	Dates	Names	Sharh	Bedehkari	Bestankari

Form Footer

Asnad Query

Report Header

پیش نمایش سند ریز مشتریان

Page Header

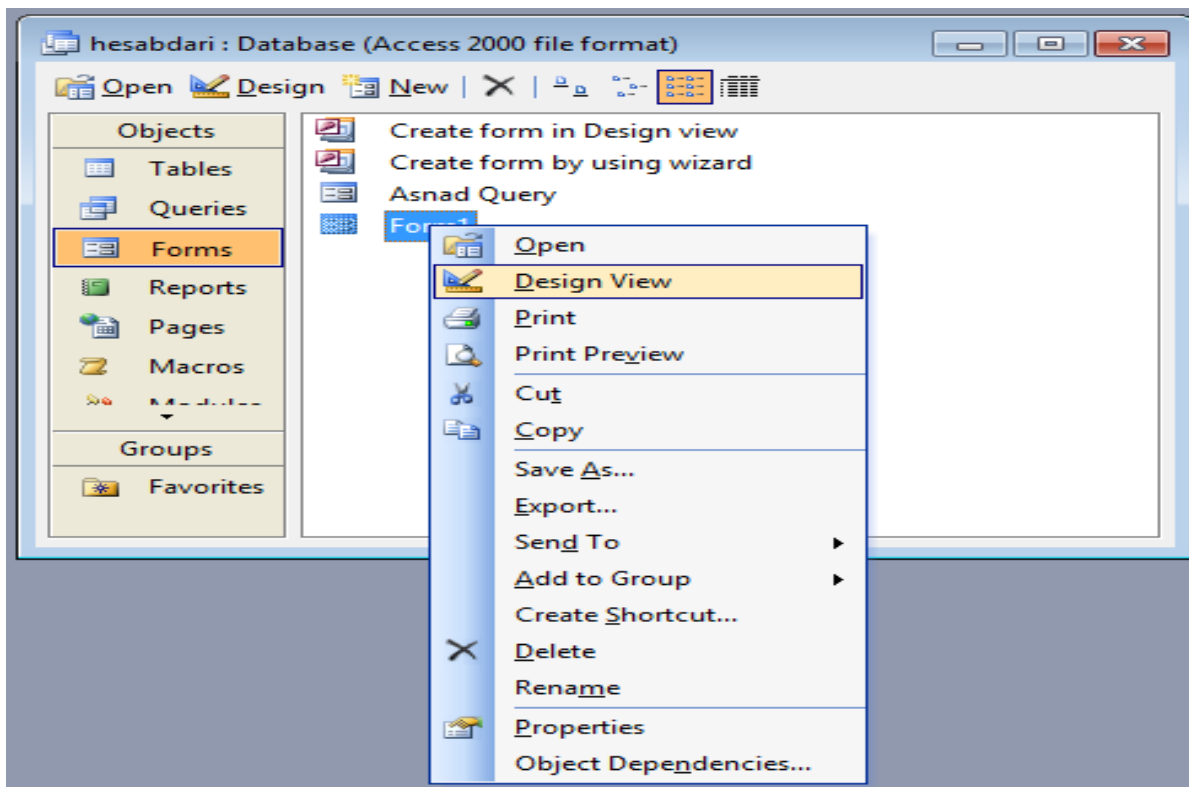
ردیف	تاریخ	نام شخص	شرح	بدهکاری	بستانکاری
Row	Dates	Names	Sharh	Bedehkari	Bestankari

Page Footer

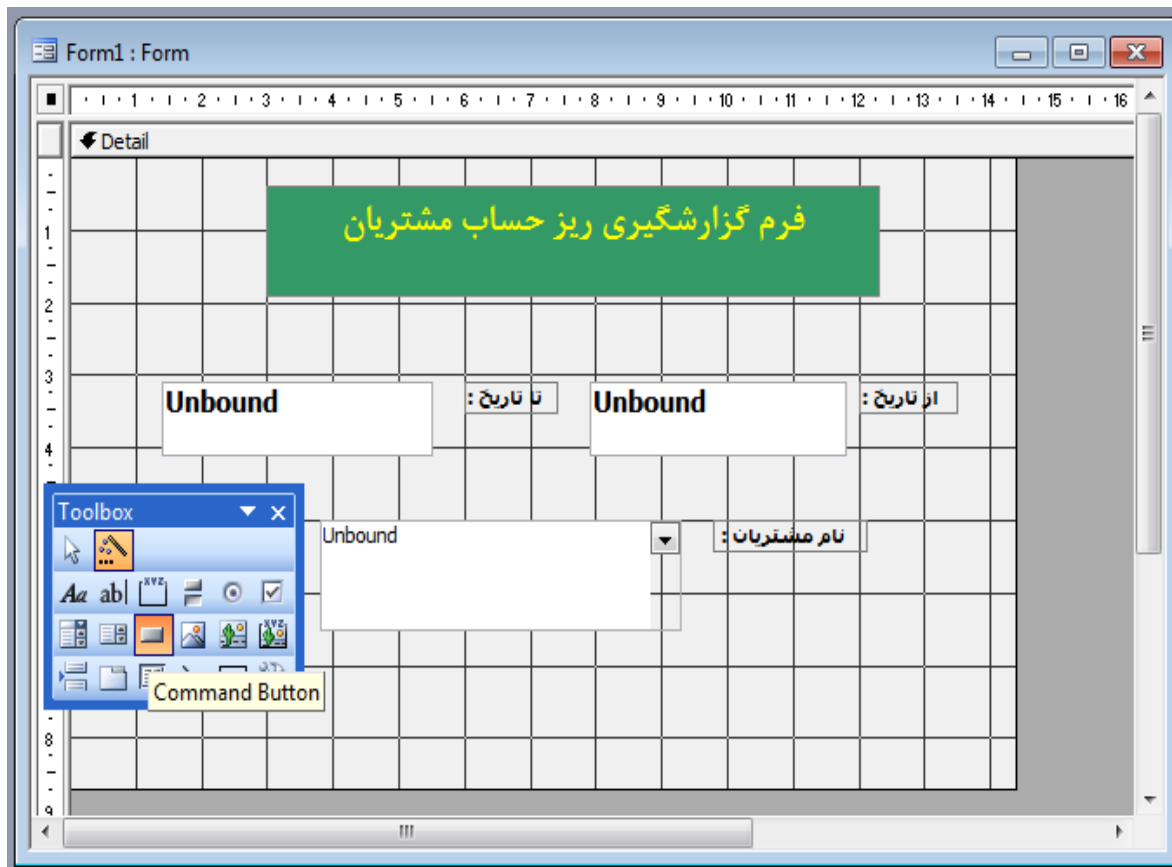
=Now()      ="Page " & [Page] & " of " & [Pages]

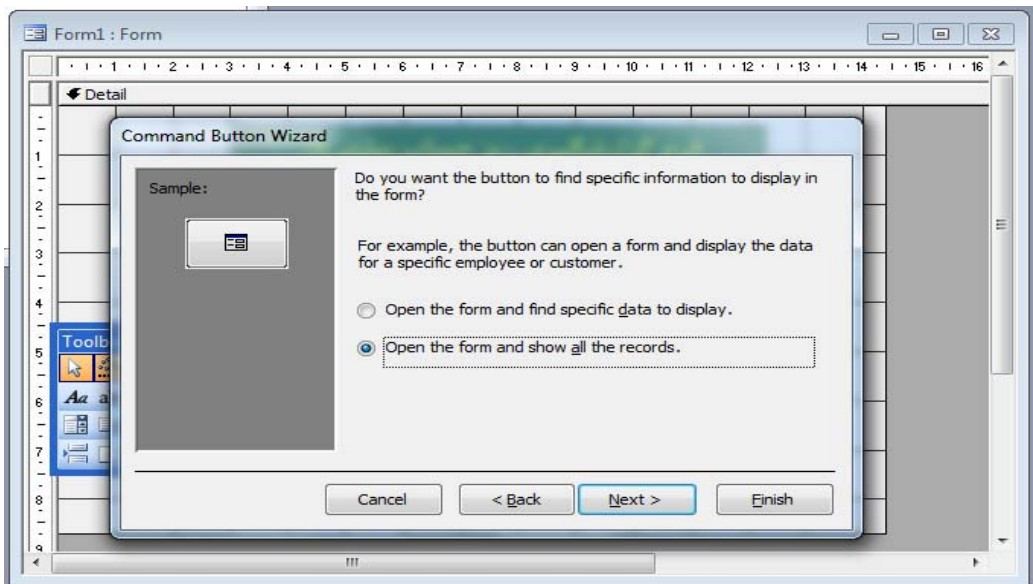
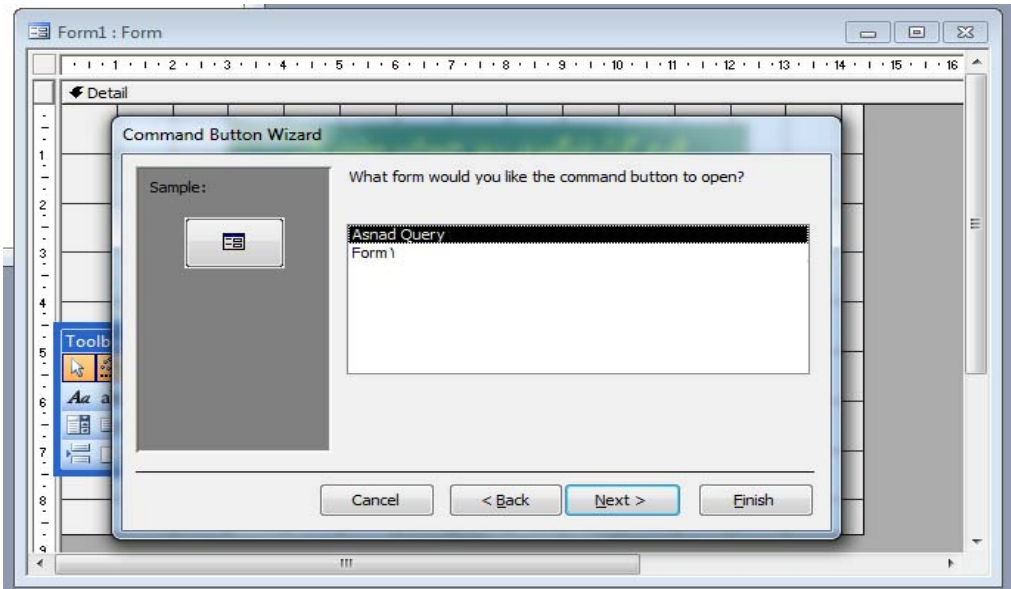
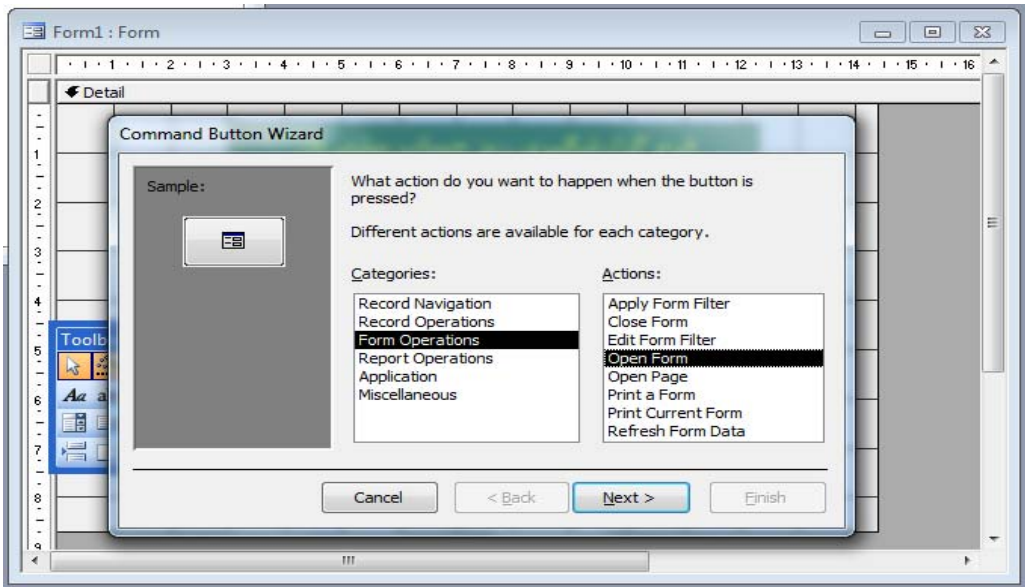
Report Footer

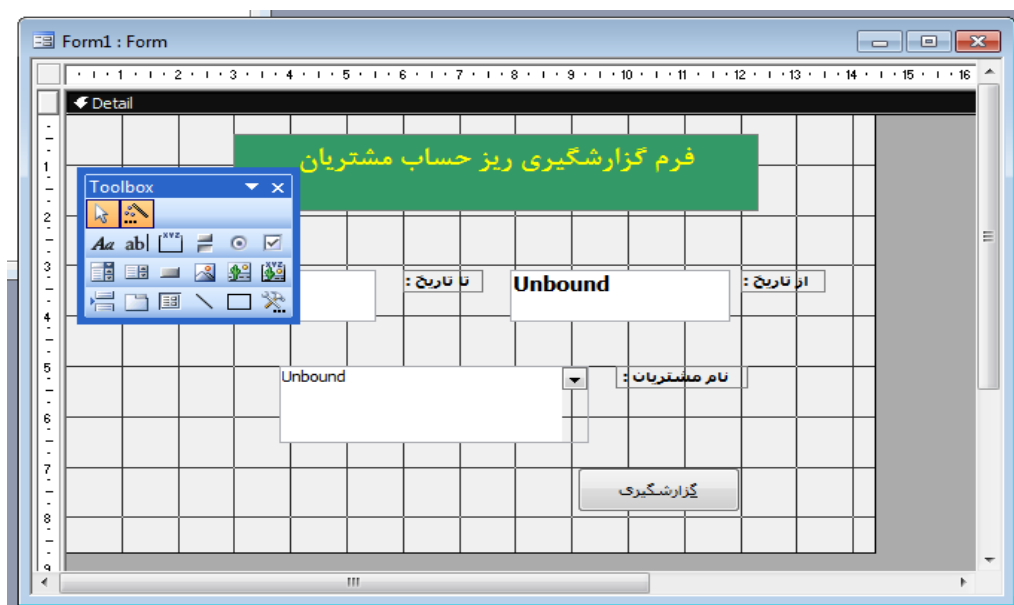
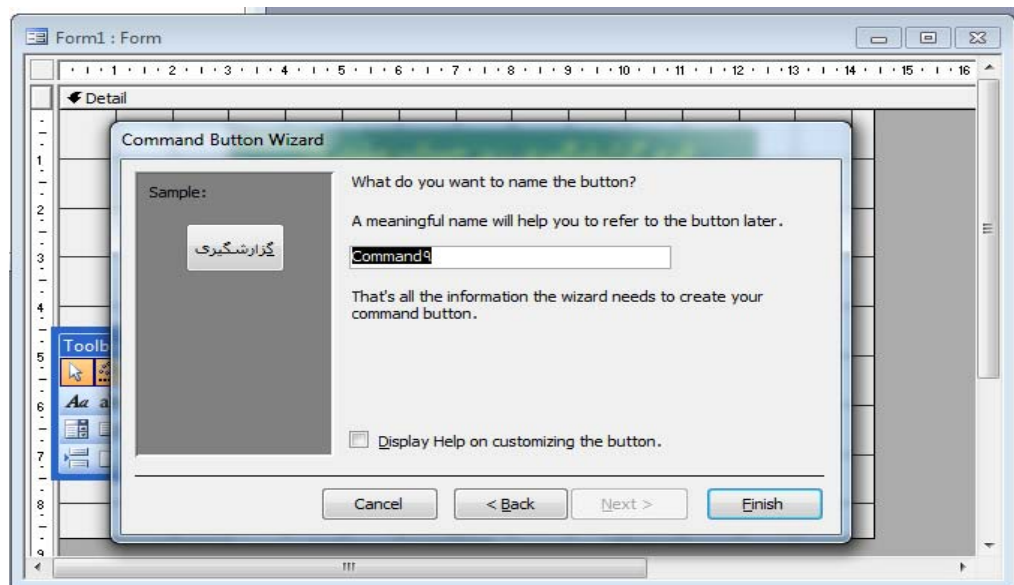
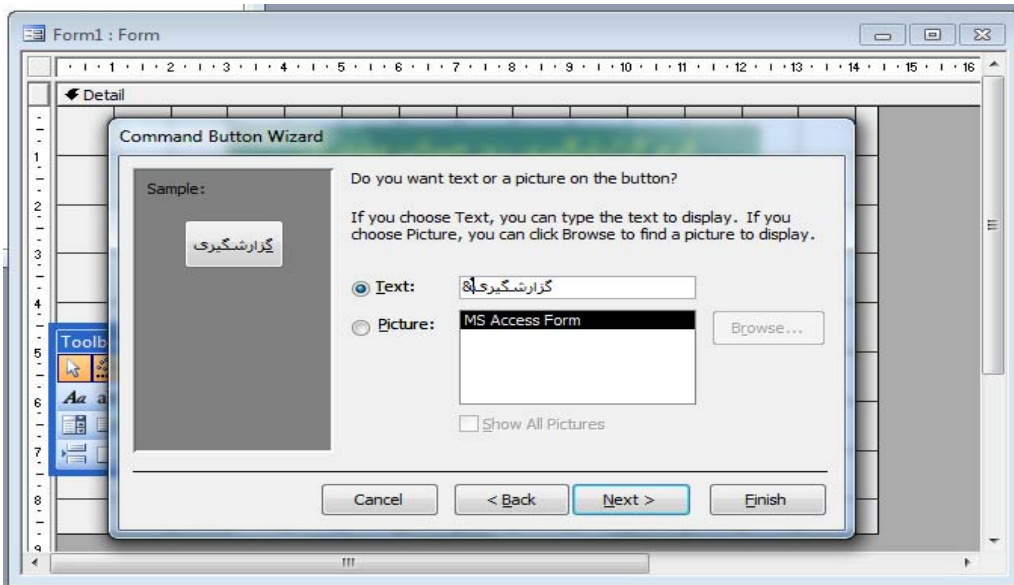
۲۹. جهت طراحی دو کلید آخر در فرم (Form) آن را در حالت Design با می کنیم.



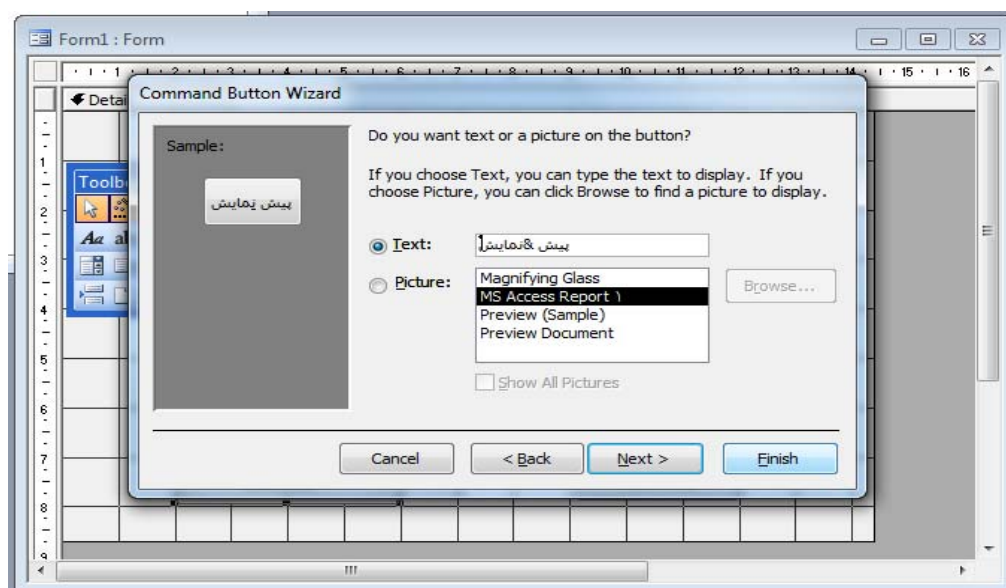
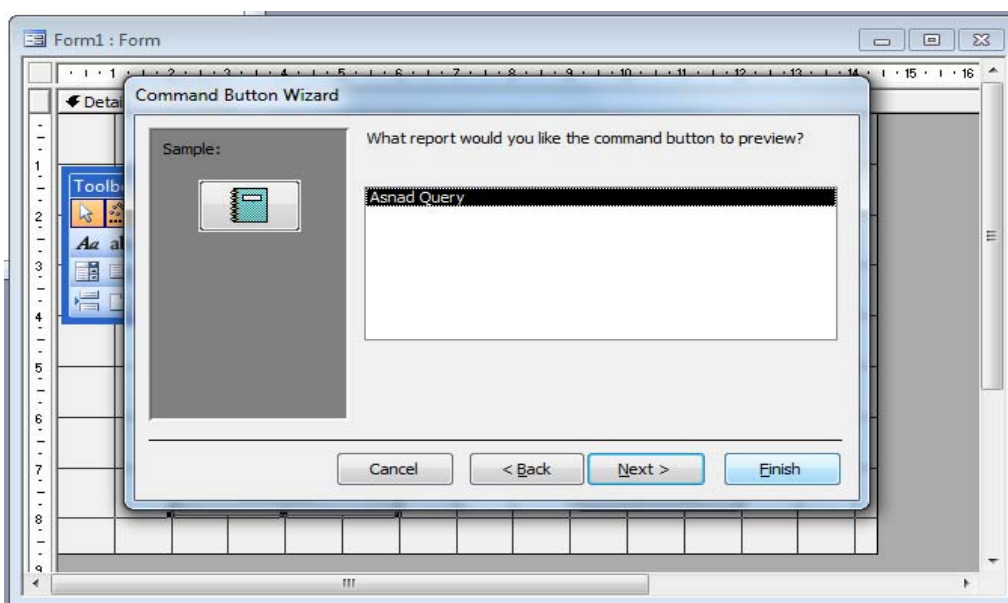
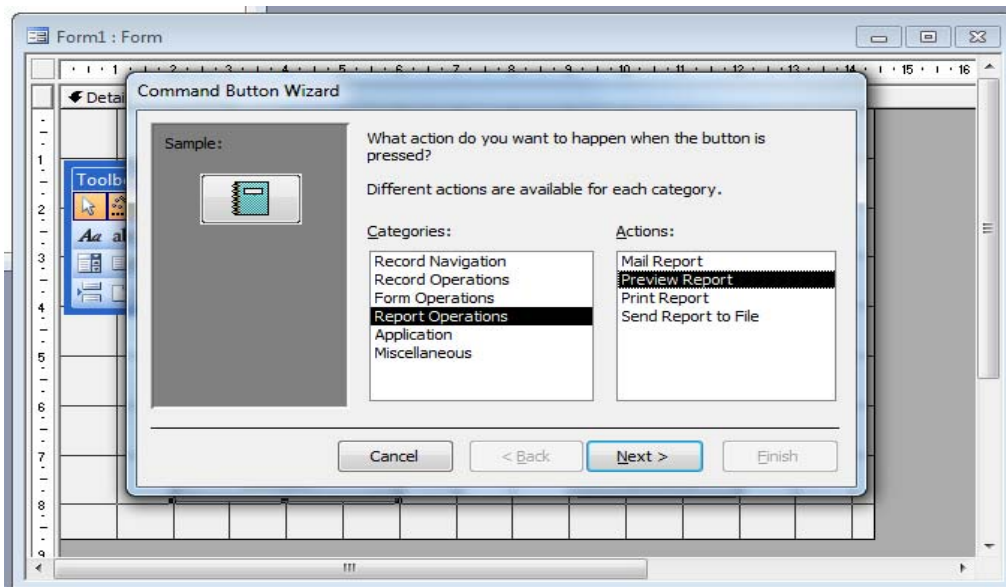
۳۰. در مد با استفاده از Toolbox دو کلید گزارشگیری و پیش نمایش در فرم (Form) ایجاد می کنیم.

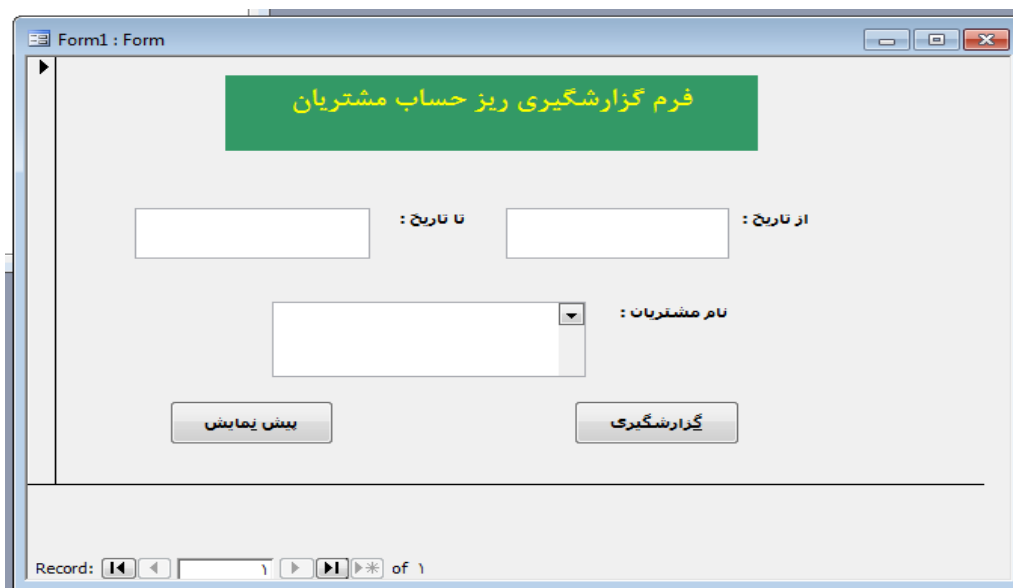
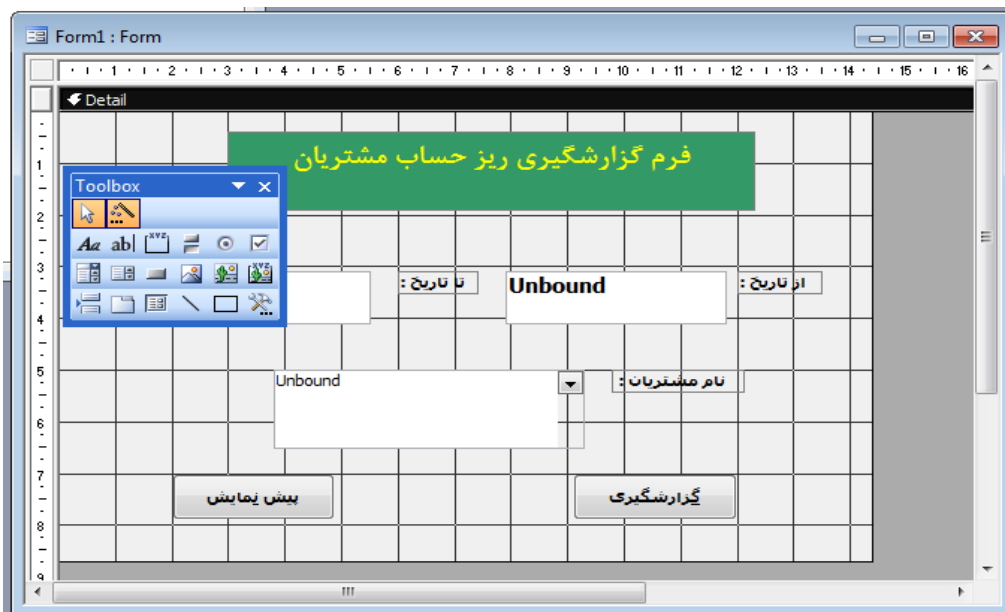
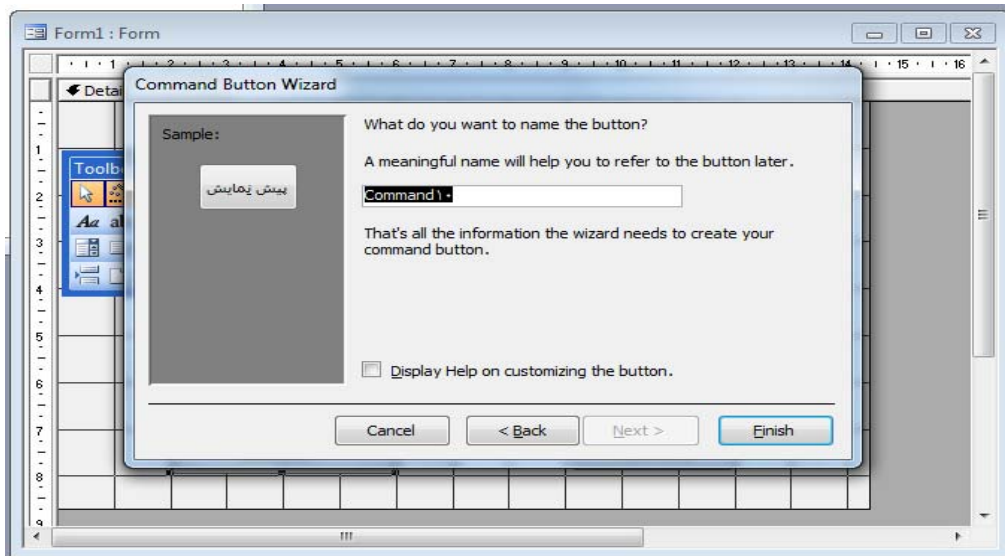












۳۱. حال به تاریخ ها مقدار داده و نام مشتری را انتخاب و کلید گزارشگیری را بزنید.

Form1 : Form

### فرم گزارشگیری ریز حساب مشتریان

از تاریخ:  تا تاریخ:

نام مشتریان:

Record: ۱ of ۱

و در نهایت گزارش نهایی ریز حساب مشتریان :

Asnad Query

### سند گزارش از ریز حساب مشتریان

ردیف	تاریخ	نام شخص	شرح	بدهکاری	بستانکاری
۴	۱۳۸۹/۰۱/۰	رضا ترابی	خرید هندوانه	۴۰۰,۰۰۰	۰
۵	۱۳۸۹/۰۱/۲	رضا ترابی	واریز	۰	۲,۰۰۰,۰۰۰
۶	۱۳۸۹/۰۱/۲	رضا ترابی	خرید غذا	۲۰۰,۰۰۰	۰
۷	۱۳۸۹/۰۱/۲	رضا ترابی	واریز	۰	۵,۰۰۰,۰۰۰
۹	۱۳۸۹/۰۱/۳	رضا ترابی	خرید بوجرخه	۸,۰۰۰,۰۰۰	۰
۱۰	۱۳۸۹/۰۱/۳	رضا ترابی	کلاه ایمنی	۸۰۰,۰۰۰	۰
*	AutoNumt			۰	۰

Record: ۱ of ۶

## پیش نمایش سند ریز مشتریان

ردیف	تاریخ	نام شخص	شرح	بدهکاری	بستانکاری
۴	۱۳۹۴/۰۱/۰۱	رضا نوری	خرید ادواته	۴۰,۰۰۰	۰
۵	۱۳۹۴/۰۱/۲۰	رضا نوری	واریز	۰	۴۰,۰۰۰,۰۰۰
۶	۱۳۹۴/۰۱/۲۵	رضا نوری	خرید عطا	۴۰۰,۰۰۰	۰
۷	۱۳۹۴/۰۱/۲۸	رضا نوری	واریز	۰	۴۰۰,۰۰۰,۰۰۰
۸	۱۳۹۴/۰۱/۳۰	رضا نوری	خرید لوجیستیک	۸,۰۰۰,۰۰۰	۰
۹	۱۳۹۴/۰۱/۳۰	رضا نوری	کلاه ایمنی	۸۰۰,۰۰۰	۰

## پیوست ۴ : کتابخانه الکترونیکی

### فصل اول : کتابخانه و کتابداری

#### تعریف کتابداری :

کتابداری دانشی است که کارکردهای ویژه گردآوری ، سازماندهی و اشاعه دانش را با اعمال روشهای مدیریت برعهده دارد.

#### خدمات فنی کتابداری :

مجموع فرآیندها و کارهایی را که برای تهیه و خرید مواد کتابخانه ای ، فهرست نویسی ، رده بندی و آماده سازی منابع برای گذاشتن در قفسه ها و دیگر فضا های فیزیکی انجام می شود ، خدمات فنی کتابداری می گویند.

#### خدمات عمومی کتابداری :

در کتابخانه چهار کارکرد عمده مجموعه سازی ، سازماندهی ، اشاعه و مدیریت صورت می گیرد.

#### منابع مرجع :

منابع مرجع منابعی هستند که کتابدار مرجع به کمک آنها به سؤال مراجعه کننده پاسخ می گوید. منابع مرجع در بخش مرجع نگهداری می شود و امانت هم داده نمی شود.

اینک منابع مرجع می تواند به صورت چاپی ، الکترونیکی ، دیداری ، شنیداری و چند رسانه ای در اختیار مراجعه کنندگان قرار گیرد.

#### کتابدار مرجع :

کتابداری را که در بخش مرجع ، مسئول ارائه خدمات مرجع به مراجعه کنندگان است را کتابدار مرجع می نامند.

#### خدمات مرجع :

مجموعه خدمات گوناگونی را که در بخش مرجع به مراجعه کنندگان حضوری و غیر حضوری ارائه می شود را خدمات مرجع می گویند.

عموماً "منابع مرجع به چند گروه تقسیم می شود:

۱- منابع مرجع دسته اول ، یا ردیف اول که به طور مستقیم پاسخ سؤال مراجعه کننده را در اختیار او قرار می دهد. مانند دائرةالمعارف ، فرهنگ نامه ها، شرح حالها و غیره .

۲- منابع دسته دوم یا منابع ردیف دوم به منابعی گفته می شود که به جای پاسخگویی مستقیم به سؤال، مراجعه کننده را به منابع دیگر ارجاع می دهند ، مانند برگه دان کتابخانه ها ، کتابشناسی ها نمایه ها ، مقاله ها ، چکیده نامه ها و فهرستگان ها.

تقسیم بندی منابع مرجع از دیدگاه اطلاعاتی :

۱- منابع اطلاعاتی دست اول : منظور یافته های اطلاعاتی هستند که پژوهشگر در پایان پژوهش و تحقیق خود به آن دست یافته و یا برای اولین بار توسط فردی نخبه ( نخبگان علمی ، سیاسی ، هنری ، فرهنگی و غیره ) در اختیار عموم قرار گرفته و در نشریات و دیگر رسانه های گروهی منتشر شده است .

۲- منابع اطلاعاتی دست دوم : وقتی که فردی بر اساس نوشته ها یا گفته های دیگران و یا با استفاده از مقالات و دست نوشته های دیگران اقدام به تدوین مقاله یا کتاب می کند ، در واقع مجموعه ای از اطلاعات دست دوم را فراهم می آورد.

از دیدگاه شکلی می توان منابع مرجع را به این گروه ها تقسیم کرد :

۱- کتابشناسیها که از دیدگاههای مختلف و متعددی می توان آن ها را تقسیم کرد.

۲- دائرةالمعارف ها

۳- فرهنگ ها

۴- زندگی نامه ها

۵- منابع جغرافیایی

۶- منابع مرجع فوری

منابع مرجع فوری شامل موارد زیر است :

- سالنامه ها و سالنها
- دستور نامه ها
- راهنما ها
- گاهنامه ها و غیره

ارائه خدمات مرجع به طور کلی به عوامل زیر بستگی دارد:

۱- شرایط جامعه

۲- سنتهای جامعه

۳- انواع مراجعه کنندگان

۴- نوع و اندازه کتابخانه

۵- منابع کتابخانه

۶- فلسفه کتابدار ، با در نظر گرفتن سازمان و مدیریت کتابخانه

خدمات مرجع را از نظر سطح ، نوع و گروه های خاص کاربران می توان گروه بندی کرد .

**گروه بندی خدمات مرجع از نظر سطح :**

۱- خدمات مرجع حداقل

۲- خدمات مرجع حداکثر

۳- خدمات مرجع حد متوسط

**گروه بندی خدمات مرجع از نظر نوع :**

۱- خدمات مرجع پایه

۲- خدمات مرجع عمومی

۳- خدمات مرجع موردی

**خدمات مرجع پایه عبارتند از :**

- کمک در استفاده از منابع مرجع
- کمک در استفاده از برگه دان کتابخانه
- کمک و یاری در یافتن اسناد و مدارک
- تهیه و تدارک اطلاعات تخصصی
- تهیه و تدارک اطلاعات عمومی

## خدمات مرجع از نظر ارائه خدمات به گروه های خاص :

۱- ارائه خدمات به گروه های خاص قومی

۲- ارائه خدمات به بیسوادان و نوسوادان

۳- ارائه خدمات به معلولان

## خدمات اطلاع رسانی و آموزش کاربران :

- بانکها و پایگاه های اطلاعاتی در قالب سی دی رام

- بانکها و پایگاه های اطلاعاتی پیوسته از جمله متن کامل مقالات و نشریات الکترونیکی

- اطلاعات الکترونیکی

## سیستم خبره :

یک سیستم خبره جهت جایگزین شدن با یک فرد متخصص در یک عملیات تعریف شده ، طراحی می شود. بدین معنی که اولاً" این سیستم در حوزه تخصصی خود دارای اطلاعات است ، ثانیاً" در مواردی ضروری می تواند سؤالات مطرح شده را پاسخ دهد.

سیستم خبره برای گرفتن دانش و رفتار یک فرد خبره در زمینه موضوعی خاص طراحی شده است ، به همین خاطر سیستم های مبتنی بر دانش نیز نامیده شده اند.

یک سیستم خبره معمول از یک موتور استنتاج و یک پایگاه دانشی ساخته شده است .

برای ایجاد یک سیستم خبره ، ارتباط بین برنامه نویسان کامپیوتری و خبرگان موضوعی مورد نیاز است . این برنامه نویسان کامپیوتری مهندسان دانش نامیده می شوند.

## مسائل و مشکلات سیستم های خبره :

هر سیستم خبره برای یک کار خاص طراحی می شود و نمی توان از تجربیات به دست آمده برای استفاده و تطبیق در کار های دیگر استفاده کرد.

انعطاف پذیری کم و نبود قابلیت اطمینان ، از دیگر مشکلات سیستم های خبره است .

## کاربرد سیستم های خبره در کتابخانه ها :

- اولاً" دانش خبرگان را در همه زمانها برای کاربران کتابخانه در دسترس قرار دهد.
- ثانیاً" وقت کتابدار را در پاسخ به سؤالات روزمره و تکراری هدر ندهد.



## عوامل هوشمند Intelligent Agents :

۱- عوامل فیلتر گذار ( مرور کردن اطلاعات )

۲- عوامل جستجو

### وظایف یک عامل هوشمند :

۱- یاد گرفتن تایپ ، فاصله گذاری ، غلط های املایی هر کاربر و تصحیح آنها

۲- یادگیری واژگان استفاده شده به وسیله هر کاربر

۳- انجام دادن تعویض لغات و اصطلاحات بر اساس نتایج بازیابی گذشته و بازخور کاربر

۴- بازبینی منابع اطلاعاتی که به وسیله هر کاربر استفاده می شود و بررسی برای افزایش علایق جدید مرتبط در آن منابع و منابع جدیدی که حوزه های مورد علاقه کاربر را در بر دارد.

۵- نگهداری یک سابقه به روز از نیازهای اطلاعاتی کاربر ، به وسیله حوزه های اطلاعاتی قدیمی ، بر اساس تاریخ آخرین استفاده .

### یک سیستم خبره مرجع باید نقش میانجی انسانی را بازی کند که مراحل زیر را در بر می گیرد:

۱- ترجمه نیازهای کاربر به شکل پذیرفته شده

۲- تأیید و در صورت لزوم تعریف دوباره نیازهای کاربر

۳- شناخت سطحی مناسب از اطلاعات مورد نیاز

۴- به دست آوردن ابزار مناسب برای پیدا کردن اطلاعات با استفاده از یک راهبرد مناسب

۵- اصلاح اطلاعات به دست آمده

۶- تکرار مراحل فوق در صورت لزوم

در حال حاضر بسیاری از سیستم های خبره در بخش مرجع کتابخانه ها به کار می روند.

## فصل دوم : مدیریت دانش و خدمات مرجع

### مدیریت دانش :

تعریف مدیریت دانش به زبان ساده عبارت است از : سازمان دادن برای دانستن است .

نظامهای مدیریت دانش فواید دیگری نیز برای کار مرجع در بردارند .

این نظامها می توانند در موارد ذیل به کتابدار مرجع کمک کنند :

- گرد آوری ، سازماندهی و ثبت نظامند دانش نهان و عیان کتابداران متخصص مرجع
- پاسخ دهی سریع و کار آمد به پرسشهای متداول
- بازننگری و اصلاح فهرست بهترین منابع مرجعی که برای پاسخگویی به یک سؤال خاص مرجع در نظر گرفته شده اند .
- بهینه سازی دسترسی کاربران به اطلاعات – استفاده کنندگان از یک سیستم مدیریت دانش می توانند خودشان کاوش کنند و سیستم آنها را به سمت پاسخ سؤال و یا معرفی منابع بدون نیاز به کمک کتابدار رهنمون کند .

سیستم های خبره از ۳ قسمت تشکیل شده اند:

۱- یک واحد رابط

۲- یک مجموعه دانش محور

۳- موتور استنتاج

## فصل سوم : خدمات مرجع الکترونیکی

### خدمات مرجع الکترونیکی :

خدمات مرجع الکترونیکی عبارت است از خدمات مرجعی که با استفاده از امکانات و ابزار های الکترونیکی و شبکه های رایانه ای به مراجعه کنندگان ارائه می شود.

### خدمات مرجع از طریق پست الکترونیکی :

خدمات مرجع از طریق پست الکترونیکی خدماتی است که در آن ، کاربر کتابخانه درخواست اطلاعات را از طریق پست الکترونیکی ارسال می کند.

### ویژگی های خدمات مرجع از طریق پست الکترونیکی عبارتند از :

- بیشتر کتابخانه ها فرم الکترونیکی تهیه کرده اند که مراجعه کننده ، در خواست اطلاعات خود را در آن فرم می نویسد و از طریق پست الکترونیکی ارسال می کند.
- خدمات مرجع از طریق پست الکترونیکی از طریق وب سایت وب کتابخانه ارائه می شود.
- خدمات مرجع از طریق پست الکترونیکی باعث ایجاد رابطه اینترنتی بین کاربران و متخصصان می گردد.

### مزایای خدمات مرجع از طریق پست الکترونیکی را می توان به شرح زیر بیان کرد :

- ۱- کتابدار و کاربر پاسخ ها و پرسش ها را با دقت و تفصیل بیشتر و با حوصله کافی ارائه خواهند داد
- ۲- کتابداران وقت زیادی برای تحلیل پرسش ها خواهند داشت
- ۳- با توجه به اینکه در این شیوه ، کتابداران پاسخ کتبی ارائه می دهند، در کار خود دقیق تر هستند.
- ۴- با ذخیره کردن پاسخ ها ، در مواقع دیگر هم از آن ها استفاده کنند
- ۵- کاربران می توانند از هر مکان و در هر زمان ، پرسش خود را مطرح کنند.
- ۶- نامه هایی را که به پاسخ تخصصی و دانش ویژه نیاز دارند، می توان برای کتابداران خاص یا متخصصین امر ارسال کرد.
- ۷- افراد می توانند پرسش های خود را به چندین مرکز ارسال کنند.

## کاربرد پست الکترونیکی در کتابخانه ها و مراکز اطلاع رسانی :

پست الکترونیکی به عنوان یکی از ابزارهای مهم ارتباطی دارای مزایا و کاربردهای ویژه ای در کتابخانه ها و مراکز اطلاع رسانی می باشد :

- امکان ارسال هم زمان پیام برای چند نفر از طریق گروه های مباحثه و فهرست های پستی
- امکان عضویت در گروه های مباحثه ، گروه های خبری و فهرست های پستی
- سرعت بالای ارسال و دریافت نامه ها و پیام برای گروه ها و اشخاص
- عضویت در خدمات آگاهی رسانی جاری و اشاعه اطلاعات گزیده ناشران و ....
- ارسال درخواست های امانت بین کتابخانه ای و سایر امکانات اداری مربوط به درخواست و ارسال منابع اطلاعاتی
- ایجاد گروه های مباحثه و خبری در موضوعات مختلف کتابخانه ، کتابداری و اطلاع رسانی و تبادل تجربیات علمی و کاربردی
- پایین بودن هزینه
- حذف محدودیت های زمانی و جغرافیایی
- دریافت بعضی از مجلات الکترونیکی رایگان از طریق پست الکترونیکی
- تولید بولتن خبری و یا فهرست تازه های کتابخانه و ارسال آن از طریق فهرست های پستی به مشترکین
- استفاده از پست الکترونیکی برای دسترسی به بسیاری از اطلاعات موجود در اینترنت

## کاربرد پست الکترونیکی در واحد مرجع :

در ابتدایی ترین طریقه ارائه خدمات مرجع از طریق پست الکترونیکی پیوندی در سایت اصلی کتابخانه یا مؤسسه ای که می خواهد خدمات مرجع از طریق آن ارائه شود ، قرار داده می شود که با انتخاب آن یک صفحه ساده پست الکترونیکی باز می شود و کاربر می تواند نیازهای اطلاعاتی خود را در آن صفحه وارد کند.

## مشکلات روند خدمات مرجع از طریق پست الکترونیکی عبارتند از :

- نداشتن دسترسی به پست الکترونیکی
- نیاز به ارتباط دو سویه و دو جانبه در برخی از پرسش ها
- عدم امکان مشاهده حرکات بدنی که در مصاحبه مرجع از اهمیت خاصی برخوردار است
- ممکن است بعد از ارسال پرسش ، کاربر مجبور باشد مطلبی را به آن اضافه کند ، که در صورت تکرار ، این کار باعث اتلاف وقت می شود.

## خدمات مرجع بیدرنگ :

خدمات مرجع بیدرنگ یا خدمات مرجع زنده خدماتی است که در طی آن ، کاربر کتابخانه با استفاده از نرم افزار مخصوص بخش مرجع کتابخانه که در سایت کتابخانه قرار داده شده ، پرسش مرجع خود را مطرح می کند و کتابدار مرجع نیز به طور هم زمان پاسخ ممکن را ارائه می دهد.

به عنوان مثال دانشگاه میشیگان در سال ۱۹۹۷ خدمات مرجع بیدرنگ خود را با یک کتابدار شروع کرد و در آن از نرم افزاری به نام " کیو آرسی " استفاده می کرد.

خدمات مرجع کتابخانه در صورتی کار آیی خواهد داشت که مثل ماشین های کاوش به راحتی از راه دور در دسترس مراجعین قرار گیرد.

## یک نمونه خدمات :

برروی مرور گر دکمه ای چشمک می زند که با عنوان " در صورت نیاز فوری به کتابدار این دکمه را فشار دهید " مشخص گردیده .

## شیوه های خدمات مرجع بیدرنگ :

خدمات مرجع بیدرنگ به شیوه های مختلفی ارائه می شود ، توجه به این شیوه ها ، نوعی حرکت تعاملی را نیز نشان می دهد که خدمات مرجع را به سوی تعامل بیشتر بین کاربر و کتابدار به پیش می برد.

## یکی از شیوه های خدمات مرجع بیدرنگ عبارت است از : فناوری چت :

این فناوری از سال ۱۹۹۵ در ارائه خدمات مرجع الکترونیکی به کار گرفته شده است . کاربران با اجرای برنامه ویژه ای که در بخش مرجع کتابخانه ها قرار داده شده ، با کتابدار ارتباط برقرار می سازند و پرسش خود را مطرح می کنند ، کتابدار نیز به طور همزمان پاسخ و منابع موجود را ارائه می دهد.

## استانداردهای خدمات مرجع الکترونیکی :

تدوین و ارزیابی استانداردهای خدمات مرجع الکترونیکی به یکی از حوزه های پژوهشی تحت حمایت مرکز کتابخانه کامپیوتری آنلاین " OCLC " و " فدراسیون کتابخانه های دیجیتال " تبدیل شده و این سازمان ها از تحقیقات در زمینه تعیین استاندارد های خدمات مرجع الکترونیکی پشتیبانی می کنند.

تعیین استاندارد برای فناوری مرجع الکترونیکی ، مواردی از قبیل تعیین ویژگی های ارائه ، اشتراک و ذخیره داده های مربوط به تعاملات مرجع الکترونیکی ( تبادیل پرسش و پاسخ ) را شامل می شود.

## توسعه خدمات مرجع الکترونیکی :

- کتابخانه باید جامعه استفاده کننده را تعریف و تعیین کند و این سیاست را در وب سایت خدمات الکترونیکی و یا در جای دیگر که جامعه استفاده کننده به آن دسترسی داشته باشند منتشر سازد.
- سطح خدمت فراهم شده ، باید مشخص و اعلام شود تا کارکنان و مراجعین وظایف خدمات را درک کنند.
- راهنماها باید سؤالاتی را که خارج از سطوح خدماتی هستند را مشخص و همچنین شیوه پاسخگویی در چنین مواردی را مشخص کنند.
- قبل از شروع ارائه خدمات باید در مورد این که آیا خدمات ، تحویل مدرک را نیز شامل می شود ، یا نه و آیا مشتریان برای این سرویس پولی پرداخت خواهند کرد ، یا نه ، تصمیم گیری شود.
- پارامتر های زمانی بایستی تعیین و به استفاده کنندگان و کارکنان ، اعلام شود.
- باید زمان هایی که در آن کارمند مرجع الکترونیکی در محل کار خود حاضر است تا در همان زمان پاسخگویی مراجعین باشد، مشخص شوند.
- مرجع الکترونیکی به بسیاری از همان ارتباطات و مهارت های فردی کارکنان کتابخانه نیاز دارد. حضور نداشتن یک مراجعه کننده ( به صورت فیزیکی ) در محل کتابخانه و مدل های متفاوت ارتباط ، ممکن است مهارت ها ، تلاش ها یا آموزش بیشتری را بطلبد تا خدمات با کیفیت برابر با خدمات مرجع رو در رو ارائه شود.
- کارکنان باید مهارت های تخصصی خود را برای ارائه موفق خدمات مرجع عرضه کنند.
- راهنماهای استاندارد خدمات مرجع باید ارائه شود . ( مانند مصاحبه ، رد و بدل کردن سؤالات بین خدمات و غیره )
- کارکنان باید تمرینات ارتباط اجتماعی را انجام تا خدمات مرجع مؤثر و کار آمد تری را ارائه دهند.
- از کارکنان باید خواسته شود که مهارت هایشان در استفاده مؤثر و کارآمد از و سایل ارتباطی پیوسته ، را به نمایش بگذارند و ارائه دهند.

## فصل چهارم : کتابخانه الکترونیکی

### فرآیند تغییر کتابخانه های سنتی :

با ظهور و ابداع رایانه در سال ۱۹۶۹ کتابخانه ها در صدد مکانیزه نمودن فرآیندهای کاری خود برآمدند.

همگام با ظهور فهرست های رایانه ای گام های بعدی به سوی خودکار سازی سایر فرآیندهای کتابخانه ای نیز آغاز شد. خودکار سازی و رایانه ای کردن فرآیند امانت گام بعدی در استفاده از رایانه در امور کتابخانه ای

است . در ابتدا ، این فرآیند فقط برای برقراری ارتباط ساده بین شماره عضویت کاربر و شماره ثبت کتاب را شامل می شد ، اما با رواج و توسعه بیش از پیش فناوری های رایانه ای ، امکان ارتباط بین پایگاه های مختلف کتابخانه ای فراهم آمد و در نتیجه خودکار سازی برخی از فرآیند های امانت مثل یادداشت های دیر کرد نیز فراهم آمد.

### کمبود منابع مالی مانع جدی کتابخانه های الکترونیکی :

اینترنت ، به عنوان یک شبکه قابل انعطاف ارزان و جهانی ، عامل مؤثر در توسعه کتابخانه های الکترونیکی است .

### رایانه های موجود در شبکه سه عملکرد اصلی دارند :

۱- کمک به کاربران برای ارتباط با کتابخانه

۲- ذخیره منظم اطلاعات جهت نگهداری بلند مدت

۳- خدمات جست و جو و مکان یابی اطلاعات با ارائه فهرست ها و نمایه ها برای ایجاد کتابخانه

الکترونیکی

### تعریف کتابخانه الکترونیکی :

تعاریف متعددی از کتابخانه الکترونیکی ارائه شده است و در بسیاری موارد از آن با اصطلاحات کتابخانه الکترونیکی ، دیجیتالی و بدون دیوار نام برده شده است ، ولی باید گفت که این عناصر به شکلی مشترک در تعاریف متعدد به کار رفته است .

در تعریف کتابخانه الکترونیکی :

دابن کرافورد می گوید :

به موضوع خاصی در زمینه اطلاعات و چگونگی الکترونیکی کردن آن ، دستکاری ، دستیابی ، اتصال ذخیره حفاظت و سهیم بودن در آن با بقیه جهان خوش آمدید. ما می توانیم کل این مجموعه را کتابخانه الکترونیکی بنامیم .

ویلیامز آرم متخصص کتابخانه الکترونیکی می گوید :

کتابخانه الکترونیکی مجموعه ای سازمان یافته از اطلاعات - با خدمات وابسته - است که بر روی یک شبکه و با ساختار الکترونیکی در دسترس قرار دارد.

فدراسیون کتابخانه الکترونیکی می گوید :

کتابخانه های الکترونیکی تشکیلاتی هستند که برای گزینش ، ساختار بندی ، ایجاد امکان دسترسی به منابع فکری ، تفسیر ، حفظ انسجام و تداوم گرد آوری آثار قومی ، منابعی - شامل کارکنان متخصص - را فراهم می آورند . به گونه ای که این منابع برای اجتماعات یا مجموعه ای از جوامع به سهولت در دسترس قرار داشته باشد.

کتابخانه الکترونیکی عبارت است از مجموعه ای سازمان یافته از اطلاعات همراه با خدمات مربوطه که در آن اطلاعات در قالب های الکترونیکی ذخیره گردیده و از طریق شبکه قابل دسترسی است .

کتابخانه الکترونیکی عبارت است از مجموعه ای سازمان یافته از اطلاعات همراه با خدمات مربوطه که در آن اطلاعات در قالب های الکترونیکی ذخیره شده و از طریق شبکه قابل دسترسی است .

کتابخانه الکترونیکی ، مجموعه های گوناگونی از اطلاعات را برای استفاده کاربران در بر دارند ، این کتابخانه ها از نظر اندازه از کوچک تا عظیم گسترش دارند .

آنها قادر به استفاده از هر نوع وسیله رایانه ای و هر نوع نرم افزار مناسب هستند .



## کتابخانه الکترونیکی چیست ؟

بدعت گذاران کتابخانه الکترونیکی نظیر NASA ، ARPA ، NSF درباره کتابخانه الکترونیکی در سال ۱۹۹۴ اظهار داشته اند :

منابع اطلاعاتی که از طریق اینترنت به آنها دست می یابیم جزئی جدا نشدنی از کتابخانه های الکترونیکی محسوب می شوند.

آنها شامل کتاب های مرجع ، کتاب ها ، نشریه ها ، روزنامه ها ، راهنما های تلفن، آهنگ ها، آواها، تصویر ها ویدئو کلیپ های ضبط شده ، داده های علمی و اطلاعات سازمان های خصوصی نظیر گزارش بورس سهام و خبرنامه های اختصاصی هستند.

### ویژگیهای کتابخانه های الکترونیکی :

۱- کتابخانه های الکترونیکی ، وجه الکترونیکی کتابخانه های سنتی هستند که مجموعه های الکترونیکی و سنتی ، یعنی مواد الکترونیکی و کاغذی را در بر می گیرند.

۲- کتابخانه های الکترونیکی هم به مهارت کتابداران احتیاج دارند و هم مهارت کار آمد رایانه .

۳- در کتابخانه های الکترونیکی فاصله میان جویندگان و پدید آورندگان تقریباً" به صفر می رسد.

۴- کتابخانه الکترونیکی علاقه مندان به موضوعات مشترک را از سراسر کشور کنار هم می آورد و به آنها امکان می دهد که در هر جا در موضوعات مورد علاقه شان به بحث بنشینند و این موضوع باعث می شود که فاصله میان نخبگان کشور از بین برود و آنها بتوانند مسائل خودشان را با دانشمندان و نخبگان جهان مطرح کنند.

۵- در کتابخانه های الکترونیکی ، امکان استفاده هم زمان از یک منبع خاص برای استفاده کنندگان فراهم است .

۶- برخلاف کتابخانه های فیزیکی ، برای کتابخانه های مجازی دو مقوله زمان و مکان ، هر دو کلمات بی مفهومی هستند.

چنانچه کتابداران بخواهند به عنوان واسط اطلاعاتی بین کاربران و منابع اطلاع رسانی در محیطی الکترونیک مانند اینترنت ، ایفای نقش کنند ، به مهارت های خاصی نیاز دارند ، این مهارت ها را می توان در ۷ سطح زیر رده بندی کرد:

۱- شناخت خدمات و منابع اطلاع رسانی

۲- درک ساختار اطلاعات

۳- تجزیه و تحلیل نیاز به اطلاعات

۴- بازیابی اطلاعات

۵- ارزیابی اطلاعات

۶- مدیریت اطلاعات

۷- کمک به ذخیره دانش

### سیر تحول کتابخانه ها : از سنتی تا مجازی در یک نگاه

ویژگی	نوع کتابخانه
موجودی به صورت چاپی که هیچ کار رایانه ای روی آن انجام نشده است	کتابخانه سنتی
خودکار کردن فعالیت های کتابخانه ، فهرست نویسی ، امانت ، فراهم آوری ... رایانه ای ، موجودی به طور عمده به شکل چاپی با تعدادی محدود منابع	کتابخانه خودکار
فعالیت ها کاملاً "خودکار، شبکه دیسک فشرده، منابع به دو شکل الکترونیکی و متعارف است	کتابخانه الکترونیکی
کتابخانه بدون دیوار با تأمین دستیابی به منابع ، کتابخانه بدون منابع	کتابخانه مجازی

### مشکلات ایجاد کتابخانه های الکترونیکی :

- معماری ساختارهای فنی : برای آماده سازی مواد و منابع الکترونیک ، کتابخانه ها باید معماری خود را اصلاح کنند و ارتقاء بخشند.
- ایجاد مجموعه های الکترونیکی : از قبیل تبدیل نسخه های چاپی منابع به شکل الکترونیکی ، سرمایه گذاری و تهیه منابع مالی مورد نیاز برای رفع کمبود نیرو های متخصص و حرفه ای .
- کنترل و بررسی: مجموعه ها قبل از تبدیل آنها به شکل الکترونیکی بایستی کنترل و بررسی گردند.
- حق مؤلف : بزرگترین مشکل برسر راه کتابخانه های الکترونیک است .

اهداف رشد کتابخانه های الکترونیکی :

این اهداف عبارتند از :

- شناسایی روش های عملی برای غلبه بر مشکلاتی که در راه تبدیل کتابخانه های سنتی به الکترونیکی به وجود می آیند.
- شناخت نقش کتابخانه های الکترونیکی برای ایجاد دانش جهانی قابل دسترس
- بررسی امکانات مشارکت خصوصی برای تأمین منابع مالی و انسانی
- جست و جوی راه هایی برای ایجاد رشد بدون وقفه کتابخانه های الکترونیکی

الزامات مهاجرت کتابخانه ای از شکل سنتی به صورت الکترونیکی :

برای تغییر سیستم ها از شکل سنتی به صورت الکترونیکی لازم است تدابیری اتخاذ شود . از جمله نکاتی که بایستی مورد توجه قرار گیرند ، عبارتند از :

- پیش بینی ترافیک در شبکه ها و بهره گیری از انواع برنامه هایی که از طریق شبکه قابل دسترسی هستند.
- تنظیم روش هایی به منظور ادامه کار چنانچه بخشی یا تمام شبکه از کار بیفتد.

جلوگیری از استفاده غیر مجاز :

حمایت از حقوق مالکیت معنوی و مطالب محرمانه دو حوزه ای هستند که " سازمان جهانی مالکیت معنوی یا WIPO " به طور جدی با آن مواجه می باشد.

در دسامبر سال ۱۹۹۶ کنفرانس دیپلماتیک دو عهد نامه منعقد کرد:

- عهد نامه WIPO در زمینه حق مؤلف
- و عهد نامه WIPO در زمینه اجرا و ضبط صدا

## سرویس های کتابخانه الکترونیکی

- ۱) سرویس جست و جو :
- ۲) سرویس بازیابی اطلاعات الکترونیکی
- ۳) سرویس Browse
- ۴) نتایج جست و جوی ذخیره شونده
- ۵) سرویس تصدیق اصالت
- ۶) ثبت نام در سیستم
- ۷) سرویس افزایش اعتبار
- ۸) سرویس های مدیریتی بخش الکترونیکی
- ۹) سفارش سند

### ۱) سرویس جست و جو :

این سرویس به کاربر کمک می کند که به چهار روش ، جست و جودر سیستم را انجام بدهد:

#### روش جست و جوی هدایت شونده توسط کاربر :

در این روش کاربر خود انتخاب می کند که سیستم در میان چه مجموعه ( collection ) هایی باید جست و جو انجام دهد. سپس سیستم جست و جو را انجام داده ، امکان دسترسی کاربر به اشیای الکترونیکی مربوطه را فراهم می کند ، علاوه بر مجموعه ، کاربر می تواند پارامتر های دیگری از قبیل تعداد پاسخ ها و سایر پارامتر ها را مشخص کند.

#### روش جست و جوی بدون هدایت :

در این روش این سیستم است که برحسب شرایط ، بخشی یا تمام مجموعه ها را مورد جست و جو قرار داده ، نتایج را در اختیار کاربر قرار می دهد.

## روش جست و جوی مبتنی بر پروتکل Z۳۹.۵۰:

در این روش جست و جو ، بر اساس پروتکل بازیابی اطلاعات Z۳۹.۵۰ ، جست و جو بر روی سیستم دیگری انجام می گیرد و نتایج به کاربر ارائه می گردد.

سیستم دوم که بر روی آن جست و جو انجام می گیرد باید Z۳۹.۵۰ را پیاده سازی کرده باشد.

### روش جستجوی Full Text :

در این روش جست و جو ، محتویات الکترونیکی سیستم بر اساس یک کلمه کلیدی مورد جستجو قرار می گیرند و اقلام اطلاعاتی که آن کلمه کلیدی را دربر دارند ظاهر می شوند و کاربر می تواند آن اقلام را بر مبنای پرداخت وجه مشاهده کند. دیدن عنوان سندهای یافته شده نیاز به پرداخت پول ندارد ، بلکه دیدن محتوا به همراه کلمه کلیدی مورد نظر بها دارد.

### (۲) سرویس بازیابی اطلاعات الکترونیکی :

این سرویس این امکان را فراهم می کند که بر اساس تصدیق اصالت مجدد ( و پرداخت وجه ) کاربر به محتویات الکترونیکی اطلاعات دست یابد ، که این دستیابی می تواند به صورت مشاهده یا Download کردن محتویات الکترونیکی باشد.

### (۳) سرویس Browse :

در این سرویس کاربر می تواند بر روی فیلدهای مختلف سیستم Browse کند ، بدین معنا که بر حسب فیلد خاصی بصورت صعودی یا نزولی اطلاعات در ابتدا برای کاربر به نمایش در آورده می شود . سپس او قلم اطلاعاتی مربوطه را انتخاب کرده به مشاهده متن آن می پردازد.

#### ۴) نتایج جست و جوی ذخیره شونده :

به کمک این سرویس کاربر می تواند نتایج جست و جوی خاصی را ذخیره کند و سپس در مواقع بعدی با استفاده از این نتایج جست و جو ، جست و جو در سیستم را پی بگیرد.

#### ۵) سرویس تصدیق اصالت :

بر اساس این سرویس کاربر برای ورود به سیستم وامکان استفاده از فضای تخصص یافته به او توسط سیستم و استفاده از سایر امکانات سیستم ، به سیستم Login می کند.

#### ۶) ثبت نام در سیستم :

این سرویس دو بخش دارد، یکی ثبت نام در سیستم برای استفاده از امکانات مجانی سیستم و دیگری ثبت نام در سیستم برای استفاده از امکانات پولی سیستم .

#### ۷) سرویس افزایش اعتبار :

این سرویس این امکان را فراهم می سازد که کاربر اعتبار موجود خود در سیستم را افزایش دهد.

#### ۸) سرویس های مدیریتی بخش الکترونیکی :

این سرویسها مخصوص مدیر سیستم است و عبارتند از :

- ورود اطلاعات تزاروس سیستم
- تعریف نقش برای کتابداران سیستم
- ایجاد مخزن اطلاعاتی الکترونیکی
- ایجاد مجموعه
- ایجاد تزاروس
- ایجاد فرا داده یا متا دیتا

## ۹) سفارش سند :

در این روش بعد از اینکه کاربر جست و جویی را در میان اطلاعات الکترونیکی سیستم انجام داد ، از سیستم درخواست می کند که قلم اطلاعاتی خاصی را در اختیار او قرار دهد .

### سرویس های عمومی :

- ۱- سرویس گزارش ها
- ۲- برقراری ارتباط با سیستم
- ۳- نمایشگاه ها
- ۴- راهنما
- ۵- اطلاعات
- ۶- انتشارات
- ۷- اتاق مطالعه آزاد
- ۸- جست و جوی بر روی اینترنت
- ۹- نقشه سایت
- ۱۰- جست و جو بر روی سایت

### چالش های ایجاد کتابخانه الکترونیکی :

- چالش اول : فراهم آوردن فناوری پیشرفته به منظور تبدیل مطالب و مواد رایج امروزی به داده های الکترونیکی
- چالش دوم : ابداع برنامه ها و لوازمی که قادر باشند جست و جو و بازیابی را به نحوی انجام دهند که از عهده فهرست نویسی مختصر یا توصیفی اطلاعات بر آیند.
- چالش سوم : ابداع برنامه ها و لوازمی که قادر باشند شیوه های فهرست نویسی توصیفی را از طریق جلب همکاری افراد داوطلب اعتلا بخشند.
- چالش چهارم : وضع کردن پروتکل ها و استانداردهایی که موجب تسهیل در امر ارتباط متقابل میان کتابخانه های الکترونیکی پراکنده و دور از هم گردد.

## اصولی که در طراحی باید رعایت شوند:

- ۱- چار چوب فنی چنین سازه ای باید منطبق با چارچوب قانونی و اجتماعی باشد.
- ۲- برای درک بهتر مفهوم های مربوط به کتابخانه الکترونیکی نیاز مبرمی به پدید آوردن اصطلاح های فنی و تعبیر نامه های استاندارد ها می باشد.
- ۳- پی ریزی چنین ساختاری باید بدون در نظر گرفتن محتویات و مطالب قابل ذخیره در کتابخانه های سنتی باشد.
- ۴- نام و شناسه ها از اجزای متشکله کتابخانه های الکترونیکی هستند.
- ۵- کتابخانه های الکترونیکی حیطة ای وسیع تر و فراتر از جایی برای ذخیره بیت ها هستند.
- ۶- اهداف کتابخانه های الکترونیکی نباید وابسته به یک نوع فناوری محدود و خاص باشند.
- ۷- مخازن اینگونه اطلاعات باید دارای امکانات قدرتمندی برای حفظ و نگهداری از ذخیره های خود باشند.

### نمونه CRADDL :

نمونه ساختار " کردل " به عنوان الگوی کتابخانه های الکترونیکی غیر متمرکز ، نمونه ساختاری نظام یافته بر پایه خدمات می باشد.

استانداردهای کتابخانه الکترونیکی که در پروژه یک کتابخانه الکترونیکی عرضه می گردد :

- وب برازر مشترک ( Common Web Browser )
- فرمت های گرافیکی به صورت Group\Fax,CGN,PNG,GIF,TIFF&JPEG
- فرمت های نوع تصویری و فیلمی اعم از ثابت و متحرک و سه بعدی به صورت VRML,Quick Time,GIF,&qA,Real Video,Vivi Active ,MPEG&AVI
- فرمت های سند و مدرکی به صورت SGML,HTML&XML
- توصیف منابع به صورت TEI,TemplatesUS,MARC.WHOIS++&Dubli Core
- سر صفحه ها ( Headers ) ، منابع در دسترس عام ( Open Sources ) و استاندارد های ویژه قلمرو های الکترونیکی
- تعیین هویت منابع به صورت SICI,DOI,PURL&URN
- خدمات حراست ، تأیید اعتبار و مبادلات پول الکترونیکی
- ایجاد استاندارد هایی برای تجارت الکترونیکی



نرم افزار Z۳۹.۵۰:

Z۳۹.۵۰ یکی دیگر از استانداردهای مهم کتابخانه ها می باشد ، در اینجا سیستم های ساخت فروشنده های مختلف باید بتوانند به سوابق یکدیگر دست یابند ، بنا براین پروتکلی به نام Z۳۹.۵۰ طراحی شد که سیستم های مختلف بتوانند برای جستجوی نتایج به سوابق MARC یکدیگر دسترسی یابند.

### ملاک های گزینش :

عمده ترین عواملی که در گزینش مواد و منابع الکترونیکی باید مورد توجه قرار گیرند ، عبارتند از :

- محتویات و کیفیت و روز آمد بودن
- وسائل و لوازم مورد نیاز نظیر سخت افزار و نرم افزار و تسهیلات شبکه ای
- ویرایش محصول ، نوع کاربرد اطلاعات از نظر وابسته بودن به شبکه و یا وابسته نبودن
- ظرفیت پاسخگویی به انبوه کاربران در حال استفاده
- شیوه های نظارت و کنترل روی کاربران نظیر رمز عبور
- توافق نامه ها و قرار داد های مربوط به پرداخت ها و کسب مجوز های لازم برای حق تألیف
- ویژگی های مربوط به پایگاه داده ها و موتورهای جست و جو ( پیوست ۵ ) و رابط های کاربران
- شیوه های متداول مربوط به بایگانی کردن اطلاعات

معمولاً" برای تبدیل مدارک کاغذی به نوع الکترونیکی مراحل زیر صورت می گیرد:

۱- اسکن برداری از مدارک

۲- تشخیص وضعیت ظاهری صفحه طی فرآیند الکترونیکی

۳- تشخیص کاراکترها

۴- اسکن برداری از تصویرها و جدولها

## تعریف کتابخانه مجازی :

- ۱- یک کتابخانه مجازی ، مجموعه سازمان یافته ای از پیوندها به اسناد ، نرم افزارها ، تصاویر ، پایگاه های داده ای و در یک شبکه و یا یک مجموعه ای از شبکه های رایانه ای می باشد.
- ۲- کتابخانه بدون دیواری که مجموعه آن فاقد منابع چاپی ، میکرو فیلم و یا هر نوع ماده ملموس دیگری است ولی منابع آن فرمت الکترونیکی قابل دسترس است .
- ۳- کتابخانه مجازی مفهومی است مبنی بر دسترسی از راه دور به محتوا و خدمات کتابخانه ها و دیگر مراکز اطلاع رسانی ، مرکب از مجموعه ای از منابع روز آمدو پر استفاده در قالب الکترونیکی و به همراه شبکه الکترونیکی دسترس پذیر به کتابخانه های جهانی و اطلاعات تجاری و منابع دانش به منظور کاربری و اشاعه آنها .

## ویژگیهای کتابخانه مجازی :

- ۱- حضور بروی اینترنت
- ۲- ارائه مجموعه ها به صورت مجازی
- ۳- ارائه مجموعه های مجازی
- ۴- سرویس دهی به مراجعه کننده مجازی در دنیای مجازی
- ۵- به کار گیری کارکنان مجازی

## معایب کتابخانه های مجازی شامل موارد زیر می باشد:

- کاربران برای محصولات مختلف باید رمز عبور های متفاوتی را به خاطر بیاورند
- حوزه پوشش و در دسترس بودن آرشیو ها ، اغلب محدود می باشد
- اغلب موارد انتقال و چاپ مطالب با مشکل رو به رو می باشد
- همه منابع به شکل الکترونیکی در دسترس نیستند
- محدودیت هایی برای چگونگی استفاده از محصولات از کارگزاری به کارگزاری دیگر وجود دارد
- کتابخانه های مجازی جهت قابل دسترس بودن ، به شبکه های کامپیوتری نیاز دارند
- کاربران در استفاده از کتاب ها راحت تر هستند

## کتابخانه های مجازی هوشمند :

نسل بعدی کتابخانه های مجازی " هوشمند" خواهند شد ، بدین معنا که این مراکز بر اثر

پرسشهای کاربران توسعه می یابند.

## تفاوت میان کتابخانه های مجازی با کتابخانه های الکترونیکی :

۱- منابع کتابخانه الکترونیکی شامل منابع چاپی و غیر چاپی از قبیل فیلم ، میکروفیلم ، میکرو

فیش ، انواع لوح فشرده و منابع اینترنتی است ، در حالی که کتابخانه مجازی ، فاقد هر نوع ماده

ملموس ، خواه چاپی و خواه غیر چاپی است .

۲- کتابخانه الکترونیکی درست مانند کتابخانه سنتی و خودکار دارای ساختمان ، تجهیزات ، مکان

و منابع فیزیکی است ، ولی کتابخانه مجازی ، کتابخانه ای بدون دیوار و بدون منابع و موجودیت

فیزیکی است .

۳- در کتابخانه الکترونیکی کاربر می تواند به محل کتابخانه مراجعه کند و از خدمات و منابع آن

چه به شکل چاپی و چه به شکل غیر چاپی استفاده کند. ولی مراجعه کاربر به کتابخانه مجازی معنا

ندارد و فقط از راه دور و از طریق شبکه ارتباطی می توانند به منابع آن دسترسی داشته باشند.

۴- در کتابخانه الکترونیکی ، کتابدار همان نقش واسط میان کاربران و منابع اطلاعاتی را دارد، ولی

در کتابخانه مجازی فقط میان تولید کننده اطلاعات و دریافت کننده آن ارتباط برقرار می شود و از

این رو به کتابدار در نقش واسط نیازی نیست .

۵- کتابخانه مجازی شرایطی را فراهم می آورد که چندین کتابخانه الکترونیکی در یک فضای

مجازی و در یک شبکه الکترونیکی منابعشان را به اشتراک می گذارند.

## فصل پنجم : کتابدار الکترونیکی

نیاز به کتابدار در مدیریت سیستم اطلاعات الکترونیکی :

گفته می شود که نیاز ، مادر اختراع می باشد، لذا پیدایش کتابخانه های الکترونیکی جهانی یا مراکز اطلاعات الکترونیکی در سراسر جهان به ایجاد یک عنوان شغلی تحت عنوان " کتابدار الکترونیکی " برای مدیریت منابع دانش انجامید .

نیاز به کتابداران الکترونیکی در جهت دستیابی به اهداف زیر مطرح گردیده است :

- مدیریت کتابخانه های الکترونیکی
- سازماندهی اطلاعات و دانش الکترونیکی
- انتشار اطلاعات الکترونیکی
- ارائه خدمات مرجع الکترونیکی و خدمات اطلاع رسانی الکترونیکی
- استخراج دانش از میان مخازن دانش

### مهارتهای کتابدار الکترونیکی

۱- اینترنت

۲- فرآیند رسانه الکترونیکی

۳- سیستم اطلاعات الکترونیکی پیوسته

آموزش از راه دور و کتابداران مرجع در عصر فناوری اطلاعات

- آموزش از راه دور بدون وجود کتابخانه های الکترونیکی ناممکن است
- به کار گیری کتابداران در آموزش از راه دور ، آشنایی آن ها را با فناوری رایانه ای اینترنتی به عنوان استفاده کنندگان با سابقه از این ابزارها فراهم می سازد

## پیوست ۵: نحوه عملکرد موتور های جستجو

وقتی جستجویی در یک موتور جستجوگر انجام و نتایج جستجو ارائه می شود، کاربران در واقع نتیجه کار بخش های متفاوت موتور جستجوگر را می بینند. موتور جستجو گر قبلاً" پایگاه داده اش را آماده کرده است و این گونه نیست که درست در همان لحظه جستجو، تمام وب را بگردد. بسیاری از خود می پرسند که چگونه امکان دارد که گوگل در کمتر از یک ثانیه تمام سایت های وب را بگردد و میلیون ها صفحه را در نتایج جستجوی خود ارائه کند؟ نه گوگل و نه هیچ موتور جستجوگر دیگری توانایی انجام این کار را ندارند. همه آنها در زمان پاسخ گویی به کاربران، تنها در پایگاه داده ای که در اختیار دارند به جستجو می پردازند و نه در وب. موتور جستجو گر به کمک بخش های متفاوت خود، اطلاعات مورد نیاز را قبلاً" جمع آوری، تجزیه و تحلیل می کند و آن را در پایگاه داده اش ذخیره می نماید و به هنگام جستجوی کاربر تنها در همین پایگاه داده می گردد.

بخش های مجزای یک موتور جستجو گر عبارتند از:

- Spider یا عنکبوت
- Crawler یا خزنده
- Indexer یا بایگانی کننده
- Database یا پایگاه داده
- Ranker یا سیستم رتبه بندی

الف - Spider یا عنکبوت

اسپایدر یا روبات ( Robot )، نرم افزاری است که کار جمع آوری اطلاعات مورد نیاز یک موتور جستجو گر را بر عهده دارد، اسپایدر به صفحات مختلف سر می زند، محتوای آنها را می خواند، اطلاعات مورد نیاز را جمع آوری می کند و آنها را در اختیار سایر بخش های موتور جستجو گر قرار می دهد.

کار یک اسپایدر ، بسیار شبیه به کار کاربران وب است . همانطور که کاربران ، صفحات مختلف را بازدید می کنند ، اسپایدر هم درست این کار را انجام می دهد ، با این تفاوت که اسپایدر کدهای HTML صفحات را می بیند، اما کاربران نتیجه حاصل از کنار هم قرار گرفتن این کدها را .

اسپایدر ، به هنگام مشاهده صفحات ، از خود بروی سرور ها رد پا برجای می گذارد . شما اگر اجازه دسترسی به آمار دید و بازدیدهای صورت گرفته از یک سایت و اتفاقات انجام شده در آن را داشته باشید ، می توانید مشخص کنید که اسپایدر کدام یک از موتور های جستجوگر صفحات سایت را مورد بازدید قرار داده اند.

اسپایدر ها کاربردهای دیگری نیز دارند ، به عنوان مثال عده ای از آنها به سایت های مختلف مراجعه میکنند و فقط به بررسی فعال بودن لینک های مربوط به آنها می پردازند و یا به دنبال آدرس پست الکترونیکی ( Email ) می گردند.

#### ب – Crawler یا خزنده

کراولر ، نرم افزاری است که به عنوان یک فرمانده برای اسپایدر عمل می کند و مشخص می کند که ، اسپایدر کدام صفحات را مورد بازدید قرار دهد.

در واقع کراولر تصمیم می گیرد که کدام یک از لینک های صفحه ای که اسپایدر در حال حاضر در آن قرار دارد، دنبال شود. ممکن است که همه آنها را دنبال کند ، بعضی ها را دنبال کند و یا هیچ کدام را دنبال نکند. کراولر ممکن است ، قبلاً" برنامه ریزی شده باشد که آدرس های خاصی را طبق برنامه در اختیار اسپایدر قرار دهد ، تا از آنها دیدن کند . دنبال کردن لینک های یک صفحه به این بستگی دارد که موتور های جستجو گر چه حجمی از اطلاعات یک سایت را می تواند در پایگاه داده اش ذخیره کند و همچنین ممکن است اجازه دسترسی به بعضی از صفحات به موتور های جستجو گر داده نشده باشد.

شما به عنوان دارنده سایت ، همان طور که دوست دارید موتور های جستجو گر اطلاعات سایت شما را با خود ببرند ، می توانید آنها را از بعضی از صفحات سایت خود دور کنید و اجازه دسترسی به محتوای آن

صفحات را به آنها ندهید. تنظیم میزان دسترسی موتور های جستجو گر به محتوای یک سایت توسط پروتکل Robots انجام می شود. به عمل کراولر ، خزش ( Crawling ) می گویند.

### ج – Indexe یا بایگانی کننده

تمام اطلاعات جمع آوری شده توسط اسپایدر در اختیار ایندکسر قرار می گیرد. در این بخش اطلاعات ارسالی مورد تجزیه و تحلیل قرار می گیرند و به بخش های متفاوتی تقسیم می شوند . تجزیه و تحلیل بدین معنی است که مشخص می شود اطلاعات از کدام صفحه ارسال شده است . کلمات در کجای صفحه قرار دارند و .....

در حقیقت ایندکسر ، صفحه را به پارامتر های آن خرد می کند و تمام این پارامتر ها را به یک مقیاس عددی تبدیل می کند تا سیستم رتبه بندی بتواند پارامتر های صفحات مختلف را با هم مقایسه کند. در زمان تجزیه و تحلیل اطلاعات ، ایندکسر برای کاهش حجم داده ها از بعضی کلمات که بسیار رایج هستند صرفنظر می کند . کلماتی نظیر a ، an ، the ، www ، is و ..... از این گونه کلمات هستند.

### د – Database ( پایگاه داده )

تمام داده های تجزیه و تحلیل شده در ایندکسر ، به پایگاه داده ارسال می گردد . در این بخش داده ها گروه بندی ، کد گذاری و ذخیره می شوند. همچنین داده ها قبل از آنکه ذخیره شوند ، طبق تکنیک های خاصی فشرده می شوند تا حجم کمی از پایگاه داده را اشغال کنند. یک موتور جستجو گر باید پایگاه داده عظیمی داشته باشد و به طور مداوم حجم محتوای آنرا گسترش دهد و البته اطلاعات قدیمی را هم به روز رسانی نماید . بزرگی و به روز بودن پایگاه داده یک موتور جستجو گر برای آن امتیاز محسوب می گردد. یکی از تفاوت های اصلی موتور جستجو گر در حجم پایگاه داده آنها و همچنین روش ذخیره سازی داده ها در پایگاه داده است .

بعد از آنکه تمام مراحل قبل انجام شد ، موتور جستجوگر آماده پاسخ گویی به سؤالات کاربران است  
کاربران چند کلمه را در جعبه جستجوی ( Serch Box ) آن وارد می کنند و سپس با فشردن Enter  
منتظر پاسخ می مانند.

برای پاسخگویی به درخواست کاربر ، ابتدا تمام صفحات موجود در پایگاه داده که به موضوع جستجو شده  
مرتبط هستند ، مشخص می شوند . پس از آن سیستم رتبه بندی وارد عمل شده ، آنها را از بیشترین  
ارتباط تا کمترین ارتباط مرتب می کند و به عنوان نتایج جستجو به کاربر نمایش می دهد.

حتی اگر موتور جستجوگر بهترین و کامل ترین پایگاه داده را داشته باشد ، اما نتواند پاسخ های مرتبطی را  
ارائه کند ، یک موتور جستجو گر ضعیف خواهد بود. در حقیقت سیستم رتبه بندی قلب تپنده یک موتور  
جستجو گر است و تفاوت اصلی موتور های جستجو گر در این بخش قرار دارد.

سیستم رتبه بندی برای پاسخ گویی به سؤالات کاربران ، پارامتر های بسیاری را در نظر می گیرد تا بتواند  
بهترین پاسخ را در اختیار آنها قرار دهد .

الگوریتم ، مجموعه ای از دستورالعمل ها است که موتور جستجو گر با اعمال آنها بر پارامتر های صفحات  
موجود در پایگاه داده اش ، تصمیم می گیرد که صفحات مرتبط را چگونه در نتایج جستجو مرتب کند.  
در حال حاضر قدرتمند ترین سیستم رتبه بندی را گوگل در اختیار دارد.

می توان با ادغام کردن اسپایدر با کراولر و همچنین ایندکسر با پایگاه داده ، موتور جستجو گر را شامل ۳  
بخش زیر دانست که این گونه تقسیم بندی هم درست می باشد :

- کراولر
- بایگانی
- سیستم رتبه بندی



## روش های جستجو در گوگل

اگر می خواهید کلمات تایپ شده در جستجو باشند ، آنها را بلافاصله و به صورت ساده در کنار هم بنویسید.

مانند: سایت طراحی وب

اگر می خواهید یکی از کلمات شما حداقل در نتایج باشد ، بین آنها OR بگذارید.

مانند: Web Design OR "tarahi"

اگر بخواهید یک کلمه حتماً باشد و همراه با یکی از دو کلمه دیگر بیاید ، از حالت پرانتز می توانید استفاده کنید.

مانند: Persian (yahoo OR blog)

این جستجو دو عبارت را جستجو می کند : Persian blog و Persian yahoo

اگر بخواهید کلمه خاص شما در title صفحه موجود باشد ، باید از کلمه کلیدی intitle استفاده کنید.

مانند: intitle: "web services"

اگر بخواهید کلمه ای که وارد کرده اید بخشی از آدرس url صفحه ای باشد از inurl استفاده کنید.

مانند: inurl: " Persian"

این جستجو تمام کلماتی که در آدرس سایت آنها کلمه Persian موجو باشد را پیدا می کند .

اگر می خواهید گوگل فقط بدنه متون را جستجو کند از کلمه intext استفاده کنید.

مانند: intext: "access ۲۰۰۰ problems"

اگر می خواهید که گوگل فقط در یک سایت خاص دنبال مطلب وارد شده شما بگردد ، بعد از نوشتن

موضوع جستجو از کلمه site به شکل زیر استفاده کنید:

Site:thissite.ir

این کد در ادامه جستجوی شما باعث می شود تا گوگل به دنبال مقاله مورد نظر شما در سایت

thissite.ir بگردد.

اگر شما یک سایت و یا وبلاگ دارید و می خواهید بدانید که چه کسانی لینک سایت شما را در سایتشان قرار داده اند از کلمه **link** با آدرس خودتان به شکل زیر استفاده کنید:

Link:www.mysite.com

اگر می خواهید دنبال فایلهایی خاص که با موضوع شما مرتبط هستند بگردید ، از کلمه **filetype** به صورت زیر استفاده کنید :

Programming vb filetype:pdf

این کد جستجوی فایل های pdf که درباره برنامه نویسی vb باشند را به شما نشان می دهد.

**یک نکته در جستجو :**

گوگل حد اکثر ۱۰ کلمه را برای جستجو می پذیرد و اگر بیشتر باشد از بقیه کلمات صرف نظر می کند. گوگل کلمات ربطی ساده مانند **of ، the ، and ، an ، a** را نادیده می گیرد. برای اینکه این کلمات را هم جزو متن کلمات در خواستی شما در نظر بگیرد ، باید قبل از این کلمات یک علامت بعلاوه + قرار دهید. اگر بخواهید چیزی در جستجوی شما اصلاً نباشد ، از علامت منفی در کنار آن کلمه خاص استفاده کنید. این حالت برای کدها هم تأثیر دارد. مثلاً اگر میخواهید که کلمه **Programming** را جستجو کنید ، اما نمی خواهید کلمه **web** در نتایج جستجوی شما باشد ، می نویسید:

Programming-web

اگر بخواهیم تعداد نتایج در یک جستجو را مثلاً به صد نتیجه در یک صفحه افزایش دهید از **num=۱۰۰** در متن جستجوی خود استفاده می کنیم .

برای فراهم سازی نتایج دقیق ، گوگل از بند آوردن یا **"wildcard"** برای حمایت از جستجو های خود استفاده نمی کند. برای مثال اگر شما **"google"** را جستجو کنید ، نتایج ارائه شونده شامل **"googler"** یا **"googlin"** نمی شود.

جستجو های گوگل حساس به حالت نوشتن ( **case sensitive** ) آنها نیست . تمامی کلمات بعد از ورود به حروف کوچک تبدیل می شوند. برای مثال اگر وارد کنید **"GOOGLE"** یا **"GoOgLe"** هر دو ره به عنوان **google** می شناسد .

## **ABSTRACT:**

In today's world conditions, doing jobs based on scientific principles and using advanced knowledge of the day .Visual Studio Is a collection of software programs that are linked very closely together and Microsoft It the application developers and programmers to be donated to them in NET development environment on Windows platforms to build their programs .

Visual Studio can write console applications, Windows, Windows services, mobile applications, ASP.NET applications and ASP.NET Web services built with your choice of languages such as C + +, C #, VB.NET, J # is used. I really do with Visual Studio "What can be done, and what it used to be and how they do things in a nutshell is that" it will.

That the civil code unique features and special attractions of this software than other similar software is more and more significant.

The importance of education the application of the for those that can be easily customized for specific programming can be used with regard to customer demands.

Step by step training provided in Visual Studio can work as a workshop and training for those who desire to learn and also to provide them with programming.





University of Applied and Scientific  
Education & Exploration Department Of Iranian Industries

Project Title:

**INSTRUCTION Of Microsoft Visual Studio**

**VC#**

Instructor:

**Eng. Amir Abas Karami**

By:

**Fakhrolsadat Mirzamani**

Subject:

**Information Technology Engineering**

Year: ۹۰