بنام انکه جان را فکرت آموخت

با بررسی های و تجربیاتی که بنده در زمینه برنامه نویسی های پایگاههای اطلاعاتی انجام دادم مشکلات بزرگ برنامه نویسان را در عوامل متعددی پیدا کردم که درطی سال گذشته (از دی ماه ۸۱ تا خرداد ۸۲) سعی کردم برای این مشکلات راه حل های جامع پیدا کنم .برای حل برخی از این مشکلات به راهنمایی و کمکهای ارزنده دوستانی(که هرگز ندیدمشان ولی خود را وامدار آنها می دانم) مدیون هستم و برخی دیگر را لطف الهی می دانم . به هر حال از انجایی که این نتایج را نمی خواهم تنها در اختیار خود داشته باشم و آنرا متعلق به همه می دانم ، سعی خواهم کرد طی مقالاتی گزارشی از راهی که پیموده ام را خدمت دوستان عزیز تقدیم کنم.

۱) انچه در شروع باید بدانیم

الف) بنابر یک اصل کلی در حل مسائل "**ابتدا باید مقصد را مشخص سپس ابزار رسیدن را انتخاب نمود** "

اگر به این مسئله معتقد باشیم باشیم انگاه دیگر درگیر معضلاتی همچون "تعصب بروی یک نرم افزار خاص" و یا "چون من فقط همین نرم افزار را می دانم پس بهترین راه همین است " نخواهیم شد. وصد البته اینکار نیازمند مطالعه و تحقیق مداوم برای رسیدن به بهترین راه حل خواهد بود.

ب) "**همه کس همه چیز را نمی داند**"

... اعتقاد به این اصل باعث خواهد شد تا شما اولا سعی در تیمی کردن پروژه های بزرگ کنید و ثانیا انجام نرم افزارهایی که از حیطه توانایی شما خارج است (ومدت کوتاهی برای اتمام آن دارید) را قبول نکنید چراکه با شکست در انجام آن اعتبار خود را زیر سوال خواهید برد

ج) "**چند کار کوچک بهتر از ۱ کار بزرگ است** "

این به معنای اینست که

- اگر تازه کار هستید با پروژه های کوچک شروع کنید و پس از کسب تجربه به سراغ اهداف بزرگتر بروید
- اگر پروژه بزرگی دارید (بزرگی پروژه بنابر منطق فازی به خودتان بستگی دارد) آنرا به قسمتهای کوچکتری تقسیم کنید تارسیدن به انها ساده تر باشد

د) "**مشکلترین راه حل همیشه بهترین نیست**"

همیشه اینگونه نیست که بتوانید همه چیز را بنابر مصالحی (همچون "استقلال برنامه از ابزار") با کد نویسی حل کنید چرا که اینکار نه تنها باعث افزایش زمان پروژه خواهد شد که احتمال خطا را نیز افزایش می دهد . بنده از سال ۷۴ تا ۷۵ در حدود ۱۲۰۰۰ خط برنامه بزبان پاسکال نوشتم که Utilityمحسوب می شد و باعث گردید که مشکلاتی همچون فارسی نویسی، ارتباط با شبکه مرتب سازی بانکهای اطلاعاتی و غیره را حل کنم و باکمک آنها چند برنامه مفید نوشتم اما اکنون از عیب یابی این برنامه ها وحشت دارم چه رسد به کسترش این برنامه ها

ه) "**خیال پرداز باشید** "

در سینما به این اصطلاح "حس گرفتن " گفته می شود. در واقع تخیل شما در باره محصول نهایی همیشه مشوق شما برای پیدا کردن راه حلها و فهم ندانسته های شما خواهد بود . این حس ذاتی است و انهایی که از این حس کمتر بهره مند هستند تا زمان تحویل پروژه متوجه نواقص ان نخواهند شد

و) "**دست پیش را بگیرید تا پس نیفتید**"

در مورد ۲ عامل مهم تاکید می کنم این ضرب المثل را فراموش نکنید

- زمان انجام پروژه
- مبلغ قرارداد

همواره پس از شنیدن صورت مسئله ،اعداد و ارقام ذهنتان را ۲ تا ۳ برابر اعلام کنید چراکه در این صورت نه تنها با ارامش خیال بدون واهمه زمان کافی برای انجام پروژ خواهید داشت که در اخر کار پس از کسر صورت هزینه هایی که حتی تصورشان را هم نمی کردید مقداری استفاده برایتان

ز) "**شما ۲ گوش داريد و ۱ زبان** "

مشتری همیشه می داند که چه کار می خواهد انجام شود ولی قرار نیست شما با حرفهای بزرگ زدن مسئله را برای او وخودتان پیچیده تر از آنچه که هست کنید . همیشه گوش کنید و نکات ظریف مسئله را با دقت تمام دنبال کنید . شاید در لحظه ای که شما فکر می کنید "مسئله ساده ای است " یک "فاجعه" در انتظار شما باشد در ان لحظه به این مسئله فکر کنید "زبان سرخ سرسبز می دهد برباد"

ح) "**من نمی دانم** "

همیشه با همه برای فهم انچه نمی دانید در ارتباط باشید .

- همواره با مشتری خود مشورت کنید . شاید او از مسائل برنامه نویسی چیزی نداند ولی مطمئننا از انچه که می خواهد با اطلاع است . نگذارید این دانش او با ایراد گرفتن به شما منتقل شود.
- همیشه در حال جستجوی اطلاعات جدید در اینترنت و یا کتب ویا حتی دوستانی که دارید باشید .

ط) "**اول کوچکترها بعد بزرگترها**"

یکی از عوامل موفقیت شما در شبیه سازی پروژه های بزرگ به صورت کوچک و تعمیم آن به برنامه های بزرگتر است . بیاد داشته باشید که در پروژه های بزرگ شما باید مسائل حاشیه ای را (که حتی ممکن است به اصل صورت مسئله ارتباط مستقیم ندارد) در نظر بگیرید . اما شبیه سازی در مقیاس کوچک به شما فرصت حل اصل مسئله را خواهد داد.

ی)"**سریع و کثیف** "

زیبا سازی برنامه را برای بعد بگذارید .. مهم نیست اول چه فونتی با چه رنگی انتخاب می شود . مهم اینست که ایا برنامه درست عمل می کند یا خیر . مشتری در هنگام استفاده دیدن یک برنامه زیبا را مهم می داند نه در هنگام تولید .

ک) "**عقل تابع چشم است** "

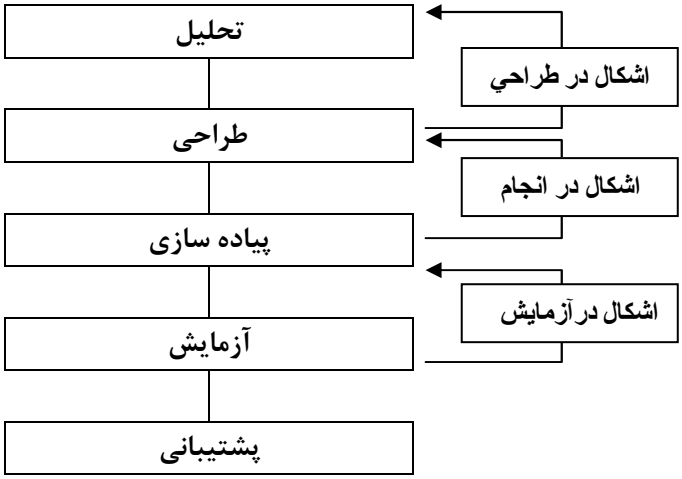
- (در مورد مشتری)یک برنامه زیبا بیشتر به دل می نشیند
- (در مورد برنامه نویس) هرچه بیشتر نمونه برنامه ببینید ، ذهن شما قدرت خیالپردازی بیشتری پیدا میکند و بالطبع دید وسیع تری را در عمل کردن خواهید داشت

۲) چگونه شروع کنیم

برای شروع لازم است در مورد مسائلی که به ترتیب عنوانهای آنها قید می شود ... علاوه بر توضیحات این مقاله مطالعه وسیع و مداومی نیز داشته باشید

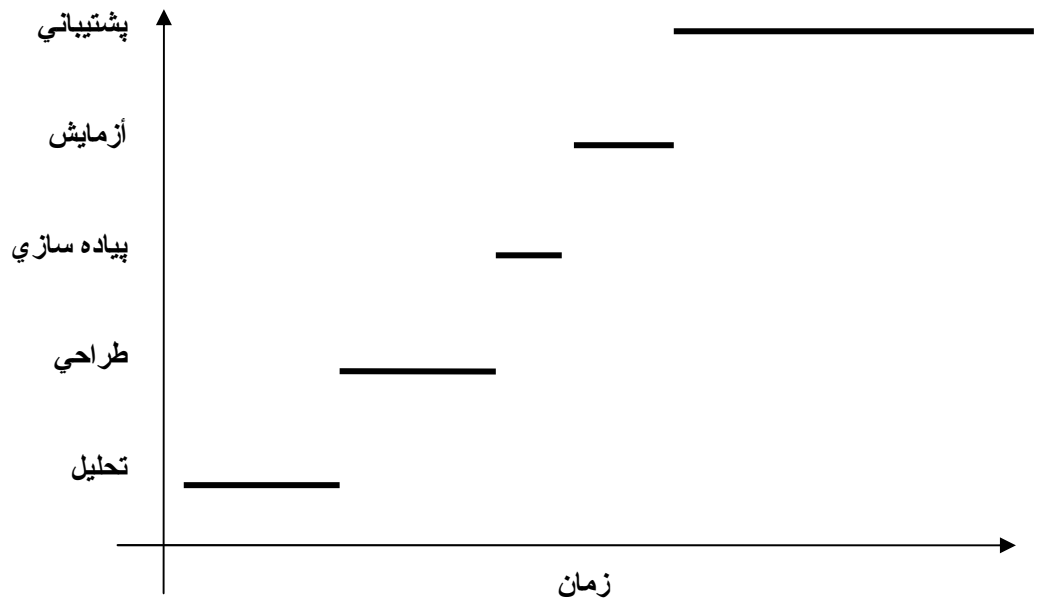
البته لازم به ذکر است که **فرض این مقاله بر این است که مشتری یا خود شما امکان و یا به صرفه بودن تولید نرم افزار را براورد و درمورد ان مطمئن هستید**.

حل هر مسئله و خصوصا برنامه های کامپیوتری مستلزم گذراندن ۵ مرحله است که در شکل نشان داده شده است



همانگونه که از شکل مشخص است هرچه ۴ مرحله اول دقیق تر انجام شود احتمال تکرار مراحل قبلی کمتر و بالطبع زمان و در پی ان هزینه تولید نرم افزار کاهش می یابد . در میان ۵ مرحله تحلیل مهمترین بخش است چراکه تعریف صورت مساله در تحلیل انجام میشود. نمی خواهم بحث را طولانی کنم .فقط این قضیه را خاطر نشان می کنم که اهمیت "تحلیل " بقدری زیاد است که در خارج از کشور شرکتها ی تولیدکننده نرم افزار حتما داراری چندین تحلیلگر ماهر هستند که از ابتدا تا انتهای پروژه هدف پروژه را کنترل و مسیر انرا اصلاح میکنند.

بعنوان یک حاشیه می توان زمانبندی ۵ مرحله را نیز بصورت زیر ترسیم نمود که با مراجعه به کتابها و نرم افرارهای مختلف می توانید این نمودار را دقیق تر کنید .



الف) **تحلیل**

"فهم مساله نیمی از حل مساله می باشد"

شما تاوقتی که خودتان ندانید چه می خواهید انجام دهید نخواهید توانست مساله را حل و بالطبع دیگر اعضای گروه را رهبری کنید .

پس قدم اول شناخت صورت مساله است .هدف از انجام این قدم را می توان پرداختن به ۴ موضوع زیر دانست

۱. تعیین اطلاعاتی که قابل مکانیزه کردن می باشد
۲. افرادی که لازم است با این ابزار(نرم افزارها) کار کنند.در پروژه های بزرگ این مسئله شامل سلسله مراتب اولویت افراد نیز می گردد
۳. انتخاب نمونه های اصلی برای کار روی انها(فرمها پرونده ها اسناد و غیره.....)
۴. میزان گستردگی نمونه های انتخاب شده و همچنین کاهش میزان پیچیدگی نرم افزار در موارد کم اهمیت

برای انجام این کار راههای متعددی وجود دارد مانند :

- **مصاحبه** : با یک یا چند کارشناس مسلط به کار (هر چند درکی از کامپیوتر نداشته باشند) صحبت کنید . این کار را گاه لازم است بطور مداوم تکرار کنید تا مطمئن شوید که می دانید چه می خواهید .
- **فرمهای نظر سنجی** : ممکن است برنامه ای قبلا نوشته شده باشد و یا ذهن یک جامعه (منظور مجموعه ای از افراد است) نسبت به یک مسئله روشن باشد ، بهتر است نظر انها را در مورد نواقص سیستم قبلی(حتی ممکن است سیستم قبلی یک سیستم دستی باشد) و یا پیشنهاد سیستم جدید بدانید .
- **مرور انچه که هست** : برخی موارد (بجز مواردی که یک نرم افزار خاصی وجود دارد) می توانید با مراجعه به انچه انجام شده(نرم افزارهای قبلی) به بررسی نقاط ضعف و قوت نرم افزارها بپردازید و از این راه تحلیل خود را محکم تر کنید
- **مشاهده عینی** : بایستی بطور مداوم با حضور در محل روند کار وانچه گفته می شود را با انچه می دانید ترکیب و مقایسه کنید تا بتوانید به نکات ریز و ظریفی که وجود دارد پی ببرید . گاه نکته ای که از دید کارفرما بدیهی است از دید شما ناشناخته است
- **عکسبرداری و فیلمبرداری و یا ضبط صوت** : انسان هیچگاه نمی تواند به حافظه خود جهت یادآوری اتکا کند . بهتر است از این سه ابزار ساده و ارزان قیمت (در مقایسه با نتایجی که بدست می آورید) استفاده کنید تا در هنگام تحلیل بتوانید با مراجعه مکرر امکان خطا را کاهش دهید
- **ثبت انچه بدست می اورید** : مطالبی که شما جمع اوری میکنید دارای ارزش زیادی هستند . انها را بطور مداوم دسته بندی و یادداشت کنید . اینکار باعث می گردد تا اولا براحتی در مراجعه بعدی مطلب را پیدا کنید و ثانیا با حذف یک شخص خاص از پروژه (اخراج ، عدم تمایل به کار، بیماری و یا ....) کل پروژه از کار نیفتاد

همانگونه که می بینید در این مورد این مساله نکات ظریف بسار زیاد است .

از جمله تجربیاتی که من در حین کار متوجه شدم اینست که وقتی من میتوانم بگویم یک **تحلیل جامع و کامل** ارائه کردم ویا در واقع صورت مساله را فهمیدم **که بتوانم تک تک کارهای مشتری را در سیستم دستی بدون اشکال خودم انجام دهم** .

البته در حین مراحل تحقیق چند نکته را از یاد نبرید

- **پررو باشید :** از سه عامل پول ، پارتی و پررویی اگر دوتای اول را ندارید حداقل سومی را داشته باشید . البته این به معنای بی ادب بودن نیست بلکه به معنای سمج بودن یا در اصطلاح ادبی "پشتکار داشتن " برای دست یافتن به مطلب مورد نظر است
- **ایجاد وحشت نکنید :** با فیلمبرداری و ضبط صدا و یا حتی یادداشتها کاری نکنید که کارمندان احساس کنند شما درحال جمع اوری مدرک علیه او هستید

یک تحلیل گر علاوه بر ارائه انچه که وجوددارد باید بتواند راهکارهای اصلاح سیستم فعلی را نیز ارائه کند که به این مقوله در بخش بعدی خواهیم پرداخت

شماي سيستم

چگونه تحليل اطلاعات و طراحي كنيم

در ادامه بحث قبل اکنون میرسیم به اینکه با نتایج مرحله قبل چه کنیم

مواردی که اکنون در اختیار ما هست عبارتند از

- کلیه فرمهای ورودی و خروجی سیستم
- توضیحات مربوط به هریک از فرمها بصورت کامل
- نوارهای صوتی و تصویری احتمالی از مصاحبه ها

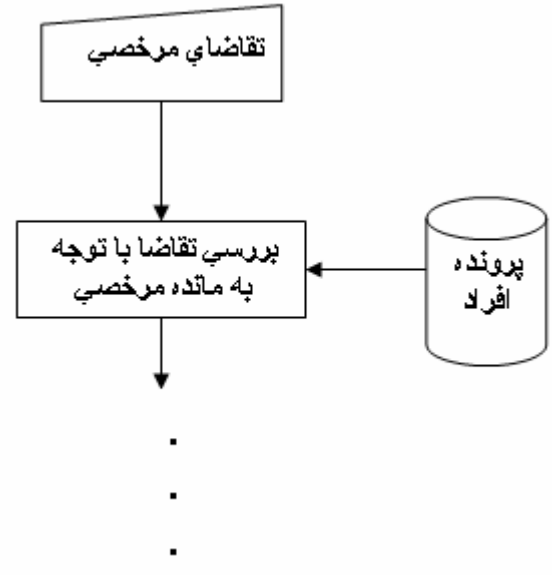
طراحی شامل قسمت های مختلفی است که می توان اولین قسمت انرا **ترسیم وضعیت موجود** و دسته بندی عنوان کرد

حال بترتیب عملیات زیر را انجام دهید

۱<<< انچه در اختیار دارید را دسته بندی کنید

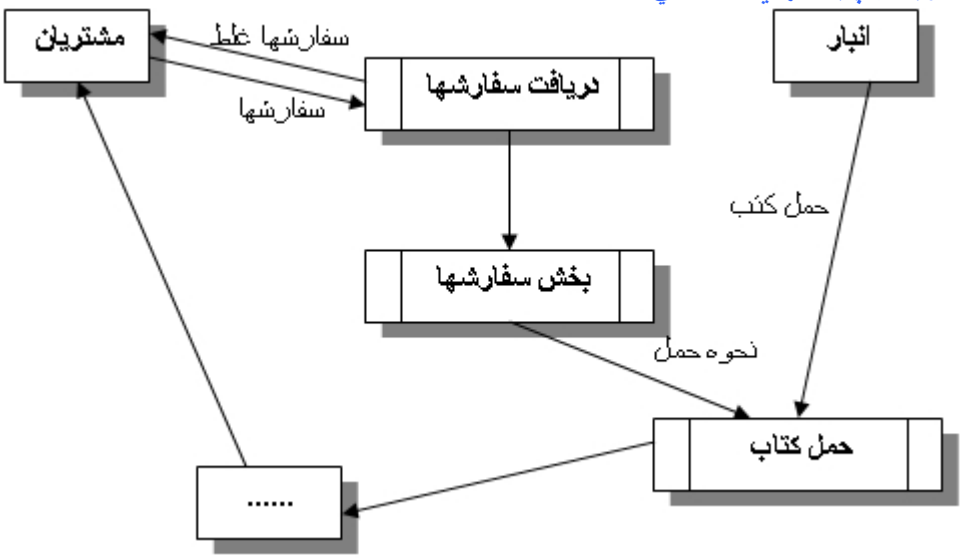
مثلا در مورد یک سیستم پرسنلی دسته ها می توانند بصورت : اطلاعات پرسنل، مدارک ، مرخصی ، مزایا و... باشند

۲<<<بر اساس دسته بندی اولیه خود نمودار کلی نظام را ترسیم کنید(System Narrative)



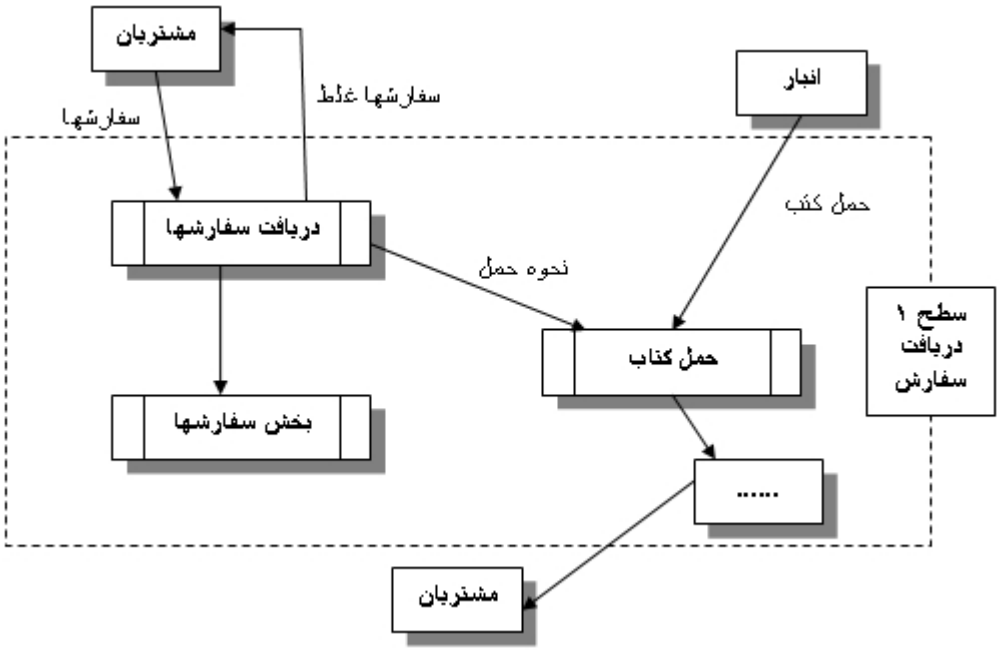
این نمودارها در واقع عملکرد سیستم فعلی را برای انجام کار نشان می دهند

۳<<<نمودار **گردش داده**(Data Flow Diagramیا DFD) را ترسیم کنید این نمودار می تواند به ما نشان دهد که دادهای خروجی و ورودی هریک از اجزاء چیست و وضعیت گردش انها به چه ترتیب است . در مورد ترسیم این نمودار می بایستی تمامی گردش کار بدقت ترسیم شود چرا که به عنوان یک نمودار عملیاتی بعدا مورد استفاده مکرر شما و مشتری خواهد بود



۴>>> نمودارهای DFDخود را از سطح صفر به سطح یک و ... ببرید. هر سطح نشانه داخلی تر بودن پردازش خواهد بود :

مثال سطح ۱



ما اکنون کار مدل سازی سیستم را به صورت شماتیک انجام داده ایم و می دانیم اطلاعات خام بدست امده از بخش قبل چگونه با یکدیگر ارتباط دارند . اما هنوز در مورد اصلاح این سیستم و شروع طراحی کاری نکرده ایم . اکنون برای بررسی اطلاعات آماده می شویم .

پايگاهها

اصلی ترین منبع برای اطلاعات فرمهایی است که در یک سیستم رد و بدل می شوند.برای پردازش این فرمها می بایستی تمامی اطلاعات قید شده در انها را بصورت یک جدول لیست کنیم . بالای جدول نیز می توانیم نام را قید کنیم

مثال:

فاکتور فروش				
فروشنده				شماره فاکتور : ۵۴۵
خریدار				تاریخ
ردیف	شرح	فی	تعداد	مبلغ
۱				
۲				
۳				

پایگاه فاکتور فروش

شماره فاکتور	فروشنده	خریدار	تاریخ	شماره تلفن
ردیف	شرح	فی	تعداد	مبلغ

نکته

در هنگام بررسی Dataها متوجه خواهید شد که برخی مطالب دارای درجه اولویت بالاتری نسبت به برخی دیگر هستند . به عنوان مثال در همان مثال شماره تلفن دارای اهمیت بالایی نیست.

حال در قدم بعدی می توانید اطلاعات موجود در جدول را بصورت زیر در یک Data Catalogثبت نمود . این کاتالوگ پیشنهادی است که اطلاح شده فرمهای مفصل و دست و پاگیر (ولی قدرتمند) اصلی (Data Dictionary)می باشد. در Data Dictionaryهای اصلی نه تنها همین مطالب که مطالب بیشتری مانند محلهای استفاده ، محدودیتهای محل و غیره را میتوان قید نمود با مراجعه به کتب طراحی می توانید به مطالب بیشتری دست پیدا کنید .

همچنین یک نام نیز برای پایگاه انتخاب کنید

Factor(پایگاه فاکتور فروش)

ردیف	نام	نوع	طول	شرح	توضیحات
۱.	FacNo	N	۶	شماره فاکتور	شماره چاپی فاکتور
۲.	Seller	A	۲۵	فروشنده	
۳.	Buyer	A	۲۵	خریدار	
۴.	FDate	A	۱۰	تاریخ فاکتور	
۵.	.....	.....	...	.....	.....

این طرح خام و اولیه پایگاه شماست

چگونه تحليل اطلاعات و طراحي كنيم ۲

در قسمتهای قبلی گفتیم که چگونه می توان به طرح خام از پایگاه رسید

حال به این مسئله می پردازیم که چگونه می باید ارتباط بین این طرههای خام را برقرار کرد

در کل سه نوع ارتباط بین پایگاهها وجود دارد

۱. **یک به یک** : (One To One) ارتباطی است که به ازای یک رکورد در یک پایگاه فقط و فقط یک رکورد در پایگاه مرتبط موجود یاشد

مثال : یک مشتری یک سفارش دارد

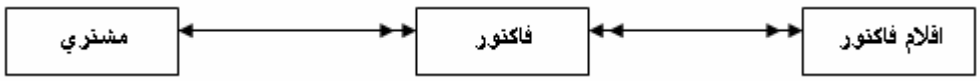
۲. **یک به چند** : (One To Many) ارتباطی است که به ازای یک رکورد در یک پایگاه حداقل یک یا بیشتر رکورد در پایگاه مرتبط موجود یاشد

مثال : یک سفارش چند قلم کالا دارد

۳. **چند به چند** : (Many To Many) ارتباطی است که به ازای یک رکورد در یک پایگاه حداقل یک یا بیشتر رکورد در پایگاه مرتبط موجود یاشد و بالعکس

مثال : چند مشتری چند سفارش دارند

شکل زیر نحو ترسیم اینگونه ارتباط ها را نشان می دهد



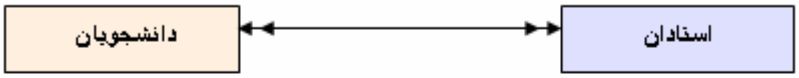
اما تشخیص و ترسیم رابطهای کلی نظام مستلزم طی کردن قدم به قرم مراحل زیر است :

- ابتدا تمامی رابطه های واقعی موجود در سیستم را تک به تک بنویسید
- شکل هریک از رابطه ها را بکشید
- اشکال را برهم منطبق کنید تا موجودیتهای تکراری حذف شوند(شکل کلی ارتباط)
- از طریق تحلیل شکل کلی ارتباط های اضافی را حذف کنید

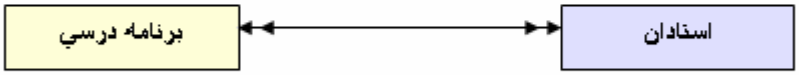
یک مثال : نظام **دانشجو ، استاد ، درس ، برنامه درسی**

رابطه ها : (دو قدم اول)

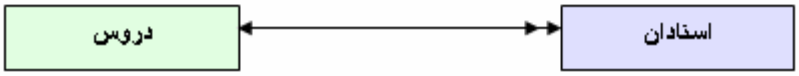
- استادان – دانشجویان



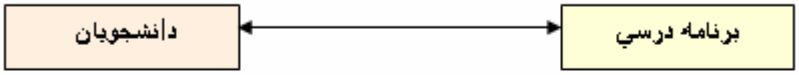
- استادان – برنامه درسی



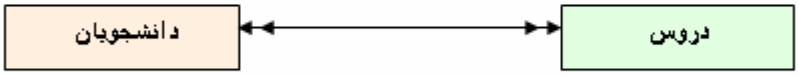
- استادان به دروس



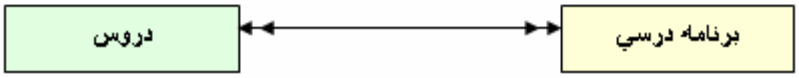
- دانشجویان به برنامه درسی



- دانشجویان به دروس

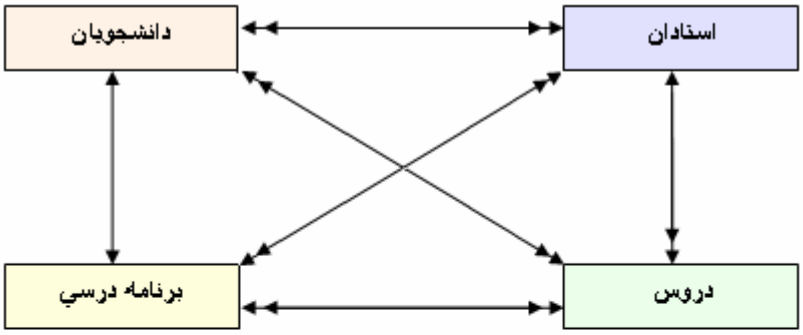


- برنامه درسی به دروس

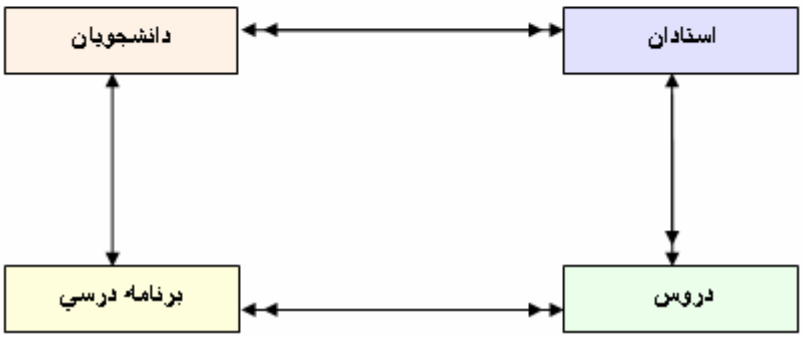


تکمیل (دو قدم دوم)

رابطه هاي كلي نظام



رابطه هاي كلي نظام پس از اصلاح



نرمالایز

چگونه تحلیل اطلاعات و طراحي كنيم ۳

تا این مرحله ما یاد گرفتیم که چگونه روند کار سیستم را نشان دهیم و چگونه پایگاههای موجود را از فرمها بصورت خام تهیه کنیم

قبلا گفته شد که می بایستی از پیچیدگی سیستم بکاهیم . این وظیفه را نرمالایز(Data Normalaization) انجام می دهد.
بترتیب قدمهای زیر را انجام دهید

۱) **داده های تکراری** را **از داده های ثابت** جدا کنید و در پایگاههای جداگانه بگذارید
مثال از فاکتور خرید زیر می توان ۲ پایگاه را ایجاد کرد

پایگاه فاکتور فروش

شماره فاکتور	فروشنده	خریدار	تاریخ	شماره تلفن
ردیف	شرح	فی	تعداد	مبلغ

پایگاه اول

شماره فاکتور	فروشنده	خریدار	تاریخ	شماره تلفن
				مبلغ

پایگاه دوم

ردیف	شرح	فی	تعداد

۲) **کلید شناسایی** تهیه کنید . تهیه کلید شناسایی رکورد دارای روشهای بسیار متنوعی است که اگر توانستم بعدا در این زمینه توضیحات بیشتری خواهم داد

اما روش کلی برای ساخت کلید را می توان بصورت زیر عنوان نمود : پیدا کردن فیلدیا ترکیبی از فیلدها که در هیچ حالتی نتواند تکرار داشته باشد . بعنوان مثال فیلد نام خریدار نمی تواند کلید باشد اما شماره فاکتور یک پیشنهاد برای کلید پایگاه اول است.و در پایگاه دوم میتوان ردیف را پیشنهاد نمود . اشکالی که در مثال وجود داردایننست که در پایگاه دوم ردیف تکراری خواهد بود برای حذف این تکرار می بایستی ترکیب این فیلد را با شماره فاکتور پایگاه اول در نظر گرفت . بنابر این یک مجموعه Master / Detail خواهیم داشت . شکل زیر این مسئله را روشن می کند

پایگاه اول

شماره فاکتور	فروشنده	خریدار	تاریخ	شماره تلفن
				مبلغ

پایگاه دوم

شماره فاکتور	ردیف	شرح	فی	تعداد

۳) **داده های قابل استنتاج** و غیر کلیدی را حذف کنید (Calculated Fields)

این گونه داده ها داده هایی هستند که مثلا از جمع مقادیر یک فیلد ، تفاضا ۲ فیلد و یا محاسبه فرمولی یک فیلد دیگر(تاریخ تولد و سن یا تاریخ استخدام و سابقه کاری) بدست می آید .

در مثال فوق مبلغ کل فاکتور در پایگاه اول از این دسته فیلدها است . بنابر این پایگاه را بصوت زیر تکمیل می کنیم

پایگاه اول

شماره فاکتور	فروشنده	خریدار	تاریخ	شماره تلفن

پایگاه دوم

شماره فاکتور	ردیف	شرح	فی	تعداد

۴) **دادههایی که در چند پایگاه تکرار** میشوند را حذف کنید (LookUp Fields)

ممکن است شما برای مشخصات مشتریان خود بخواهید یک پایگاه جداگانه تعریف کنید و یا حتی بخواهید فروشندگان خود را به تفکیک مشخص کنید در این صورت لازم نیست در هر رکورد پایگاه اول تمامی مشخصات انهارا قید کنید ، بلکه کافی است کد مربوطه را به این پایگاه منتقل کنید . شکل زیر تمامی پایگاههای تشکیل شده را نشان میدهد

پایگاه اول

شماره فاکتور	کد فروشنده	کدخریدار	تاریخ

پایگاه دوم

شماره فاکتور	ردیف	شرح	فی	تعداد

پایگاه سوم

کد فروشنده/خریدار	نام	تلفن	.....	.....

حال پایگاه شما در حالت کامل نرمالایز قرار دارد

گفتن چند نکته شاید بد نباشد :

۱) بهنگام طی کردن مرحله ۳ می بایستی در مورد فیلدهایی که قابل استنتاج هستند ولی استنتاج انها زمان زیادی می طلبد قدری با احتیاط رفتار کرد . این مساله به نحوه استنتاج شما ارتباط مستقیمی دارد .شاید توضیحات زیر تا حدودی قضیه را روشنتر کند :

- استنتاج زیر ۱۰۰ رکورد با حلقه While را شاید بتوان تحمل کرد، هر چند پیشنهاد نمی شود
- ولی این روش را نمی توان در مورد بالاتر از ان روش درستی نامید . بهتر است با یک دستور SQL این زمان را در مورد تعداد بالاتر کاهش داد .
- اما در برخی موارد دستورات SQL نه تنها در مورد تعداد بالا بلکه در مورد پیچیدگی دستور نیز دچار مشکل و بالطبع زمانبر می شوند در این حالت بهتر است از روشهای بهینه سازی دستورات SQL (تعریف Index ها، تودرتو کردن دستورات و...)استفاده کنید

- اگر با روشها ی فوق به نتیجه نرسیدید و یا علاقه مند به وارد شدن به این مقولات نیستید بهتر است اصلا ایـن فیلـد را حذف نکنید

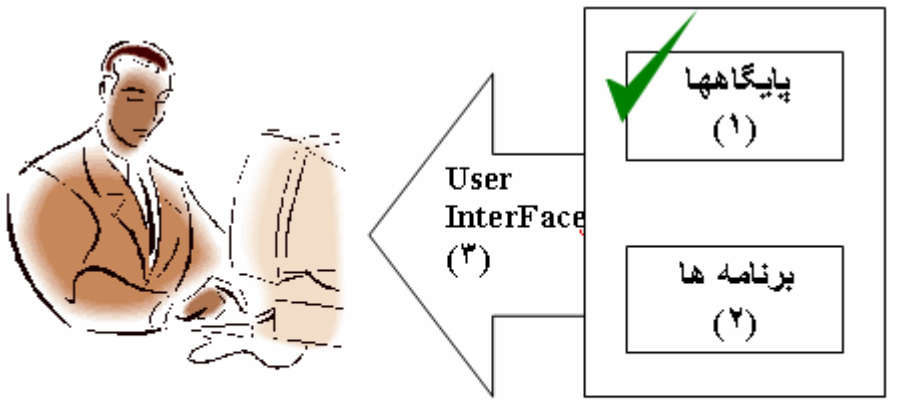
۲) در مورد مرحله ۲ می بایستی تجربه و نگاه خوبی به مساله داشته باشید .

- ممکن است در برخی موارد کلیدی بظاهر یکتا باشدولی بعدا متوجه شوید که یکتا بودن فقط در یک دوره خاص است . این مساله تمامی طراحی های شما را بهم خواهد ریخت .
- ممکن است برخی موارد کلید ها را بیش از پیچیده کنید . مثلا ترکیب چند فیلد برای کلید بودن .این مساله عـلاوه بـر احتمال کاهش سرعت در فیلدهای کلیدی بزرگ باعث خواهد شد که اگر برخی از موارد مورد نیاز شما در حالات خاص وارد نشود شما در تولید کلید دچار مشکل شوید

شرايط محيطي

چگونه تحليل اطلاعات و طراحي كنيم ٤

انچه كه تا اينجا قيد شد تنها طراحي بخش داده ها بود . ما هنوز چند قسمت ديگر را در پيش رو داريم . به شكل زير دقت كنيد



شماي كلي تعريف يك سيستم ADT

اگر شماي كلي يک سيستم را شامل ۳ بخش فوق بدانيم ما هم اكنون بخش اول انرا تهيه كرديم .

برنامه ها در بخش دوم قرار دارند كه با UIمتفاوت هستند . برنامه ها در هنگام طراحي صرفا يكسري **كدهاي بدون وابستگي به زبان خاص**(Pseudo Codes)مي باشند . شايد اطلاق **شرايط محيطي** به اين برنامه ها بـراي جلـوگيري از اشـتباه مفيـد باشد .

براي نوشتن اين شرايط طراحان معمولا از كلمات اشنايي نزديک به معاني اصلي استفاده مي كنند .نمونه اين كلمات عبارتند از

Read / Get	خواندن از ورودي
Write / Print / Message	خروجي
Find	جستجو
Add , Sub , Mul , Div , Mod	عمليات رياضي
Compare	مقايسه
Delete , Copy	عمليات (فايل / رشته /....)
Sort	مرتب سازي
If , Else , Case , Switch	شرط ها
While , For , Repeat	حلقه ها
Begin , End , { , }	بلاک بندي

مثال :

```
Read   No
Read   K
If  No > 100
    For 1 to No
         { k = k Mod 10
           Print K
         }
Else
   S = Add (No,5)
```

اما چگونه از این کد ها می توان استفاده نمود

فرض کنید در برنامه فاکتور می خواهید حتما یکسری تصمیم گیری ها قید و یا محاسباتی انجام شود . بـه مثالهـای زیـر توجـه کنید

١) تاریخ فاکتور به ماقبل بر نمی گردد

```
If Input Date < Current Date
   Print "Error in Date"
```

٢) جمع حاصل ضرب فی در تعداد در مبلغ کل ریخته شود

```
S = 0
For All rows
     S=S+Fi * Numner
Monye = S
```

٣) تعداد درخواستی باید کمتر از میزان موجود باشد

```
If ReqNumber< Remind in stock
     ……
Else
    Message "Not Enough In Stock "
```

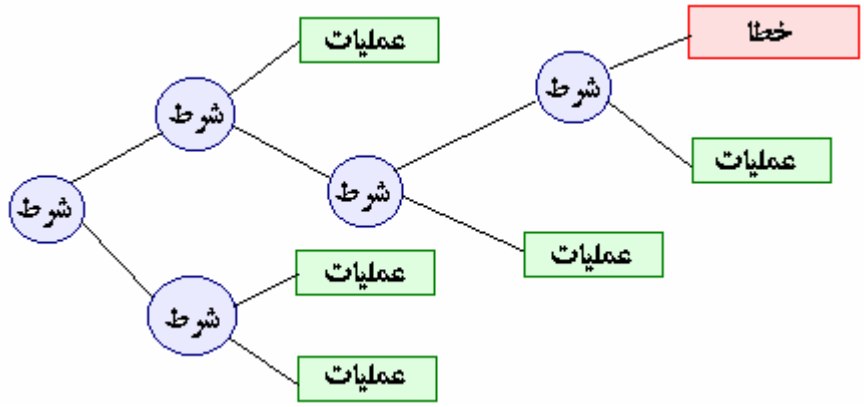
اما چرا نیاز است که این مطالب را به این صورت بنویسیم . زیرا که اولا امکان خواندن برنامه ها بدون تـلاش در فهـم متغیـر هـا امکان پذیر می گرددو ثانیا اهداف برنامه ها را می توان بسادگی پیدا کرد

اما در برخی موارد در نوشتن این برنامه ها در شرطهای تودر تو دچار مشکل می شویم در اینجا بسادگی می توانیم از درختهـای تصمیم گیری استفاده کنیم .

بعنوان مثال بجای نوشتن این گونه کد ها

```
If ……..
   ……
Else
    If ……..
        ……
•
•
•
```

مي توان از درخت زير استفاده كرد



محیط پیاده سازی

تا اینجا توانستیم ۲ قسمت از ADT را تکمیل کنیم

۱) پایگاههای اطلاعاتی

۲) شرایط محیطی کار روی اطلاعات (برنامه ها بدون وابستگی به زبان)

حال بایستی به سراغ قسمت سوم یعنی رابط کاربر(User Inter Face) برویم . پیش شرط ان بررسی و شناخت محیط پیاده سازی است . این به معنای **بررسی سخت افزار قبل از پیاده سازی نرم افزار** می باشد.

برای این کار بایستی به چند سوال پاسخ دهید

۱. ورود اطلاعات به کامپوتر و احتمالا تبادل اطلاعات با دیگر کامپیوتر ها چگونه است
۲. نحوه گرفتن پشتیبان از اطلاعات به چه ترتیب است
۳. نحوه دریافت خروجی های برنامه ما توسط کاربر چگونه است
۴. چه کامپیوتر (هایی) مورد نیاز است که برای پاسخ به ان بایستی به سوالات زیر توجه کنید
   a. حجم عملیات به چه میزان می باشد
   b. حجم اطلاعاتی فوق در بهترین و بدترین شرایط چه زمانی به می دهد
   c. فعالیت های ما (Process های برنامه)به چه میزان زمان نیاز دارند
   d. زمان بازیابی اطلاعات ما چقدر است

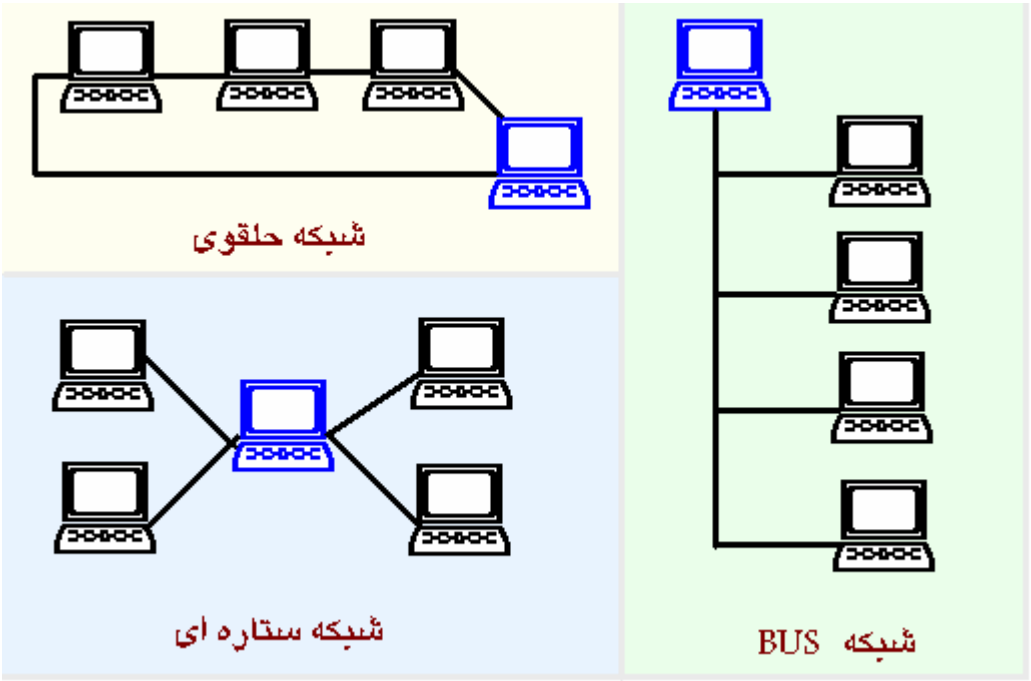
همانگونه که مشاهده می کنید

- این سوالات را در مورد یک **سیستم ساده** می توان براحتی جواب داد اما در مورد سیستمهای وسیع قدری باید دقت کرد . هرچند توصیه می کنم هرگز در جایی که یک شبکه وجود دارد هیچ سیستمی را به صورت تک کاربره در نظر نگیرید ، چون بعد از مدتی یادشان می اید که به شما درمورد شبکه بودن ان قبلا تذکر داده بوده اند!

- در مورد **سیستمهای وسیع** (*مجموعه نرم افزارها تحت شبکه و یا ۱ نرم افزار که روی چند سیستم کار میکند*) شاید دانستن چند تعریف باعث شود مسائل را بهتر متوجه شوید

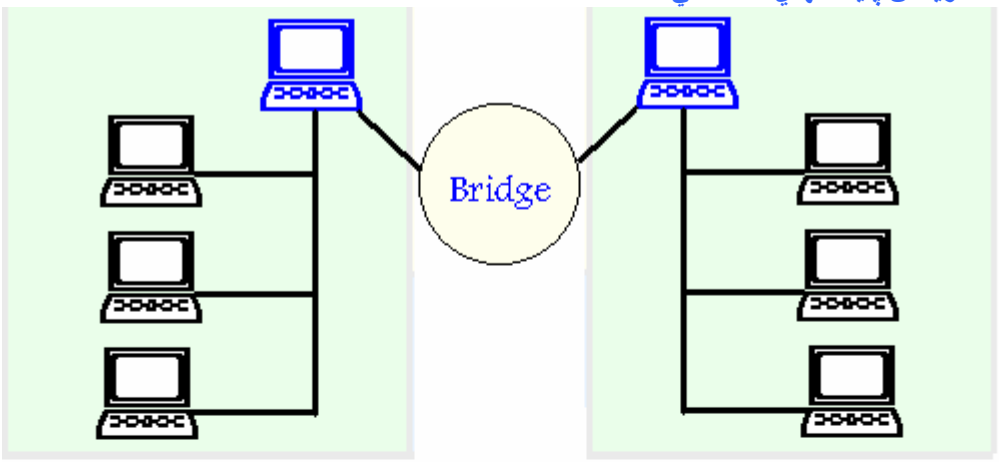
١. **شبکه ها محلی** (LAN = Local Area Network) شبکه هایی هستند که معمولا در یک ساختمان و یا محوطه کوچک پیاده سازی می شوند.
٢. **شبکه های وسیع** (WAN = Wide Area Network) شبکه هایی هستند که وابستگی جغرافیایی در انها معنی ندارد
٣. **میزبان** (Host/Server) : به کامپیوتری می گویند که وظیفه نگهداری اطلاعات را برعهده دارند . این کامپیوتر ها معمولا از قدرت پردازش بالاتری برخوردارند
٤. **میهمان** (Guest / Client) کامپیوتر هایی که از داده های موجود در سرور استفاده / حذف / تغییر ویا اضافه می کنند

همچنین در مورد نحوه اتصال کامپیوتر ها به یک دیگر نیز با توجه به اشکال زیر می توان برخی از مفاهیم را شناسایی کرد

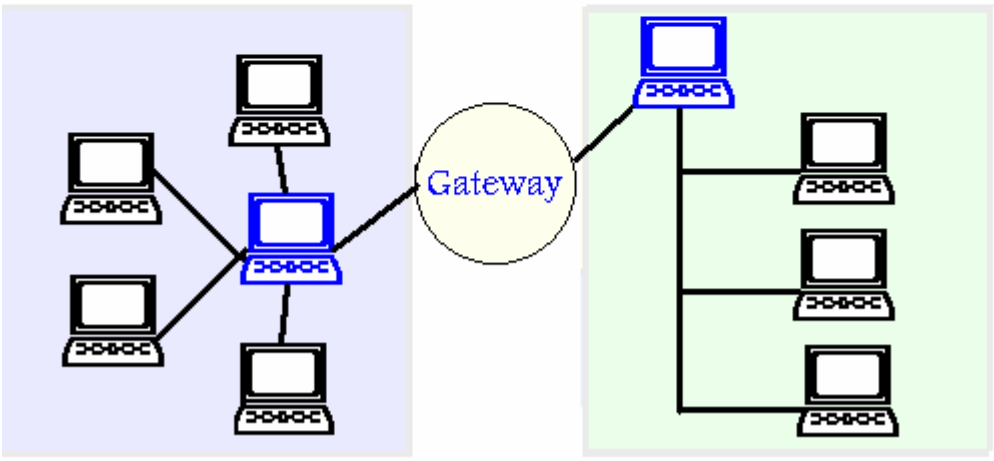


کامپیوتر آبی همان میزبان و مشکی میهمان است

البته ممکن است که چند شبکه به هم متصل باشند



اتصال دوشبکه مشابه



اتصال دو شبکه متفاوت

تمامی مطالبی که گفته شد برای این بود که بگوییم

"وظیفه شما به عنوان **یک تحلیل گر و طراح** ایجاب می کند که شما بتوانید ابتدا **طرح شبکه** را به عنوان کسی کـه **شناخت جامع و کاملی از نرم افزار** دارید به **مسئولان پیاده سازی شبکه پیشنهاد** دهید "

اما چگونه ؟

۱. ابتدا به مساله پایگاهها با دقت توجه کنید . برخی از پایگاهها را لازم است در کامپیوتر میهمـان و برخـی را در کـامپیوتر میزبان قرار دهید . نحوه این تقسیم بندی را بایستی با توجه به

a. میزان زمان مورد نیاز برای پردازشها
b. اهمیت دادهها
c. اشتراک داده ها
d. رده بندی داده ها

مشخص نمود .

فرض کنید در یک سیستم برخی از پایگههای شما موقتی هستند و نتیجه انها در یک سیستم دیگر ثبت می شوند و یا لازم است نتیجه یکسری از مقادیر در اختیار دیگران قرار گیرد . هردو این مثالها پایگاههایی را نشان می دهند کـه مـی توانند روی کامپیوتر های میهمان قرار گیرند

۲. به مساله پردازشها توجه کنید . این مساله ارتباط مستقیمی با محل نگهداری اطلاعات پایگاهها دارد . برخی از پردازشها را باید روی کامپیوتر میهمان انجام داد . مثلا نتیجه گیری از دادههای وارد شده در کامپیوتر اپراتور برای ارسال به کامپیوتر میزبان . اما برخی دیگر را می بایستی در کامپیوتر میزبان انجام داد.

یک مثال ساده می تواند به شما کمک کند . فرض کنید می خواهید جمع کل یکسری داده را محاسبه کنید . این پردازش بایستی در نزدیکترین محل به کامپیوتر نگهداری ان داده ها باشد . اگر داده ها در کامپیوتر میزبان هستند ، در انجا و باالعکس . اینکار باعث کاهش ترافیک شبکه و سرعت پاسخگویی می شود .اما اگر این کار امکان پذیر نیست بایستی مشتری هزینه بیشتر برای شبکه متحمل شود

۳. کنترل ، نه تنها به مفهوم امنیت که به معنای تقسیم کار هم می باشد . فرض کنید در یک سیستم ، دادههای ورود و خروج اجناس را قرار است ثبت کنند . ایا بهتر است یک کامپیوتر (برای ثبت اطلاعات) در محل قرار دهند و یا می توان اطلاعات را به یک قسمت دیگر برای ثبت فرستاد ؟ و ایا بهتر است بانک اطلاعاتی در محل باشد و سپس با دیسکت (یا هر انتقال Off Line دیگر) به کامپیوتر مادر منتقل شود و یا انتقال باید همزمان ثبت باشد؟ ایا امنیت (چه در مورد دستکاری اطلاعات Off lineو چه در مورد ثبت اشتباه Online) تامین می شود ؟ و سوالات دیگری که با تجربه و مطالعه می توان جواب انها را پیدا کرد

شروع پیاده سازی ۱

برای پیاده سازی راههای متعددی وجود دارد .من در ادامه این نوشته ها راهی را که بنظرم دارای کمترین اشکال وکد وهمچنین بیشتری بازه است را خواهم گفت .

اما قبل از آنکه بخواهید شروع به نوشتن برنامه ها کنید چند نکته کوچک و پر اهمیت را برای دوستانی که نمی دانند عنوان می کنم .

تقسیم بندی پایگاهها :

پایگاهها همیشه یک نوع نیستند. بلکه دارای تقسیم بندی برحسب نوع عملکرد دارند. این تقسیم بندی (که تقریبا همجا یکسان است) بصورت زیر می باشد

۱. **پایگاههای اصلی (Master)**: پایگاههایی هستند که اطلاعات مادر رانگهداری می کنند . مثلا پایگاه اطلاعات پرسنل و یا اطلاعات دانش آموزان

تشخیص : این نوع پایگاه دارای تغییرات کمی هستند مثلا تصحیح نام یا تاریخ استخدام و برخی از فیلد ها هرگز تغییر نمی کنند . مثلا شماره شناسایی

۲. **پایگاههای ثبت تغییرات (Detail)** : پایگاههایی که تغییرات مرتبط با پایگاههای مادر را ثبت می کنند. مثلا ورود و خروج کارمندان و یا نمرات دانش آموزان

تشخیص : این نوع پایگاه دارای تغییرات وسیعی می باشد و بیشترین کار کاربر روی فعالیت این نوع پایگاه استوار است

۳. **پایگاههای مرجع (Reference)** : اطلاعاتی که در این نوع پایگاهها ثبت می شوند تقریبا ثابت هستند. مثلا پایگاه واحدهای اندازه گیری یا نحوه محاسبه بیمه و مالیات

نکته : این نوع پایگاهها را نباید با پایگاههای اصلی اشتباه گرفت و همچنین در برخی از موارد این پایگاهها فاقد کلید اصلی هستند

۴. **پایگاههای راکد (Archive)** : پایگاههایی برای نگهداری اطلاعات دوره های قبلی کار سیستم هستند. مثلا بایگانی سالهای مالی و یا سوابق افراد

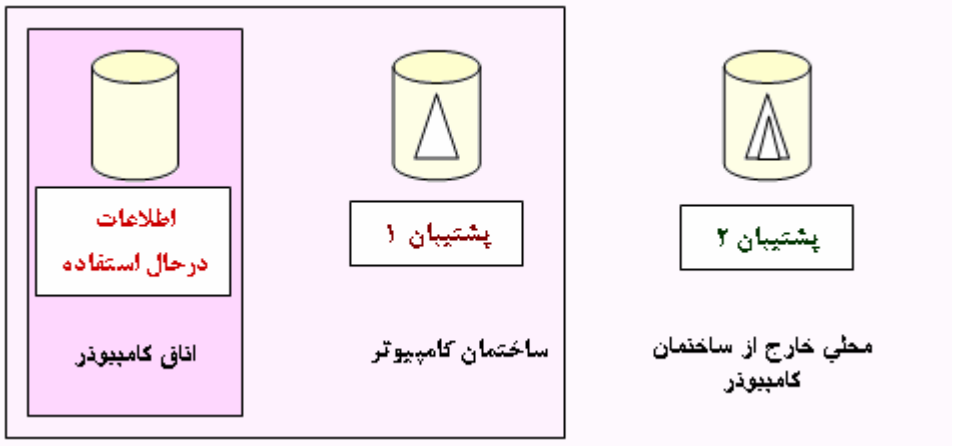
نکته : در بیشتر موارد اطلاعات موجود در این پایگاهها می بایستی غیر قابل تغییر باشند. همچنین در مورد نحوه تشکیل این پایگاهها می توان به ۲ صورت زیر عمل نمود

- **یک فایل ثابت که اطلاعات هردوره به این فایل اضافه می شود**
- **از طریق تقسیم بندی محل نگهداری (دیسکها ، شاخه ها و ...) این اطلاعات را برحسب دوره از یکدیگر منفک نمود**

۵. **پایگاه های پشتیبان (Back Up)** : پایگاههایی برای پشتیبانی از پایگاههای اصلی یا تغییرات هستند و در صورت خرابی فایل ها اصلی مورد استفاده قرار می گیرند .

نکته : انواع روشهای زمانبندی برای پشتیبان گیری وجوددارد . به عنوان مثال روش پدربزرگ که شکل زیر میتواند تاحدودی به نحوه کار ان اشاره کند.

توجه داشته باشید که شدت رنگ میزان خطر از دست رفتن اطلاعات را مشخص می کند



۶. **فایل یا پایگاه ثبت تغییرات (Transaction Log)** : پایگاه و یا فایلهایی که هرگونه تغییرات در دیگر انواع پایگاهها را ثبت می کنند و بهنگام بروز اشکال ، کاربر می تواند پایگاهها را به زمان پشتیبان گیری برگردانده و از طریق این فایلها تمامی عملیات را مجدد اعمال نماید

نکته : چنانچه شما سیستمی را مورد استفاده قرار می دهید که تعدادی اپراتور در ان ورود و یا اصلاح اطلاعات انجام می دهند ، اینگونه فایلها جزء مهمترین دسته فایلهایی هستند که به شما می توانند کمک کنند.

روشهای بازیابی اطلاعات :

۱. **روش متوالی (Physical Sequential File Access)**

در این روش اطلاعات به ترتیب و رکورد به رکورد خوانده و یا نوشته می شوند

۲. **روش متوالی منطقی (Logical Sequential Access)**

در این روش کلیه رکورد ها بر اساس یکی از فیلد مرتب شده است و پردازش رکورد به رکورد انجام می شود

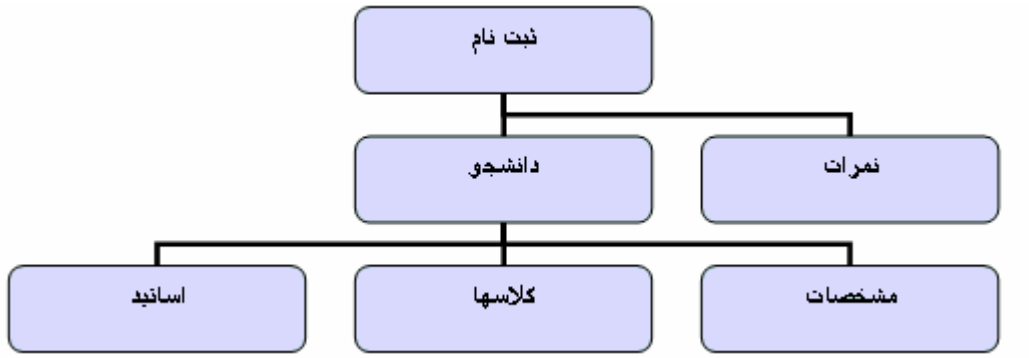
۳. **روش دسترسی مستقیم (Direct Access/Random Access)**

در این روش دستیابی به یک رکورد فقط از طریق کلید انجام می شود

انواع بانکهای اطلاعاتی :

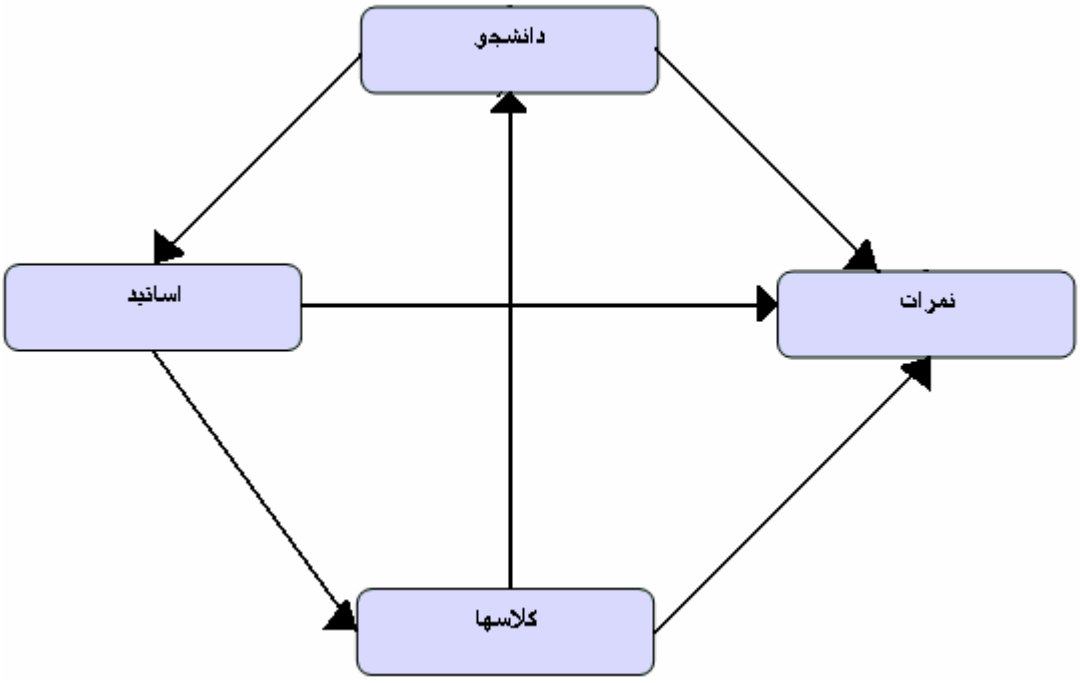
۱. طبقاتی (Hierarchical)

در این روش هر پایگاهی یا بصورت Parent و یا Child است . دستیابی به اطلاعات حتما می بایستی از ریشه شروع شود . مزیت این نوع مشابهت ان با طبیعت سازمانهای اطلاعات است
شکل زیر این نوع را نمایش می دهد



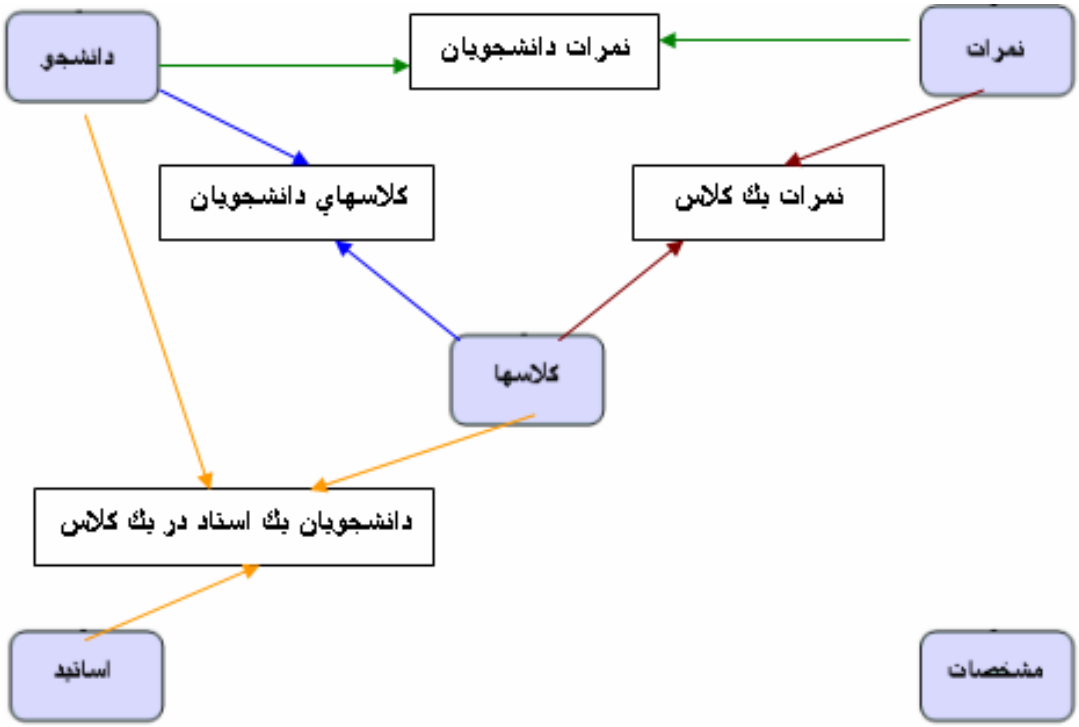
۲. شبکه ای (Network)

در این روش هر پایگاه عضوی (Member) از یک پایگاه دیگر(Owner) است . این روابط را مجموعه ها (Set) تعریف می کنند . از انجا که هر پایگاهی می تواند داراری این رابطه باشد شکل ناشی از این نوع پایگاهها دارار انعطاف بسیار زیادی است . برای پیدا کردن یک رکورد در این مجموعه ابتدا باید به سراغ مالک مجموعه و از طریق ان بسراغ عضو رفت . به شکل زیر دقت کنید



۳. نوع رابطه ای (Relational)

در این روش برخلاف دو روش قبل رابطها قطعیت ندارند ، بلکه با روابط ریاضی تعریف می شوند . نمونه این پایگاهها SQL است .در این روش میتوان از پایگاههای اصلی هر نوع جداولی که اطلاعات را به تفکیک یکسری صفات (فیلد ها) نگهداری می کنند ایجاد و روابط بین این جداول را از طریق دستوراتی مانند Select وJoin و... برقرار نمود .توانایی بالای این نوع پایگاهها در ایجاد جداول ، ارتباط بین جداول و یا با پایگاهها و مرتب سازی قدرت انعطاف بسیار بالایی را به این نوع بانکهای اطلاعاتی می دهد. شکل زیر را ببینید



در پرانتز

آنچه که تا اینجا عنوان شد حاصل دروس دوران دانشگاه ، کتب مختلف طراحی و تجربه های شخصی بنده بود . کتب مختلفی بعنوان مرجع استفاده شدند که کتاب "تحلیل و طراحی نظامهای کامپیوتری " تالیف دکترمحمود جهانی برای دسته بندی وترتیب مطالب و برخی مثالها بیشترمورد استفاده قرار گرفت. خواندن این کتاب را به همه توصیه می کنم و از همینجا از ایشان تشکروقدردانی می نمایم . در ادامه مطلب از مراجع دیگر استفاده ومعرفی خواهند شد.

شروع

با یک جمع بندی از قبل که انجام دهیم ، می توانیم به این نکته برسیم که ما برنامه را از نظر تحلیل و طراحی آماده کـردیم امـا در زمان پیاده سازی نیاز است که به چند سوال مهم جواب دهیم

۱. برنامه را چگونه پیاده کنیم
۲. در چه محیطی
۳. با چه ابزاری
۴. در چه مدت زمان
۵. چه نوع تکنیکهای برنامه نویسی استفاده کنیم

بنظرمن جواب این سوالها به ۲ سوال مهم از خود شما برمی گردد :

۱. با چه محیطهایی آشنا هستید
۲. تا چه حد به محیطهای مذکور مسلط هستید

اما بطور کلی از دید من جواب سوالهای فوق بدین صورت می باشد

۱. سه نوع پیاده سازی وجود دارد

a. **پیاده سازی از طریق مدل سازها** مانند Rational Rose/Modal Maker که در این حالـت بایسـتی از سالها تجربه در زمینه تحلیل ، طراحی و برنامه نویسی برخوردار باشید تا بتوانیـد هـم بـه سـادگی برنامـه ای را تولید هم سریعا خطاهای احتمالی سیستم را بررسی و رفع کنید . انشـاا... وقتـی ایـن بحـث خاتمـه پیـدا کـرد مقالاتی راجب به این مطلب خواهم نوشت .
b. **پیاده سازی از طریق الگوهای آماده** ماننـد Microsoft Access کـه در ایـن حالـت نیـازی آنچنـانی بـه دانستن علوم کامپیوتر احساس نمی شود . شما می توانید با یک دوره آموزش ۱۰ ساعته به تهیه و تولید هرنوع برنامه بانک اطلاعاتی بپردازید . این روش بنابر قدرتهای خوبی که در اختیار کاربر (و یا تولید کننده نـرم افـزار) قرار میدهد، دارای محبوبیت فراوانی است . اما این نوع برنامه ها نیز دارای مشکلات مخصوص به خود هسـتند. درمورد این روشها نیز می توانید مطالب زیادی (و یا افراد زیادی) پیدا کنید
c. **پیاده سازی از طریق کد نویسی** (برنامه نویسی) این روش که زمان زیادتری (به نسـبت ۲ روش قبـل)نیـز طلب می کند مورد بحث ما است که باالطبع توضیح کامل در ادامه بحث داده خواهد شد

۲. کد نویسی را می توان در نرم افزارهای مختلفی انجام داد. مروری اجمالی بر این نرم افزار ها خواهیم داشت یاد اوری می کنم سطرهایی که می نویسم تنها نظر شخصی من است ولی تحقیق در باره انها را به عهده خودتان می گذارم

a. نرم افزار هایی که Interpreter هستند مانند Fox/Basic : این نرم افزار ها بـدلائلی کـه عنـوان مـی کـنم محیط مناسبی برای کار حرفه ای نیستند.
   i. سرعت پایین : این گونه نرم افزار ها در واقع کد اجرایی تولید نمی کنند بلکه در هنگـام اجـرا خـط بـه خط انها از طریق برنامه مادر اجرا می شود. بدلیل اینکه یک یا دولایه ما بـین برنامـه و سیسـتم عامـل

ii. محیط راحت ولی محدود : توابعی که این برنامه ها در اختیار برنامه نویس می گذارند در ۹۰درصد موارد جوابگوی نیاز وی می باشد ، اما اینگونه برنامه نویسان اگر به این راحتی عادت کنند در تولید توابع خاص دچار مشکل خواهند شد

iii. محدودیت سیستم عامل : بسیاری از این برنامه ها دارای Versionهایی هستند که متکی به یک سیستم عامل خاص می باشند مثلا ویندوز . بدین جهت دورنمای تغییر سیستم عامل یک دورنمای تاریک خواهد بود (علی الخصوص برای برنامه های سنگین و بزرگ)

iv. محدودیت پارامترهای ورودی و خروجی سیستم : اینگونه برنامه ها بدلیل بسته بودن محیط Componentهای مورد استفاده در مورد ورودی ها و خروجی ها متکی به شرکت تولید کننده می باشند و حتی برخی مانند Visual Basic در کتب خود به ضعف در مورد گزارشگیری از پایگاههای اطلاعاتی ویا ارتباط ناقص از طریق OLEبا دیگر نرم افزار ها اعتراف دارند.

v. مشکلات دیگری مانند محدودیت حجم رکورد ها در پردازش ، سرعت پایین در اطلاعات بالا ، عدم کنترل برنامه نویس در پردازش خطاها ، نیاز به برنامه و یا درایورهای مادر و غیره نیز از جمله مشکلاتی است که این نرم افزار ها با آن مواجه هستند . اما در قبال آن می توان برخی نقاط قوت از جمله تغذیه شدن توسط برخی از برنامه های بزرگ مدل ساز مانند Rational Rose و همچنین امکان انتقال و یا ارتباط برخی از نرم افزار های تک منظوره مانند Auto Mapرا نیز ذکر کرد .

b. نرم افزارهای Interpreterتحت وب مانند Java / Asp : اینگونه نرم افزار ها را می توان دارای مزایای زیادی از جمله سرعت خوب در اینترنت ، عدم اتکا به محیط توزیع (سیستم عامل و یا Browser) ، قدرت فراوان در تولید و حمایت از طرف تعدادی از برنامه های مدل سازو غیره دانست . اما آنچه که باعث می شود تا این دسته از نرم افزار ها در مقوله صحبت ما کنار روند همان بحث DataBaseها در محیط تک کاربره و شبکه است .

c. نرم افزار های تولید کننده کد قابل اجرا مانند Delphi/C++ : اینگونه نرم افزارها

i. نیازی به حمایت یک برنامه مادر و یا نصب برنامه خاصی برای اجرا ندارند .

ii. دارای سرعت خوبی از نظر پردازش اطلاعات هستند ،

iii. وابستگی به ابزار پیاده سازی خود ندارند و قابلیت دریافت هرگونه Componentهای دیگر را نیز دارند

iv. برنامه نویس می تواند با نرم افزارهای تولیدی را برای استفاده در یک سیستم عامل دیگر در کوتاهترین زمان تبدیل کند .

v. بنابر دلیل ذکر شده در iii قدرت بالایی از نظر ارتباط نه تنها از طریق OLEکه ارتباط مستقیم با دیگر نرم افزارها را دارند

vi. وابستگی به یک درایور خاص بانک اطلاعاتی ندارند و تقریبا همه Data Providerها تحت پوشش قرار می دهند .

vii. وقدرتهای دیگر مانند تغییر Source Codeابزار مورد استفاده ، کنترل بالای برنامه نویس در پردازش خطا ، کنترل بالا در استفاده از Resourceهای سیستم ، قدرت بالا در طراحی و اجرای اجزاء مورد استفاده به صورت Dynamic و غیره . که بنده دلفی را میان این نرم افزارها انتخاب نموده ام.

۳. در مورد ابزار و تکنیکهای مورد استفاده می بایستی مشروحا در دلفی بحث کنم که جلوتر ضمن دسته بندی مشکلات و مسائل به این مساله خواهیم پرداخت .

۴. من همواره به کسانی که در مورد تولید نرم افزار با من مشورت دارند توصیه می کنم اگر مدت زمان اجرای پروژه به اختیار خودتان است زمان را با محاسبه دقیق پس از تحلیل اولیه صورت مساله و مشخص شدن دقیق صورت مساله تعیین کنید .و اگر حرفه ای هستید ویا سیستم را قبلا نوشته اید و اکنون می خواهید برای یک مشتری دیگر تغییراتی در آن بدهید ضریب حوادث غیرمترقبه را در اعلام زمان درنظر بگیرید .

۵. همانگونه که قبلا در شماره ۳ عنوان کردم در ادامه بحث به تکنیک ها نیز خواهیم رسید

اگر دوستانی هستند که در مورد مقاله ها سوال و یا ایراد و یا انتقادی دارند خواهش می کنم سوالهای خود را بفرمایند تا در صورت امکان به انها پاسخ دهیم .دلیل اینکه این مساله را اینجا قید کردم اینست که برنامه نویسان درمورد اصول توافق و در مورد پیاده سازی و علی الخصوص محیطهای پیاده سازی با یکدیگر بحث دارند . خوشحال می شوم نظرهای انها را نیز داشته باشم

برای پیاده سازی هر سیستمی ما نیاز داریم که عملیاتی که روی هر بخش یک سیستم تعریف می گردد را بشناسیم بطورکلی عملیات روی یک سیستمی که با پایگاههای اطلاعاتی کار می کند بصورت زیر می باشد :

۱. ساخت پایگاه در Data Providerمورد نظر
۲. ساخت فرمها ی اصلی و حاشیه ای
۳. اضافه نمودن اطلاعات جدید Insert
۴. اصلاح اطلاعات وارد شده Update/Edit
۵. حذف اطلاعات وارد شده Delete
۶. جستجو در اطلاعات Search
   a. تک مورد Find
   b. مورد به مورد Find Next
   c. مجموعه ای از اطلاعات Filter/Select
   d. .....
۷. گزارش گیری و خروجی
   a. از اطلاعات خام List
   b. از مجموعه اطلاعات M/D
   c. از جستجوها
   d. از نتایج محاسبات و یا استنتاج ها
   e. برحسب دسته بندی
   f. چارتهاو نمودارها

g. .....

٨. پشتيبان گيری Back Up

٩. بازيابی پشتيبان Restore

١٠. امنيت و تعيين ميزان دسترسی Security

١١. زيبا سازی برنامه

١٢. مسائل حاشيه ای ديگر (امنيت برنامه / تست /)

بگذاريد يک به يک پيش برويم

اما قبل از شروع خدمتتان عرض می کنم که Data Providerهايی که ما با آن کار می کنيم از نوع

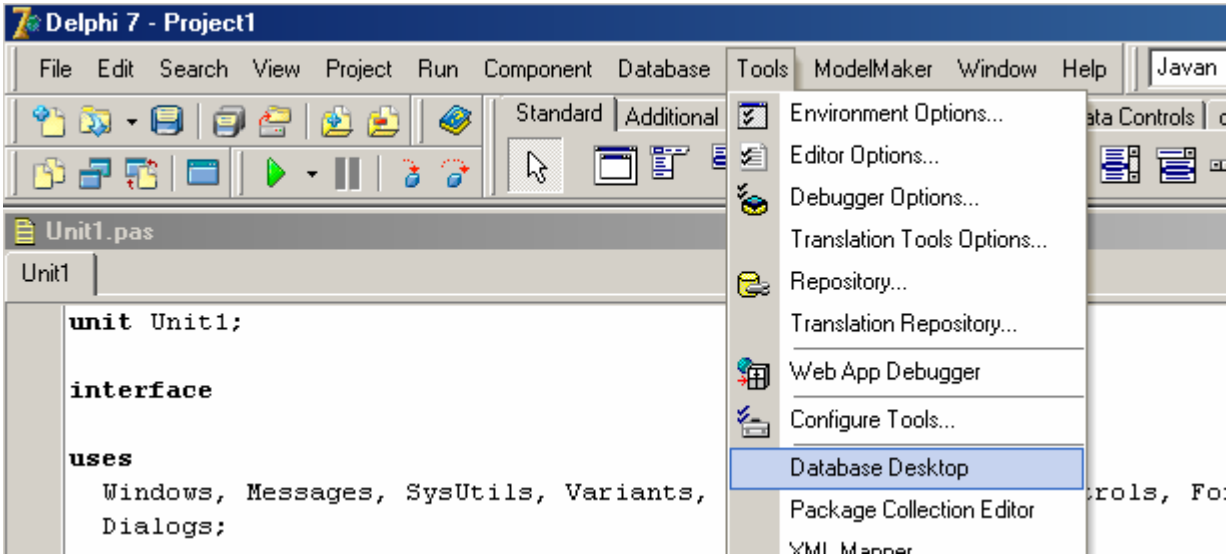
- Paradox 7.0
- SQL Server 7.0/2000

می باشند .

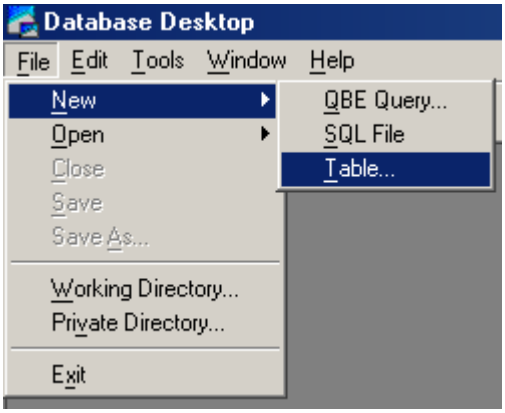
ساخت پایگاه در Data Provider مورد نظر

برای ساخت پایگاههای پارادوکس در دلفی می توانید از Data Base Desktop 7.0 که روی خود دلفی نصب است استفاده کنید برای اینکار به شکلهای زیر دقت کنید :

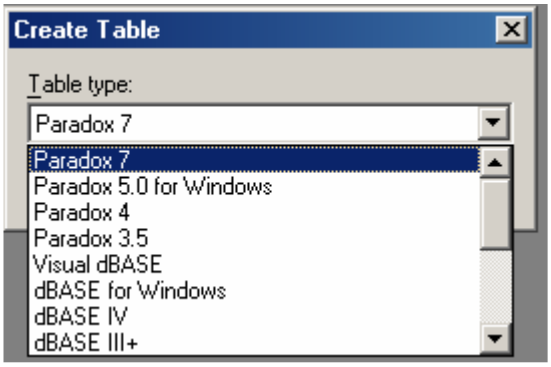
۱. احضار DatabaseDesktop 70



۲. ایجاد یک Tableجدید

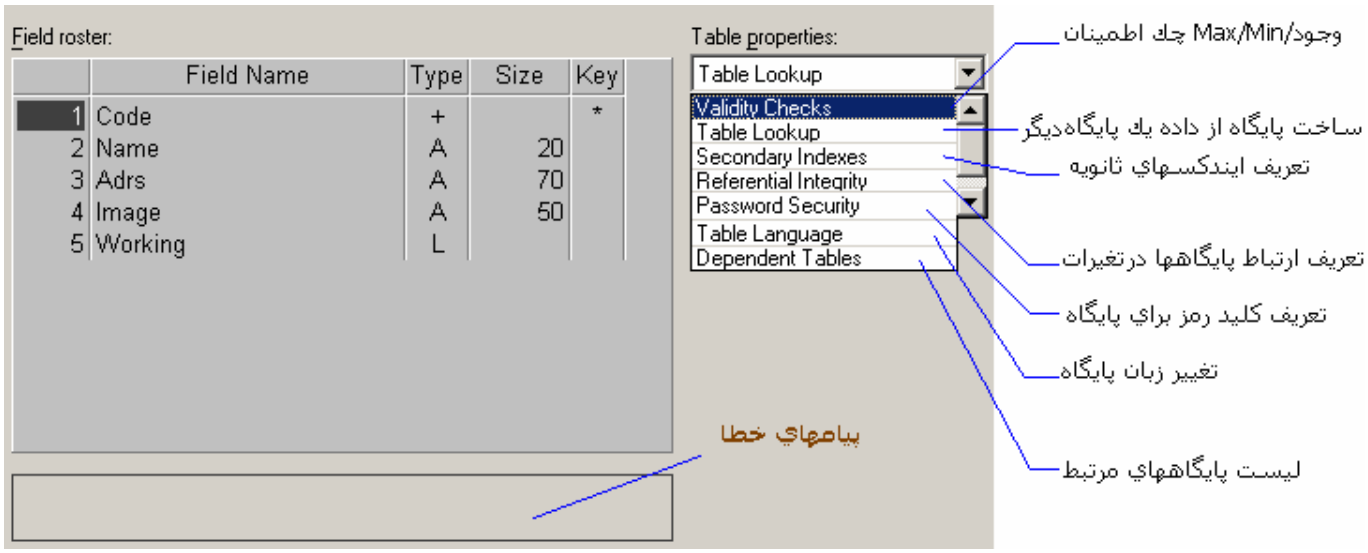


۳. انتخاب نوع پایگاه



۴. نحوه تعریف فیلدهای پایگاه

۵. امکانات اضافه در DataBase Desktop



توجه

۱. در شکل ۲ مشاهده می شود که می توان علاوه بر ایجاد یک پایگاه جدید
   a. یک Query By Example (پرس و جو با کمک مثال)
   b. یک فایل SQL

   نیز تشکیل داد که در آینده درباره استفاده از این دو ابزار سودمند توضیح خواهم داد

۲. در شکل ۳ می توانید علاوه بر Paradoxنوع دیگری را نیز انتخاب کنید که در این صورت صفحات تشکیل پایگاه و امکانات جانبی ان متفاوت خواهد بود

۳. در شکل ۴ با وجود طیف گسترده ای از انواع متغیر های پیشنهادی ، بالشخصه پیشنهاد می کنم از انواع زیر استفاده کنید
   a. N : برای کلید ها ، مبالغ ، اعداد (صحیح و اعشاری)و غیره

b. A : نام ، فاميل ، تاريخ ، ساعت ، نام عكس ، نام فايل توضيحي و غيره (اصولا فكر ميكنم هر چيزي را كه با Alphabetic تعريف قدرت خوبي براي كنترل خواهيم داشت به شرط اينكه محدوده طول را درست تعريف كنيم مثلا تاريخ را A(10) يا نام عكس را بدون مسير A(30))

c. + : كليد هاي مرتب كه نيازي به نمايش ندارند و فقط ترتيب ركورد ها را برحسب انها مي توان مرتب نمود

d. L : فيلدهايي كه يا درست يا نادرست يا تعيين نشده هستند مانند جنسيت

۴. در مورد امكانات اضافي مي توانم چند توضيح كوتاه بدهم . اما كار حرفه اي روي انرا از طريق Help يا اگر نياز شد از طريق يک مقاله ديگر توضيح خواهم داد

a. Validity Checks : در اين قسمت به ازاي هر فيلدي مي توان يكسري پارامترهاي كنترلي را يكبار براي هميشه (One Time For Ever) تعريف نمود . مانند ميزان مجاز (MinوMax) ، الزامي براي ثبت اطلاعات (Required) ، فرمت دريافت و يا نمايش اطلاعات (Picture) . از انجا كه برنامه نويسان تمايل دارند كنترل ها را در برنامه تعيين كنند بعدا در مورد شبيه سازي اين قسمت در برنامه مفصلا صحبت خواهيم كرد

b. Table Lookup : مي توان از اين قسمت با كمک پايگاه جاري و تركيب ان با يک پايگاه ديگر پايگاهي تشكيل داد كه تبادل فيلد داشته باشند . اين قسمت را نيز در برنامه خواهيم گفت

c. Secondary Indexes : در اين قسمت شما مي توانيد ايندكسهاي ديگري را براي ترتيب نمايش و يا جستجوي خود تعريف كنيد . بعدا اين قسمت را نيز به روش ديگري شبيه سازي خواهيم كرد

d. Referential Integrity : مي توانيد در اين بخش پايگاههاي مرتبط را به هنگام تغيير مقادير اين پايگاه مشخص كنيد . درواقع نوعي Update هوشمند و بي دردسر . اين بخش را نيز در تكنيكها شبيه سازي مي كنيم

e. Password Security : گذاشتن كلمه رمز براي پايگاه . اين امكان قدرتمند پارادوكس براي آماتورها بد نيست ولي چون همه مي دانند چگونه اين امكان را بشكنند از اين قسمت صرفنظر مي كنيم

f. Table Language : تعيين Char Set مورد استفاده در ثبت اطلاعات . اين امكان براي تبادل پايگاهها بين دو سيستم عامل با Char Set نامتناجنس(Win98/Win 2000) بنظر مفيد است . من تا حالا تست نكرده ام

g. Dependent Tables : نمايش ليست پايگاههايي كه در Referential Integrity معرفي كرده اند

۵. نكات مهم در نام فيلد ها

a. در نام فيلدها فاصله بكار نبريد

b. نام فيلدها را فارسي ننويسيد

c. سعي كنيد يک قاعده ثابت براي نامگذاري فيلدها پيدا كنيد مثلا "TellNumber"

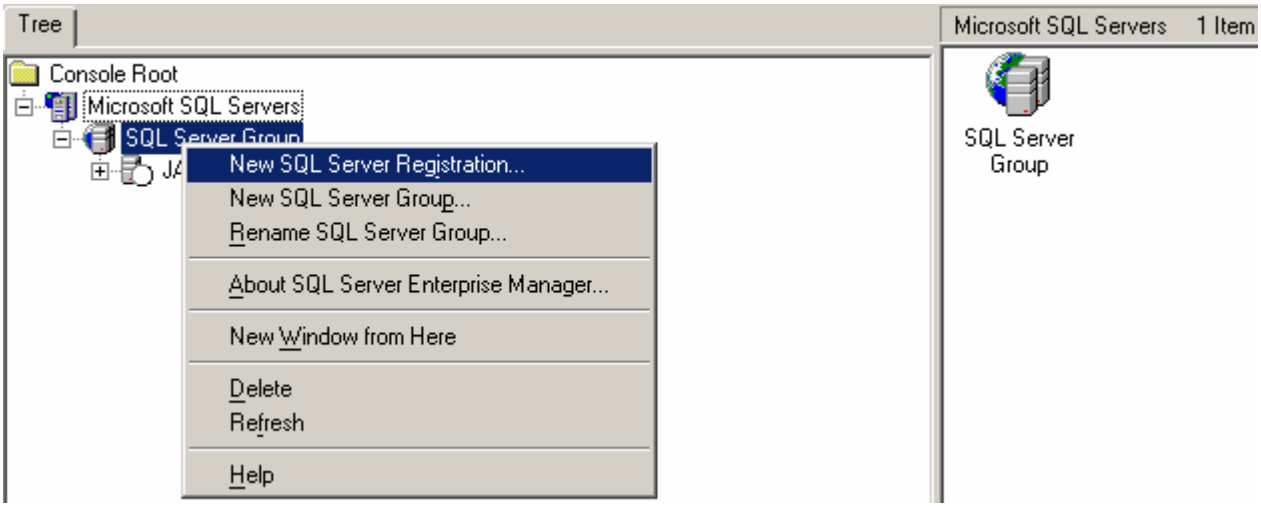
d. فيلدهاي عددي كه با انها محاسبه انجام نمي دهيد بصورت Aدر نظر بگيريد

e. سعي كنيد از Reserve Wordها استفاده نكنيد مثلا Date / Table / IF بجاي ان از OutDate / InTable / IFMale استفاده كنيد

ساخت پایگاه در Data Provider مورد نظر ۲

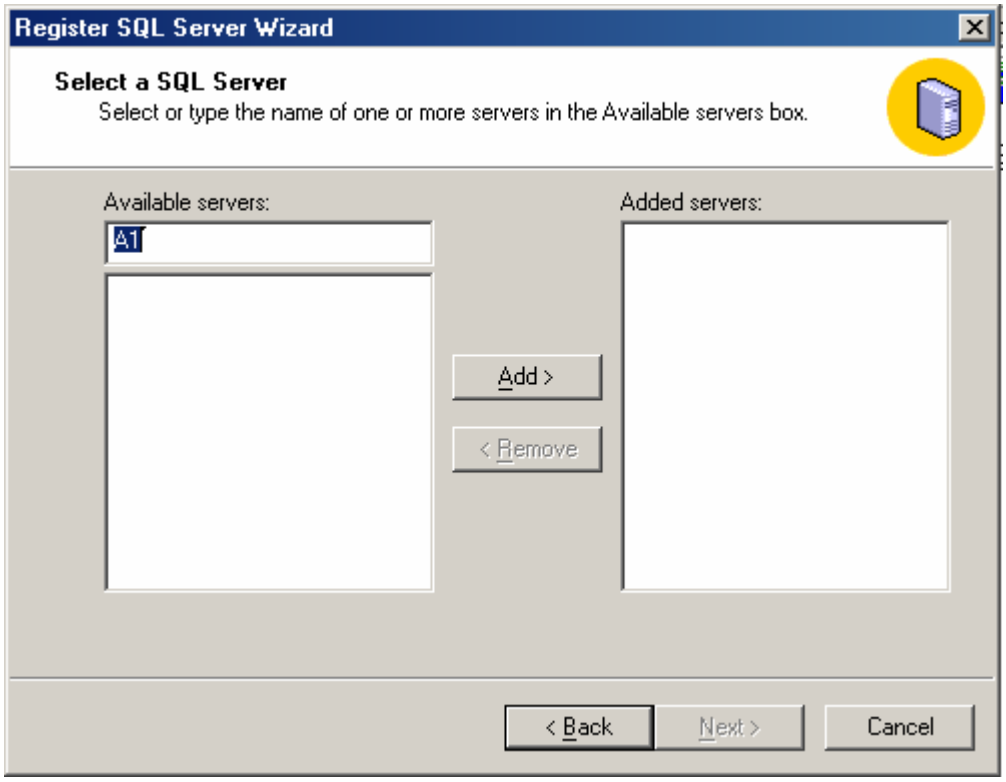
برای ساخت یک پایگاه ابتدا بایستی یک SQL Server را رجیستر کرده باشید به اشکال زیر دقت کنید

۱. شروع

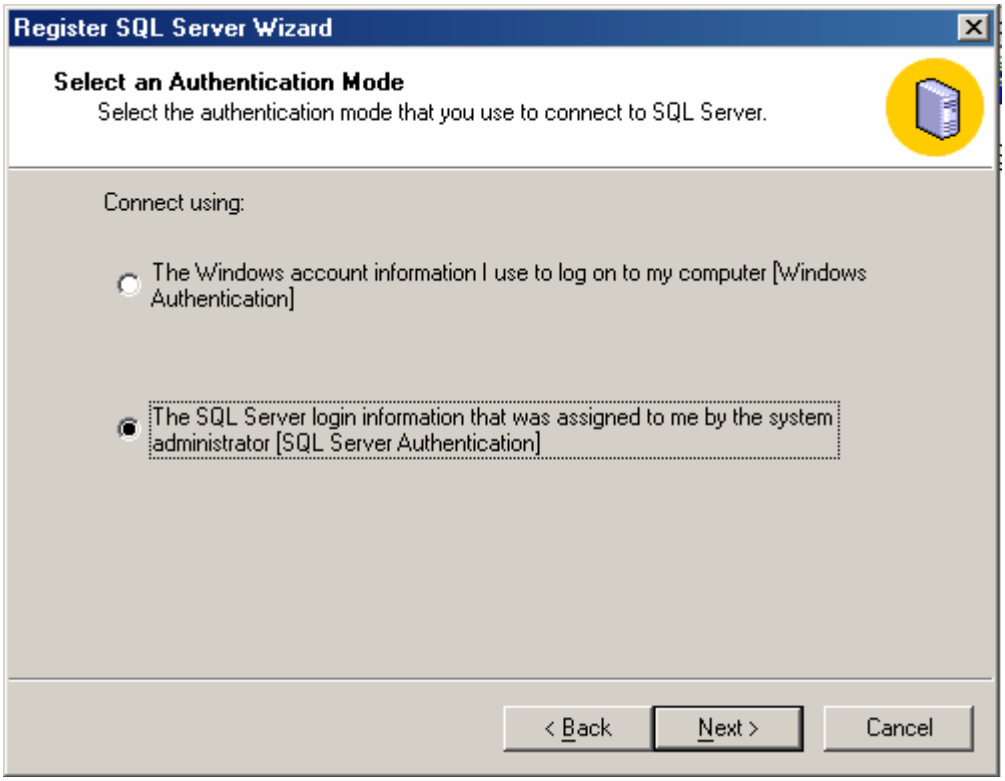


۲. راهنمای قدم به قدم

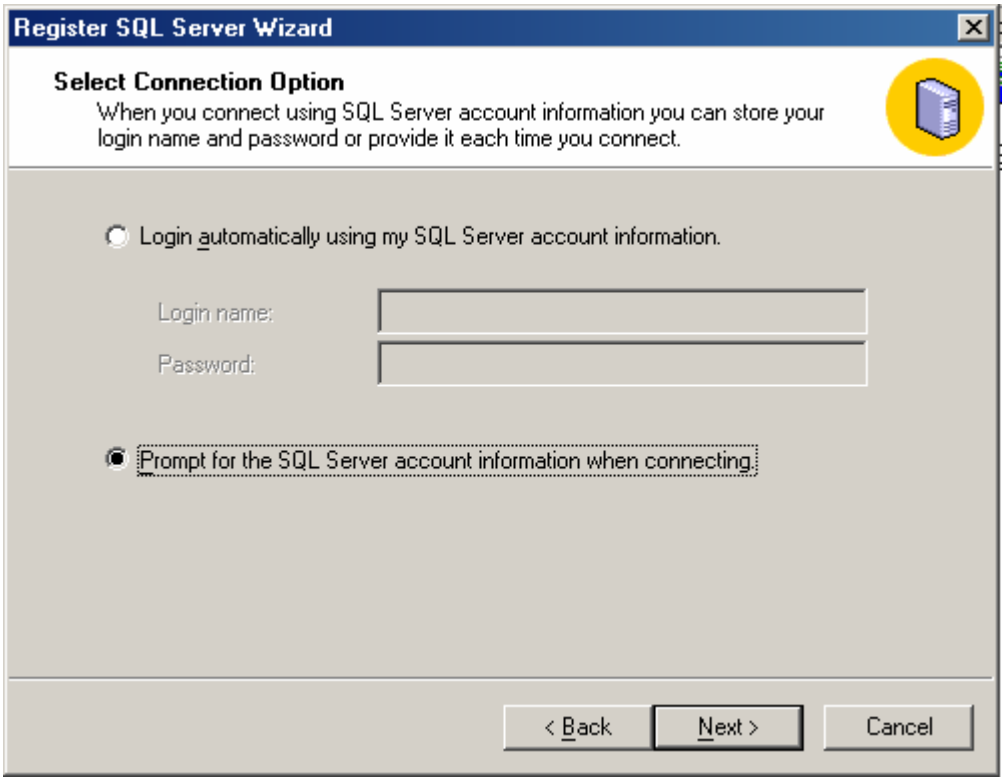
۳. مشخص کردن نام سرور (Local/Network) باکمک تایپ یا انتخاب از لیست



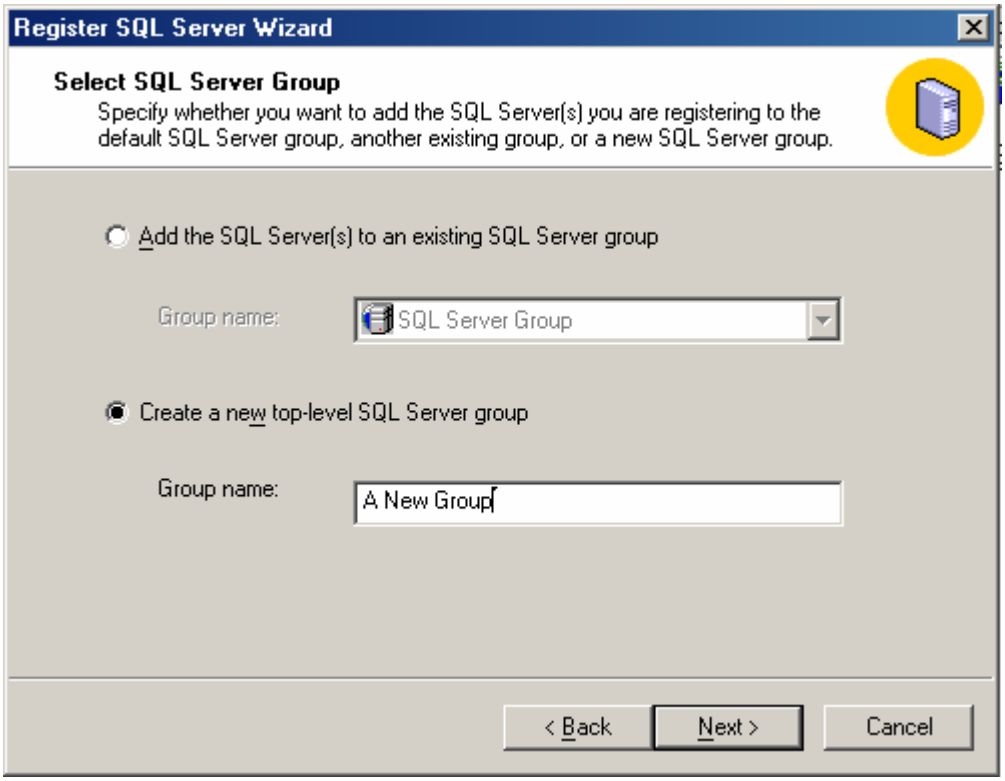
۴. انتخاب نوع ورود به Accountاز طریق الف) Accountهـای ویندوز ۲) از طریـق تعـاریف کـاربران در SqlServer



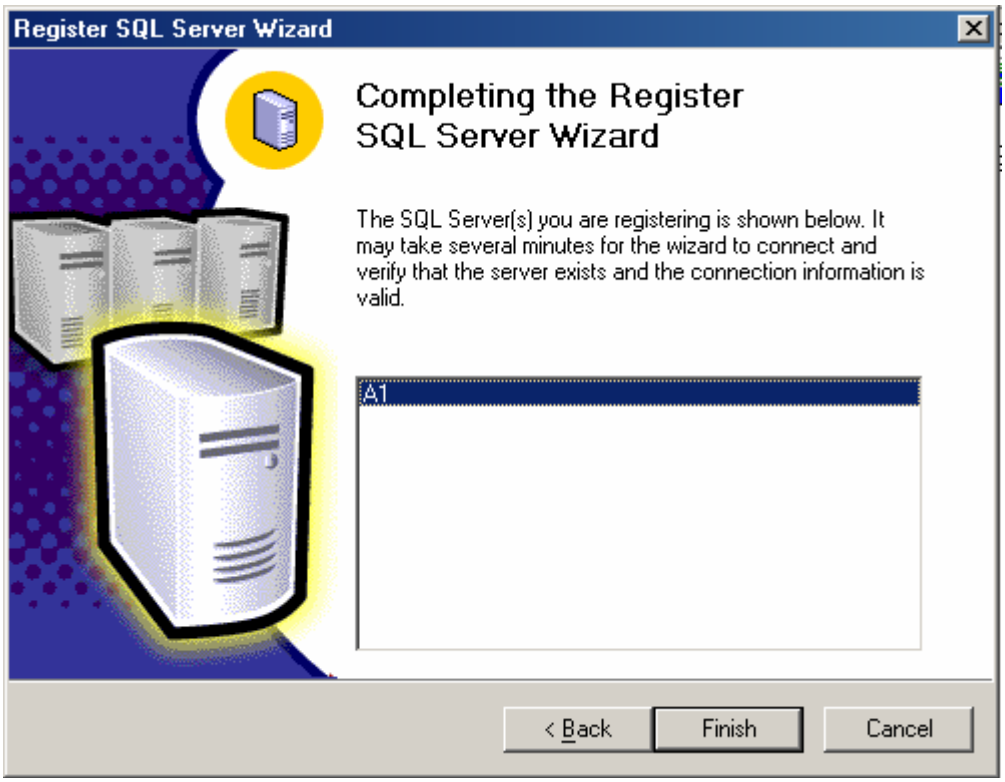
۵. نحوه ورود به اطلاعات Account بصورت پيش فرض يا در هنگام ورود هر بار سوال شود



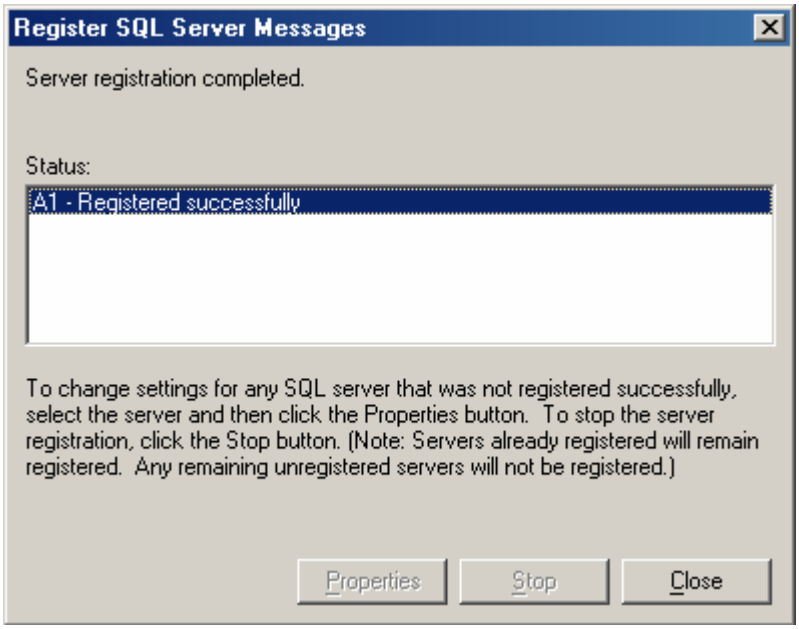
۶. ايجاد اين سرور در گروه قبلی يا در گروه جديد



۷. اماده شدن برای ثبت سرور

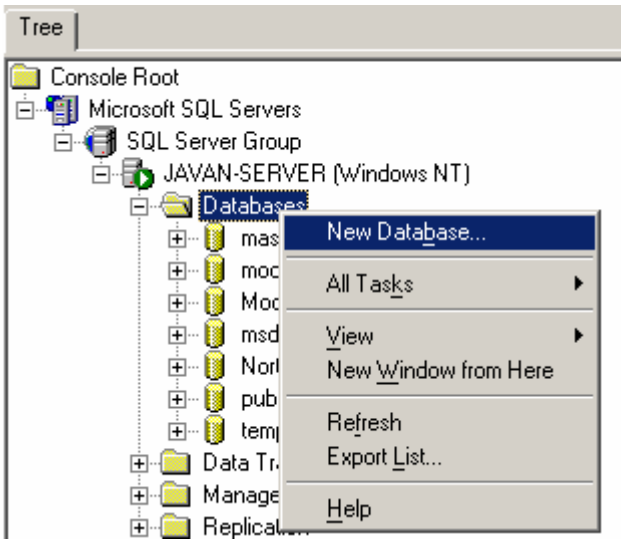


۸. پیام موفقیت در صورت درست بودن عملیات

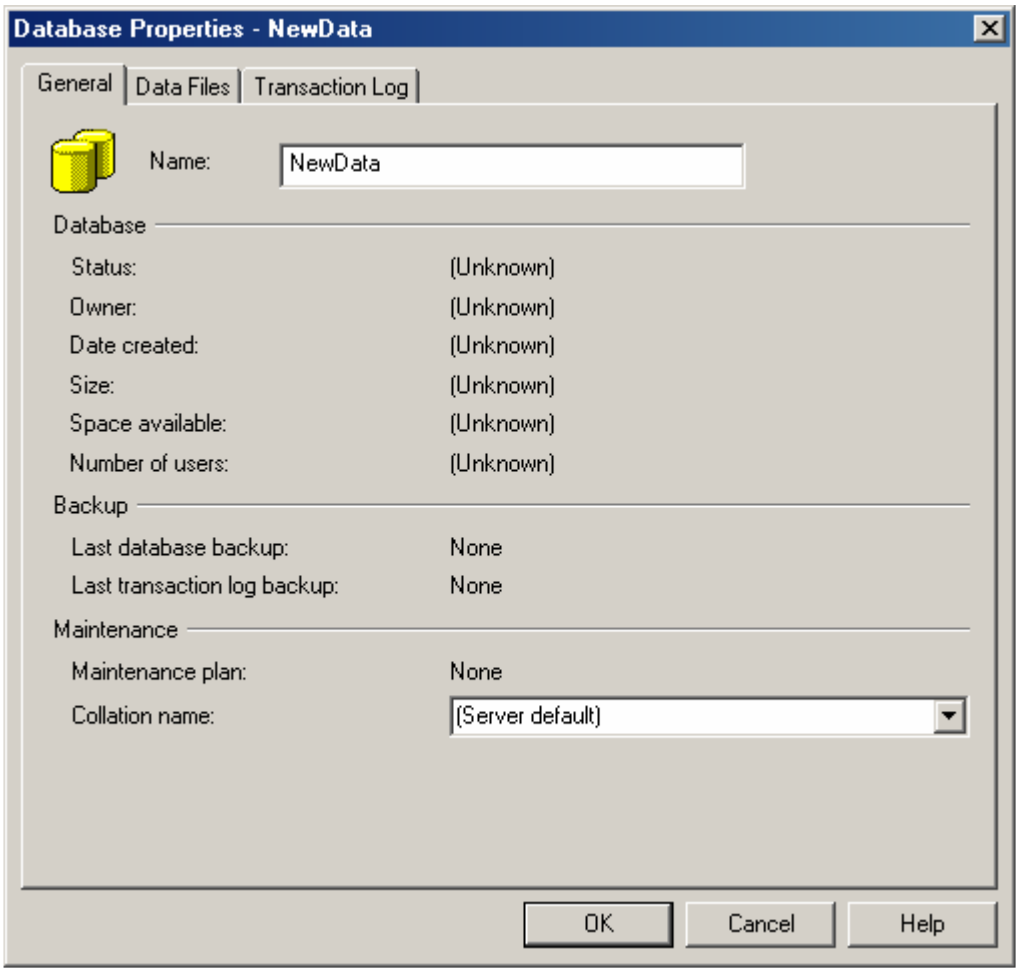


حالا سرور شما آماده ثبت Data Base شما است. Data Base محلی است که شما می توانید در ان پایگاههای خـود را ثبت کنید برای اینکار بترتیب قدمهای زیر را دنبال کنید .هدف ما ثبت یک DataBaseبنام NewData است

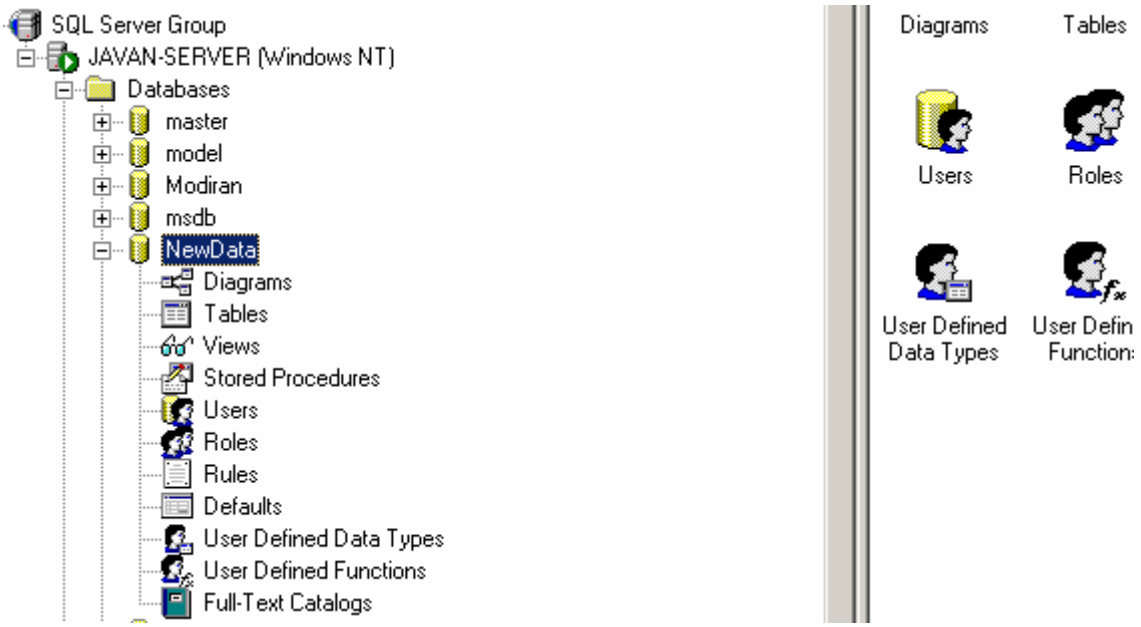
١. ايجاد DataBase



٢. وارد كردن نام DataBase



۳. DataBase ایجاد شده است



چند نکته

بحث های مربوط به Sql Server بسیار مفصل است اما آنچه را که می توان در ۲ جدول بصورت خلاصه مطرح نمود بدین صورت است

پایگاههای موجود در SQL SERVER

master	این پایگاه تمامی اطلاعات مدیریتی SQL Server را در خود دارد تمامی تعاریفی که در بخشهای : User/Store Proc /Tables و ... می نمایید در این قسمت نگهداری می شود(System Catalog)
model	پایگاه مدل برای ایجاد دیگر پایگاهها ، به این معنی که هر انچه در این پایگاه اضافه می شود بطور اتوماتیک در دیگر پایگاهها نیز اضافه خواهد شد(مثال یک User به این پایگاه اضافه و در همه ببینید)
Northwind	پایگاه مثال
msdb	پایگاه تعیین زمانبندی سیستم و ثبت تاریخ پشتیبان گیری
tempdb	پایگاه نگهداری اطلاعات موقت سیستم . این محتویات این پایگاه پس از قطع ارتباط اتوماتیک از بین می رود

ICONهای مورد استفاده در SQL Server و معانی آن

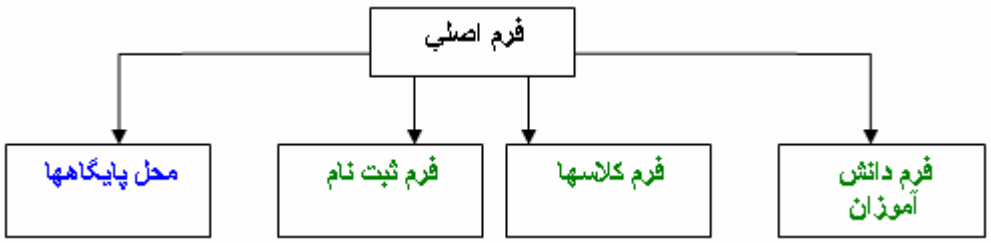
 Tables	جداول : محل نگهداری اطلاعات
 Views	تعریف پایگاههای مجازی که از ارتباط روی چند پایگاه بدست می آید
 Stored Procedures	برنامه های ذخیره شده در SQL Server که باعث افزایش سرعت در پردازشها در سرور می گردد
 Users	تعریف کاربران
 Roles	تعریف حق دسترسی کاربران به پایگاهها
 Rules	تعیین مقادیر مجاز برای یک فیلد خاص از جدول اطلاعاتی
 Defaults	مقدار پیش فرض که در صورت خالی ماندن فیلد مذکور در آن فیلد قرار می گیرد

فرمها ، ابزار و ارتباط

انچه تا اینجا گفتیم در خارج از محیط دلفی بود . می خواهیم در داخل محیط دلفی شروع به برنامه نویسی بانکهای اطلاعاتی کنیم . یاد آور می شوم من فرض را بر این گذاشته ام که شما حداقل آشنایی را با دلفی دارید .

فرمهای برنامه

ترتیب فرمهای برنامه دارای اهمیت خاصی نیست . اما اگر انها را به صورت منظم بگذارید بنفع خودتان خواهد بود .
مثال : فرض کنید می خواهید یک سیستم ساده ثبت نام دانش آموز را پیاده سازی کنید . پیشنهاد من برای فرمها بشکل زیر است .



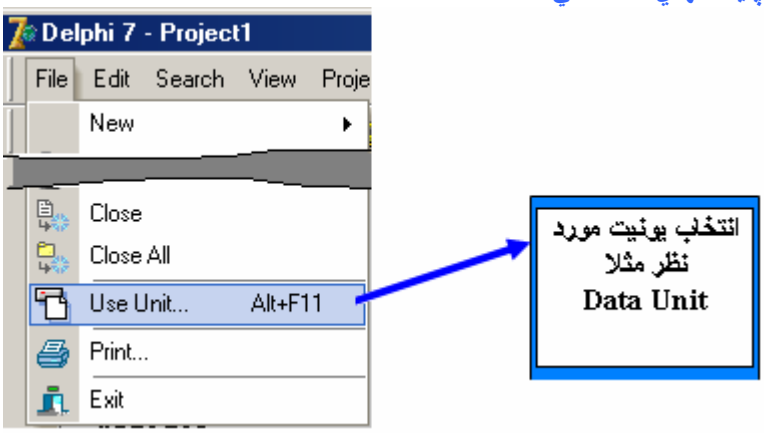
در شکل فوق می توان چند نکته را مشاهده نمود

۱. فرم ورود اطلاعات دانش آموزان ، فرم ورود اطلاعات کلاسها و فرم ثبت نام دانش آموزان در کلاسها جدا گانه قرار گرفته اند
۲. محل پایگاهها نوع خاصی از فرمها بنام **Data Module** می باشد که بعدا ابزار پایگاهها را در آن قرار می دهیم
۳. ارتباط با هر یک از فرمها زیرین در فرم اصلی تعریف شده است
۴. فرمهای دانش آموزان / کلاسها و ثبت نام از **DM** برای تغذیه اطلاعات استفاده می کنند.

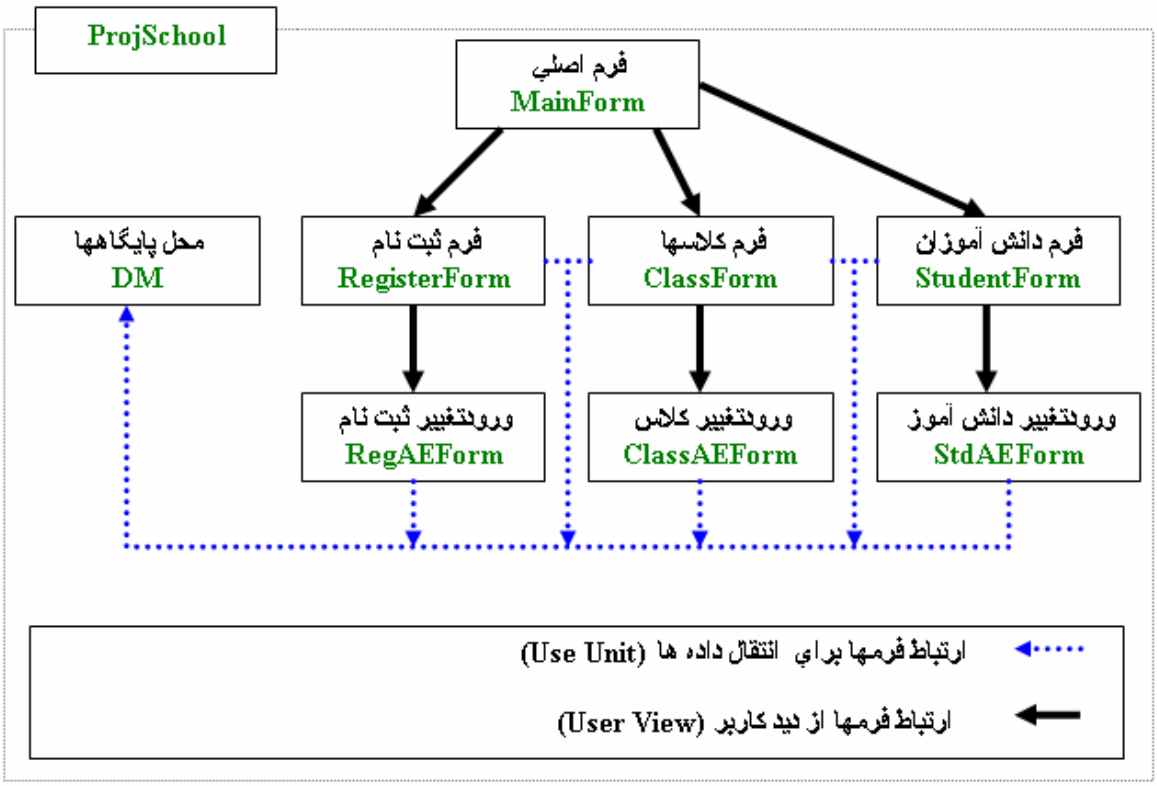
نامگذاری فرمها (علی الخصوص در پروژههای واقعی و بزرگ) دارای اهمیت فراوانی است . من معمولا از این نحوه نامگذاری استفاده می کنم

	قاعده نام گذاری	مثال
فرمها	**Name**Form	**StudentsForm**.Dfm
یونیتها	**Name**Unit	**StudentsUnit**.Pas
DataModule	**DM**	**DM**.Dfm
DM Unit	**DataNameUnit**	**DataUnit.Pas**
پروژه	**Proj**Name	**ProjSchool**.Dpr

همچنین در فرمهایی که از DM بعنوان مرکز اطلاعات برنامه استفاده می کنند می توانید از طریق منوی Fileو سپس Use Unit ارتباط را برقرار کنید . همچنین این کار را برای فرم اصلی نسبت به فرمهای زیرین نیز باید انجام دهید .
البته دلفی اگر اینکار را انجام ندهید و نیازی به ارتباط پیدا کند خود بصورت هوشمند اینکار را انجام می دهد. شکل زیر نحوه کار را به شما نشان می دهد.



بنابر این شکل اولیه بصورت زیر تبدیل خواهد شد.



من روشهای مختلفی را برای ورود و اصلاح اطلاعات امتحان کردم اما در نهایت روشی که در شکل فوق می بینید را انتخاب کردم که بیشترین امنیت و کمترین میزان به هدر رفتن Resource ها را دارددر عین اینکه به برنامه نـویس امکـان کنتـرل بـالا بـا کمترین کد می دهد . این روش را بعدا مفصلا خواهیم دید

پایگاهها و ارتباط

همانگونه که می دانید روشهای متفاوتی برای ارتباط با پایگاههای اطلاعاتی وجود دارد . این روشها در دلفی به صورت زیر تقسیم بندی شده است :

- **BDE** : این مدل از ساختار Borland DataBase Engine استفاده می کند و اعمال مدیریتی توسط توابع API انجام می شود.این مدل بدلیل اینکه بخش عمده ای از منابع موجود روی سرور و سرویس گیرنده را در اختیار می گیرد مورد انتقاد است
- **ADO** : در این مدل از اجزاء ActiveX استفاده می شود . در اغلب موارد (بجز تعریف Alias) شبیه به BDE است و تفاوتی در آنها نه از بابت سرعت و نه امکانات و نه منابع دیده نمی شود .
- **dbExpress** : این مدل همانند BDEمی باشد با این تفاوت که دارای سرعت بالاتری در مورد پردازش می باشند .دلیل این مسئله :

۱. استفاده از DataSetهای یکطرفه است یعنی شما فقط در جهت رکورد بعدی می توانید حرکت کنید ولی نمی توانید به رکورد قبلی برگردید .
۲. انجام عملیات اضافه ، تغییر و حذف بدون استفاده از حافظه موقت یعنی شما مستقیما عملیات را در پایگاه ثبت می کنید و امکان بازگشت انچه انجام داده اید وجود ندارد
۳. فیلتر گذاری روی رکوردها امکان پذیر نیست

همانگونه که می بینید در ۳ نکته گفته شده هم دلیل سرعت بالای dbExpress قید شده است و هم اینکه مشخص شده که نقاط ضعف dbExpress در کجاست . ولی درمقایسه با ADO و BDE این مدل دارای سرعت بسیار بالاتری در Query ها می باشد

- **InterBase** : این مدل برای ارتباط با بانکهای اطلاعاتی مبتنی بر InterBase می باشد.

یک برنامه ساده و ادامه بحث

با یک مثال ساده بحث تمرین و بعد مطالب بیشتری را درمقایسه عنوان می کنیم

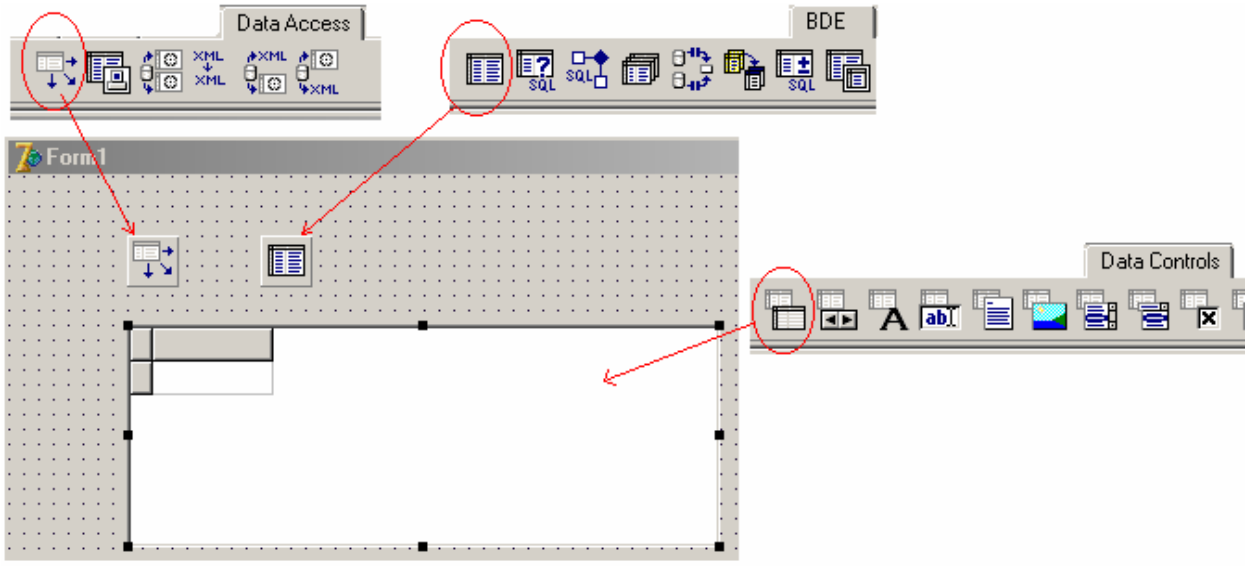
یک فرم ساده تشکیل دهید و اجزاء مورد نیاز را از بخشهایی که در شکل نشان داده شده است بروی فرم قرار دهید

این اجزاء عبارتند از

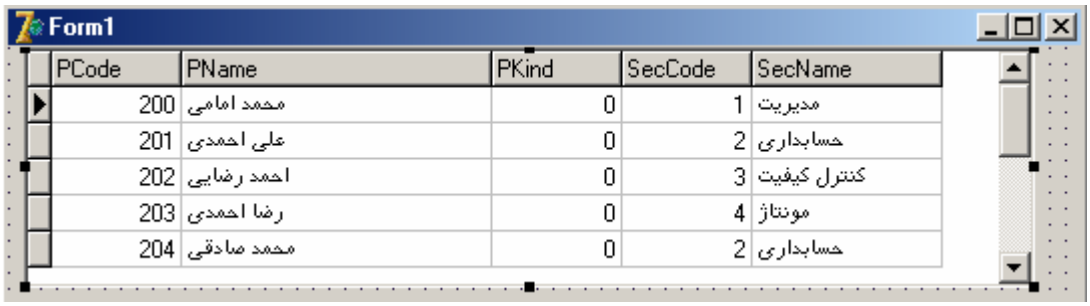
۱. TTable : برای ارتباط برنامه با بانکی که در DataBase Desktop و یا SQL Server ساخته ایـد . پـس از قـرار دادن ان بروی فرم خواص زیر را در Object Inspector تغییر دهید
   a. DataBaseName : مـی توانـد Alias تعریـف شـده و یـا مسـیر پایگـاه بـرروی کـامپیوتر باشـد . مـثلا C:\Test\Db
   b. TableName : اگر DataBaseName را درست تعریف کرده باشید می توانید لیست این بخش را باز کرده و نام پایگاه مورد نظر را انتخاب کنید مثلا Books.Db و یا Dbo.Books (Sql)
   c. Active : پایگاه را برای فعالیت باز می کند

۲. TDataSource : پخش کننده TTable بروی اجزاء بانک اطلاعاتی است . پس از قرار دادن آن فقط کافی است خاصیت DataSet انرا به نام TTableخود تغییر دهید . مثلا Table1 که اینکار را از طریق لیست همین قسمت نیز می توانید انجام دهید

۳. DbGrid : نمایش دهنده اطلاعات در پایگاه می باشد . برای استفاده از این عنصر نیز فقط خاصیت DataSource آنرا به نام TDataSourceخود تغییر دهید .مثلا DataSource1 که باز از طریق لیست بازشونده نیز قابلیت انتخاب وجود دارد



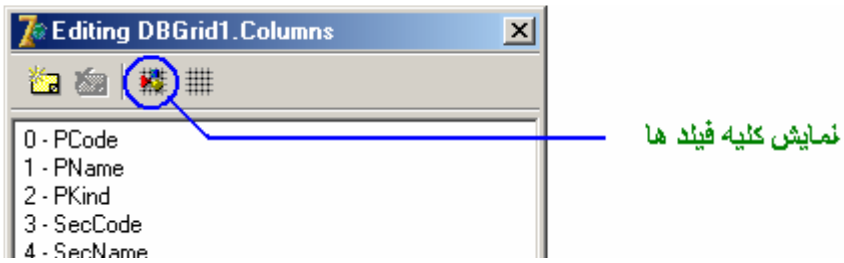
پس از تنظیم پارامترهای گفته شده می توانید تیتر فیلدها را در DBGrid ببینید.



همه چیز در یک DBGrid نهفته است

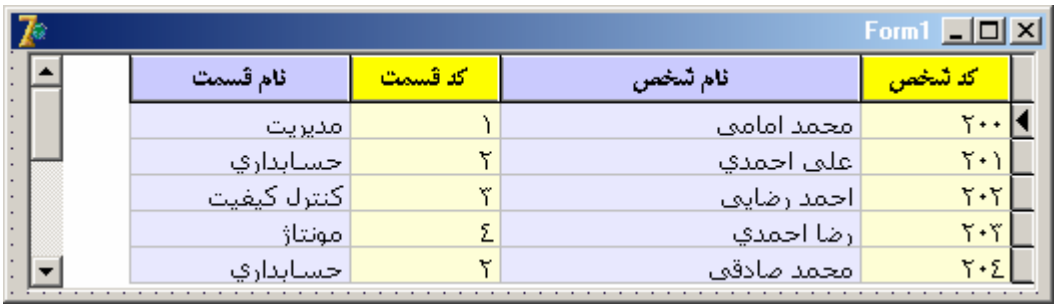
اما هنوز کارهای زیادی مانده است . در مرحله بعد بهتر است کمی به DbGrid خود برسید.

با دو کیلک روی DbGrid فرم زیر نمایش داده خواهد شد. که البته خالی است . بعد از با فشار کلید Add All Fields که در شکل نمایش داده شده است ، می توانید کلیه فیلدهای پایگاه را ببینید . ممکن است نخواهید برخی از فیلدها توسط کاربر دیده شود ، انها را حذف کنید .



حالابروی تک تک موارد این فرم می توانید کلیک کرده و در Object Inspector این موارد را به سلیقه خود تغییر دهید .برخی از خصوصیاتی که می توانید تغییر دهید عبارتند از

من برخی از خصوصیات را تغییر دادم و DbGrid مثال را به شکل زیر در آوردم



البته برای فارسی شدن اعداد روی فرم کلیک کنید و BiDiMode را برابر RightToLeft قرار دهید

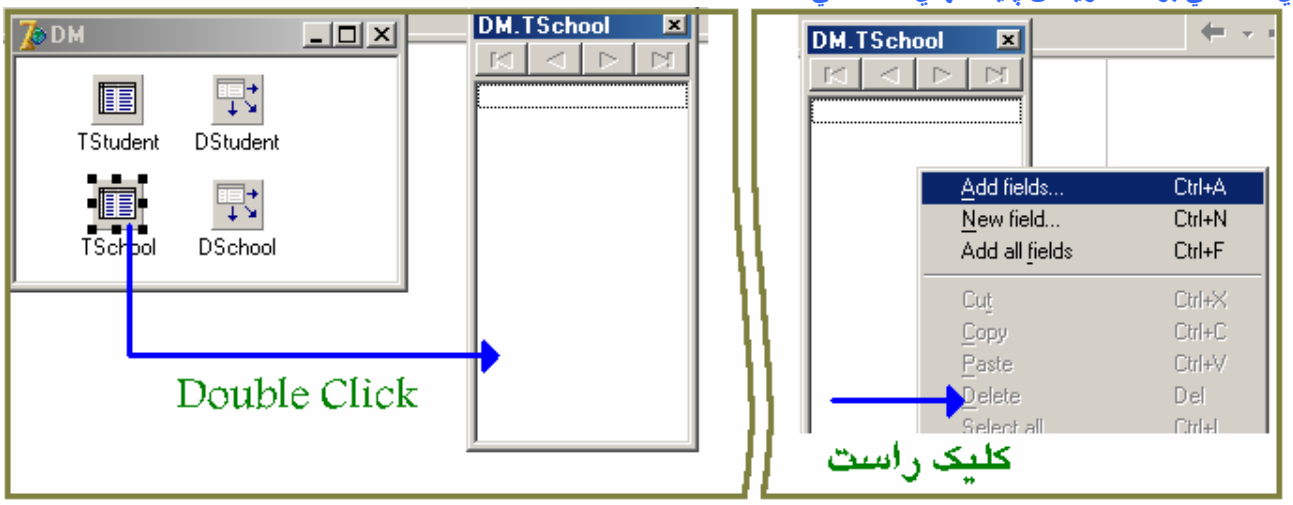
DbGrid از دید من یکی از مهمترین عناصر مرتبط با بانکهای اطلاعاتی است . از این عنصر بعدا استفاده های زیادی خواهیم کرد

Calculated Fields/Lookup Fields

فیلدهای Look Up

اگر خاطرتان باشد بهنگام نرمالایز کردن پایگاهها گفتیم که فیلدهای تکراری در پایگاههای مختلف را حذف می کنیم . اکنون می خواهیم این مساله را بررسی کنیم که اگر بخواهیم یک فیلد از یک پایگاه را در پایگاه دیگر نمایش دهیم چگونه می توان این کار را انجام داد . این مساله را با توضیح یک مثال نمایش می دهیم .

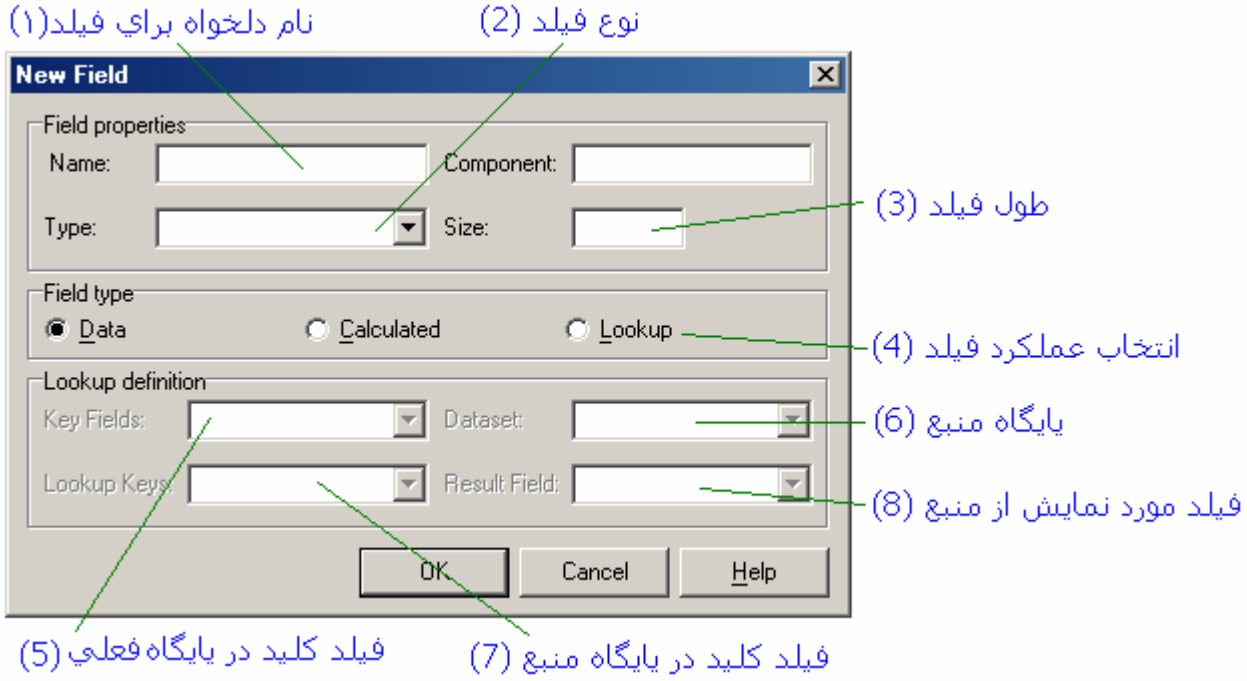
فرض کنید یک پایگاه برای ثبت اطلاعات دانش آموزان داریم(TStudent) . پایگاه دوم نیز وضعیت تحصیلی این دانش آموز است . چیزی که در پایگاه دوم(TSchool) ثبت می شود تنها کد دانش آموز است . برای نمایش دیگر اطلاعات دانش آموز می



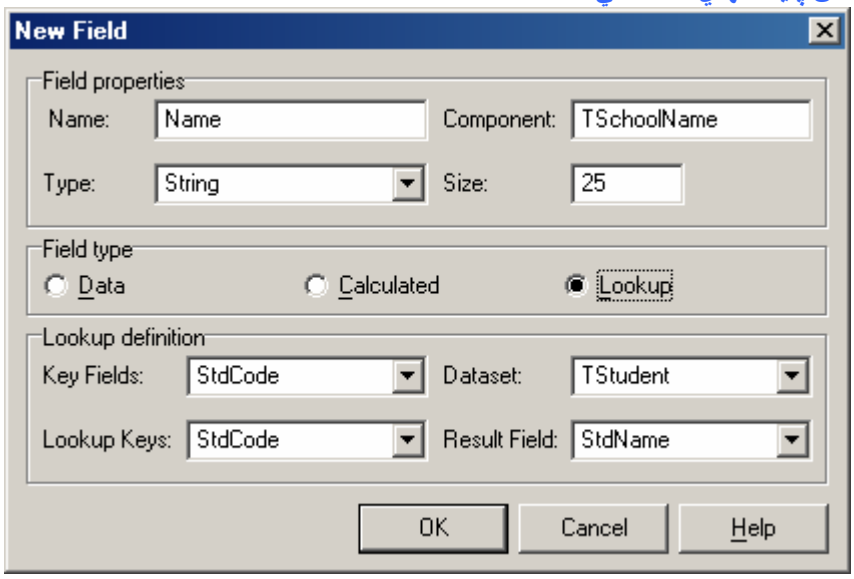
شکل (۱)

بعد از با انتخاب Add All Fields تمامی فیلدها را می توانید نمایش دهید .

حال دوباره کلیک راست روی همان پنجره انجام دهید و New Field را انتخاب کنید . پنجره ای مشابه فرم زیر خواهید دید :



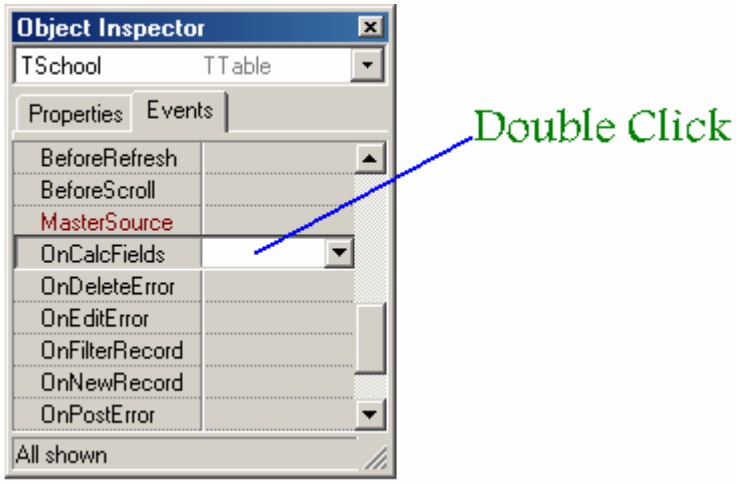
بترتیب شماره های شکل موارد را پر کنید . شکل بعدی پرشده این فرم را نمایش می دهد



حالا شما فیلدی را در اختیار دارید که بصورت LookUp تعریف شده است .

فیلدهای Calculated

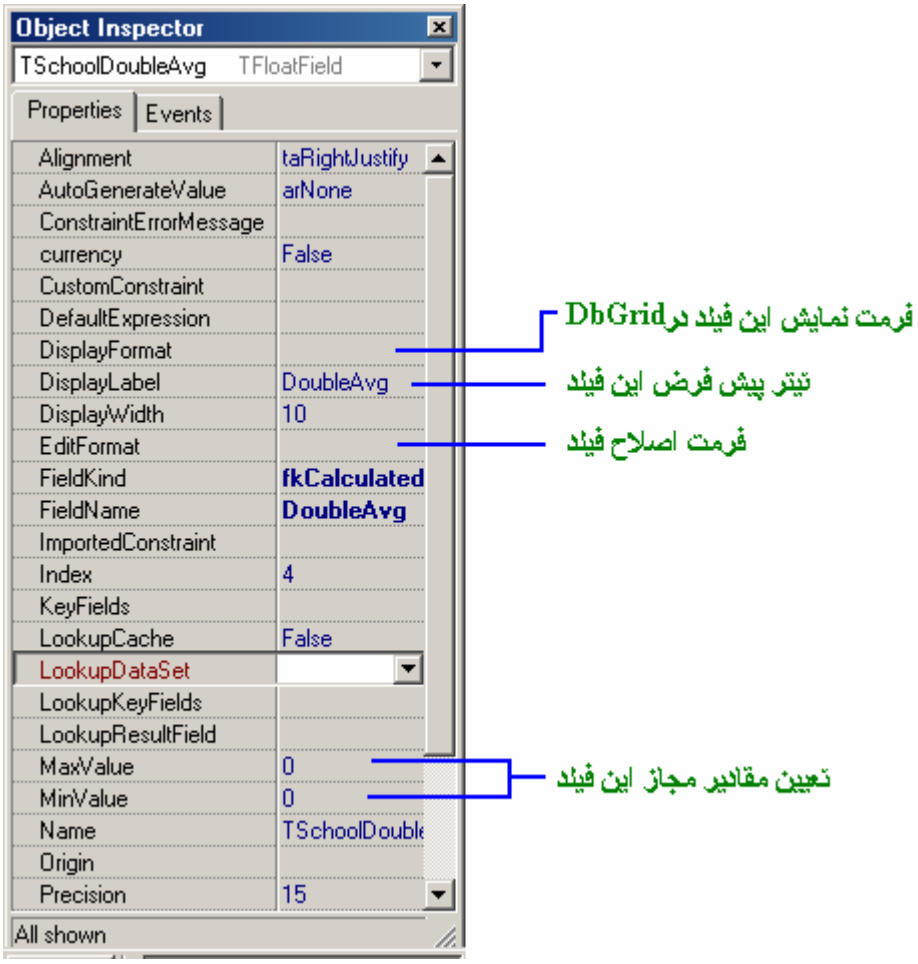
در مرحله بعدی بسراغ فیلدهای محاسباتی یا قابل استنتاج می رویم . یاد آور می شوم اینگونه فیلدها فقط در مورد یک رکورد عمل می کند . برای ایجاد اینگونه فیلدها ، مانند شکل ۱ New Field را انتخاب و سپس همان روندی که در شکل ۲ نمایش داده شد را طی می کنیم ، با این تفاوت که نحوه عملکرد فیلد را Calculated انتخاب می کنیم . پس از همانگونه که در شکل می بینید در Object Inspector در OnCalcFields کلیک کرده و



کد موردنظر را می نویسیم، مثلا اگر بخواهیم فیلدی داشته باشیم(DoubleAvg) که به طور اتوماتیک ۲ برابر فیلد معدل(Avg) را نمایش دهد :

```
Procedure TDM.TSchoolCalcFields(DataSet: TDataSet);
Begin
  DataSet['DoubleAvg']:=DataSet['StdAvg']*2;
End;
```

در شکل زیر می توان برخی دیگر از امکانات Field Editor را مشاهده نمود



شکل (۶)

بعد از این مثالهایی که در هر بخش وجوددارد را در سایت www.Javan-Soft.Com خواهم گذاشت . برنامه فوق را با نام **Ch06Ex1** دریافت کنید .

توضیح ۱

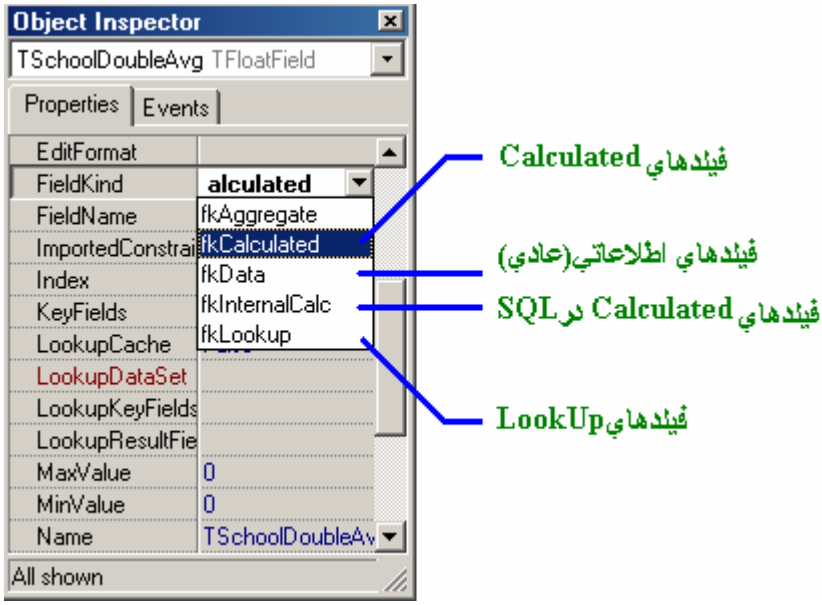
در برنامه مثال می بینید که از کد زیر استفاده شده است. توضیح هر خط به شما در فهم آن کمک می کند.

```
procedure TDM.DataModuleCreate(Sender: TObject);  // درهنگام ایجاد DM
```

```
begin
   TStudent.DatabaseName:=GetCurrentDir;//مسير پايگاه را برابر مسير جاري قرار بده
   TSchool.DatabaseName:=GetCurrentDir;
   TStudent.Open;
   TSchool.Open;                          //  پايگاه را باز كن
end;
```

توضيح ٢

در شكل ۶ چنانچه خصوصيت **Field Kind** را نگاه كنيد ، برخي انتخابها را خواهيد ديد . اين انتخابها را در شكل توضيح داده ام



Data Control

امروز می خواهیم مرور سریعی داشته باشیم بر ابزاری که در کنترل اطلاعات بانکها کاربر دارند

ابتدا جدول زیر را ببینید

	1. DbNavigator	کنترل کننده عملیات روی اطلاعات (اضافه ،حذف، بعدی و...)
	2. DbText	نمایش دهنده مقدار ثبت شده در یک فیلد
	3. DbEdit	علاوه بر نمایش تغییر دهنده مقدار ثبت شده در یک فیلد
	4. DbMemo	نمایش و تغییر یک فیلد از نوع توضیحی
	5. DbImage	نمایش و تغییر یک فیلد از نوع عکس
	6. DbListBox	نمایش یکسری مقادیر و انتخاب یکی از آنها برای ثبت در فیلد
	7. DbComboBox	نمایش یکسری مقادیر و انتخاب یکی از آنها برای ثبت در فیلد
	8. DbCheckBox	انتخاب وضعیت True/False در یک فیلد از پایگاه
	9. DbRadioGroup	نمایش یکسری مقادیر و انتخاب یکی از آنها برای ثبت در فیلد
	10. DbLookUpListBox	نمایش محتویات یک فیلد از پایگاه دیگر و ثبت در یک فیلد
	11. DbLookUpComboBox	نمایش محتویات یک فیلد از پایگاه دیگر و ثبت در یک فیلد
	12. DbRichEdit	نمایش و تغییر یک فیلد از نوع توضیحی با خصوصیات متن
	13. DbCtrlGrid	اجازه نمایش اجزاء فوق بصورت رکورد به رکورد
	14. DbChart	ترسیم یک چارت با توجه به مقادیر یک یا چند فیلد

همه ابزارهای فوق در رابطه با بانکهای اطلاعاتی می باشند. کارآیی هریک از آنها متفاوت است .

۱. به عنوان مثال DbText در مورد نمایش مقدار یک فیلد بسیار مفید است . اگر همین عنصر را با DbCtrlGrid ترکیب کنیم منویی از مقادیر یک فیلد بوجود می آید . معادل اینکار استفاده از DbLookUpComboBox ویا DbLookUpListBox می باشد اما حسن استفاده از این ۲ عنصر را در تعدد موارد مورد نمایش است.بعدا مثالی در این مورد حل خواهیم کرد.

۲. بیشترین کاربرد DbLookUpComboBox ویا DbLookUpListBox در فیلدهای LookUp می باشد .

۳. در مورد استفاده از DbNavigator فقط برای کارهای سبک فوری توصیه مـی شـود و DbImage و DbMemo هم بایستی با احتیاط استفاده شوند.

برای ایجاد این ارتباط نیاز است برخی تنظیمها را در Object Inspector انجام دهیم . این تنظیمها در جدول زیر قیـد شـده اند.

DbNavigation DBCtrlGrid	DataSource : پایگاه مورد کنترل در مورد DBCtrlGrid این DataSource برای تمامی فرزندان تعمیم می یابد
DbText DbEdit DbMemo DbImage DbCheckBox DbRadioGroup DbRichEdit	DataSource : پایگاه مقصد DataField : فیلدمورد نظر در پایگاه مقصد
DbListBox DbComboBox	DataSource : پایگاه مقصد DataField : فیلدمورد نظر در پایگاه مقصد Items : مواردی که کاربر می تواندانتخاب کند
DbLookUpListBox DbLookUpCombobox	DataSource : پایگاه مقصد DataField : فیلدمورد نظر در پایگاه مقصد KeyField : فیلدی که مقدار آن از پایگاه مبداء به پایگاه مقصدمنتقل می شود ListSource : پایگاه مبداء ListField : فیلدمورد نمایش در پایگاه مبداء
DbChart	بعدا مقاله مفصلی در باره آن خواهم نوشت

حال بگذاریم نگاه سریعی به یک برنامه کوچک به عنوان مثال داشته باشیم . بترتیب قدمهای زیر را انجام دهید:

۱. File → New→Application
۲. File → New → Data Module
۳. نام DataModule1 را به DMتغییر دهید
۴. در DM یک Table و یک DataSourceقراردهید
۵. نتظیمهای مربوط به این دو عنصر را طبق آنچه در قسمت قبل گفته شد انجام دهید
۶. بروی فرم اصلی یک DbGrid کار بگذارید
۷. File → Use Unit و سپس Unitمربوط به DMرا انتخاب کنید
۸. DbGrid را به DataSource متصل کنید
۹. بروی فرم یک کلید بنام "**اضافه** " بگذارید
۱۰. یک فرم جدید بسازید File → New → Form

۱۱. بروی این فرم عنصر DBEdit را بـرای اتصـال بـه پایگـاه و ۲ کلیـد "**ثبـت**"و"**انصـراف**" را قـرار دهیـد. در Object Inspector خصوصیت Modal Result کلید "**ثبت** " را برابر MrOk و "**انصـراف**" را برابـر MrCancel قراردهیـد .همچنین برای DbEdit کار گذاشته شده طبق جدول بالا خصوصیات را تنظیم کنید

۱۲. در رویداد OnClick فرم اول کلید "**اضافه**" کد زیر را تایپ کنید

```
Procedure TForm1.Button1OnClick(Sender:TObject);
Begin
    Dm.Table1.Insert;
    If Form2.ShowModal=MrOk Then
      Dm.Table1.Post
    Else
      Dm.Table1.Cancel;
End;
```

۱۳. همانگونه که پس از اجرای برنامه خواهید دید با فشار کلید "**اضافه**" فرم دوم نمایش داده خواهد شد و اطلاعـات شـما را خواهد گرفت و چنانچه کلید "**ثبت** " را فشار دهید اطلاعات ذخیره و در غیر اینصورت عمل ثبت انجام نخواهد شد

۱۴. با ایجاد یک کلید بنام "تغییر" همان کد را دوباره بنویسید با این تفاوت که بجای کلمـه Insert از کلمـه Editاسـتفاده کنید

۱۵. با ایجاد یک کلید بنام "حذف" کد زیر را بنویسید

```
Procedure TForm1.Button3OnClick(Sender:TObject);
Begin
    If    MessageDlg('آیـا بـرای حـذف مطمـئن هسـتید؟',MtConfrimation,[MbYes,MbNo],0)  =MrYes
Then
        Dm.Table1.Delete;
End;
```

اتمام پاییز ۸۲

محمد وکیلی