

## پیشگفتار :

ما در عصری زندگی می کنیم که آن را عصر اطلاع رسانی یا قرن ارتباطات نامیده اند، لذا می توان اذعان نمود که کامپیوتر بهترین ابزار ارتباطی در آن عصر می باشد. امروزه کامپیوتر نه تنها به عنوان یک وسیله شخصی یا ابزار گروهی خاص مورد استفاده قرار می گیرد، بلکه به عنوان یک وسیله ارتباطی مهم در جهان مطرح می باشد و به همین دلیل است که کاربرد آن روز به روز در جهان گسترش می یابد به گونه ای که در همه کشورهای جهان از جایگاه ویژه ای برخوردار می باشد. با گسترش کاربرد کامپیوتر در جهان این رشته به صورت گسترده ای در زمینه های مختلف توسعه یافته و هر روز نیز به این گسترش افزوده می گردد.

پیشرفت سریع تکنولوژی، به ویژه فناوری اطلاعات و ارتباطات (ICT) روز به روز چشم اندازها و افق های روشن تری را جهت تسخیر قلم علمی، فنی و صنعتی و حل مشکلات و مسائل بشر ارائه می کند و تک تک افراد جامعه را به تلاش مضاعف در کسب مهارت های رایانه ای و کاربرد آنها در سایر علوم ملزم می سازد، به نحوی که امروزه افراد و جوامع ناتوان در بکارگیری فن آوری های جدید رایانه را بی سواد تلقی می کنند. یکی از رشته های علمی مفید در زمینه کامپیوتر کار با پایگاه داده ها و نرم افزارهای بانک اطلاعاتی است که در زمینه های مختلف تجاری و اداری و وب سایت و ... کاربرد دارد.

از این نرم افزارهای می توان به ORACLE ، MS SQL ، ACCESS ، MS SQL ,SERVER اشاره نمود.

ACCESS یک نرم افزار بانک اطلاعاتی قوی و آسان است که بسیاری از کاربران بانک اطلاعاتی تمایل زیادی به استفاده از آن نشان می دهند. در این پروژه درسی نیز از این نرم افزار خوب و سودمند بهره گرفته شده است.

زبان برنامه نویسی BASIC نیز به دلیل راحتی و آموزش سریع و آسان از گذشته دور مورد استقبال بسیاری از کاربران و برنامه نویسان بوده است.

زبان VISUAL BASIC نسخه تحت ویندوز زبان Basic است که البته بسیار قوی تر و پیشرفته تر از نسخه تحت DOS آن عمل می کند و بسیاری از برنامه های پیشرفته و سطح بالا و

UML SSADM طراحی و بسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
پیچیده مورد نیاز شرکت های بزرگ به ویژه برنامه های چند رسانه ای (Multi Media) با استفاده  
از آن نوشته می شود.

من نیز در این پروژه درسی از این زبان قوی که دارای قابلیت ارتباط با بانک اطلاعاتی به نحو  
مطلوبی است، بهره گرفته ام.

اصل و پایه پروژه من نحوه ارتباط با پایگاه داده ها، در یک زبان سطح بالا است که شامل باز  
کردن بانک اطلاعاتی، دیدن اطلاعات موجود در بانک اطلاعاتی، ویرایش یا حذف داده های بانک  
اطلاعاتی، ذخیره داده های جدید در بانک اطلاعاتی و لیست کردن داده های بانک اطلاعاتی طبق  
خواسته و نظر کاربر می باشد.

بانک اطلاعاتی برنامه من مربوط به یک فروشگاه نوت بوک می باشد. در این فروشگاه علاوه بر  
خود نوت بوک، کیف نوت بوک، لوازم جانبی نوت بوک یا PC، اجزاء اصلی داخل نوت بوک یا PC و  
نیز خود PC عرضه می شود.

برنامه VB در این پروژه طبق Query های خاصی که در ارتباطش با بانک اطلاعاتی ACCESS  
درخواست می کند، لیست های متفاوتی از اطلاعات آن را به نمایش در می آورد.

### نرم افزارهای مورد استفاده :

۱- Microsoft Access 2003

از برنامه های مجموعه Microsoft office 2003

۲- Microsoft Visual Basic

از برنامه های مجموعه Microsoft Visual studio 6.0

امید آنکه روزی بتوانم این پروژه را تا سطح خیلی پیشرفته تر و وسیعتری گسترش دهم تا در  
زمینه های واقعی قابلیت استفاده داشته باشد.

همچنین امیدوارم این کتابچه و اطلاعات ارائه شده در آن بتواند برای سایر دانشجویان رشته  
کامپیوتر و علاقه مند به یادگیری VB , Access مفید واقع شود.

فاطمه مهدوی، تیر ۱۳۸۶

## مقدمات Visual Basic

### مفاهیم اولیه

روش های برنامه نویسی معمولاً به دو نوع کلی تقسیم می شوند:

\* برنامه نویسی تابع گرا یا ساخت یافته

\* برنامه نویسی شیء گرا

**الف) برنامه نویسی تابع گرا یا Procedural:** این روش برنامه نویسی متن گرا نیز نامیده

می شود روش قدیمی تری است که مبتنی بر توابع می باشد و معمولاً یکی از توابع، اصلی است. با شروع برنامه، کنترل برنامه به اولین دستور تابع اصلی رفته و به ترتیب شروع به اجرای دستورات، از اولین دستور تا آخرین دستور تابع، می کند و مواقعی هم ممکن است فراخوانی هایی به یک یا چند زیر برنامه داشته باشد که با هر بار فراخوانی، کنترل برنامه به ابتدای تابع مورد نظر رفته و دستورات آن تابع را به ترتیب اجرا می نماید آن گاه با اتمام دستورات تابع مذکور، کنترل برنامه به ابتدای تابع مورد نظر رفته و دستورات آن تابع را به ترتیب اجرا می نماید آن گاه با اتمام دستورات تابع مذکور، کنترل برنامه به دستور بعد از دستور فراخوانی برمی گردد و روال برنامه تا رسیدن به آخرین دستور تابع اصلی ادامه می یابد. بعد از اجرای آخرین دستور تابع اصلی یا رسیدن به دستور خاتمه ی برنامه، برنامه پایان می پذیرد؛ مانند اکثر زبان های برنامه نویسی که تا به حال با آن کار کرده ایم، که از جمله بیسیک، پاسکال و C.

**ب) برنامه نویسی شیء گرا (Object Oriented):** برنامه نویسی شیء گرا دارای اصول و

ویژگی هایی است که با پیدایش ویندوز و مفاهیمی چون «چند وظیفه ای» که در آن سیستم عامل وجود دارد، مطرح شد. در این روش، بر خلاف روش قبلی، به جای استفاده از تابع اصلی که وظیفه ی کنترل تمام برنامه را بر عهده دارد از مفاهیمی هم چون کلاس، خصوصیت و شیء استفاده می شود. بنابراین برای یادگیری بهتر این روش نخست باید با مفاهیم گفته شده آشنا شویم.

در محیط زندگی با مفهوم شیء آشنا هستیم. همان طور که می دانید یک شیء چیزی است مادی که دارای خصوصیتی است، از جمله این که می تواند در مقابل برخی اتفاقات که ممکن است

برایش رخ دهد از خود واکنش‌هایی نشان دهد. پس با این حال می‌توان گفت که هر شیء به همراه

سه جنبه‌ی ذیل شناخته می‌شود :

- خواص
- رفتار یا متد
- روابط

خواص ویژگی‌هایی است که مشخص کننده‌ی حالت فعلی شیء است؛ برای مثال می‌توان گفت رنگ یک میز قهوه‌ای است یا قد یک شخص ۱۷۰ سانتی‌متر است. در این مثال میز و انسان شیء هستند و قهوه ای بودن و ۱۷۰ سانتی متر بودن قد خواص آنهاست.

رفتار یک شیء نحوه‌ی پاسخ آن شیء در مقابل رویدادهایی است که ممکن است برایش رخ دهد. برای مثال شیء میز در مقابل رویداد وارد آمدن نیروی بیش از حد بر آن می‌شکند (رفتار). توجه داشته باشید که ممکن است شیء خاصی در مقابل بعضی رویدادها هیچ رفتاری از خود نشان ندهد. این در صورتی است که برای رویداد مورد نظر هیچ رفتاری تعریف نشده باشد. روابط هر شیء نیز نشان دهنده‌ی ارتباط آن شیء با اشیای دیگر است. به عنوان مثال، یک شخص می‌تواند مالک یک شیء مانند میز باشد که در این صورت رابطه‌ی مالکیت بین شیء و شخص برقرار است. اشیایی که در دنیای واقعی وجود دارند، از انواع متفاوتند. حتی اشیای هم نوع ممکن است خصوصیات، رفتارها و روابط متفاوتی داشته باشند. با توجه به این نکته، برای شیء‌ها تقسیم بندی خاصی را در نظر می‌گیریم و اصطلاح کلاس را تعریف می‌کنیم. کلاس مجموعه‌ی تمام اشیای هم نوع است. هر چند این شیء‌ها خواص، رفتار و روابط متفاوتی داشته باشند.

مثلاً انسان یک کلاس است و هر شخص به خصوصی از این مجموعه شیئی از کلاس مذکور تلقی می‌شود. با توجه به تعاریف و مفاهیمی که در بالا ذکر شد می‌توان روش برنامه نویسی شیء گرا را به صورت زیر بیان کرد:

هر برنامه‌ی شیء‌گرا شامل تعدادی شیء با خواص و متدهای متفاوت است به طوری که روابط خاص بین آنها برقرار می‌باشد.

متدها مجموعه‌ای از دستورالعمل‌های برنامه نویسی هستند که باید در هنگام بروز رویدادهایی آشکار شوند. مجموعه‌ی این دستورالعمل‌ها رفتار آن شیء را در برابر رویداد به خصوصی نشان می‌دهند.

## تاریخچه‌ی زبان BASIC

BASIC ، سر نام کلمات Beginner's All-purpose Symbolic Instruction Code به معنی زبان همه منظوره برای افراد مبتدی است. این زبان برنامه نویسی، به دلیل سادگی ساختاری، از محبوبیت زیادی برخوردار است. یک هنرجوی مبتدی که آشنایی زیادی با رایانه و برنامه نویسی ندارد، پس از آموزشی کوتاه، قادر خواهد بود که این زبان را یاد گرفته، امکان نوشتن برنامه در محیط این زبان برنامه نویسی را به دست آورد.

زبان برنامه سازی BASIC، در سال ۱۹۶۴ میلادی، از سوی John Kenney و Thomas kurts در کالج دارتموث (Dartmouth) پدید آمد. این زبان، نخستین زبان برنامه سازی نبود ولی هدف این افراد، فراهم نمودن یک زبان ساده برای دانشجویان رشته‌های مختلف بود. تا به امروز نسخه‌های متعددی از زبان BASIC ارائه شده است که می‌توان از آن جمله به QUICK BASIC , ANSI BASIC , BASICA , GW BASIC , QBASIC , اشاره کرد.

زبان برنامه نویسی BASIC با ارائه‌ی VISUAL BASIC جان تازه‌ای گرفت و دوباره رونق یافت. به همین دلیل، به عنوان یک زبان برنامه سازی پایه‌ای در دوره‌ی آموزش رایانه شناخته شده است. هنرجو، با آموختن این زبان، با اصول برنامه سازی و همچنین برنامه نویسی در محیط VISUAL BASIC و QUICK – BASIC آشنا خواهد شد.

در هر زبان برنامه سازی اگر مقدمات آن زبان را بیاموزید، نوشتن برنامه‌ها در آن ساده خواهد بود.

مقدمات یک زبان عبارتند از : انواع داده‌های موجود، چگونگی تعریف متغیرها، انواع عملگرها، دستورهای شرطی، انواع حلقه‌های تکرار و دستورهای ورودی / خروجی.

## کار با ویژوال بیسیک

محیط ویژوال بیسیک ساده است. این محیط که از جمله محیط‌های توسعه یافته‌ی مجتمع یعنی IDE (Integrated Development Environment) می‌باشد، به برنامه نویسان امکان می‌دهد

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
که برنامه‌های تحت ویندوز خود را بدون نیاز به استفاده از برنامه‌های کاربردی دیگر، اجرا و  
خطایابی کند.

## آشنایی با محیط ویژوال بیسیک

بعد از اجرای برنامه ویژوال بیسیک، کادر محاوره‌ای New project به نمایش در می‌آید که در  
این کادر به برنامه نویس امکان انتخاب یکی از انواع برنامه‌هایی را می‌دهد که می‌توان در VB  
ایجاد کرد.

نوع Standard EXE که به طور پیش فرض در این کادر انتخاب شده است به برنامه نویس  
امکان می‌دهد که برنامه اجرایی استاندارد را ایجاد نماید (برنامه‌های اجرایی استاندارد از اکثر  
خصوصیه‌های ویژوال بیسیک استفاده می‌کنند).

کادر محاوره‌ای New project شامل سه زبانه (Tab) است :

- برگه New : جهت ایجاد یک پروژه جدید
  - برگه Existing : برای باز کردن پروژه‌ای که از قبل وجود دارد.
  - برگه Recent : لیستی از آخرین پروژه‌های باز شده یا ایجاد شده را نشان می‌دهد.
- تعریف پروژه (Project) :** پروژه عبارت است از مجموعه‌ی فایل‌هایی (فرم، برنامه و ...) که  
در کل یک هدف واحد را دنبال می‌کنند. کدهای برنامه، مشخصات ظاهری برنامه و احتمالاً  
فایل‌های بانک اطلاعاتی در این مجموعه از فایل‌ها قرار دارند.

برای باز کردن یک پروژه بر روی آیکن Standard . exe دابل کلیک کرده و یا روی آیکن،  
کلیک کنید، سپس کلید Enter و یا دکمه‌ی Open را فشار دهید. با باز شدن پروژه، کادر تبدالی  
بسته شده و وارد محیط IDE می‌شویم. این محیط دارای چندین پنجره؛ یک میله‌ی منو و یک  
میله‌ی ابزار است که مشابه میله‌های منو و ابزار در اکثر برنامه‌های تحت ویندوز است.

پروژه‌ی Standard EXE شامل پنجره‌های زیر می‌باشد.

۱- پنجره‌ی پروژه (PROJECT)

۲- پنجره‌ی (Form Layout)

۳- جعبه ابزار (ToolBox)

۴- پنجره‌ی خصوصیات (Properties)

۵- پنجره‌ی فرم (Form)

**پنجره‌ی پروژه :** پنجره‌ای است که معمولاً به نام Project Explorer نیز معروف است شامل تمام فایل‌های مربوط به پروژه می‌باشد.

نوار ابزار این پنجره شامل سه دکمه به نام‌های : View object , View code و Toggle Folder است.

دکمه‌ی View code برای نمایش پنجره‌ای که در آن کد VB (دستورات برنامه) فایل‌ی که در پنجره‌ی پروژه‌ی فعال وجود دارد، به کار می‌رود. هم چنین View code برای نمایش شکل ظاهری فرم فعال در پنجره‌ی پروژه به کار می‌رود.

نکته : اگر در پنجره‌ی پروژه هیچ فایل‌ی فعال نباشد هر دو دکمه‌ی View و View code object به صورت غیر فعال درمی‌آیند.

دکمه‌ی Toggle Folders باعث می‌شود که با هر بار فشار آن، پوشه‌ی Forms به صورت متناوب به نمایش درآمده و پنهان شود.

پنجره Project یکی از مهمترین ابزارهای مدیریت پروژه است.

**پنجره (Form Layout) :** این پنجره محل فرم را به هنگام اجرای برنامه (Run time) بر روی صفحه‌ی نمایش مشخص می‌کند.

این پنجره یک صفحه‌ی نمایش را نشان می‌دهد که در داخل آن محل قرار گرفتن فرم مشخص شده است.

با قرار دادن نشان گر ماوس بر روی شکل فرم و پایین نگه داشتن دکمه‌ی سمت چپ ماوس و جابه‌جا کردن آن (Drag) می‌توان فرم را در محل جدید خود قرار داد. به این ترتیب در زمان اجرا فرم مورد نظر در محل مشخص شده ظاهر می‌شود.

**پنجره‌ی خصوصیات (properties window) :** این پنجره ویژگی‌ها و خصوصیات فرم یا کنترل را نشان می‌دهد که به ترتیب الفبایی<sup>۱</sup> مرتب شده اند.

---

<sup>۱</sup> - خصوصیت Name از این قاعده مستثنی است.

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار در قسمت بالای پنجره، جعبه لیست مانندی (combo box) وجود دارد که در آن نام کنترل یا فرمی که خصوصیات آن در این پنجره آورده شده است. داخل این لیست نام تمامی کنترل ها و همچنین نام فرمی که فعلاً فعال است آورده شده است. با انتخاب هر کنترل یا فرم دیگری از این لیست خصوصیات مربوط به آن در پنجره نشان داده می شود. توجه کنید که بعضی از این خصوصیات مشترک هستند مانند خصوصیت Name که در هر مورد نشان دهنده ی نام کنترل یا فرم است، برخی دیگر برای کنترل ها یا فرم مشترک نیستند بلکه منحصر به فردند.

**جعبه ی ابزار (Tool box):** این جعبه شامل کنترل هایی است که از هر کدام آنها می توان به تعداد دلخواه بر فرم های مربوط به پروژه اضافه کرد. به این شرط که نام هر شیء ایجاد شده منحصر به فرد باشد.

کنترل های مربوط به جعبه ی ابزار نسبت به نوع پروژه ای که در ابتدای ایجاد پروژه ی جدید مشخص می شود، متغیر است و معمولاً در نوع Standard EXE تعداد این کنترل ها بیشتر می باشد. با این حال روش هایی برای اضافه کردن کنترل هایی که به صورت استاندارد در جعبه ی ابزار قرار داده نشده اند ولی در VB موجود می باشد وجود دارد.

**پنجره ی فرم ها (Forms):** این پنجره، فرم فعال در پنجره ی پروژه، با تمام اشیای مربوط به آن را در یک رابط گرافیکی کاربر (Graphic user interface) نشان می دهد.

در ابتدای ایجاد یک فرم جدید هیچ شیئی در آن وجود ندارد. البته در صورتی که دکمه ی View code در پنجره ی پروژه انتخاب شود و یا بر روی شیء یا فرم Double Click شود، پنجره ی مربوط به کد در این قسمت نمایش داده می شود.

**نوار منو (Menu bar):** نوار منو مکانی است که در اکثر برنامه های تحت ویندوز وجود دارد و شامل دستوراتی برای ساخت، نگهداری و راه اندازی برنامه هاست. جدول ۱ وظایف هر بخش از منو را به طور خلاصه مشخص نموده است.

### جدول ۱

File	برای باز کردن، ذخیره و چاپ پروژه
------	----------------------------------



برای Paste کپی، حذف و غیره.	Edit
برای نحوه‌ی نمایش پنجره‌های محیط IDE	View
برای افزودن خصیصه‌هایی مانند فرم‌ها به یک پروژه	Project
برای تنظیم کنترل‌های موجود بر روی فرم	Format
برای خطایابی	Debug
برای اجرا، متوقف کردن برنامه و ...	Run
برای بازیابی داده‌ها از پایگاه داده‌ها	Query
برای ابزارهای IDE و بهینه سازی محیط کار	Tools
برای ویرایش و اصلاح در طراحی پایگاه داده ها	Diagram
برای نصب و حذف	Add_ins
برای مرتب کردن و نمایش پنجره‌ها	Windows
راهنمای کاملی برای استفاده کننده	Help

در پایین نوار منو، نوار ابزاری وجود دارد که توسط آن می‌توان به سرعت به برخی از گزینه‌های منو دسترسی پیدا کرد.

### طریقه‌ی کار با جعبه ابزار

برای اضافه کردن یک شیء از هر کنترل موجود در جعبه‌ی ابزار می‌توان به دو روش عمل نمود:

الف) با Double Click کردن روی هر کنترل، یک نمونه از شیء مربوطه که در وسط فرم فعال ظاهر می‌شود، همیشه از نظر اندازه و موقعیت ثابت است. با پایین نگه داشتن کلید ماوس روی هر شیء اضافه شده و حرکت دادن ماوس (Drag) می‌توان محل شیء را تغییر داد و نیز می‌توان با قرار دادن مکان نما در گوشه‌های شیء و Drag کردن، اندازه‌ی آن شیء را نیز به دلخواه تعیین نمود.

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
ب ) کنترل مربوط به آن شیء را در جعبه ابزار توسط کلیک سمت چپ ماوس انتخاب نمود و سپس بر روی فرم مورد نظر در محل دلخواه کلیک کرده و با Drag کردن، شیء را به اندازه‌ی دلخواه بر روی فرم اضافه نمود.

## کلیات

هنگام برنامه نویسی باید بدانید که کدها چگونه با فرم و کنترل‌های برنامه ارتباط برقرار می‌کنند. به یاد دارید که پنجره‌ی خواص تمام فایل‌های برنامه را نشان می‌دهد. کدهای برنامه در پنجره‌ی کد نوشته می‌شوند ولی می‌دانید که روال‌های رویداد به فایل‌های خاص خود نیاز ندارند؛ آنها در همان فایل فرم ذخیره می‌شوند. یک فرم می‌تواند کدهایی داشته باشد که روال رویداد نیستند، اینها کدهایی هستند که وظایف کلی‌تری بر عهده دارند.

پروژه‌ها می‌توانند انواع دیگری از کد داشته باشند. روال‌های Visual Basic می‌توانید سابروتین یا تابع باشند. تمام کدهایی که روال رویداد نیستند را می‌توان در یک محل جداگانه، به نام ماژول (module)، قرار داد. ماژول یک فایل جداگانه است که در پنجره‌ی خواص ظاهر خواهد شد. از نظر فنی، کدهای موجود در فرم هم در یک ماژول (به نام ماژول فرم) قرار دارند. هر پروژه به تعداد فرم‌هایشان دارای ماژول فرم است. وقتی برنامه‌ای دارای چند فرم است. برنامه نویس باید تصمیم بگیرد که کدام فرم در شروع برنامه روی صفحه ظاهر شود. فرم شروع (start form) اولین فرمی است که در برنامه ایجاد می‌شود، اما می‌توان این وضع را عوض کرد. برای این کار، آیتم Project Name Properties را کلیک کنید (<Project Name> نام پروژه است) تا Visual Basic کادر محاوره‌ای خواص پروژه را نشان دهد.

## پنجره‌ی Code

اما یک ماژول فرم علاوه بر روال‌های رویداد می‌تواند حاوی قسمت دیگری هم باشد: قسمت تعاریف (Declaration Section). قسمت تعاریف فضایی است که نام‌های مورد استفاده در سایر قسمت‌های ماژول تعریف می‌شوند. کنترل‌ها نیازی به تعریف ندارند، اما سایر انواع داده‌های Basic Visual را باید در این قسمت تعریف کنید.

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار هر ماژولی هم که داشته باشد می تواند دارای یک قسمت تعاریف باشد؛ این قسمت همیشه در ابتدای ماژول ظاهر می شود. هر کدی که قبل از اولین روال رویداد نوشته شود جزء این قسمت محسوب خواهد شد.

متن انتخاب شده (که با دستور Option Explicit شروع شده) قسمت تعاریف ماژول است. دقت کنید که در لیست های Object , Procedure به ترتیب (General) و (Declarations) دیده می شود؛ دو لیست در هر لحظه نشان می دهند که شما در کدام قسمت از ماژول فرم قرار دارید. دو روال بعدی روال رویداد نیستند و این واقعیت را از نام آنها می توان دریافت. به یاد دارید که نام یک روال رویداد از دو قسمت، نام کنترل و نام رویداد که با یک زیر خط به هم متصل شده اند، تشکیل می شود. با این که نام روال Update \_ Count () دارای دو قسمت و یک زیر خط است ولی اگر مکان نما را در بدنه ی این روال قرار دهید در لیست Object همچنان کلمه ی General را خواهید دید، چون روال مزبور جزء روال های عمومی ماژول است. (اما نام روال Update \_ Count در لیست Procedure ظاهر خواهد شد، چون این لیست حاوی نام تمام روال های ماژول از جمله روال های عمومی است).

### داده ها در ویژوال بیسیک

هر زبان برنامه نویسی برای پردازش اطلاعات به انواع مختلفی از داده ها نیاز دارد و Basic Visual هم از این قاعده مستثنی نیست. Visual Basic از انواع داده های مختلفی پشتیبانی می کند که با آنها می توان نیازهای مختلف برنامه نویس را برآورده کرد. Visual Basic هم مانند سایر زبان های برنامه نویسی درباره ی داده های خود سخت گیر است (البته نه به شدت برخی از آنها) و باید دقیقاً به او بگویید که از چه نوع داده ای می خواهید استفاده کنید. Visual Basic از دوازده (۱۲) نوع داده (Data type) پشتیبانی می کند.

### داده های عددی

تمام انواع داده های عددی در یکی از مقوله های زیر جای می گیرند:

- اعداد صحیح (Integer). اعداد صحیح بدون نقطه ی اعشاری : مانند ۶۱۴، ۰، -۹۳۴

• اعداد اعشاری (Decimal). اعداد با نقطه‌ی اعشاری (ممیز)؛ مانند ۰،۰۰۵، ۰،۳۵۵، ۴۰۲

– به اعداد اعشاری اعداد ممیز شناور هم گفته می‌شود. در تمام اعداد اعشاری باید نقطه

اعشاری وجود داشته باشد حتی اگر ارقام بعد از آن صفر باشند.

Visual Basic اعداد اعشاری و صحیح را به روش‌های مختلف ذخیره و با آنها کار می‌کند. با آنکه

برای یک کاربر بین ۸ و ۸/۰۰ هیچ فرقی وجود ندارد ولی از نظر Visual Basic آنها متفاوتند.

مقدار حافظه‌ای که انواع داده‌ی مختلف به خود اختصاص می‌دهند یکسان نیست. با نگاه کردن به

یک عدد نمی‌توان گفت که چقدر حافظه اشغال کرده است. مثلاً، اعداد ۹۹۹، ۲۹، ۷۰۱ هر دو به

یک مقدار حافظه می‌گیرند. با آن که امروزه دیگر حافظه یک مشکل کلیدی نیست و شما هم به

عنوان برنامه نویس نباید زیاد نگران آن باشید، ولی همیشه سعی کنید برای داده‌هایتان نوعی

انتخاب کنید که حافظه‌ی کمتری را اشغال می‌کند.

در جدول ۲ هفت نوع داده‌ی عددی Visual Basic، مقدار حافظه‌ی مورد نیاز هر کدام و

محدوده‌ای می‌توانند در خود جای دهند را می‌بینید. هنگام تعریف داده‌ها این جدول را مد نظر

داشته باشید. به عنوان مثال، اگر می‌خواهید با اعداد منفی کار کنید نباید از نوع Byte استفاده

کنید، اما اگر با سن افراد سر و کار دارید این نوع بهترین انتخاب ممکن است.

### جدول ۲) هفت نوع داده‌ی عددی Visual Basic

۰ تا ۲۵۵	۱ بایت	Byte
۷۶۷/۳۲ – تا ۷۶۸/۳۳	۲ بایت	Integer
۶۴۷/۴۸۳/۱۴۷/۲ – تا ۶۴۸/۴۸۳/۱۴۷/۲	۴ بایت	Long
اعداد منفی: - ۴۰۲۸۲۳ و E + ۳۸۳ تا E - ۴۵۱ – ۴۰۱۲۹۸ اعداد مثبت: ۴۰۱۲۹۸ و E - ۴۵۱ تا E + ۳۸۳ و ۴۰۲۸۲۳	۸ بایت	Single
اعداد منفی: - ۷۹۷۶۹۳۱۳۴۸۶۲۳۲ و E+۳۰۸۱ تا - E - ۳۲۴۴ و ۹۴۰۶۵۶۴۵۸۴۱۲۴۷ اعداد مثبت: ۹۴۰۶۵۶۴۵۸۴۱۲۴۷ و E - ۳۲۴۴ تا E+۳۰۸۱ و ۷۹۷۶۹۳۱۳۴۸۶۲۳۲	۸ بایت	Double
- ۴۷۷ و ۵۸۰۸/۶۸۵/۲۰۳/۳۳۷/۹۲۲ تا	۸ بایت	Currency

۴۷۷ و ۵۸۰۷/۶۸۵/۲۰۳/۳۳۷/۹۲۲ (چهار رقم اعشار برای حفظ دقت محاسبات است)		
۳۳۵/۹۵۰/۵۴۳/۵۹۳/۳۳۷/۲۶۴/۵۱۴/۱۶۲/۲۲۸/۷۹ ± بدون اعشار	۱۲ بایت	Decimal
۷ و ۹۲۲۸۱۶۲۵۱۴۲۶۴۳۳۷۵۹۳۵۴۳۹۵۰۳۳۵ رقم اعشاری ( Visual Basic هنوز به طور کامل از این نوع پشتیبانی نمی کند).		

**بایت (byte)** معادل یک واحد ذخیره سازی در حافظه PC است. در نوع Single برای نوشتن نمای عدد از E یا e استفاده می شود؛ در نوع Double می توان از D یا d برای نوشتن نما استفاده کرد. برای تبدیل یک عدد از عدد نویسی معمولی باید عدد بعد از E (یا D) را به توان ۱۰ رسانده و آن را در عدد قبل از E (یا D) ضرب کنید. مثلاً،  $E+55,83$  معادل  $10^{55} \times 5,83$  یا  $583000$  است. هنگام کار با اعداد بسیار بزرگ (یا بسیار کوچک) عدد نویسی علمی کمک قابل توجهی به دقت و صرفه جویی در جا خواهد کرد.

وقتی در برنامه عددی را صریحاً می نویسید (به این قبیل اعداد واژه‌ی عددی – Numeric Literal – گفته می شود)، Visual Basic مناسب ترین نوع را برای آن برمی گزیند ولی گاهی لازم است تا واژه‌ی مورد استفاده از نوعی باشد که شما دارید نه آنچه که Visual Basic تعیین می کند. در چنین مواردی می توانید نوع واژه را صریحاً به Visual Basic معرفی کنید، این کار با استفاده از پسوند نوع داده (data – type suffix) امکان پذیر است. جدول ۳ انواع پسوندهای عددی را در Visual Basic نشان می دهد.

### جدول ۳) پسوندهای عددی Visual Basic

نوع داده	پسوند
Long	&
Single	!
Double	≠
Currency	@

توجه داشته باشید که اگر از عدد نویسی علمی استفاده می کنید، حروف D,E به ترتیب معرف انواع Double , Single هستند و دیگر نباید از پسوند نوع داده استفاده کنید.

به یک مثال توجه کنید. اگر در برنامه‌ای از واژه‌ی عددی ۵/۸ استفاده کرده باشید، Visual Basic به طور خودکار نوع Single را برای آن برمی‌گزیند و ۴ بایت حافظه به آن اختصاص می‌دهد. اما با نوشتن این عدد به صورت 5.8 ≠، Visual Basic را وادار کنید تا به آن چشم یک عدد Double نگاه کند و ۸ بایت حافظه برای آن کنار بگذارد.

### سایر انواع داده

درک داده‌های غیر عددی (برای کسانی که علاقه‌ای به ریاضیات ندارند) آسان‌تر است. یکی از دلایلی که BASIC، علیرغم حضور زبان‌ها پیشرفته‌تر، همچنان مطرح مانده، توانایی‌های آن در کار با رشته‌های متن است. رشته (String) ترکیبی است از چند کاراکتر، که حتی می‌تواند عدد باشد ولی نمی‌توان روی آنها محاسبه انجام داد. نام، آدرس، شماره تلفن یا حساب بانکی را می‌توان به صورت رشته نمایش داد. همواره سعی کنید فقط برای اعدادی که نیاز به محاسبه دارند از انواع عددی استفاده کنید. در جدول ۴ انواع غیر عددی Visual Basic را می‌بینید.

جدول ۴) انواع داده‌ی غیر عددی Visual Basic

نوع داده	مقدار حافظه	محدوده
String (طول ثابت)	طول رشته	از ۱ تقریباً ۶۵۴۰۰ کاراکتر
String (طول متغیر)	طول رشته + ۱۰ بایت	۰ تا ۲ میلیارد کاراکتر
Date	۸ بایت	از اول ژانویه ۱۰۰ تا ۳۱ دسامبر ۹۹۹۹
Boolean	۲ بایت	True یا False
Object	۴ بایت	معادل شیء تعریف شده
Variant (عددی)	۱۶ بایت	هر عددی تا Double
Variant (متن)	طول رشته + ۲۲ بایت	مانند (طول متغیر) String

متغیر Boolean فقط می‌تواند دو مقدار بگیرد: True (درست) یا False (نادرست).

واژه‌های رشته‌ای (string literal) همیشه بین دو علامت نقل قول (") قرار می‌گیرند و می‌توانند شامل هر کاراکتری باشند. مثال‌های زیر همگی رشته هستند.

"Oh me, oh my"

"543 - 00 - 0324"

"1020 S.Yale Avenue"

""

هر چیزی که بین دو علامت " قرار گیرد، یک رشته است حتی اگر (مانند مثال آخر) هیچ کاراکتری در آن نباشد. رشته‌ای که طول آن صفر باشد، رشته‌ی Null نامیده می‌شود. Visual Basic از رشته‌های خاصی به نام رشته‌های خالی (که با کلمه‌ی کلیدی Empty تعریف می‌شوند) پشتیبانی می‌کند. رشته‌ی خالی رشته‌ای است که هیچ مقداری (حتی Null) ندارد.

هنگام تعریف واژه‌هایی که حاوی تاریخ و زمان هستند از علامت ≠ استفاده کنید. Visual Basic از تمام فرمت‌های تاریخ و زمان پشتیبانی می‌کند. به مثال‌های زیر توجه کنید:

July 4,1776 ≠ ≠

7:11 pm ≠ ≠

19:11:22 ≠ ≠

1-2-2003 ≠ ≠

≠ 5-Dec-99 ≠

نوع داده‌ای Boolean برای مواردی برای مواردی مناسب است که فقط دو مقدار متضاد (True یا False، Yes یا No و از این قبیل) دارید. خاصیت Enabled کنترل‌ها از این نمونه است.

نوع داده‌ای Variant می‌تواند هر مقداری (به جز رشته‌های با طول ثابت) را در خود جای دهد. از این نوع داده زمانی استفاده کنید که از قبل دقیقاً ندانید با چه نوع داده‌ای سر و کار خواهد داشت.

### کار با متغیرها

متغیر (variable) مکانی است برای نگهداری یک مقدار، مقداری که در متغیر قرار داده می‌شود، قابل تغییر است (نام آن هم بر همین ویژگی دلالت دارد). وقتی مقداری را در یک متغیر قرار می‌دهید مقدار قبلی آن از بین خواهد رفت. متغیر مکانی است در حافظه برای ذخیره کردن داده‌های موقتی برنامه.

متغیرها با نامشان شناخته می‌شوند، بنابراین در یک روال استفاده از دو متغیر با یک نام مجاز نیست چون Visual Basic قادر به تشخیص آنها نخواهد بود. بر خلاف کنترل‌ها که Visual Basic خود به آنها یک نام پیش فرض می‌دهد، نامگذاری متغیرها از همان ابتدا برعهده‌ی برنامه‌نویس

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار است. قبل از استفاده از یک متغیر باید آن را تعریف (declare) کنید، تعریف یک متغیر یعنی نامگذاری آن و تعیین نوع مقداری که می تواند بگیرد. متغیرها می توانند فقط از همان نوعی که تعریف شده اند مقدار بگیرند (به استثنای متغیرهای variant که می توانند از تمام انواع داده مقدار بگیرند)

## تعریف متغیرها

برای تعریف یک متغیر (نامگذاری و تعیین نوع آن) از کلمه ی کلیدی Dim استفاده می شود. قبل از استفاده از یک متغیر حتماً باید آن را تعریف کرد. البته Visual Basic اجازه می دهد که این قاعده ی کلی را زیر پا بگذارید ولی تخلف از این قاعده می تواند به سردرگمی منجر شود. الزام (یا عدم الزام) به تعریف متغیرها را می توانید در منوی Tools، گزینه options، برگه ی Editor و گزینه ی Require Variable Declaration مشخص کنید. اگر دقت کرده باشید، در قسمت تعاریف پنجره ی کد که دستور به صورت پیش فرض وجود دارد:

### Option Explicit

این دستور به Visual Basic می گوید که در کل ماژول مورد بحث، متغیرها قبل از استفاده باید تعریف شوند. در چنین ماژول هایی هر گاه نام یک متغیر را اشتباه بنویسید، Visual Basic آن را به شما گوشزد خواهد کرد. اما اگر این گزینه را غیر فعال کرده باشید، Visual Basic اشتباه در نوشتن نام یک متغیر را متغیر جدیدی تلقی کرده و به کار خود ادامه خواهد داد. در این حالت تمام متغیرهایی که صریحاً تعریف نشوند از نوع Variant در نظر گرفته خواهند شد. شکل کلی استفاده از دستور Dim برای تعریف یک متغیر چنین است :

Dim Var Name As Data Type

که در آن Var Name نام متغیر و Data Type یکی از انواع داده های Visual Basic است. متغیرهای مورد استفاده در یک روال باید در همان ابتدای روال تعریف شوند. متغیرهای تعریف شده در یک روال فقط در همان روال قابل استفاده اند و در هیچ روال دیگری قابل دسترسی نیستند؛ به این قبیل متغیرها، متغیر محلی (local variable) گفته می شود. متغیرهایی که در قسمت تعاریف ماژول تعریف شوند از تمام روال های آن ماژول قابل دسترس خواهند بود؛ به این گونه متغیرها، متغیر عمومی (Global Variable) می گویند. متغیرهای عمومی فقط در همان



ماژولی که تعریف شده‌اند دیده می‌شوند. متغیرها را می‌توان به گونه‌ای تعریف کرد که در تمام ماژول‌های پروژه قابل دسترسی باشند.

چون نام گذاری متغیرها بر عهده‌ی برنامه نویس است، باید قواعد نام گذاری آنها را بدانید:

- نام متغیر باید با یکی از حروف الفبا شروع شود.
  - استفاده از حروف و اعداد در نام متغیرها مجاز است.
  - نام یک متغیر می‌تواند تا ۲۵۵ کاراکتر طول داشته باشد.
  - سعی کنید حتی الامکان از حروف خاص (غیر الفبایی - عددی) استفاده نکنید؛ زیر خط ( ) در این میان یک استثناست.
  - فاصله در نام متغیرها مجاز نیست.
- علاوه بر قواعد الزامی فوق، سعی کنید هنگام نام گذاری متغیرها نکات زیر را هم رعایت کنید:
- در نام متغیرها از پیشوندهایی استفاده کنید که نوع آن را مشخص کند. بدین ترتیب دیگر نیازی نیست مدام برای اطلاع از نوع یک متغیر به قسمت تعریف متغیرها مراجعه کنید.
  - اگر از نام‌های بامعنی استفاده کنید، برنامه برنامه‌تان قابل فهم‌تر خواهد بود و به مستندسازی کمتری نیاز خواهد داشت.

در هر دستور Dim می‌توان بیش از یک متغیر را تعریف کرد؛ تعریف‌ها با کاما (,) از هم جدا می‌شوند :

```
Dim intTotal As integer, CurSales99 As Currency
```

اگر نوع داده‌ی یک متغیر ذکر نشود، Visual Basic آن را از نوع variant تعریف خواهد کرد؛ بنابراین هر دو دستور زیر معادل یکدیگرند:

```
Dim vntControlVal As Variant
```

```
Dim vntControlVal
```

### تعریف رشته‌ها

هنگام تعریف رشته‌ها یک مشکل ثانویه پیش می‌آید، چون دو نوع String وجود دارد: با طول ثابت، با طول متغیر. متداول‌ترین رشته‌ها رشته‌های با طول متغیر هستند چون روش تعریف آنها درست شبیه تعریف متغیرهای دیگر است. در مثال‌های زیر دو رشته‌ی با طول متغیر تعریف شده‌اند :

```
Dim strCityName As String
```

### Dim strStateName As String

هر دوی این متغیرها می توانند رشته‌هایی با طول‌های متفاوت را در خود نگه دارند. مثلاً، اگر ابتدا در متغیر strCityName رشته‌ی "Tehran" و سپس رشته‌ای "Yazd" را ذخیره کنیم، این متغیر طول خود را متناسب با آن تغییر خواهد داد. در اکثر موارد این همان چیزی است که ما می خواهیم، ولی گاهی پیش می آید (مثلاً هنگام کار با فایل‌ها) که بخواهیم طول رشته‌ها ثابت بماند. در این موارد باید طول آنها را مشخص کنیم :

### Dim strZipCode As String \*5

اگر رشته‌ای که طول آن بیش از پنج حرف باشد به متغیر strZipCode نسبت داده شود، Visual Basic فقط پنج حروف اول آن را ذخیره می کند و ما بقی را دور خواهد انداخت.

#### مقدار دادن به متغیرها

بعد از تعریف یک متغیر، می توان داده‌ها را در آن ذخیره کرد. ساده‌ترین راه برای ذخیره کردن یک مقدار در یک متغیر، دستور انتساب (assignment statement) است. شکل کلی این دستور چنین است :

### Item Name = Expression

که در آن Item Name نام یک متغیر (یا خصوصیت) است و Expression می تواند یکی از موارد زیر باشد :

- یک عبارت محاسباتی
- یک واژه
- یک عبارت منطقی یا رشته‌ای
- مقدار خصوصیت یک کنترل (خواص کنترل‌ها از نوع Variant هستند ولی Visual Basic هنگام ذخیره کرده آنها در یک متغیر نوع آنها را تبدیل خواهد کرد)
- ترکیبی از عبارات محاسباتی یا منطقی، واژه‌ها، متغیرها و مقدار خواص کنترل‌ها هر چیزی که بتواند یک مقدار تولید کند، عبارت (expression) است. به مثال‌های زیر توجه کنید :

curSales : 571275

strFirstName="Herry"

blnPassedTest=True

blnsEnabled = lblTitle.Enabled

dblValue=45.1#

intcount=intNumber

dteOld= #4-1-92#

sngOld97Tptal=sngNew98Total-100000

مهمترین نکته درباره‌ی یک عبارت آن است که مقدار سمت راست عبارت به متغیر سمت چپ آن نسبت داده می‌شود. توجه کنید که مقدار سمت راست عبارت بایستی با نوع متغیر سمت چپ عبارت متناسب باشد، یا اینکه امکان تبدیل آن برای Visual Basic وجود داشته باشد. مثلاً، Visual Basic می‌تواند یک عدد کوچکتر را در متغیری از نوع Long (که قاعداً برای اعداد بزرگ به کار می‌رود) قرار دهد، ولی نمی‌تواند یک رشته را به متغیر عددی نسبت دهد. در Visual Basic، برای حفظ سازگاری با بیسیک‌های قدیمی، می‌توان از کلمه کلیدی let برای مقدار دادن به متغیرها استفاده کرد؛ دو دستور زیر معادل یکدیگرند:

```
Let intCount =1
```

```
inCount= 1
```

## عملگرهای ویژه‌ی بیسیک

Visual Basic از عملگر (operator) های محاسباتی و رشته‌ای متعددی پشتیبانی می‌کند. جدول ۵ عملگرهای متداول Visual Basic را نشان می‌دهد. عملگر ابزاری است برای ترکیب کردن داده‌های مورد نظر.

عملگر توان (exponentiation) یک عدد را به توان عدد دیگر می‌رساند؛  $2^3$  یعنی ۲ توان ۳، عملگرهای ضرب، تقسیم، جمع و تفریق هم دقیقاً همان کاری را می‌کنند که در محاسبات معمول با آن آشنا هستید. عملگر Mod برای محاسبه‌ی باقی مانده یک تقسیم است، بنابراین  $11 \bmod 3$  معادل ۲ خواهد شد. چون باقی مانده‌ی تقسیم ۱۱ بر ۳ معادل ۲ است. عملگر Mod فقط برای اعداد صحیح است و اگر از اعداد اعشاری استفاده کنید، Visual Basic ابتدا آنها را به صحیح تبدیل کرده و سپس باقی مانده را محاسبه خواهد کرد. عملگر تقسیم صحیح (۱) خارج قسمت صحیح تقسیم را برمی‌گرداند و باقی مانده‌ی تقسیم را دور می‌اندازد.

## جدول ۵ عملگرهای Visual Basic

عملگر	مفهوم	مثال	نتیجه
^	توان	$2^3$	8
*	ضرب	$2*3$	6
/	تقسیم	$6/2$	3
+	جمع	$2+3$	5
-	تفریق	$6-3$	3
Mod	مدول یا نهشت	$11 \text{ mod } 3$	2
\	تقسیم صحیح	$11/3$	3
& یا +	ترکیب رشته‌ها	"Hi," & "There"	"Hi, There"

نکته جالب در جدول ۵ آن است که عملگر + دو کار متفاوت انجام می‌دهد: جمع معمولی و به هم چسباندن (ترکیب) رشته‌ها. این عملگرها با توجه به محلی که مورد استفاده قرار گرفته (بین دو عدد یا بین دو رشته) واکنش مناسب را نشان می‌دهد. هنگام ترکیب رشته‌ها، Visual Basic هیچ چیز به آنها اضافه نخواهد کرد. بنابراین اگر می‌خواهید بین دو رشته یک فاصله وجود داشته باشد باید خودتان آن را اضافه کنید. به مثال زیر توجه کنید:

`StrCompleteName=lblFirst.Caption & " " & lblLast.Caption`

به این نوع عملگرها عملگر تحریف شده می‌گویند چون عملکرد آنها بسته به مکان استفاده از آن متفاوت خواهد بود.

### نکاتی درباره‌ی تقدم عملگرها

Visual Basic اعمال ریاضی را به ترتیب خاصی، که در جدول ۶ آمده، انجام می‌دهد. اعمال درون پرانتز ابتدا اجرا می‌شوند و سپس بالاترین تقدم را توان (^) دارد و بعد نوبت به ضرب و تقسیم می‌رسد و بعد از آن نوبت جمع و تفریق است.

### جدول ۶) تقدم عملگرها در محاسبات

عملگر	مفهوم	مثال	نتیجه
۱	پرانتز	$(2+3)*7$	۳۵

۹	$2^3+1$	^	۲
۲۳	$2+3*7$	Mod , \ , / , *	۳
۳	$10-4*2+1$	- و +	۴

اگر از پرانتزها استفاده نکنید، Visual Basic همیشه ابتدا توان، سپس ضرب و تقسیم و بعد از آن جمع و تفریق را انجام خواهد داد. اگر در یک عبارت عملگرهایی با تقدم یکسان وجود داشته باشند، Visual Basic محاسبات را از چپ به راست انجام می دهد. به مثال زیر نگاه کنید.

$$10/2*3$$

در این عبارت چون ضرب و تقسیم دارای تقدم یکسان هستند. Visual Basic ابتدا تقسیم را انجام داده و سپس حاصل تقسیم را در ۳ ضرب خواهد کرد؛ بدین ترتیب حاصل عبارت فوق ۱۵ خواهد شد. اگر می خواهید ترتیب انجام محاسبات را تغییر دهید باید از پرانتزها استفاده کنید. در پرانتزهای تودرتو، Visual Basic از داخلی ترین جفت پرانتز شروع کرده و رو به بیرون حرکت می کند. در مثال زیر

$$(10+2-(8-3))+1$$

Visual Basic قبل از هر کاری (8-3) را محاسبه خواهد کرد.

### - عملگرهای شرطی یا مقایسه ای

در جدول ۷ عملگرهای شرطی (Visual Basic (conditional operator) را می بینید. این عملگرها هیچ گونه عملیات ریاضی انجام نمی دهند، بلکه داده ها را مقایسه می کنند. با این عملگرها برنامه هوشمندتر خواهد شد و خواهد توانست داده ها را مقایسه کرده و بر اساس نتایج آن، عملکرد مناسب را در پیش گیرد.

### جدول ۷) عملگرهای شرطی Visual Basic

عملگر	توضیح	مثال	نتیجه
=	تساوی	$7=2$	نادرست
<	بزرگتر است از	$6>3$	درست

درست	$5 < 11$	کوچکتر است از	$>$
درست	$23 >= 23$	بزرگتر یا مساوی	$= <$
درست	$4 <= 21$	کوچکتر یا مساوی	$= >$
نادرست	$3 < > 3$	نامساوی	$< >$

توجه کنید که نتیجه‌ی عمل عملگرهای شرطی همیشه False (نادرست) یا True (درست) خواهد بود؛ نکته‌ی دیگر جدول ۷ آن است که عملگر = در آن تحریف شده است؛ یعنی این عملگر هم می‌تواند در دستورات انتساب مورد استفاده قرار گیرد، هم در دستورات مقایسه و این Visual Basic است که کارکرد آن را تشخیص داده و به طور مناسب عمل خواهد کرد.

عملگرهای شرطی، نه فقط روی واژه‌ها بلکه روی عبارات، متغیرها، کنترل‌ها و یا ترکیبی از آنها هم عمل می‌کنند. با توجه به تنوع داده‌ها در Visual Basic، این برنامه نویسی است که باید تصمیم بگیرد داده‌ها را چگونه مقایسه کرده و چگونه نتیجه گیری کند.

هنگام استفاده از عملگرهای شرطی ممکن است حالت خاصی پیش آید و آن Null بودن یکی از اجزاء مقایسه است. در این حالت Visual Basic مقدار Null برمی‌گرداند نه True یا False. اگر امکان Null بودن یکی از عناصر مقایسه وجود دارد، باید این حالت را هم در نظر بگیرید و برای آن تمهیدی بیندیشید.

عملگرهای شرطی، علاوه بر مقادیر عددی، رشته‌ها را هم می‌توانند مقایسه کنند. در هنگام مقایسه‌ی رشته‌ها به نکات زیر توجه داشته باشید :

- حروف بزرگ کوچکتر از حروف کوچک هستند، یعنی "ST.JOHN" قبل از "st.john" قرار خواهد گرفت.
  - حروف الفبا به همان ترتیبی که هستند مقایسه می‌شوند، یعنی "A" کوچکتر از "B" است و نام "Walter" قبل از "Williams" قرار خواهد گرفت.
  - اعداد کوچکتر از حروف هستند، یعنی "3" از "Three" کوچکتر خواهد بود.
- برای درک بهتر این نکات فقط کافیست بدانید که Visual Basic رشته‌ها را مانند یک کتاب لغت مرتب خواهد کرد. با این قابلیت برنامه‌ی شما هم می‌تواند روی انواع مختلف داده‌ها عمل کند.

UML SSADM طراحی و بسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
Visual Basic هنگام مقایسه‌ی رشته‌ها به بزرگ یا کوچک بودن حروف هم توجه دارد  
یعنی بین آنها تفاوت قایل خواهد شد. اگر میل دارید Visual Basic حروف بزرگ و  
کوچک را یکسان فرض کند می‌توانید در قسمت تعاریف ماجول از دستور Option  
Compare Text استفاده کنید. اما اگر این دستور وجود نداشته باشد، Visual Basic  
برای مرتب کردن رشته‌ها از جدول ASCII استفاده خواهد کرد. جدول ASCII شامل  
۲۵۶ کاراکتر موجود در PC هاست و هر کاراکتر در این جدول دارای یک عدد متناظر (و  
منحصر به فرد) است.

به دستورات مقایسه‌ی زیر توجه کنید:

"abcdef" > "ABCDEF"

"Yes!" < "Yes?"

"Computers are fun!" = "Computers are fun!"

"PC" <> "pc"

"Books, Books, Books" > = "Books, Books"

تمام دستورات فوق مقدار True را برمی‌گردانند.

Visual Basic عملگر شرطی دیگری دارد به نام LIKE، که با آن می‌توان مقایسه‌های  
کلی تری انجام داد. شاید هنگام کار با سیستم عامل با کاراکترهای \* و ؟ برخورد کرده  
باشید. \* یعنی صفر یا چند کاراکتر و ؟ به جای یک کاراکتر می‌نشیند. علاوه بر این دو  
کاراکتر، عملگر Like از کاراکتر # هم (که به معنای یک رقم است) استفاده می‌کند. در  
زیر به کمک چند مثال با این عملگر بیشتر آشنا می‌شوید. تمام این عبارات مقدار True  
دارند :

"Sams Publishing Like "Sa\*"

"Data### "Like "Data 98"

"X" Like "[XYZ]"

آخرین مثال شکل خاصی از دستور Like است. اگر کاراکتر مشخص شده مانند هر یک  
از کاراکترهای درون [] باشد، مقدار True برگشت داده خواهد شد. [] برای یافتن کاراکتر  
از میان چند کاراکتر است. رشته‌ی "Code [12345] RED" هنگام مقایسه با تمام  
رشته‌های زیر مقدار True برمی‌گرداند:

"Code3Red" , "Code2Red" , "CodeRed"

. "Code5Red" , "Code 4Red"

## داده‌های شرطی

هنگام مقایسه‌ی دو مقدار، نوع آنها باید با هم سازگار باشد، مثلاً، دو عدد یا دو رشته را می‌توان با هم مقایسه کرد. ولی نمی‌توان یک عدد را با یک رشته یا یک مقدار منطقی (Boolean) مقایسه کرد. حتی مقایسه‌ای انواع داده‌ی دقیق (مانند Single یا Double) هم در اکثر موارد به اشتباه منجر خواهد شد چون Visual Basic این قبیل اعداد را گرد می‌کند و مقایسه‌ی آنها نمی‌توانند دقیق باشد. برای مقایسه‌ی این قبیل داده‌ها از تفریق استفاده کنید (یا حتی الامکان از این کار اجتناب کنید!).

Visual Basic هنگام مقایسه‌ای داده‌های Variant (که خواص کنترل‌ها هم از این نوع است) بسیار خوب عمل می‌کند. Visual Basic هنگام مقایسه‌ی داده‌های Variant، ابتدا آنها را به نوع مناسب تبدیل کرده و سپس مقایسه را انجام خواهد داد. اما مقایسه‌ی داده‌های ناهمگن همیشه می‌تواند با خطا همراه باشد.

## ترکیب عملگرهای شرطی و منطقی

از لحاظ تئوری، شش عملگر شرطی Visual Basic قدرت کافی برای هر نوع مقایسه‌ای را دارند، ولی قدرت آنها را با استفاده از عملگرهای منطقی (Logical operator) باز هم می‌توان بالاتر برد. جدول ۸ عملگرهای منطقی Visual Basic را نشان می‌دهد.



## جدول ۸) عملگرهای منطقی Visual Basic

عملگر	توضیح	مثال	نتیجه
And	هر دو گزاره باید درست باشند	(2<3) And (4<5)	True
Or	حداقل یکی از دو گزاره باید درست باشد	(2<3) Or (6<7)	True
Xor	فقط یکی از دو گزاره باید درست باشد یا نه هر دو	(2<3) Xor (7>4)	False
Not	نقیض	Not (3=3)	False

با عملگرهای منطقی می توان دو یا چند مقایسه ای شرطی را با هم ترکیب کرد. کاربرد Or , And , بسیار بیشتر از دو عملگر دیگر است. با Xor می توان دو گزینه ای انحصاری را مقایسه کرد. اگر بیش از یکی از گزینه های Xor درست باشد، کل عبارت نادرست خواهد شد.

عملگر Not هم برای نقیض یک عبارت است. این عملگر ارزش عبارت را به عکس آن تبدیل می کند. به عبارت زیر، که ترکیب دو عملگر شرطی با And منطقی است، توجه کنید :

(CurSals< CurMinSales) And (intYrsEmp>10)

در این عبارت، اگر فروش سال جاری یک کارمند از حداقل فروش تعیین شده کمتر باشد و تعداد سال های کاری وی هم از ۱۰ سال کمتر باشد (به انتخاب نام متغیرها دقت کنید!) کل عبارت درست خواهد شد. البته می توانستید این دو عبارت را جداگانه مقایسه کنید ولی And امکان می دهد تا آنها را در یک عبارت متمرکز کنید. البته ترکیب بیش از حد عبارات آنها را گیج کننده خواهد کرد.

در جدول ۹ تقدم کاملتر عملگرها را مشاهده می کنید.

## جدول ۹) ترتیب عملگرهای محاسباتی، مقایسه ای و منطقی

عملگر	ترتیب
پرانتز ()	۱
^	۲

Mod ، \ ، / ، *	۳
- ، +	۴
عملگرهای شرطی از قبیل Like	۵
عملگر منطقی Not	۶
And	۷
Or	۸
Xor	۹

## -مقدمات بانک اطلاعاتی Access

### Access امروزی مایکروسافت

Access فقط یک قسمت از استراتژی کلی مدیریت داده های مایکروسافت است. مانند همه پایگاه های داده ای رابطه ای خوب، در Access به شما اجازه داده می شود که به آسانی اطلاعات مربوط به هم را به هم پیوند دهید - مانند پیوند دادن مشتری به اطلاعاتی که شما وارد می کنید. ولی Access با داشتن ویژگی های برقراری ارتباطات قدرتمند دیگر، از سایر مزایای پایگاه های داده ها نیز برخوردار است. Access می تواند منابع دیگر، مانند بسیاری از برنامه های پرطرفدار بانک اطلاعات PC : (مانند Paradox, dBASE, Structured Query Language-SQL (زبان پرس و جوی ساختار یافته) در میز کار، در سرورها، در مینی کامپیوترها، یا در کامپیوترهای بزرگ، و داده های ذخیره شده در سرورهای اینترنت یا اینترنت Web را مستقیماً به کار ببرد. Access به طور کامل از فن آوری ActiveX مایکروسافت نیز پشتیبانی می کند. بنابراین یک برنامه کاربردی آن می تواند برای سایر برنامه های کاربردی Office مانند برنامه های Word, Excel, Powerpoint, Publisher, FrontPage, Outlook, Microsoft OneNote و برنامه جدید Microsoft Access یک برنامه خدمات گیرنده و خدمات دهنده محسوب شود.

Access یک برنامه کاملاً پیشرفته برای کار تحت سیستم عامل ویندوز مایکروسافت محسوب می شود. و به کمک آن میتوان با هر نوع منبع داده ای، برنامه های کاربردی را سریعاً ساخت. در واقع می توانید برنامه های کاربردی ساده را با تعریف فرم ها و گزارش ها بر اساس داده هایتان ساخته و به کمک چند عبارت ویژوال بیسیک آنها را به یکدیگر پیوند دهید، لذا دیگر نیازی به نوشتن کدهای پیچیده در محتوای یک برنامه نویسی کلاسیک نیست. از آنجایی که در Access زبان ویژوال بیسیک به کار می رود، می توانید از مهارت های سایر برنامه های Office یا ویژوال بیسیک مایکروسافت در ساخت این برنامه ها استفاده کنید.

در مشاغل کوچک اداری تجاری (و برای مشاوره درباره ساخت برنامه برای این مشاغل کوچک)، ویژگی های میز کار Access کل نیازهای شما را برای ذخیره و مدیریت داده های به کار رفته در یک کسب کوچک نوعی برآورده می سازند. Access همراه با SQL Server میکروسافت - در روی میز کار یا بر روی یک سرور - یک روش ایده آل برای بسیاری از شرکت های نه چندان بزرگ جهت ساخت برنامه های کاربردی جدید برای ویندوز به طرز سریع و کم هزینه می باشد. برای شرکت های بزرگ با سرمایه گذاری های کلان بر روی برنامه های کاربردی بانک اطلاعات / پایگاه داده های رابطه ای و نیز تکثیر برنامه های کاربردی میزکار کد قبلی بر بانک های اطلاعاتی PC می باشند، Access ابزارهایی جهت تسهیل پیوند داده های موجود از روی کامپیوترهای بزرگ و PC را در یک برنامه کاربردی مبتنی بر ویندوز ارائه داده است.

و نیز Access میکروسافت یک منبع مستقیم اطلاعات است که بر روی یک اینترنت یا بر روی World Wide Web (شبکه جهانی وب) انتشار داده می شود. صفحه های دسترسی به داده ها به شما اجازه ساخت سریع و به کار بردن برنامه های کاربردی اینترنت را که مستقیماً در Access ساخته اید می دهند، درست شبیه اینکه این برنامه ها را در Access ساخته باشید. صفحه های دسترسی به داده ها می توانند داده های ذخیره شده در بانک اطلاعاتی Access یا در SQL Server میکروسافت را نشان دهند. Access 2003 میکروسافت شامل ویژگی های جدید و بهینه شده ای است که وارد و صادر کردن داده ها به فرمت XML (زبان ذخیره سازی داده ها در Web) را ممکن می سازد و برای پیوند مستقیم به یک منبع داده ای XML در پایگاه وب Microsoft SharePoint Services به کار می رود. شما می توانید داده ها (یا زیر مجموعه ای از داده های) ذخیره شده در بانک اطلاعات Microsoft Access یا SQL Server را به یک سرور SharePoint صادر کرده و سپس این پرونده ها را به هم پیوند داده و آنها را به برنامه کاربردی اصلی برگردانید.

## کاربرد برنامه Access

اگر با کامپیوترهای شخصی زیاد کار می کنید، بی تردید از واژه پردازها یا برنامه های کاربردی صفحه گسترده در کارهای خود استفاده می نمایید و احتمالاً استفاه از واژه پردازها را از زمانی که محصولات مبتنی بر کاراکتر تحت MS-DOS به کار برده می شوند شروع کرده و به تدریج آنها را به واژه پردازهایی که تحت سیستم عامل ویندوز اجرا می شوند ارتقاء داده اید . و نیز احتمال می رود که اکنون از مزیت داشتن بعضی نرم افزارهای بانک اطلاعاتی، یا به منزله جزئی از بسته های مجتمع سازی نشده مانند Microsoft Works یا به صورت یک برنامه جداگانه برخوردار هستید.

اکنون مدت های طولانی است که برنامه های بانک اطلاعاتی برای استفاده کلیه کاربران کامپیوترهای شخصی به بازار ارائه شده اند، ولی متأسفانه بیشتر این برنامه ها یا برنامه های مدیریت ذخیره سازی ساده داده ها می باشند که برای ساخت برنامه های کاربردی مناسب نمی باشند یا سیستم های پیچیده توسعه برنامه های کاربردی هستند که یادگیری و استفاده از آنها سخت می باشد. امروزه حتی افرادی که با کامپیوتر به خوبی آشنا هستند نیز از سیستم های بانک های اطلاعاتی پیچیده دوری می کنند، مگر اینکه مجبور به استفاده از یک بانک اطلاعاتی اختصاصی تکمیل شده باشند اکنون، با ارائه Access مایکروسافت، یک سیستم بانک اطلاعاتی ساده در اختیار همگان می باشد. و کاربران بسیاری برای ساخت بانک های اطلاعاتی ساده و کاملاً پیشرفته از آن استفاده می کنند.

اکنون که هفتمین نسخه Access ارائه شده است. این نرم افزار به مراتب از پنجمین نسخه خود که برای نسخه های ۳۲ بیتی ویندوز طراحی شده بود قدرتمندتر گشته و نیاز به بررسی نحوه استفاده از کامپیوتر برای انجام کار تحت Access دارد.

## بانک اطلاعاتی چیست؟

به یک زبان ساده، یک DataBase ( بانک اطلاعاتی) به مجموعه رکوردها و پرونده هایی اطلاق می شود که به منظور خاصی سازماندهی شده اند. برای مثال، می توانید اسامی و نشانی کلیه دوستان یا مشتریان را بر روی کامپیوتر خود نگه دارید. و یا کلیه

نامه هایی را که می نویسید نگهداری کرده و آنها را به ترتیب اسامی گیرندگان آنها مرتب سازید و یا مجموعه ای از پرونده های حاوی، اطلاعات مالی - پول های پرداختی و حساب های دریافتی و موازنه حساب های خود را روی کامپیوتر ذخیره کنید. اسناد واژه پرداز که بر حسب عنوان سازماندهی می شوند نیز به نوعی یک بانک اطلاعاتی را تشکیل می دهند. پرونده های صفحه گسترده ای که بر حسب کاربران آنها سازماندهی شده اند نیز نوع دیگری از یک بانک اطلاعاتی میباشند. میانبرها به کلیه برنامه ها در روی منوی Start ویندوز نیز نوعی بانک اطلاعاتی محسوب می شوند و میانبرهای اینترنت به پوشه Favorite شما هم یک بانک اطلاعاتی است.

اگر کار شما کاملاً سازماندهی شده باشد، می توانید صدها کاربردگر یا میانبر را با استفاده از پوشه ها و زیر پوشه ها اداره کنید. هنگام انجام این کار، شما مدیر بانک اطلاعاتی محسوب می شوید. ولی اگر به اشکال بزرگی برخوردید چه باید کرد؟

چگونه می توان اطلاعات مربوط به مشتریان متعدد را که در اسناد و پرونده های کاربردگر های گوناگون درج شده اند جمع آوری کرد و به آنها دسترسی داشت؟ چگونه می توان با وارد کردن اطلاعات جدید همین اطلاعات رابطه برقرار نمود؟ چگونه می توان مطمئن شد که اطلاعات وارد شده صحیح می باشند؟ چگونه می توان اطلاعات را با دیگران به اشتراک گذاشت، ولی به آنان اجازه تغییر آنها را نداد؟ برای انجام این کارها، به یک سیستم مدیریت اطلاعات یا (DataBase Management System (DBMS نیاز دارید.

### بانک های اطلاعاتی رابطه ای

تقریباً کلیه سیستم های جدید مدیریت بانک اطلاعاتی ، اطلاعات را با استفاده از الگوی مدیریت رابطه ای بانک اطلاعاتی ، ذخیره و اداره می کنند. در یک سیستم مدیریت رابطه ای بانک اطلاعاتی که به RDBMS معروف می باشد، سیستم کلیه اطلاعات را از طریق جدول ها کنترل می کند. در جدول هایی (Tables) که حتی یک عنوان مثل نام مشتری یا محصول ذخیره می شوند، هر جدول دارای ستون ها (Columns) یا فیلدها (Fields) می باشد که هر ستون حاوی اطلاعات مختلفی درباره موضوع (مثل نشانی مشتری یا شماره های تلفن) می باشد. و نیز یک جدول دارای ردیف ها (ROWS) یا رکوردها (Records) است که کلیه مشخصه های یک نمونه

از موضوع (مثل اطلاعات درباره یک محصول یا یک مشتری خاص) در آنها ثبت می شوند. ولی هنگامی که از یک بانک اطلاعاتی پرس و جویی ( Query ) می کنید (آوردن اطلاعات از یک یا چند جدول ) همیشه نتیجه شبیه جدول دیگری به نظر خواهد رسید.

اصطلاح رابطه ای ( Relational ) از این واقعیت ناشی می شود که هر جدول در یک بانک اطلاعاتی حاوی اطلاعات مربوط به یک موضوع و فقط یک موضوع می شود. اگر الگوی مدیریت رابطه بانک اطلاعاتی را مرور کنید متوجه خواهید شد که رابطه ( Relation ) برای مجموعه ای از ردیف ها ( یک جدول ) درباره یک موضوع به کار می رود. و نیز، اطلاعات مربوط به دو دسته از اطلاعات (مانند مشتری ها و سفارشات ) می توانند براساس مقدارهای مرتبط داده ها به عنوان یک ورودی منظور شوند. برای مثال، ذخیره اسم و نشانی مشتری برای هر سفارشی که مشتری می دهد کار زائدی است. در یک سیستم رابطه ای بانک اطلاعاتی ، اطلاعات مربوط به هر سفارش دارای فیلدی است که اطلاعاتی مانند شماره مشتری را ذخیره می کند و از این فیلد می توان برای برقراری ارتباط و اخذ هر نوع اطلاعات مربوط به این مشتری استفاده کرد.

در عین حال می توانید اطلاعات مقدارهای مرتبط به هم در چند جدول یا پرس وجو را با هم یکی سازید. برای مثال می توان از اطلاعات شرکت که شامل اطلاعات تماس است. برای یافتن نشانی های تماس با یک شرکت خاص استفاده کرد و یا از اطلاعات مربوط به کارکنان همراه با اطلاعات هر دپارتمانی برای پیدا کردن بخشی که کارمند خاصی در آن کار می کند استفاده نمود.

چند اصطلاح مربوط به بانک اطلاعاتی / پایگاه داده های رابطه ای؛

**Relation** : اطلاعات درباره فقط یک موضوع مانند مشتریان سفارش ها، کارکنان، تولیدات یا شرکت ها. این اطلاعات معمولاً در سیستم مدیریت رابطه ای بانک اطلاعاتی در یک جدول ( table ) نگه داری می شوند.

**Attribute** : اطلاعات ویژه درباره یک موضوع، مثل نشانی مشتری یا ارزش یک سفارش، مشخصه های فوق معمولاً در ستون ( فیلد) یک جدول قرار داده میشوند.

**Instance** : یک عضو مخصوص در یک رابطه- مثل یک مشتری یا محصول منفرد. این جزء معمولاً در جدول به صورت یک رکورد، یا در ردیف نگه داری می شود.

**Relationship** : روشی که در آن اطلاعات در یک رابطه مربوط به اطلاعات در رابطه دیگری می شود. برای مثال، هر مشتری دارای « یک- به - چند» رابطه با سفارشها می باشد. رابطه شرکت ها

با افراد مورد تماس یک رابطه «چند-به-چند» می باشد، زیرا هر مشتری می تواند چند نشانی تماس داشته باشد و یک مشتری نیز می تواند با چند شرکت تماس داشته باشد.

**Join** : پروسه به هم پیوستگی جدول ها یا پرس و جوها در جدول ها از طریق مقادارهای داده‌های مربوط به هم. برای مثال، مشتریان را می توان با تطابق شماره شناسایی مشتری ( customer ID) در یک جدول مشتری به سفارشها در جدول سفارشات به هم مربوط ساخت.

### قابلیت های یک بانک اطلاعاتی

یک RDBMS کنترل کامل برای چگونگی تعریف داده ها، کار با آنها و اشتراک گذاری داده ها با دیگران را ممکن می سازد. این سیستم ویژگی های پیشرفته برای کاتالوگ و مدیریت داده های بسیار در جدول های گوناگون را نیز ارائه می دهد. یک RDBMS از سه قابلیت عمده تعریف داده ها، دستکاری داده ها و کنترل داده برخوردار است.

**تعریف داده ها:** در RDBMS می توانید داده های ذخیره شده در بانک اطلاعاتی خود را از نظر نوع داده ( مثلاً داده عددی یا داده حرفی ) و چگونگی ارتباط داده ها مشخص سازید. در بعضی موارد شما می توانید نحوه قالب بندی داده و چگونگی تعیین صحت آن را نیز مشخص سازید.

**دستکاری داده ها:** به روش های گوناگون م میتوان با داده ها کار کرد. شما می توانید فیلدهایی از داده را که لازم دارید مشخص سازید، داده ها را از فیلتر بگذرانید و آنها را مرتب سازید. شما می توانید داده ها را با اطلاعات مربوطه مرتبط ساخته و و خلاصه ای از داده ها را تهیه نمایید و یا می توانید مجموعه ای از داده ها را انتخاب کرده و از RDBMS بخواهید تا این داده ها را به روز رسانده ، حذف نموده و بر روی جدول دیگر کپی نماید یا جدول جدیدی برای این داده ها بسازید.

**کنترل داده ها:** شما می توانید تعیین کنید که چه افرادی مجاز هستند تا داده‌های شما را خوانده به روز رسانده یا داده های جدیدی را در پرونده های شما وارد کنند. در بسیاری از موارد می توانید در عین حال مشخص کنید که داده هایتان چگونه به اشتراک گذاشته شوند، و توسط چه کاربرانی به روزرسانی شوند.

کلیه این عملیات به کمک ویژگی های قدرتمند Access میکروسافت انجام میشوند. در زیر به بررسی چگونگی انجام این عملیات می پردازیم و نحوه انجام آنها را با روش های مورد استفاده در برنامه های واژه پرداز و صفحه گسترده مقایسه می کنیم.



## Access مایکروسافت به منزله یک RDBMS

یک بانک اطلاعاتی میز کار Access ساخت شرکت مایکروسافت ( که در آن از پرونده ها با پسوند .mdb . استفاده می شود) یک RDBMS با قابلیت کامل عملیاتی می باشد. در این برنامه کلیه عملیات تعریف داده ها، دستکاری داده ها و کنترل داده برای داده های حجیم را می توان به خوبی انجام داد.

یک بانک اطلاعاتی میز کار Access را می توان یا به صورت RDBMS مستقل در روی یک ایستگاه کاری و یا در حالت اشتراکی خدمات گیرنده / خدمات دهنده در یک شبکه به کار برد. یک بانک اطلاعاتی میز کار یک منبع داده ها برای داده های نمایش داده شده در صفحه های Web روی اینترنت شرکت شما نیز میباشد. هنگامی که یک برنامه کاربردی را با بانک اطلاعاتی میز کار Access میسازید، Access یک RDBMS است. از Access می توانید در عین حال برای ساخت برنامه های کاربردی در یک پرونده پروژه ( با پسوند .adp ) مرتبط به SQL server مایکروسافت نیز استفاده کنید و می توانید داده های سرور را با دیگر برنامه های کاربردی یا با سایر کاربران web به اشتراک بگذارید.

هنگامی که یک پرونده پروژه Access ( با پسوند .adp ) را می سازید، SQL server ( یا SQL server Desktop Engine – Microsoft MDSE ) یک RDBMS میباشد.

### تعریف داده ها و ذخیره سازی

در ضمن کار با یک سند یا یک کاربرگ، معمولاً به آزادی می توانید محتویات سند یا خانه کاربرگ را تعریف کنید. در یک صفحه مشخص از یک سند، می توانید پاراگراف، جدول، نمودار یا چند ستون از داده ها با قلم های مختلف را به متن اضافه کنید. در یک ستون داده شده بر روی یک کاربرگ، می توانید در بالای ستون یک عنوان با داده های متنی برای چاپ یا نمایش بر روی صفحه مانیتور قرار دهید. و نیز می توانید از قالب های متعدد عددی، بسته به عملیات هر ردیف در یک ستون استفاده کنید. شما به این انعطاف پذیری نیاز دارید، زیرا سند واژه پرداز شما باید قادر به انتقال پیام بر روی یک صفحه چاپی باشد و کاربرگ باید داده هایی را که تحلیل می کند بتواند ذخیره کند و محاسبات مربوطه را انجام داده و نتیجه را نمایش دهد.

این انعطاف پذیری به خصوص برای حل مشکلات کارهای به نسبت کوچک ولی به خوبی تعریف شده بسیار سودمند است. در صورتی که تعداد صفحه های اسناد به چند دوجین برسند، سندها دست و پاگیر شده و انجام کار مشکل خواهد شد. در کاربرگها نیز هنگامی که تعداد خانه ها به چند صد ردیف اطلاعات برسد، نیز کار سخت می شود. با افزایش مقدار داده ها، فضای ذخیره سازی داده ها، فراتر از حد در نظر گرفته شده برای واژه پرداز یا صفحه گسترده در کامپیوترتان خواهد رفت. اگر هدف شما اشتراک گذاری سند یا کاربرگ می باشد، کنترل چگونگی استفاده دیگران از این اسناد و وارد کردن داده ها به این اسناد مشکل ( البته نه غیرممکن ) می گردد. برای مثال، در یک کاربرگ، حتی اگر فقط به دو خانه برای درج تاریخ و یک مقدار پولی نیاز داشته باشد، باز این احتمال وجود دارد که کاربری به اشتباه یک مقدار حرفی را در این دو فیلد وارد کند. در بعضی برنامه های صفحه گسترده به شما اجازه داده می شود تا یک ناحیه، بانک اطلاعاتی را در یک کاربرگ برای مدیریت اطلاعات مورد نیاز جهت ارائه نتایج مورد نظر در نظر بگیرید. ولی به هر حال باید محدودیت فضای ذخیره سازی در یک برنامه صفحه گسترده را منظور دارید و در هر صورت کنترل کاملی بر روی آنچه که در ردیف ها و ستون های ناحیه بانک اطلاعاتی وارد می شود نخواهید داشت. و نیز، اگر باید داده هایی به جز داده های عددی و حرفی را کنترل نمایید، باید مطمئن شوید که برنامه صفحه گسترده شما این اطلاعات را به منزله یک پرونده صوتی یا تصویری تلقی ننماید.

یک سیستم RDBMS به شما اجازه تعریف نوع داده هایتان و تعیین نحوه ذخیره سازی آنها را می دهد و نیز می توانید قواعدی را که RDBMS می تواند برای کسب اطمینان از یکپارچگی داده هایتان به کار برد، مشخص سازید. در ساده ترین شکل آن، یک قاعده تعیین صحت داده ها (Validilation Rule) می تواند این اطمینان را ایجاد کند که کاربران به طور تصادفی کاراکترهای الفبایی را به جای عدد و فیلدی وارد نکنند. در قواعد دیگر ، می توان مقادیرهای مورد قبول یا دامنه مقدار برای داده ها را مشخص ساخت. در بیشتر سیستم های پیشرفته، می توانید رابطه بین مجموعه ای از داده ها (معمولاً جدول ها و پرونده ها) را مشخص ساخته و از RDBMS بخواهید تا یکپارچگی داده ها را کنترل کند. برای مثال، می توانید از سیستم بخواهید به طور خودکار کار بازمینی ترتیب ورودی ها برای مشتریان معتبر را انجام دهد.

UML SSADM طراحی و بسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
در یک بانک اطلاعاتی میزکار Access (mdb) شما از انعطاف پذیری کاملی برای  
تعریف داده های خود (به صورت متن، عدد، تاریخ، زمان، پول، فرآیندهای اینترنتی،  
تصویر، صوت، مستندات و صفحه های گسترده) و تعیین نحوه ذخیره داده هایتان (طول  
رشته، دقت عدد، و دقت تاریخ / زمان) و تعیین نحوه نمایش داده ها در روی صفحه  
مانیتور یا در چاپ برخوردار هستید. شما می توانید قواعد ساده یا پیچیده ای برای تأیید  
صحت داده ها را نیز برقرارسازید. شما می توانید از Access بخواهید که رابطه بین  
پرونده ها یا جدول هادر بانک اطلاعاتی شما را بررسی و صحت آنها را بازبین نماید.  
هنگامی که پروژه Access (.adp) به بانک اطلاعاتی SQL Server وصل می شود،  
SQL Server نیز کلیه قابلیت های فوق را ارائه می دهد.

از آنجایی که Access یک برنامه کاربردی با فن آوری های جدید برای ویندوز  
مایکروسافت است شما می توانید از کلیه تسهیلات اشیاء ActiveX و کنترل های  
سفارشی ActiveX نیز استفاده کنید. ActiveX یک فن آوری پیشرفته ویندوز است.  
که به نوبه خود اجازه پیوند اشیاء به اشیاء تعبیه شده در میزکار بانک اطلاعاتی  
Access یا SQL Server را می دهد. این اشیاء شامل تصاویر صفحه های گسترده و  
مستندات برنامه های کاربردی دیگر مبتنی بر ویندوز است که در آنها نیز از ActiveX  
پشتیبانی می شود.

Access 2003 مایکروسافت را می توان به منزله یک سرور ActiveX نیز به کار  
برد و اجازه داد تا بتوانید از برنامه های دیگر مبتنی بر ویندوز، بانک اطلاعاتی Access  
را باز کرده و اشیاء آن (مانند جدول ها، پرس و جوها، و فرم ها) را دستکاری کنید.  
در فرم ها و گزارش های Access می توانید کنترل های سفارشی ActiveX را به  
کار ببرید تا عملیات برنامه خود را بهتر سازید. کنترل های ActiveX اشیاء طراحی  
بهینه ای را در اختیار می گذارد که امکان ارائه داده های پیچیده به فرم ساده تر و به  
طرز گرافیکی را فراهم می سازد. بیشتر کنترل های ActiveX مجموعه ای غنی از  
«عملیات» (موسوم به روش ها (Methods)) در اصطلاح شناسی شیء ) را فراهم می  
سازند که این عملیات را می توان در یک پروسجر فراخواند و مشخصه ها را برای  
چگونگی نمایش و رفتار کنترل تنظیم نمود. برای مثال، اگر بخواهید به کاربران خود

اجازه درج تاریخی را با انتخاب تاریخ از یک تصویر تقویم بدهید، می توانید به روش فوق عمل کنید.

یکی از کنترل های ActiveX که می توانید در برنامه های کاربردی Access به کار ببرید ، کنترل تقویم است که یک رابط گرافیکی از این نوع را ارائه می دهد.

## معماری Microsoft Access

شالوده معماری Access بر چیزی موسوم به Object (شیء) بنا شده است. در یک بانک اطلاعاتی Access اشیاء اصلی عبارتند از جدول ها، پرس و جوها، فرم ها، گزارش ها ، صفحه های دسترسی به داده ها، ماکروها و ماجول ها.

اگر قبلاً با کامپیوترهای رومیزی یا دیگر برنامه های بانک اطلاعاتی کار کردن باشید احتمالاً به واژه Database (بانک اطلاعاتی/ پایگاه داده ها) به معنای (فقط پرونده هایی که شما در آنها داده های خود را ذخیره می کنید) برخوردیده اید. ولی در Access یک پرونده بانک اطلاعاتی میز کار، که دارای پسوند .mdb می باشد در عین حال نیاز به این دارد که کلیه اشیاء عمده با داده های ذخیره شده مرتبط باشند و نیز اشیای خودکارسازی استفاده از این داده ها تعریف شوند. یک برنامه Access را می توان با به کار بردن یک پرونده پروژه (با پسوند پرونده .adp) نیز ساخت که شامل اشیایی می گردد که برنامه پیوند داده شد، شما به بانک اطلاعات SQL Server را مشخص می سازد که در این بانک اطلاعاتی جدول ها و پرس و جوها ذخیره می شوند. اشیاء اصلی در یک بانک اطلاعاتی Access به شرح زیر می باشند:

**Table (جدول) :** جدول از اشیایی است که توسط شما تعریف و برای ذخیره کردن داده ها به کار می رود. هر جدول حاوی اطلاعاتی درباره یک شیء خاص مانند شیء مشتری یا سفارش می باشد. هر جدول دارای فیلدهایی (ستون ها) برای ذخیره انواع مختلف داده ها، مانند اسم یا نشانی ، و رکوردها (ردیف ها) برای درج اطلاعات درباره یک مورد از شیء می باشد. شما می توانید یک کلید اولیه (یک یا چند فیلد حاوی

مقدارهای منحصر به فرد برای هر رکورد) و یک یا چند ایندکس بر روی هر جدول، برای بازیابی سریع داده ها داشته باشید.

**Query** (پرس و جو): پرس و جو به شیء اطلاق می شود که یک نمای سفارشی از داده های یک یا چند جدول را ارائه دهد. در **Access** می توانید یک پرس و جو گرافیکی با استفاده از تسهیلات نمونه **OBE** را به کار ببرید یا می توانید با نوشتن عبارت های **SQL** پرس و جوها را خود به وجود آورید. از پرس و جوها می توان برای انتخاب، به روز رسانی، درج یا حذف داده ها استفاده کرد. و نیز پرس و جوها را برای ساخت جدول های جدیدی از داده های موجود در یک یا چند جدول به کار برد. در صورتی که برنامه **Access** شما یک پرونده پروژه وصل شده به یک بانک اطلاعاتی **SQL Server** باشد، می توانید انواع ویژه ای از پرس و جوها را ایجاد کنید که حاوی تابع ها و روال های ذخیره شده ای باشند که عملیات پیچیده و مرکب را به طور مستقیم بر روی سرورها انجام دهند.

**Form** (فرم) – فرم یک شیء می باشد که در درجه اول به منظور وارد کردن داده ها یا نمایش یا برای کنترل اجرای برنامه به کار می رود. از فرم ها برای سفارشی کردن نمایش داده هایی که برنامه شما از پرس و جوها یا جدول ها استخراج می کند استفاده می شود.

فرم ها را می توان چاپ کرد و شما می توانید فرمی را طراحی کنید که در واکنش به بعضی رویدادها، به تعداد رویداد یک ماکرو یا یک روال ویژوال بیسیک را اجرا کند. مثلاً روالی را هر بار با تغییر داده خاصی، دوباره اجرا کند.

**Report**(گزارش): گزارش شیء است که برای قالب بندی محاسبه، چاپ و خلاصه کردن اطلاعات انتخاب شده به کار می رود. قبل از چاپ هر گزارش در **Access** می توانید آن را روی صفحه مانیتور مشاهده کنید.

**Data Access Page** (صفحه دسترسی به داده ها): این صفحه شیء است که به یک پرونده **HTML** حاوی کد ویژه و کنترل **ActiveX** پیوند خورده و نمایش و ویرایش داده ها را تحت **Internet Explorer** میکروسافت تسهیل می بخشد. شما می توانید این پرونده ها را بر روی اینترنت شرکت خود منتشر کرده و به دیگر کاربران روی

شبکه شرکت خود که برنامه های Office و Internet Explorer نسخه ۵ یا نسخه های بعدی روی کامپیوترهایشان نصب شده است ، اجازه دهید داده های شما را مشاهده جستجو و ویرایش نمایند.

**Macro (ماکرو) :** ماکرو یک تعریف ساختاری از یک یا چند عمل می باشد که شما می خواهید Access در واکنش به یک رویداد مشخص این عملیات را انجام دهد. برای مثال، می توانید ماکرویی را طراحی کنید که یک فرم ثانوی را در پاسخ به انتخاب یک ففره از فرم اصلی باز نماید. برای مشخص کردن اینکه چه هنگامی در یک ماکرو باید عملیاتی صورت گرفته و یا بعضی عملیات نادیده گرفته شوند می توانید شرایط ساده ای را در یک ماکرو به کار ببرید. از ماکروها می توان برای باز کردن و اجرای پرس و جوها، باز کردن جدولها، یا چاپ یا مشاهده گزارش ها استفاده کرد و نیز می توان ماکروها یا بعضی از روال های ویژوال بیسیک را از داخل یک ماکرو دیگر اجرا نمود.

**Module (ماجول) :** ماجول یک شیء است که حاوی روال های سفارشی می باشد. این روال ها با استفاده از ویژوال بیسیک کد می شوند. مدول ها جریان های غیروابسته عملیات را نشان می دهند و به شما اجازه به دام انداختن خطاها را می دهند یعنی کاری که با ماکروها نمی توانید آن را انجام دهید. هر مدول می تواند یک شیء مجزا حاوی تابع هایی باشد که از هر جایی در برنامه بتوان آن را فراخواند و یا مستقیماً وابسته به یک فرم یا گزارش برای واکنش نشان دادن به رویدادها در فرم یا گزارش های وابسته می باشد. ماجول ها بسیار قدرتمند تر از ماکروها هستند.

جدول ها، داده هایی را ذخیره می کنند که بتوان آنها را با پرس و جوها استخراج نمود ، در گزارش ها نمایش داد و در فرم ها یا صفحه های دسترسی به داده ها آنها را به روز رساند. البته، توجه داشته باشید که اطلاعات فرمها، گزارش ها و صفحه های دسترسی به داده ها را می توان مستقیماً از جدول ها استخراج کرد یا آنها را با گذراندن از فیلتر پرس و جو ها برای فرم ها و گزارش ها به کار برد. در پرس و جوها می توان از تابع های ویژوال بیسیک برای انجام محاسبات اختصاصی بر روی داده های بانک اطلاعاتی استفاده کرد. به علاوه در Access تابع های از پیش کار گذاشته شده

متعددی وجود دارند که در پرس و جوها به شما امکان خلاصه سازی و قالب بندی داده هایتان را می دهند.

رویدادها در فرم ها و گزارش ها می توانند روال های ویژوال بیسیک یا ماکروها را هدف قرار دهند یک رویداد (Event) به هر تغییری که در وضعیت یک شیء Access داده شود اطلاق می گردد. برای مثال می توانید ماکروها یا روال هایی به زبان ویژوال بیسیک برای واکنش به باز کردن یک فرم ، بستن یک فرم ، وارد کردن یک ردیف جدید بر روی یک فرم، یا تغییر داده ها در رکورد جاری یا در یک کنترل (Control) انفرادی (یک شیء بر روی یک فرم یا یک گزارش که حاوی داده ها است) بنویسید. و حتی می توانید ماکرو یا یک روال ویژوال بیسیک را به نحوی طراحی کنید که با فشردن کلیدهای منفرد بر روی صفحه کلید به هنگام وارد کردن داده ها واکنش نشان دهد.

**یک نرم افزار پایگاه داده امکانات زیر را در اختیار ما قرار می دهد:**

#### **۱- ذخیره اطلاعات**

نرم افزار پایگاه داده به ما امکان می دهد که اطلاعات متعدد و مرتبط با یکدیگر را در یک پایگاه ذخیره نماییم و در صورت نیاز اطلاعات وارد شده را اصلاح یا حذف نماییم و یا اطلاعات جدیدی را به پایگاه داده اضافه نماییم.

#### **۲- جستجوی اطلاعات :**

نرم افزار پایگاه داده به ما امکان می دهد که به آسانی اطلاعات ذخیره شده را جستجو نماییم.

#### **۳- تجزیه و تحلیل اطلاعات**

نرم افزار پایگاه داده امکان انجام محاسبات بر روی اطلاعات ذخیره شده در پایگاه داده را می دهد و به این طریق می توان اطلاعات را مورد تجزیه و تحلیل قرار داد.

#### **۴- مدیریت اطلاعات**

نرم افزار پایگاه داده، مدیریت اطلاعات در حجم بالا را به سادگی انجام می دهد. از نرم افزارهای پایگاه داده معروف می توان نرم افزارهای زیر را نام برد:

نرم افزار مدیریت پایگاه داده با تعداد کاربران کم و حجم اطلاعات متوسط و زیاد

### Microsoft Access

نرم افزار مدیریت پایگاه داده با تعداد کاربران زیاد و حجم اطلاعات بسیار زیاد

### Microsoft SQL Server

نرم افزار مدیریت پایگاه داده با تعداد کاربران زیاد و حجم اطلاعات بسیار زیاد

### Oracle

## آشنایی با فیلد، رکورد و جدول

اطلاعات موجود در پایگاه داده توسط مجموعه‌ای از اشیاء سازماندهی می‌شود که مهمترین شیء آن جدول (TABLE) است. هر پایگاه داده حداقل شامل یک جدول است.

هر سطر جدول یک رکورد (Record) محسوب می‌شود. یک رکورد از مجموعه‌ای از اقلام اطلاعاتی تشکیل شده است که به هر کدام از این اقلام فیلد (Field) گفته می‌شود. هر ستون جدول یک فیلد محسوب می‌شود. مثلاً نام، نام خانوادگی، نام پدر و... هر کدام، یک فیلد محسوب می‌شوند.

### فیلد (Field):

به فضایی که یک قلم اطلاعاتی در آن ذخیره می‌شود، فیلد گفته می‌شود.

### رکورد (Record):

به مجموعه‌ای از فیلدهای مربوط به یک موضوع، رکورد گفته می‌شود.

### جدول (Table):

به مجموعه‌ای از رکوردها که دارای فیلدهای یکسانی هستند، جدول گفته می‌شود.

هر فیلد ممکن است محتوای متنی، عددی، ریالی، تصویری و... داشته باشد. برای اینکه نرم افزار پایگاه داده تشخیص دهد که محتوای فیلد از چه نوعی است و چه میزان فضا برای این فیلد باید در نظر گرفته شود، برای هر فیلد خواصی در نظر گرفته می‌شود.



UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
 خواص یک فیلد مشخص می کند که آن فیلد از چه نوعی است و حداکثر فضای لازم  
 برای آن فیلد چقدر است.

### خواص فیلد (Field Properties)

به مجموعه ای از صفات فیلد نظیر نوع فیلد، طول فیلد، شرایط صحت فیلد و.. خواص فیلد گفته می شود.

خواص فیلد در پایگاه داده بسیار مهم است، زیرا به نرم افزار پایگاه داده کمک می کند صحت اطلاعات وارد شده را بررسی کند. یکی از خواص مهم فیلد، نوع داده فیلد است.

### نوع داده فیلد (Field Data Type)

نوع ساختار اطلاعاتی یک فیلد را نوع داده فیلد می گویند.  
 هر فیلد برای ذخیره کردن نوع خاصی از داده ها مورد استفاده قرار می گیرد. مثلاً اگر بخواهیم نام یا نام خانوادگی را ذخیره کنیم، نوع فیلد را متنی در نظر می گیریم ولی برای نگهداری تاریخ تولد افراد، نوع فیلد را عددی در نظر می گیریم.  
 در جدول زیر انواع داده های پایگاه داده Access نمایش داده شده است.

نوع داده	کاربرد
Text	برای متن یا ترکیب متن و عدد مناسب است. علاوه بر متن، فیلدهای عددی که نیاز به محاسبات نداشته باشند (مانند شماره تلفن) نیز می توانند در این نوع داده ذخیره شوند. حداکثر طول این نوع داده ۲۵۵ کاراکتر است.
Memo	این داده نیز از نوع متنی است، با این تفاوت که از نظر طول فیلد محدودیتی ندارد. اطلاعات توضیحی ، شرح و نکات از این نوع داده استفاده می نماید.
Number	برای داده های عددی که محاسبات ریاضی بر روی آنها انجام می شود بکار می رود.
Date/Time	جهت درج تاریخ و ساعت بکار می رود بصورتی که می توان آنها را با یکدیگر مقایسه نمود.
Currency	عددی است که بر اساس پول رایج کشورها قالب بندی می شود. با استفاده از این

گزینه می توان عددی را تا ۱۵ رقم در سمت چپ نقطه اعشاری و تا چهار رقم در سمت راست آن وارد کرد.	
برای هر رکورد عدد منحصر به فردی را که به صورت صعودی افزایش می یابد، درج می نماید.	Auto Number
جهت ایجاد فیلدی بکار می رود که پاسخ آن بصورت بله / خیر است .و در این فیلد فقط یکی از دو حالت تعیین شده بله / خیر - درست / غلط - روشن / خاموش قرار می گیرد.	YES/NO
برای قرار دادن تصویر از این فیلد استفاده میشود . همچنین برای درج و دسترسی به اشیاء از نرم افزارهای دیگر استفاده می شود.	OLE
فیلدی شامل اقلام مختلف را ایجاد می کند که این اقلام از جدول دیگری نمایش داده می شوند که کاربر با استفاده از آن می تواند مقادیر مختلف را از فهرستی بازشدنی انتخاب نماید.	Look up Wizard

جدول ۱ : انواع داده های قابل تعریف جهت فیلدها در نرم افزار Microsoft Access

### آشنایی با فیلد کلیدی (Key Field) و کلید اصلی (Primary Key)

شماره پرسنلی ، کد کالا و شماره نامه نمونه هایی از فیلدهایی هستند که معمولاً به عنوان فیلد کلیدی تعریف می شوند. مقدار فیلد کلیدی هر رکورد جدول باید یکتا باشد و نرم افزار پایگاه داده اجازه نمی دهد فیلد کلیدی در دو رکورد از جدول یکسان باشد. فیلد کلیدی می تواند برای مراجعه سریع به یک رکورد مورد استفاده قرار گیرد. مثلاً شماره پرسنلی هر کارمند عدد منحصر بفردی است که از طریق آن می توان به رکوردی از جدول مراجعه کرد که مشخصات کارمند در آن رکورد ثبت شده است. یک جدول ممکن است چند فیلد کلیدی داشته باشد ولی تنها به یکی از این فیلدها، کلیدی اصلی (Primary Key) گفته می شود ، کلید اصلی می تواند مجموعی از ۲ یا چند فیلد باشد به شرط اینکه مجموع این فیلدها مقدار منحصر بفردی را داشته باشند. همچنین مقدار کلید اصلی نمی تواند خالی باشد. نرم افزار پایگاه داده به صورت پیش فرض اطلاعات یک جدول را بر اساس فیلد کلید اصلی مرتب می کند. همه فیلدهای کلیدی

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار می توانند به عنوان کلید اصلی انتخاب شوند ولی ما باید فیلدی را که مناسبتر است انتخاب نماییم. مثلاً در جدول اطلاعات کارمندان، فیلد شماره پرسنلی و فیلد کد ملی منحصر بفرد هستند ولی فیلد شماره پرسنلی برای کلید اصلی مناسبتر است زیرا ممکن است بعضی از کارمندان کد ملی خود را دریافت نکرده باشند.

از کلید اصلی برای ایجاد ارتباط بین جداول مختلف استفاده می شود.

### فیلد کلیدی (Key Field):

فیلدی که مقدار آن در هر رکورد جدول یکتا باشد

### کلید اصلی (Primary Key):

یک (یا چند) فیلد که مقدار آن منحصر به فرد بوده و مشخص کننده یک رکورد جدول است. کلید اصلی نمی تواند مقدار خالی داشته باشد.

### آشنایی با اجرای نرم افزار Access

اجرای نرم افزار بانک اطلاعاتی Access به دو طریق زیر صورت می پذیرد.

#### ۱- اجرای نرم افزار از منوی Start

بدین منظور نرم افزار Microsoft Access را از مسیر زیر اجرا می کنیم.

Start → All Programs → Microsoft Access

#### ۲- دو بار کلیک بر روی آیکن میانبر نرم افزار

بر روی آیکن میانبر نرم افزار در Desktop دوبار کلیک می کنیم.

پس از اجرای نرم افزار Access به یکی از دو روش فوق، صفحه اصلی آن مطابق شکل صفحه بعد باز می شود. مشابه برنامه های دیگر Office محیط برنامه Access نیز از قسمتهایی مانند نوار عنوان، نوار منو، نوار ابزار و نوار وضعیت تشکیل شده است.

### - اصول باز کردن یک بانک اطلاعاتی

جهت باز نمودن یک بانک اطلاعاتی Access ، مراحل زیر را دنبال می کنیم.

برنامه Access را اجرا می کنیم، در اینصورت پنجره اصلی برنامه ظاهر خواهد شد.

اگر نام بانک اطلاعاتی دلخواه در قسمت Open a File از کادر سمت راست برنامه قرار دارد، بر روی آن کلیک می نماییم تا باز شود. در اینصورت بانک اطلاعاتی انتخاب شده، در پنجره Database نمایش داده خواهد شد.

اگر بانک اطلاعاتی دلخواه در لیست Open a File نبود، بر روی More files.. کلیک می

کنیم، در اینصورت پنجره Open ظاهر می شود.

در پنجره Open و از قسمت Look In درایو و پوشه های درون کامپیوتر را باز نموده و فایل بانک اطلاعاتی مورد نظر را پیدا می کنیم.

بر روی دکمه Open کلیک می کنیم، در اینصورت بانک اطلاعاتی انتخاب شده در پنجره Database باز می گردد.

فایل های بانک اطلاعاتی Access دارای پسوند (Microsoft Database)mdb هستند.

نکته: یک راه آسان جهت باز کردن بانک اطلاعاتی ، دوبار کلیک کردن بر روی فایل بانک اطلاعاتی دلخواه است.

### باز نمودن یک جدول

جهت باز نمودن یکی از جداول بانک اطلاعاتی مراحل زیر را دنبال می کنیم:

بانک اطلاعاتی حاوی جدول دلخواه را باز می کنیم

در پنجره Database بر روی گزینه Tables از لیست Objects کلیک می کنیم.

از لیست جداول موجود جدول دلخواه را انتخاب می کنیم.

بر روی نام جدول دلخواه دوباره کلیک می کنیم یا دکمه Open را از نوار ابزار Database کلیک می کنیم.

در اینصورت جدول به همراه داده های موجود در آن مانند یک صفحه گسترده نمایان می شود.

### - روش های مشاهده جدول

دو روش عمده برای مشاهده جدول معمول است:

Datasheet -

Design -

نمای Datasheet روش پیش فرض Access برای مشاهده جداول است در این نما داده های جدول در قالب ستونها (فیلدها) و سطرها (رکوردها) و بصورت صفحه گسترده نمایش داده می شوند این نما بیشتر جهت مشاهده و ویرایش داده های جدول کاربرد دارد.

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
در نمای Datasheet می توانیم ساختار جدول را مشاهده کرده و آنها را تغییر دهیم. تغییر نام  
فیلد، نوع فیلد، اضافه کردن فیلد جدید و... از کاربردهای این نما می باشد.

### - جهت تغییر روش نمایش جدول

از منوی View یکی از روش های نمایش Datasheet View یا Design View را انتخاب  
می کنیم.

### - نحوه پیمایش رکوردهای جدول

رکوردهای جدول بانک اطلاعاتی را می توانیم به روش های زیر مشاهده نماییم.

۱- پیمایش رکوردها توسط صفحه کلید

۲- پیمایش رکوردها توسط ماوس

۳- پیمایش رکوردها توسط گزینه Go To

۴- استفاده از نوار پیمایش رکوردها

جهت مشاهده رکوردهای جدول با استفاده از کلیدهای صفحه کلید بصورت زیر عمل می کنیم:

یکی از کلیدهای جدول زیر را با توجه به نیاز استفاده می کنیم.

کلید	کاربرد
↑	مکان نما را به رکورد قبلی منتقل می نماید.
↓	مکان نما را به رکورد بعدی انتقال می دهد.
Tab یا → یا Enter	مکان نما را به فیلد بعدی انتقال می دهد.
Shift+Tab یا ←	مکان نما را به فیلد قبلی انتقال می دهد.
Home	مکان نما را به اولین فیلد رکورد جاری منتقل می نماید.
End	مکان نما را به آخرین فیلد رکورد جاری منتقل می نماید.
Ctrl+Home	مکان نما را به اولین فیلد رکورد اول منتقل می نماید.
Ctrl+End	مکان نما را به آخرین فیلد رکورد آخر منتقل می نماید.
Page up	مکان نما را به رکورد صفحه قبل انتقال می نماید.

جدول ۲) کاربرد کلیدهای جهتی جهت پیمایش رکوردها

جهت مشاهده رکوردهای جدول با استفاده از ماوس بصورت زیر عمل می کنیم:

برای رفتن به رکورد بالاتر دکمه ↑ و رکورد پایین تر دکمه ↓ را کلیک می کنیم.

برای رفتن به چند رکورد بالاتر دکمه ↑ و چند رکورد پایین تر دکمه ↓ را کلیک کرده و پایین

نگه می داریم تا به رکورد دلخواه برسیم، سپس دکمه ماوس را رها می کنیم.

جهت پیمایش فیلدهای جدول با استفاده از ماوس بصورت زیر عمل می کنیم:

برای رفتن به فیلد بعدی دکمه → و فیلد قبلی دکمه ← را کلیک می کنیم.

برای رفتن به چند فیلد بعدی دکمه → و چند فیلد قبلی دکمه ← را کلیک کرده و پایین نگه

میداریم تا به فیلد دلخواه برسیم، سپس دکمه ماوس را رها می کنیم.

روش دیگر جهت پیمایش رکوردها ، استفاده از فرمان Go To است.

جهت استفاده از فرمان Go To مراحل زیر را دنبال می کنیم:

جدول مورد نظر را باز می کنیم.

از منوی Edit گزینه Go To را انتخاب می کنیم.

گزینه مناسب را مطابق جدول زیر انتخاب می کنیم.






گزینه	کاربرد
First	مکان نما را به اولین رکورد جدول منتقل می نماید.
Last	مکان نما را به آخرین رکورد جدول منتقل می نماید.
Next	مکان نما را به رکورد بعدی جدول منتقل می نماید.
Previous	مکان نما را به رکورد قبلی جدول منتقل می نماید.
New Record	یک رکورد جدید به جدول اضافه می نماید.

### جدول ۳) کاربرد گزینه های Go TO

- یکی دیگر از روشهای ویرایش رکوردها، استفاده از دکمه های نوار پیمایش رکوردها به شرح

جدول زیر است. این دکمه ها بطور خودکار در هنگام باز شدن جدول در پایین جدول ظاهر می

شوند.

کاربرد	دکمه
مکان نما را به اولین رکورد جدول منتقل می نماید.	
مکان نما را به رکورد قبلی منتقل می نماید.	
به شماره رکوردی که تایپ کنیم منتقل می شویم.	<input type="text" value="۱"/>
مکان نما به رکورد بعدی جدول منتقل می نماید.	
مکان نما را به آخرین رکورد منتقل می نماید.	
یک رکورد جدید به جدول اضافه می نماید.	

جدول (۴) کاربرد دکمه های نوار پیمایش

### اصول اصلاح یک رکورد

جهت هر گونه تغییر در داده های فیلدها و رکوردها، لازم است ابتدا آنها را انتخاب نماییم.

جهت انتخاب محتویات فیلدها از روشهای زیر استفاده می کنیم:

- در صورتی که با کلید Tab بین فیلدها حرکت کنیم، محتویات فیلد فعال انتخاب می شود.

- اشاره گر ماوس را به گوشه سمت چپ و بالای فیلد منتقل می کنیم تا شکل آن به صورت

علامت + تغییر یابد در اینصورت بر روی فیلد فوق کلیک می کنیم.

جهت انتخاب	نحوه انتخاب
کاراکترهای یک فیلد	بر روی کاراکتر شروع کلیک کرده و تا داده های بعدی آن را Drag می کنیم.
کل فیلد	اشاره گر ماوس را به گوشه سمت چپ و بالای فیلد حرکت داده تا شکل آن تغییر کند، سپس بر روی فیلد کلیک می کنیم.
فیلدهای مجاور	اولین فیلد را کلیک کرده و توسط عمل Drag فیلدهای بعدی را انتخاب می کنیم.
ستون	بر روی عنوان ستون کلیک می کنیم.
ستون های مجاور	اولین ستون را انتخاب کرده و بدون رها نمودن کلید ماوس، اشاره گر ماوس را بر روی ستون های مجاور Drag می کنیم.
رکورد	بر روی عنوان ردیف کلیک می کنیم.

رکوردهای مجاور	اولین رکورد را انتخاب کرده و ماوس را بر روی رکوردهای مجاور Drag می کنیم.
تمام رکوردها	کلیدهای Ctrl+A را همزمان بکار می بریم.

جدول (۵) روشهای انتخاب فیلدها و رکوردهای جدول

**جهت تغییر محتویات فیلد** از روشهای زیر استفاده می کنیم:

- جهت تغییر کل محتویات فیلد، فیلد را انتخاب کرده و مقدار جدید را وارد می کنیم.
- جهت ویرایش بخشی از محتویات فیلد، بر روی قسمت مورد نظر از فیلد کلیک کرده و تغییرات مناسب را انجام می دهیم.

### تغییر نام فیلد در نمای Datasheet

در هنگام نمایش داده های جدول در نمای Datasheet می توانیم نام فیلد را نیز تغییر دهیم.

جهت تغییر نام فیلد یک جدول در نمای Datasheet مراحل زیر را دنبال می کنیم:

- بر روی عنوان فیلد دلخواه جدول دوبار کلیک می کنیم تا نام فیلد انتخاب (رنگی) شود.
- نام جدید را وارد کرده و کلید Enter را فشار می دهیم.

### اصول ذخیره کردن بانک اطلاعاتی

نرم افزار Access هنگام ایجاد بانک اطلاعاتی ، نام آن را سؤال نموده و سپس در هنگام ایجاد جدول، فرم و سایر اشیاء نامی جهت هر یک درخواست نموده و تغییرات را در شیء ذخیره می نماید. با ذخیره تغییرات در شیء این تغییرات بطور خودکار در فایل بانک اطلاعاتی نیز اعمال خواهد شد.

### ذخیره جدول بانک اطلاعاتی

هنگامی که ساختار بانک اطلاعاتی را تغییر می دهیم (مثلاً نام یک فیلد را در حالت Design

View تغییر می دهیم)، باید تغییرات را در جدول ذخیره نماییم.

جهت ذخیره جدول بانک اطلاعاتی یکی از روش های زیر را بکار می بریم:

- کلیدهای ترکیبی Ctrl+S را بطور همزمان فشار می دهیم.



- روی دکمه به شکل دیسکت از نوار ابزار Table Design کلیک می کنیم.

- فرمان Save را از منوی File اجرا می کنیم.

جهت ثبت تغییرات جدول فعلی در جدول جدید بصورت زیر عمل می کنیم.

- از منوی View گزینه Design View را انتخاب می کنیم تا وارد نمای طراحی شویم.

- دستور Save as را از منوی File اجرا می کنیم.

- پنجره Save as ظاهر می شود ، نام جدول جدید را در کادر اولی وارد می کنیم.

- از کادر As گزینه Table را انتخاب می کنیم.

- دکمه OK را کلیک می کنیم.

اکنون جدول جدید در انتهای لیست جداول بانک اطلاعاتی فعلی اضافه شده است.

### اصول بستن بانک اطلاعاتی

نرم افزار Access این امکان را می دهد که بدون خرج از نرم افزار بانک اطلاعاتی جاری را

بسته و بانک اطلاعاتی دیگری را باز نماییم.

جهت بستن بانک اطلاعاتی جاری یکی از روشهای زیر را بکار می بریم:

- دستور Close را از منوی File اجرا می کنیم.

- کلیدهای ترکیبی Ctrl+F4 را بطور همزمان فشار می دهیم.

- دکمه X پنجره Database را کلیک می کنیم.

با انجام هر یک از روشهای فوق، بانک اطلاعاتی جاری بسته می شود ولی پنجره برنامه

Access فعال است و به صورت خالی با زمینه تیره ظاهر می گردد.

### - ایجاد بانک اطلاعاتی

جهت ایجاد بانک اطلاعاتی جدید مراحل زیر را انجام می دهیم:

در صفحه اصلی برنامه Access از منوی File گزینه New را انتخاب می کنیم در این صورت

قالب وظیفه New File ظاهر خواهد شد.

بر روی گزینه Blank Database از قاب وظیفه New File کلیک می کنیم . در این

صورت پنجره محاوره ای File New Database ظاهر خواهد شد.

در قسمت File Name نام فایل بانک اطلاعاتی را تایپ نموده و بر روی دکمه Create کلیک می کنیم.

در این صورت بانک اطلاعاتی جدیدی با نام داده شده در پوشه My document ایجاد خواهد شد.

### - ایجاد بانک اطلاعاتی جدید از فایل های بانک اطلاعاتی موجود

اگر بخواهیم یک بانک اطلاعاتی مشابه فایل بانک اطلاعاتی موجود ایجاد نماییم بصورتی که فایل بانک اطلاعاتی جدید همان جداول، فرمها، گزارشها و.. را داشته باشد، مراحل زیر را انجام می دهیم:

- در قاب وظیفه New File از قسمت New From Existing File گزینه Choose file را انتخاب می کنیم.

در این صورت پنجره محاوره ای New From Existing File ظاهر خواهد شد.

- فایل بانک اطلاعاتی موجود را انتخاب کرده و بر روی دکمه Create New کلیک می کنیم.

در این صورت بانک اطلاعاتی با نام جدید ایجاد خواهد شد. بانک اطلاعاتی جدید دقیقاً مشابه بانک اطلاعاتی انتخاب شده خواهد بود.

## اصول طراحی یک بانک اطلاعاتی

(۱) جدول و یا جداول مورد نیاز جهت طراحی بانک اطلاعاتی کدامند و نام هر یک چیست؟

(۲) فیلدهای مورد نیاز شما در هر جدول کدام است؟

(۳) فیلد کلیدی هر جدول چیست؟

(۴) مرتب سازی اطلاعات جداول بر اساس کدام فیلدها مورد نیاز است؟

(۵) گزارش های مورد نیاز جهت برنامه بانک اطلاعاتی چیست؟

یک برنامه بانک اطلاعاتی جهت مدیریت اطلاعاتی شخصی و تجاری کاربرد فراوانی دارد و این کاربرد در صورتی است که طراحی و ساخت آن بطور صحیح انجام شده باشد. یک بانک اطلاعاتی

با طراحی ضعیف ارزش اجرایی خود را از دست خواهد داد و چه بسا کاربران آن را دچار مشکلاتی بیشتر از زمانی که کار به صورت دستی انجام می گردید، خواهد نمود.

## – تحلیل سیستم موجود

نخستین موردی که باید در طراحی یک سیستم بانک اطلاعاتی مورد توجه قرار گیرد، تعریف روش های استفاده از اطلاعات و تشریح خروجی های سیستم است . در این مرحله شما باید گزارشات مورد نیاز سیستم را تعیین نموده و چگونگی فیلتر نمودن، گروه بندی و خلاصه سازی اطلاعات را مشخص نمایید. واسط کاربر (User Interface) معمولاً از یک فرم ورود اطلاعات تشکیل شده است. این فرم باید به دقت و با توجه به نیازها و مهارتهای کاربران طراحی گردد. کاربران مبتدی به راهنمایی های بیشتری نیاز دارند که باید در کادرهای توضیحی، پیغام های راهنما در اختیار شان قرار داده شود.

معمولاً هر سازمان یا شرکتی یک سیستم نیمه خودکار (بصورت دستی و یا مکانیزه) دارد. وظیفه طراح این است که سیستم موجود را بهبود و ارتقاء دهد. کاربران معمولاً در برابر تغییر ناگهانی و عمده روشهای سنتی مقاومت نشان می دهند. شما به عنوان طراح یک سیستم جدید به روشهای موجود توجه نموده تا سیستم به صورت یکنواخت تکمیل و از طرف کاربران مورد پذیرش قرار گیرد.

بطور خلاصه فرآیند طراحی بانک اطلاعاتی را می توان به مراحل زیر تقسیم نمود که هر یک اهداف خاصی را دنبال می کنند:

- تعیین نیازهای کاربران سیستم
- نحوه توزیع داده ها در جداول
- تعیین فیلدهای هر جدول
- تعیین فیلد کلیدی هر جدول
- تعیین ارتباط بین جداول و مدل سازی اولیه آنها
- ایجاد جداول و ورود داده ها
- اجرای عملیات ایجاد بانک اطلاعاتی

## - اصول ایجاد یک جدول

جهت ایجاد یک جدول در بانک اطلاعاتی مراحل زیر را انجام می دهیم:

- ۱- در پنجره Database بر روی دکمه Tables کلیک می کنیم.
- ۲- بر روی گزینه Create table in Design View دوبار کلیک می کنیم. در این صورت پنجره ایجاد ساختار جدول ظاهر خواهد شد.  
پنجره Design View از دو قسمت زیر تشکیل شده است:
  - محدوده تعیین نام، نوع و شرح فیلد
  - محدوده تعیین خواص مربوط به هر فیلد (Field Properties)اولین ردیف مربوط به وارد نمودن فیلد را کلیک نموده و نام اولین فیلد را تایپ می کنیم.  
ضوابط نام گذاری فیلدها عبارتند از:
  - اسامی فیلدها حداکثر ۶۴ کاراکتر می باشند که شامل حروف، اعداد و فاصله می باشد.
  - استفاده از کاراکترهای نقطه (.) علامت تعجب (!) علامت کروشه ( [ ] ) در نام فیلد مجاز نمی باشد و نام فیلد نباید با فاصله خالی شروع شود.
  - نرم افزار Access بین حروف کوچک و بزرگ در نام فیلد تفاوتی قائل نخواهد شد.
  - در صورتی که نام غیر مجازی جهت فیلد در نظر گرفته شود، پیغام خطای The Field name is not valid ظاهر خواهد شد.
- ۳- از لیست منویی Data Type نوع داده مناسب را انتخاب می کنیم.
- ۴- توضیحات و شرح هر فیلد را می توانیم در قسمت Description وارد کنیم. این توضیح جنبه نمایشی دارد و می تواند هر عبارتی باشد این عبارت هنگامی که در نمای Datasheet قرار داریم در نوار وضعیت نمایش داده می شود.
- ۵- در قسمت خواص فیلد (Field Properties) از پنجره محاوره ای تعیین ساختار جدول، خواص مربوط به هر فیلد مانند مقدار پیش فرض (Default Value) قانون معتبرسازی فیلد (Validation Rule) و.. را تعیین می کنیم.
- ۶- سایر فیلدها را به همین روش وارد می کنیم.
- ۷- گزینه Save را از منوی File انتخاب نموده و نامی را جهت جدول طراحی شده تایپ می کنیم.

### - تعیین اندازه فیلد (Field Size)

به دو دلیل زیر باید فیلد انتخابی به اندازه مناسب در نظر گرفته شود:

۱- صرفه جویی در فضای حافظه

۲- جلوگیری از خطای احتمالی ورود داده های نامعتبر

- در صورت انتخاب فیلد از نوع متنی (Text) ، نرم افزار Access تعداد ۵۰ کاراکتر را بصورت پیش فرض برای اندازه فیلد (Field size) در نظر خواهد گرفت. جهت تغییر این عدد، بر روی کادر Field size کلیک کرده و عدد جدید را تایپ می کنیم.
- در صورت انتخاب فیلد از نوع عددی (Number) بطور پیش فرض LongInteger برای اندازه فیلد (Field Size) در نظر گرفته می شو. جهت تغییر آن بر روی فلش رو به پایین در قسمت Field Size کلیک نموده و از منوی باز شده یکی از انواع داده ای Number را انتخاب می کنیم.

## نمای Design View

این نما به صورت افقی به دو پنجره کوچک تقسیم شده است. پنجره کوچک بالایی محدوده وارد نمودن نام فیلد، نوع داده و توضیحات اختیاری می باشد. همچنین تعیین فیلدی به عنوان کلید اصلی جدول در همین قسمت صورت می گیرد.

پنجره کوچک پایینی، مخصوص تعیین خصوصیات هر فیلد مانند اندازه فیلد (Field Size) عنوان فیلد (Caption) قانون معتبرسازی (Validation Rule) و تعداد دیگری از خصوصیات فیلد میباشد. در سمت راست این پنجره توضیحاتی از خصوصیت انتخاب شده فیلد نمایش داده میشود. جدول زیر خصوصیات رایج فیلدها را که در انواع داده ها وجود دارد، توضیح می دهد.

توضیحات	خصوصیات
حداکثر تعداد کاراکترهای فیلد را تعیین می نماید. پیش فرض آن ۵۰ کاراکتر است.	Field Size
نحوه نمایش ظاهری فیلد بکارگیری حروف کوچک و بزرگ را تعیین می نماید این فیلد پیش فرض ندارد.	Format
تعیین الگوی خاصی جهت ورود داده ها مانند الگوی شماره ملی - شماره بیمه تأمین اجتماعی و ... را ارائه می دهد.	Input Mask
نامی به غیر از نام فیلد که در نمای Datasheet فرم و گزارش ظاهر میشود.	Caption
بطور خودکار مقداری را برای فیلد در نظر می گیرد.	Default Value
تعیین عبارتی بعنوان قانون جهت ورود اطلاعات تا داده های نامعتبر را تشخیص دهد.	Validation Rule
در صورتی که داده وارد شده به هنگام ورود اطلاعات، قانون معتبرسازی را نقض کند، این متن نمایش داده می شود.	Validation Text
نشان دهنده آن است که این فیلد نمی تواند خالی باشد.	Required
تعیین کننده آن است که آیا رشته ای به طول صفر می تواند در فیلد قرار گیرد.	Allow zero length
جهت ایجاد ایندکس بر روی فیلد استفاده می شود تا سرعت جستجو بر روی فیلد بالا	Indexed
اجازه می دهد که داده های رشته ای که با فرمت Unicode ذخیره شده اند، فشرده شوند تا از فضای حافظه به صورت بهینه استفاده گردد. پیش فرض آن Yes است.	Unicode Compression
در فیلدهای عددی، تعداد ارقام بعد از اعشار را تعیین می کند.	Decimal Places
در فیلد عددی از نوع Decimal تعیین کننده تعداد کل ارقامی که در سمت چپ و راست نقطه اعشار می تواند ذخیره نمود. (از ۱ تا ۲۸ رقم)	Precision
در فیلد عددی از نوع Decimal تعداد ارقامی که بعد از نقطه اعشار میتواند ذخیره شود.	Scale

## تنظیم خصوصیات فیلد

خصوصیات فیلد، نحوه ذخیره سازی و نمایش فیلدها را مشخص می کند. هر نوع فیلد یک سری خصوصیات خاص خود را دارا است. به عنوان مثال خصوصیت Field Size فقط مخصوص داده هایی از نوع Auto Number, Number, Text است و سایر انواع داده این خصوصیت را دارا نمی باشند. Caption یکی دیگر از خصوصیات فیلد است که این خصوصیات در ستون عنوان در نمای Datasheet و به عنوان برچسب فیلد در فرم و گزارش ظاهر خواهد شد.

### ایجاد جدول با استفاده از ویزارد (Table Wizard)

برنامه ویزارد Table، گستره وسیعی از فیلدها را از طریق جداول نمونه پیشنهاد می کند.

جهت ایجاد یک جدول با استفاده از Table wizard مراحل زیر را انجام می دهیم:

۸- در پنجره Database بر روی دکمه Tables کلیک می کنیم.

۹- بر روی گزینه Create table by using wizard دوبار کلیک می کنیم. در این صورت

پنجره محاوره ای Table wizard ظاهر خواهد شد.

با انتخاب هر یک از دو گزینه Business (تجاری) Personal (پرسنلی) لیست

متفاوتی از جداول نمونه در کادر Sample Tables ظاهر خواهد شد.

جهت انتقال یک فیلد از لیست Sample Tables به لیست Fields in my new

table فیلد مورد نظر را از لیست انتخاب کرده و سپس دکمه > را کلیک می کنیم.

در صورتی که فیلدی را اشتباها به Fields In my new table منتقل کنیم،

جهت بازگرداندن آن پس از انتخاب فیلد بر روی دکمه < کلیک می کنیم.

جهت انتقال تمام فیلدها به قسمت Fields In my new Table و بالعکس به

ترتیب بر روی دکمه های <> و << کلیک می کنیم.

برای تغییر نام یک فیلد ابتدا بر روی فیلد در لیست Fields In my new table

کلیک کرده و سپس دکمه Rename Field... را کلیک می کنیم. در این صورت

پنجره محاوره ای Rename Field ظاهر خواهد شد. نام مورد نظر را در کادر

Rename تایپ نموده و دکمه OK را کلیک می کنیم.

- بر روی دکمه Next کلیک کرده نام دلخواهی را برای جدول تایپ می کنیم.

- بر روی دکمه Next کلیک می کنیم، در این صورت آخرین پنجره ایجاد جدول ظاهر خواهد شد.

- با انتخاب Modify the table design و فشردن دکمه Finish جدول ایجاد شده در نمای Design جهت تغییر و اصلاح ظاهر خواهد شد.

- با انتخاب Enter data directly into the table و فشردن دکمه Finish جدول ایجاد شده در نمای Datasheet ظاهر شده و تمامی فیلدها در یک سطر نمایش داده خواهد شد.

- با انتخاب Enter data into the table using a form و فشردن دکمه Finish جدول ایجاد شده به صورت فرم ظاهر شده و هر فیلد به همراه عنوان آن در یک سطر نمایش داده خواهد شد.

## ایجاد جدول با روش ورود اطلاعات

روش دیگر جهت ایجاد جدول، استفاده از روش مستقیم ورود اطلاعات در جداول است. در این روش ۱۰ فیلد با نامهای (Field1, Field2, Field3, ...) بطور پیش فرض توسط Access در نظر گرفته خواهد شد و در صورت نیاز خواهید توانست فیلدهای دیگری را نیز به آن اضافه نمایید.

جهت ایجاد جدول با استفاده از روش ورود اطلاعات مراحل زیر را انجام می دهیم.

- در پنجره Database بر روی دکمه Tables کلیک می کنیم.

- بر روی گزینه Create table by entering data دوبار کلیک می کنیم.

پنجره محاوره ای Table ظاهر خواهد شد.

- جهت تغییر عنوان فیلدها بر روی عنوان فیلد دوباره کلیک نموده و نام جدید را

تایپ می کنیم.



- جهت اضافه نمودن فیلد جدید بین دو فیلد دیگر، از منوی Insert گزینه Column را انتخاب می کنیم، در این صورت برنامه Access بطور خودکار نام Field11 و Field12 و... را به فیلدهای جدید خواهد داد.

- اطلاعات جداول را در دریف های بعدی و در کادرهای سفید وارد می کنیم. برای جابجایی بین فیلدها از کلید Tab استفاده می کنیم.

- اگر در فیلدی مقدار عددی وارد کنیم. آن فیلد با نوع Number و با اندازه Long Integer در نظر گرفته می شود و اگر در فیلدی مقدار متنی وارد کنیم، آن فیلد با نوع Text. و با اندازه ۵۰ کاراکتر در نظر گرفته می شود. باید توجه داشته باشید که فقط فیلدهایی که در آن اطلاعات وارد کرده اید به عنوان فیلد ذخیره می شوند و فیلدهایی که خالی هستند در ساختار جدول ذخیره نمی شوند.


- گزینه Save را انتخاب کرده و نامی را برای جدول وارد می کنیم تا جدول ذخیره گردد. جدول ایجاد شده به لیست جداول در پنجره Database اضافه خواهد شد.

### هدایت و جهت دادن به کار جدول

حال که همه تغییرات لازم بر روی ساختار جدول اعمال شد، باید داده های آن را وارد کنیم. یک جدول بدون داده کاربردی ندارد. با درج اطلاعات به جدول است که می توانیم گزارشهای متنوع، پرس و جو و فیلترسازی موردنظر را انجام دهیم. در نمای Datasheet می توانیم اطلاعات را بطور مستقیم در جداول وارد کنیم.

### ورود اطلاعات در جدول

جهت وارد نمودن اطلاعات به جدول مراحل زیر را انجام می دهیم:  
در پنجره محاوره ای Database بر روی نام جدول دلخواه دوبار کلیک می کنیم، در این صورت جدول ورود اطلاعات در نمای Database باز خواهد شد.

بر روی دکمه  واقع در نوار ابزار Table Datasheet کلیک می کنیم، در این صورت رکورد جدیدی اضافه شده و مکان نما در اولین فیلد رکورد خالی قرار خواهد گرفت. حال می توانیم اطلاعات رکورد جدید را وارد کنیم. جهت انتقال به فیلد بعدی در رکورد از کلید Tab و یا کلیدهای جهتی استفاده می کنیم.

### تعیین کلید اولیه

همانطور که می دانیم فیلد کلیدی، فیلدی است که در هیچ دو رکورد از جدول یکسان نباشد. نرم افزار Access، اجازه درج مقادیر تکراری و یا تهی (NULL) را در فیلد کلیدی نخواهد داد. در صورتیکه در جدولی یک یا چند فیلد کلیدی داشته باشیم ، می توانیم یکی از این فیلدها را به عنوان کلید اولیه تعریف کنیم.

جهت تعریف یک فیلد بعنوان کلید اولیه عملیات زیر را انجام می دهیم:  
جدولی که می خواهیم فیلد کلیدی را در آن تعریف کنیم، انتخاب می کنیم.  
دکمه Design را کلیک می کنیم.

فیلدی که می خواهیم به عنوان فیلد کلیدی تعریف گردد، انتخاب می کنیم.  
سپس یکی از روش های زیر را انجام می دهیم:

- کلیک بر روی ابزار کلید از نوار ابزار Table Design
  - انتخاب گزینه Primary Key از منوی Edit
  - راست کلیک بر روی فیلد و انتخاب گزینه Primary Key از منوی میانبر ظاهر شده
- فیلدی که به عنوان فیلد کلیدی تعریف گردد، علامت کلید در جلوی آن ظاهر خواهد شد.

- جهت حذف شاخص فیلد کلیدی، مراحل ذکر شده در بالا را تکرار می کنیم.
- در پایان Save را انتخاب می کنیم تا تغییرات داده شده ، ذخیره شود.

### اصلاح مشخصات جدول

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
بعد از آنکه فیلدهای یک جدول را ایجاد کردیم ، ممکن است بخواهیم نام فیلد یا نوع  
داده ای فیلد را تغییر داده و فیلدهایی را اضافه یا کم کنیم.

#### جهت تغییر خصوصیات یک فیلد مراحل زیر را انجام می دهیم:

- در پنجره Database دکمه Tables را کلیک می کنیم.
- جدولی را که می خواهیم بر روی آن تغییراتی بدهیم، انتخاب می کنیم.
- بر روی دکمه Design کلیک می کنیم. در این صورت جدول انتخاب شده در  
نمای Table Design نمایش داده خواهد شد.
- بر روی نام فیلدی که می خواهیم تغییر دهیم، کلیک می کنیم.
- جهت تغییر نوع داده فیلد، بر روی Data type کلیک کرده و از کادر باز شده،  
نوع داده مورد نظر را انتخاب می کنیم.
- در قسمت Field Properties خصوصیات فیلد مورد نظر را تعیین می کنیم.

#### افزودن فیلد به جدول

ممکن است بعد از طراحی جدول متوجه شویم که فیلد جدیدی را باید به جدول اضافه کنیم.  
اگر مرحله تحلیل و طراحی سیستم را به خوبی انجام داده باشیم کمتر اتفاق می افتد که نیازی  
باشد فیلد جدیدی را به جدول اضافه کنیم. اگر اضافه کردن فیلد به جدول قبل از ورود  
اطلاعات به جدول باشد، مشکل زیادی ایجاد نمی کند ولی اگر بعد از ورود اطلاعات به جدول،  
فیلدی را به جدول اضافه کنیم، مقدار این فیلد برای رکوردهای موجود، تهی (Null) خواهد  
بود و باید برای این فیلد در همه رکوردهای جدول ورود اطلاعات انجام شود.

#### جهت افزودن فیلد جدید به جداول مراحل زیر را انجام می دهیم:

- در پنجره Database دکمه Tables را کلیک می کنیم .
- جدولی را که می خواهیم بر روی آن کار کنیم، انتخاب می کنیم.
- بر روی دکمه Design کلیک می کنیم. در این صورت جدول انتخاب شده در نمای Table  
Design نمایش داده خواهد شد.
- فیلدی که قبل از محل درج فیلد جدید قرار می گیرد را انتخاب میکنیم.

گزینه Row را از منوی Insert انتخاب می کنیم.

- با انتخاب Rows و کلیک بر روی آن، یک سطر خالی جهت درج فیلد جدید اضافه خواهد شد. همچنین فیلدی که در حالت انتخاب شده قرار داشت، یک سطر به پایین تر خواهد رفت.

- نام فیلد نوع داده فیلد و در صورت تمایل شرحی برای فیلد را به ترتیب در قسمت های Field Name ، Data Type و Description وارد می کنیم.

### حذف فیلد از جدول

به سادگی می توانیم یک یا چند فیلد ایجاد شده در یک جدول را که مورد نیاز نمی باشند، حذف نماییم. اگر یک فیلد را از جدول حذف کنیم، آن فیلد به همراه اطلاعات وارد شده در آن، از جدول حذف خواهد شد.

برای حذف یک فیلد از جدول عملیات زیر را انجام می دهیم:

- در پنجره Database دکمه Tables را کلیک می کنیم.

- جدول مورد نظر را انتخاب کرده و بر روی دکمه Design کلیک می کنیم. در این صورت جدول انتخاب شده در نمای Table Design نمایش داده خواهد شد.

- بر روی قسمت سمت راست فیلد مورد نظر کلیک می کنیم تا فیلد به صورت سطری انتخاب شود.

کلید Delete را فشار می دهیم.

**نکته:** در نمای Datasheet جهت حذف فیلد، بر روی عنوان فیلد کلیک کرده و سپس گزینه Delete Column را از منوی Edit انتخاب می کنیم.

پس از انتخاب فیلدی جهت حذف و فشردن دکمه Delete و یا انتخاب گزینه Delete Column نرم افزار Access در مورد حذف دائمی فیلد و محتوای آن سوال می نماید.

- پس از انتخاب فیلدی جهت حذف و فشردن دکمه Delete و یا انتخاب Delete Cloumn ، نرم افزار Access در مورد حذف دائمی فیلد و محتوای آن سؤال می نماید.

### نحوه انتخاب رکوردها

جهت انتخاب یک یا چند رکورد از رکوردهای جدول مراحل زیر انجام می دهیم:

- در پنجره Database دکمه Tables را کلیک می کنیم.

- بر روی نام جدول مورد نظر دوبار کلیک می کنیم تا جدول در نمای Datasheet نمایش داده شود.

- برای انتخاب یک رکورد، بر روی مربع خاکستری رنگ کنار آن رکورد کلیک می کنیم . در این صورت نرم افزار Access رکورد را در حالت انتخاب شده قرار می دهد.

- برای انتخاب دو یا چند رکورد مجاور هم، بر روی مربع خاکستری رنگ کنار اولین رکورد کلیک کرده و ماوس را تا آخرین رکورد مورد نظر می کشیم. در این صورت نرم افزار Access رکوردهای مورد نظر را در حالت انتخاب شده قرار می دهد.

نکته : نمی توانیم رکوردهای غیرمجاور یک جدول را در یک مرحله انتخاب نماییم.

### آشنایی با نحوه حذف ، کپی و انتقال رکوردها در جدول

جهت انجام عملیات ویرایشی بر روی رکوردهای جدول مراحل زیر را انجام می دهیم:

- جدول مورد نظر را در نمای Datasheet باز می کنیم.
- رکورد یا رکوردهای مورد نظر را جهت ویرایش انتخاب می کنیم.
- جهت حذف رکورد یا رکوردهای انتخاب شده، کلید Delete را فشار می دهیم.
- جهت کپی کردن رکوردهای انتخاب شده در حافظه موقت، کلیدهای Ctrl+C را فشار می دهیم.

- جهت انتقال رکوردهای مورد نظر به محل دیگر، کلیدهای Ctrl+V را فشار می دهیم.

### نحوه مرتب کردن داده های جدول

اگرچه نرم افزار Access داده ها را به صورت خودکار بر اساس فیلد کلید اولیه مرتب می نماید ولی ما می توانیم فیلدها را به هر ترتیبی که می خواهیم مرتب کرده و مشاهده کنیم.

به کمک دستور Sort می توان رکوردها را براساس فیلدمورد نظر مرتب نمود. عمل مرتب نمودن می تواند به صورت صعودی (از A تا Z یا از ۱ تا ۱۰) و یا نزولی (از Z تا A و یا از ۱۰ تا ۱) باشد.

جهت مرتب نمودن داده ها در یک جدول مراحل زیر را انجام می دهیم:

- جدول مورد نظر را در نمای Datasheet باز می کنیم.
- بر روی عنوان فیلدی که می خواهیم جدول را بر اساس آن مرتب نماییم، کلیک می کنیم.

- بعد از انتخاب گزینه Sort از منوی Records ، دستور Sort Ascending (مرتب سازی

صعودی) یا Sort Descending (مرتب سازی نزولی) را انتخاب می کنیم.

- اگر بخواهیم جدول را به حالت قبل از مرتب شدن بازگردانیم، گزینه Remove Filter/ Sort

را از منوی Records انتخاب می کنیم.

- اگر بعد از عمل Remove Filter/Sort مجدداً بخواهیم جدول را به حالت مرتب سازی قبلی

برگردانیم، بر روی گزینه Apply Filter/Sort از منوی Records کلیک می کنیم.

### آشنایی با نحوه تغییر ترتیب قرارگیری فیلدهای یک جدول

بعد از آنکه فیلدهای یک جدول را ایجاد کرده و داده هایی را به آن اضافه نمودیم ، ممکن است که

بخواهیم ترتیب قرارگیری فیلدها را در جدول جابجا نماییم.

جهت تغییر ترتیب قرارگیری فیلدهای یک جدول مراحل زیر را انجام می دهیم:

- جدول مورد نظر را در نمای Datasheet باز می کنیم.

- بر روی عنوان فیلد مورد نظر کلیک می کنیم. در این صورت فیلد به صورت ستونی

انتخاب خواهد شد.

- مجدداً بر روی عنوان فیلد کلیک می کنیم ولی این بار دکمه ماوس را رها نمی کنیم. در

این صورت یک کادر مستطیل شکل کوچک در کنار اشاره گر ماوس ظاهر خواهد شد.

- اکنون فیلد انتخاب شده را به موقعیت جدید در سمت چپ و یا راست موقعیت اولیه درگ

می کنیم.

- پس از رسیدن اشاره گر ماوس به مقصد موردنظر ، یک خط ضخیم قائم ظاهر می شود.

این خط مشخص می کند که نرم افزار Access محل دلخواه ما را برای قرارگیری فیلد

تشخیص داده است در محل مورد نظر ، دکمه سمت چپ ماوس را رها می کنیم تا فیلد

مورد نظر در محل جدید ظاهر گردد.

### نحوه کپی ساختار یک جدول موجود

ممکن است بخواهیم جدول جدیدی ایجاد کنیم که فیلدهای این جدول مشابه یکی از جداول

موجود است. برای ایجاد سریعتر این جدول می توانیم ساختار جدول طراحی شده قبلی را کپی

کرده و در جدول جدید استفاده کنیم.

جهت کپی ساختار جدول ، مراحل زیر را انجام می دهیم:

- جدول مورد نظر را از پنجره Database انتخاب می کنیم.
- گزینه Copy را از منوی Edit انتخاب می کنیم .
- اگر بخواهیم جدول را در بانک اطلاعاتی دیگری کپی کنیم، بانک اطلاعاتی جاری را بسته و بانک اطلاعاتی جدید را باز می کنیم.
- گزینه Paste را از منوی Edit انتخاب می کنیم، در این صورت پنجره Paste Table As ظاهر خواهد شد.
- در قسمت Table Name نام جدول جدید را وارد می کنیم.
- گزینه Structure Only را جهت کپی ساختار جدول انتخاب کرده و بر روی دکمه OK کلیک می کنیم.

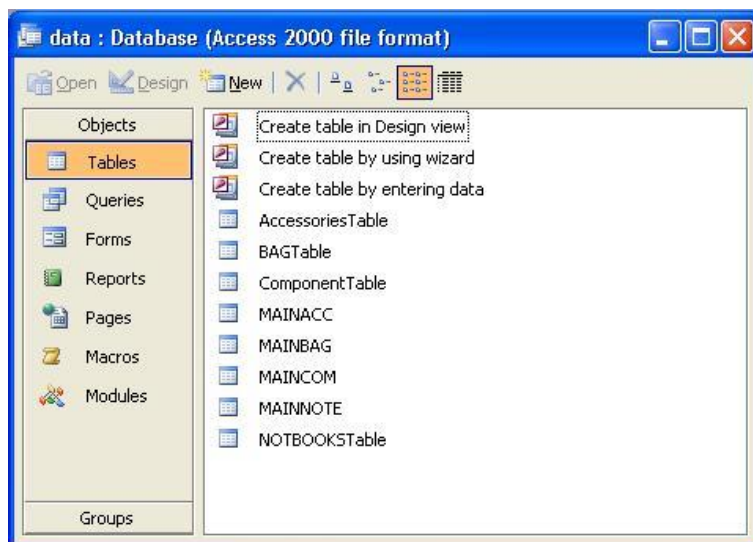
## توصیف پروژه:

### الف ( توصیف پایگاه داده

نام فایل اصلی پایگاه داده موجود در این پروژه `data . mdb` است که باید حتماً در همان آدرسی که فایل‌های دیگر مربوط به پروژه قرار دارند، موجود باشد. این بانک اطلاعاتی که در نرم افزار Access طراحی شده است، شامل جداول زیر می باشد:

Accessories table	-۱
Bagtable	-۲
Component table	-۳
Note book Table	-۴
Main Acc	-۵
Main Bag	-۶
Main Com	-۷
Main Note	-۸

چهار جدول اول، جداولی هستند که داده های اصلی در آنها قرار دارد و چهار جدول دوم برای تنظیمات مورد نیاز و ساده ترین Query گرفتن از data base طراحی شده اند. تصویر `pictures / data base / tables.JPG` اضافه شود.





### ۱- جدول AccessoriesTable

Title	Brand	Properties	Warranty	Price(Rials)
EXT HDD BOX	Abacus			۷۵۰۰۰۰۰۰
MOUSE	HP	NOTEBOOK MOUSE		۷۵۰۰۰۰۰۰
MOUSE	SONY	NOTEBOOK MOUSE		۷۵۰۰۰۰۰۰
MOUSE	TOSHIBA	NOTEBOOK MOUSE		۷۵۰۰۰۰۰۰
EXT HDD BOX	OTB			۲۲۰۰۰۰۰۰
EXT HDD BOX		USB۲		۱۴۰۰۰۰۰۰
EXT HDD BOX	VAIO			۱۲۰۰۰۰۰۰
EXT HDD BOX	VAIO B			۱۴۵۰۰۰۰۰
PCMCIA DEVICES		Convertor PCMCi to se		۲۹۰۰۰۰۰۰
PCMCIA DEVICES		Convertor PCMCi to US		۳۱۰۰۰۰۰۰
USB KIT		Convertor PCMCi to Pa		۱۲۵۰۰۰۰۰
USB KIT		Convertor PCMCi to se		۱۲۵۰۰۰۰۰
KEY BOARD	IBM Keyboard+			۱۶۵۰۰۰۰۰
KEY BOARD		Keyboard Notebook		۱۰۵۰۰۰۰۰
PCMCIA DEVICES		LAN		۳۰۰۰۰۰۰۰
PCMCIA DEVICES	Bilionton	Modem		۳۳۰۰۰۰۰۰
MOUSE	DELL	Mouse Notebook		۷۵۰۰۰۰۰۰
USB KIT		HUB USB		۹۰۰۰۰۰۰۰
MOUSE	Vaio	Mouse Notebook		۷۵۰۰۰۰۰۰
USB KIT		Sound Card USB		۱۶۰۰۰۰۰۰
USB KIT		Coble USB Kit		۹۰۰۰۰۰۰۰
WEB CAM		Videocam Messenger		۲۰۰۰۰۰۰۰
PCMCIA DEVICES	Light Wave	TV Tuner PCMCi		۶۰۰۰۰۰۰۰
USB KIT		TV Tuner USB		۵۰۰۰۰۰۰۰
PCMCIA DEVICES		Convertor PCMCi to IDI		۲۱۰۰۰۰۰۰

این جدول مربوط به لوازم جانبی کامپیوتر (نوت بوک) است، از جمله باکس هارد خارجی (Ext HDD BOX)، ماوس (Mouse)، USB KIT، web cam، صفحه کلید (key board) و PCMCIA Devices.

لیست فیلدها:

1- Title:

عنوان مربوط به لوازم جانبی، منظور مشخص کردن همان Mouse key board، USB kit یا ...

2- Brand:

مارک تجاری قطعه کامپیوتری مورد نظر مثل Generus یا vaio برای ماوس یا IBM برای کی بورد.

**: Properties (۳-۱)**

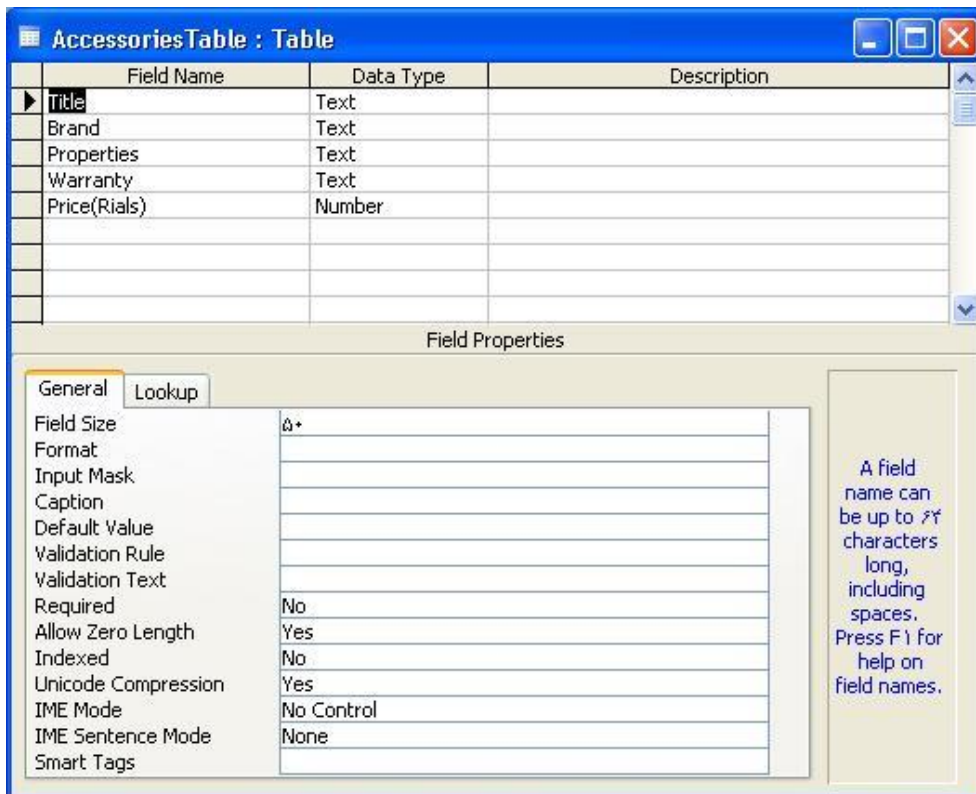
مشخصات اضافه تر

**: warranty (۴-۱)**

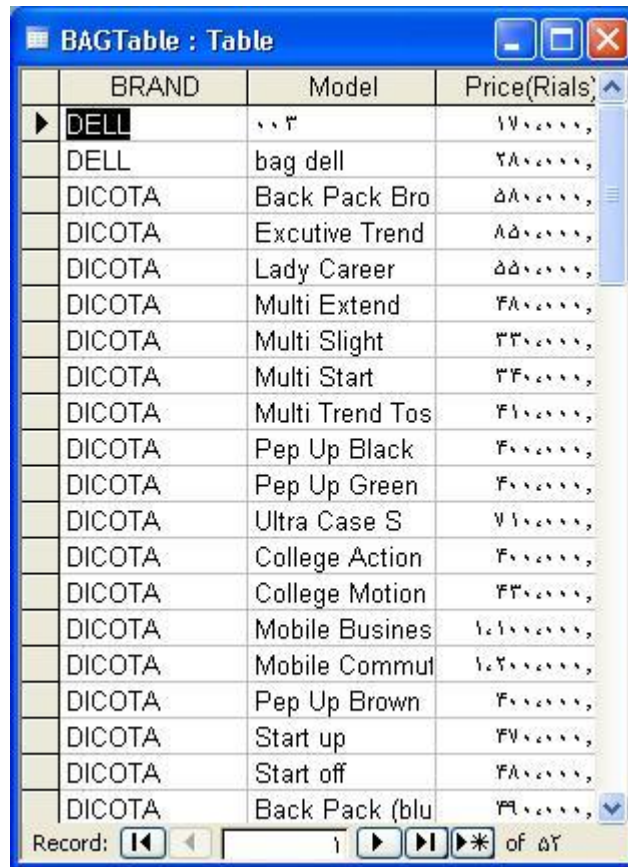
مشخص کننده طول زمان گارانتی کالای مورد نظر

**: price(Rials) (۵-۱)**

قیمت کالای مورد نظر به ریال



## ۲- جدول BagTable



BRAND	Model	Price(Rials)
DELL	۰۰۳	۱۷۰۰۰۰۰٫۰۰
DELL	bag dell	۲۸۰۰۰۰۰٫۰۰
DICOTA	Back Pack Bro	۵۸۰۰۰۰۰٫۰۰
DICOTA	Excutive Trend	۸۵۰۰۰۰۰٫۰۰
DICOTA	Lady Career	۵۵۰۰۰۰۰٫۰۰
DICOTA	Multi Extend	۴۸۰۰۰۰۰٫۰۰
DICOTA	Multi Slight	۳۳۰۰۰۰۰٫۰۰
DICOTA	Multi Start	۳۴۰۰۰۰۰٫۰۰
DICOTA	Multi Trend Tos	۴۱۰۰۰۰۰٫۰۰
DICOTA	Pep Up Black	۴۰۰۰۰۰۰٫۰۰
DICOTA	Pep Up Green	۴۰۰۰۰۰۰٫۰۰
DICOTA	Ultra Case S	۷۱۰۰۰۰۰٫۰۰
DICOTA	College Action	۴۰۰۰۰۰۰٫۰۰
DICOTA	College Motion	۴۳۰۰۰۰۰٫۰۰
DICOTA	Mobile Busines	۱۰۱۰۰۰۰۰۰٫۰۰
DICOTA	Mobile Commut	۱۰۲۰۰۰۰۰۰٫۰۰
DICOTA	Pep Up Brown	۴۰۰۰۰۰۰٫۰۰
DICOTA	Start up	۴۷۰۰۰۰۰٫۰۰
DICOTA	Start off	۴۸۰۰۰۰۰٫۰۰
DICOTA	Back Pack (blu	۳۹۰۰۰۰۰٫۰۰

این جدول مربوط به کیف نوت بوک است.

لیست فیلدها :

۱-۲ Brand :

مارک تجاری کیف مثل Dicota , Sumdex

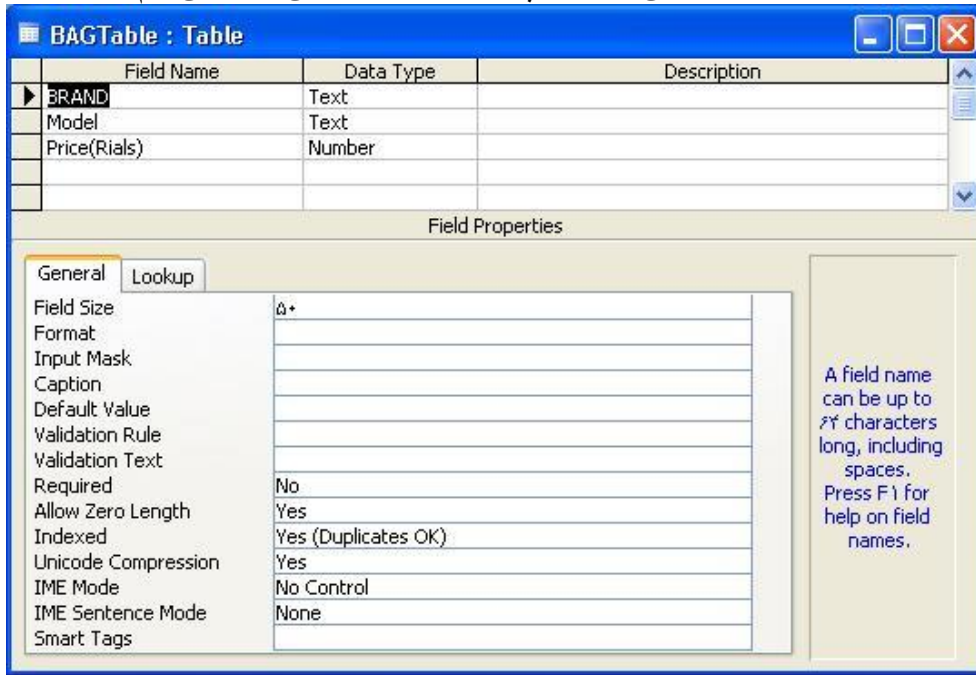
۲-۲ Model :

هر مارک کیف دارای مدل‌های مختلف است.

مثلاً Dicota Multi Start یا Sumdex 854

۳-۲ Price(Rials) :

قیمت به ریال



۳- جدول ComponentTable

Title	Brand	Properties	Warranty	Price(Ria)
Combo Drive	LG			۲۵۰,۰۰۰
DVD-Writer	Pioneer			۱,۰۱۵,۰۰۰
DVD-Writer	Suzuki	Notebook-Exter		۱,۰۷۰,۰۰۰
DVD-Writer	Philips	Notebook		۹۵۰,۰۰۰
FDD	IBM	۱,۴۴ MB USB		۱۷۵,۰۰۰
HDD	Hitachi	۱۰۰ GB	Almas	۱,۰۲۰,۰۰۰
HDD	Hitachi	۸۰ GB	Almas	۷۹۰,۰۰۰
HDD	TOSHIBA	۴۰ GB		۶۴۰,۰۰۰
HDD	Hitachi	۸۰GB		۷۹۰,۰۰۰
HDD	Seagate	۴۰GB		۴۲۰,۰۰۰
HDD	Samsung	۱۰۰GB		۹۹۰,۰۰۰
HDD	Toshiba	۱۲۰GB		۱,۰۲۰,۰۰۰
HDD	Toshiba	۸۰GB		۷۹۰,۰۰۰
RAM	TOSHIBA	SD ۱۲۸		۲۲۰,۰۰۰
RAM	Queen	DDR۲ ۲۵۶	Diagram	۲۵۰,۰۰۰
RAM	Kingstone	DDR ۲۵۶	GLC	۲۸۰,۰۰۰
RAM	Samsung	۲۵۶ DDR	---	۲۶۰,۰۰۰
RAM	Kingstone	۵۱۲ DDR	---	۳۹۰,۰۰۰
RAM	Vdata	۲۵۶ DDR	---	۲۶۰,۰۰۰
RAM	GIEL	۱G DDR۲		۱,۰۱۰,۰۰۰
RAM	Kingston	۱G DDR		۱,۰۱۵,۰۰۰
RAM	Infineon	۲۵۶ DDR۲		۲۷۰,۰۰۰

این جدول مربوط به اجزاء اصلی و داخلی کامپیوتر ( نوت بوک ) است. از جمله ( FDD ( Combo Drive , Ram , DVD Writer , HDD ( Hard) , Floppy Drive)

لیست فیلدها :

**Title (۱-۳)**

عنوان مربوط به قطعات اصلی، منظور مشخص کردن همان FDD یا HDD یا ...

**Brand (۲-۳)**

مارک تجاری قطعه مورد نظر مثل Queen برای RAM یا Hitachi یا Toshiba  
برای HDD

**Properties (۳-۳)**

مشخصات اضافه تر

**Warranty (۴-۳)**

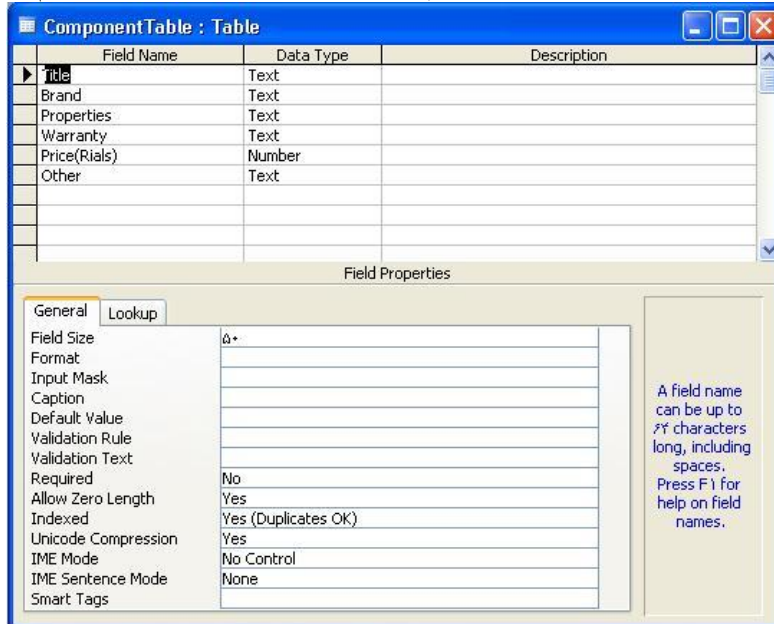
مشخص کننده طول زمان گارانتی کالای مورد نظر

**Price(Rials) (۵-۳)**

قیمت کالای مورد نظر به ریال

**Other (۶-۳)**

اطلاعات دیگری که ممکن است در مورد کالای مورد نظر مهم و مورد نیاز باشد.



#### ۴- جدول NotebookTable

	BRAND	MODEL	CPU	RAM	HDD	DRIVE
▶	ACER	AS ۳۶۲۴	Celeron M ۱,۶	۵۱۲	۶۰	DVD RW DL
	ACER	TM ۲۴۲۴	Celeron.M ۱,۶	۵۱۲	۴۰	DVD RW
	ACER	Aspire ۵۶۱۲	CoreDuo ۱,۶۶	۵۱۲	۸۰	DVD RW DL
	ACER	AS ۱۶۴۲	Centrino ۱,۷ ۲	۵۱۲	۶۰	DVD RW DL
	ACER	TM ۴۰۷۲	Centrino ۱,۷	۵۱۲	۶۰	Combo
	ACER	TM ۴۱۰۱	Centrino ۱,۶	۵۱۲	۶۰	DVD RW DL
	ACER	TM ۴۰۰۲	Centrino ۱,۶	۲۵۶	۴۰	Combo
	ACER	AS ۵۶۰۲ WLM1	CoreDuo ۱,۶۶	۵۱۲	۸۰	DVD RW DL
	ACER	AS ۵۵۶۲ WXMI	CoreDuo ۱,۶۶	۵۱۲	۸۰	DVD RW DL
	ACER	AS ۵۶۰۲ WLM1	CoreDuo ۱,۶۶	۱۰۲۴	۱۲۰	DVD RW DL
	ACER	۴۱۵۲	Centrino ۱,۷۳	۲۵۶	۶۰	Combo
	ACER	۹۴۱۱	CoreDuo ۱,۶۶	۱۰۲۴	۱۲۰	DVD RW
	ACER	TM ۴۰۷۰	Celeron.M ۱,۶	۲۵۶	۶۰	Combo
	DELL	D ۸۲۰	Core۲duo ۱,۸۳	۵۱۲	۸۰	DVD RW DL
	DELL	۶۴۰۰	Core۲duo ۲,۱۶	۱۰۲۴	۱۶۰	DVD RW DL
	DELL	۷۰۰M	Centrino ۲,۰	۱۰۲۴	۱۰۰	DVD RW DL
	DELL	۱۳۰۰	Celeron.M ۱,۶	۵۱۲	۶۰	DVD RW
	DELL	۶۴۰M	Core۲duo ۲,۰	۲G	۱۶۰	DVD RW
	DELL	۶۴۰	Core۲Duo ۱,۸۳	۵۱۲	۸۰	DVD RW
	DELL	۶۴۰۰	Core۲duo ۲,۱۶	۱۰۲۴	۱۲۰	DVD RW
	DELL	۶۴۰۰	CoreDuo ۱,۶۶	۵۱۲	۸۰	DVD RW
	DELL	۶۴۰۰	Core۲duo ۲,۰	۱۰۲۴	۱۶۰	DVD RW
	DELL	XPSM۱۲۱۰	۱,۶۶ CoreDuo	۵۱۲	۸۰	DVD RW
	DELL	D ۶۲۰	CoreDuo ۱,۸۳	۱۰۲۴	۸۰	DVD RW

Record: 1 of 202

این جدول که اصلی ترین جدول اطلاعاتی پایگاه داده های این پروژه است مربوط به نوت بوک هایی است که در این فروشگاه موجود است.

### لیست فیلدها

#### : Brand (۱-۴)

مارک تجاری نوت بوک مورد نظر مثل Acer ، Dell ، Gigabyte ، Sony ، LG ، Toshiba یا ...

#### : Model (۲-۴)

هر مارک تجاری شامل چندین مدل است.

مثلاً Acer TM2424 یا Acer 9411 یا Dell D820

#### : CPU (۳-۴)

مدل و فرکانس CPU دستگاه نوت بوک مورد نظر

#### : RAM (۴-۴)

حجم حافظه نوت بوک مورد نظر

#### : HDD (۵-۴)

حجم دیسک سخت نوت بوک مورد نظر

#### : Drive (۶-۴)

نوع درایوی که روی نوت بوک مورد نظر وجود دارد. مثلاً DVD RW قابلیت خواندن و نوشتن روی DVD را دارد.

یا Combo قابلیت خواندن و نوشتن روی CD و خواندن DVD را دارد.

یا DVD RW DL قابلیت خواندن و نوشتن دو لایه روی DVD را دارد.

یا Optional یعنی نوت بوک مورد نظر خودش درایوی ندارد و خریدار به صورت انتخابی می تواند درایو مورد نظرش را برای نصب درخواست نماید.

#### : LCD (۷-۴)

نوع و سایز ( به اینچ ) مانیتور

#### : VGA (۸-۴)

مارک و حجم حافظه کارت گرافیکی

#### : OS (۹-۴)

سیستم عامل

**: Weight (10-4)**

وزن نوت بوک

**: Madein (11-4)**

کشور سازنده

**: Price(Rials) (12-4)**

قیمت به ریال

**: Other (13-4)**

موارد دیگری که ممکن است در ارتباط با نوت بوک مورد نظر مهم باشد، مثل طول زمان گارانتی و لوازم اضافه ای که همراه نوت بوک است ، همچون TV Tuner یا مشخص کردن نوع گارانتی ( مثلا International ) یا ...

Field Name	Data Type	Description
RAM	Text	
HDD	Text	
DRIVE	Text	
LCD	Text	
VGA	Text	
OS	Text	
WEIGHT	Text	
MADE IN	Text	
PRICE(Rial)	Number	
OTHER	Text	

Field Properties

General Lookup

Field Size: 5+

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Allow Zero Length: Yes

Indexed: Yes (Duplicates OK)

Unicode Compression: Yes

IME Mode: No Control

IME Sentence Mode: None

Smart Tags:

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help.

**۴- جدول MainAcc**



	ID	TITLE
▶ +	1	EXT HDD BOX
+ +	2	KEY BOARD
+ +	3	MOUSE
+ +	4	PCMCIA DEVIC
+ +	5	USB KIT
+ +	6	WEB CAM
* *		

با جدول اصلی AccessoriesTable ارتباط برقرار می کند.

لیست فیلدها :

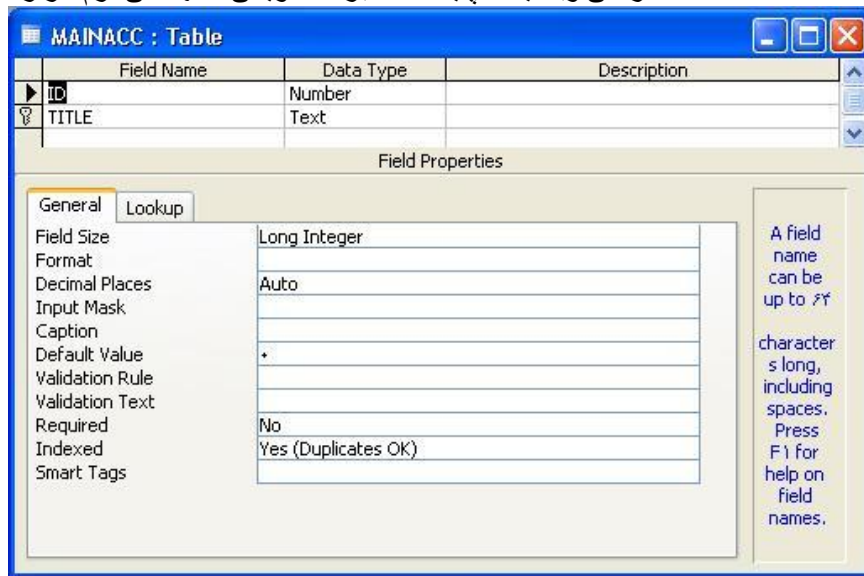
**ID (۱-۵) :**

شناسه برای فیلد Title است.

**Title (۲-۵) :**

عنوان، که همان داده های اصلی فیلد title در جدول AccessoriesTable است.

مثل Mouse ، key board یا USB kit .



فیلد کلید اصلی ( primary kay ) فیلد Title است.

## ۶- جدول MainBag



جدول اصلی که با جدول BagTable ارتباط برقرار می کند.

لیست فیلدها:

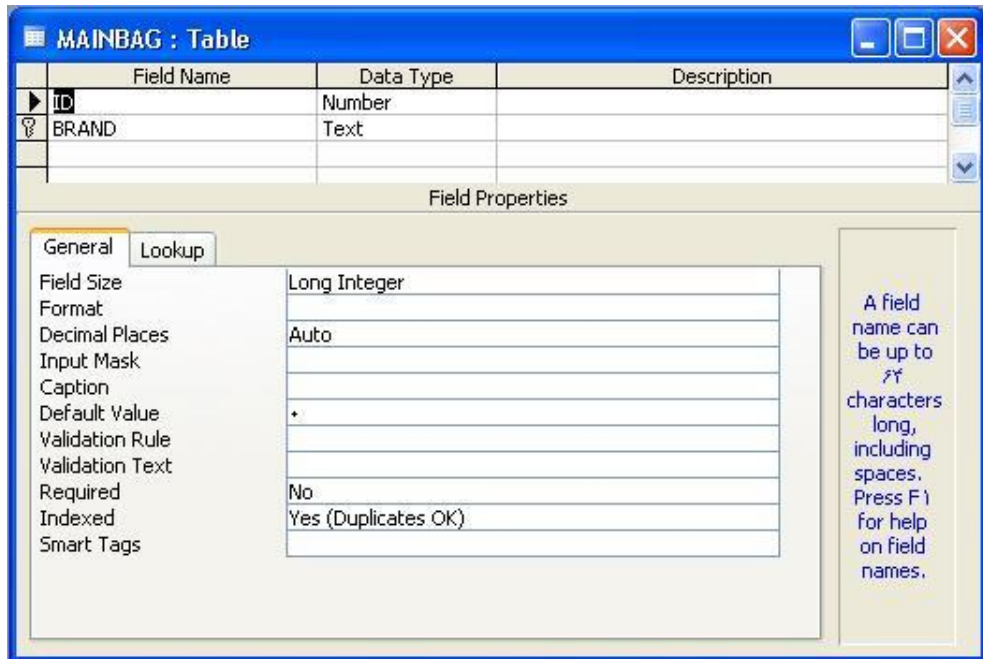
**ID (۱-۶)**

شناسه برای فیلد Brand .

**Brand (۲-۶)**

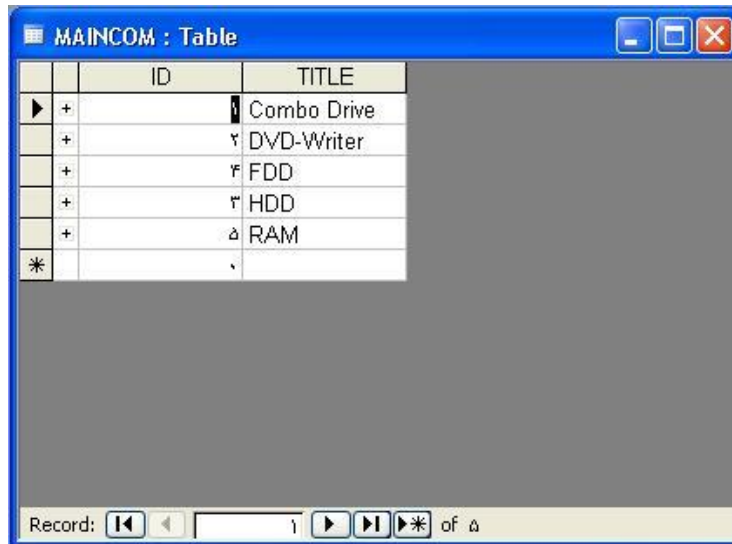
مارک (عنوان) که همان داده های اصلی فیلد Brand در جدول BagTable است.

مثل SONY یا Sumdex .



فیلد Primary key فیلد Brand است.

## ۷- جدول MainCom



	ID	TITLE
▶ +	۱	Combo Drive
+	۲	DVD-Writer
+	۳	FDD
+	۴	HDD
+	۵	RAM
*	۶	

Record: 1 of 6

جدول اصلی که با جدول Component table ارتباط برقرار می کند.

لیست فیلدها :

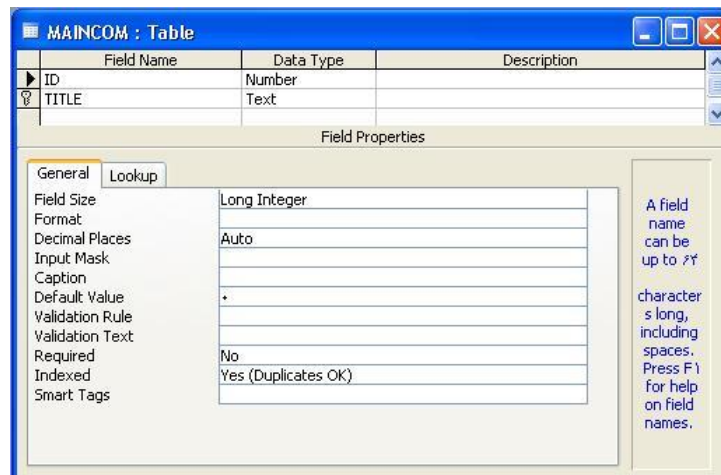
: ID (۱-۷)

شناسه برای فیلد Title

: Title (۲-۷)

عنوان، که همان داده های اصلی فیلد Title در جدول component table است،

مثل FDD یا RAM.



Field Name	Data Type	Description
ID	Number	
TITLE	Text	

Field Properties

General    Lookup

Field Size: Long Integer

Format:   

Decimal Places: Auto

Input Mask:   

Caption:   

Default Value: \*

Validation Rule:   

Validation Text:   

Required: No

Indexed: Yes (Duplicates OK)

Smart Tags:   

A field name can be up to 255 characters long, including spaces. Press F1 for help on field names.

## MainNote - ۸

ID	BRAND
۱	ACER
۲	DELL
۳	FUJITSO-SEIM
۴	GIGABYTE
۵	HP-COMPAQ
۶	IBM
۷	LENOVO
۸	LG
۹	SONY
۱۰	TOSHIBA

جدول اصلی که با جدول Note book Table ارتباط برقرار می کند.

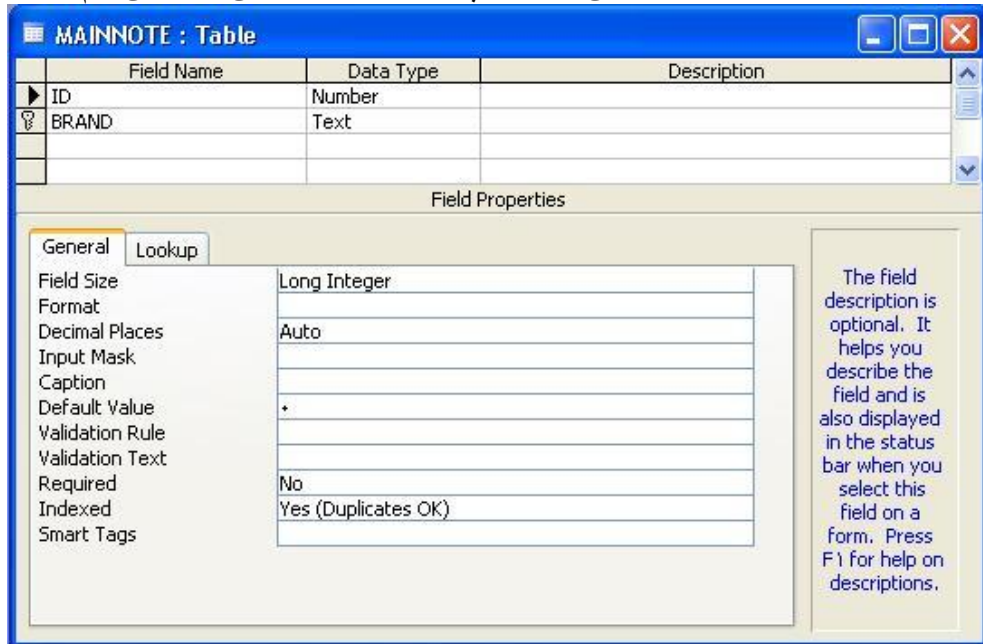
لیست فیلدها:

**ID (۱-۸**

شناسه برای فیلد Brand

**Brand (۲-۸**

مارک ( عنوان ) که همان داده های اصلی فیلد Brand در جدول Note book table است.

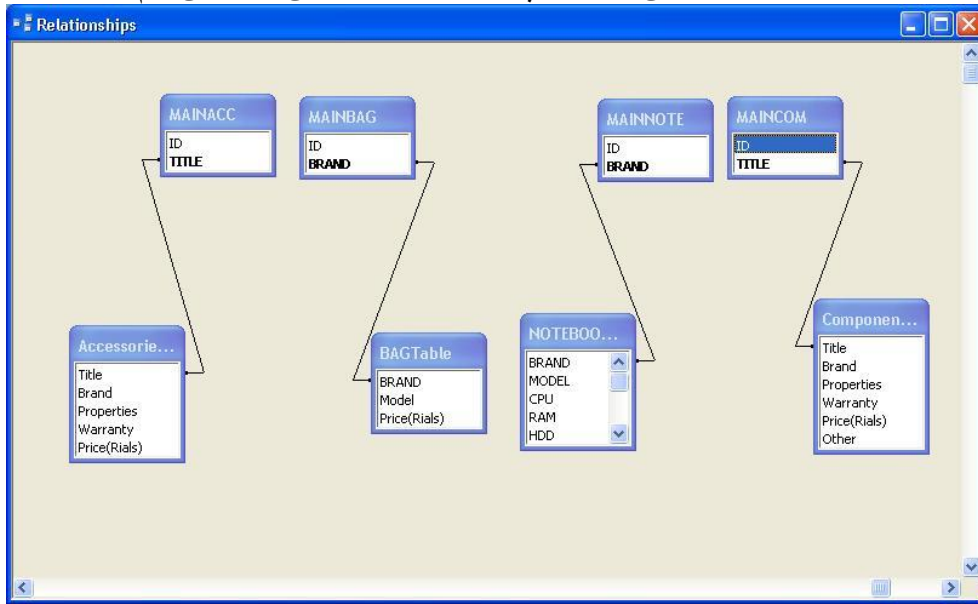


فیلد primary key فیلد Brand است.

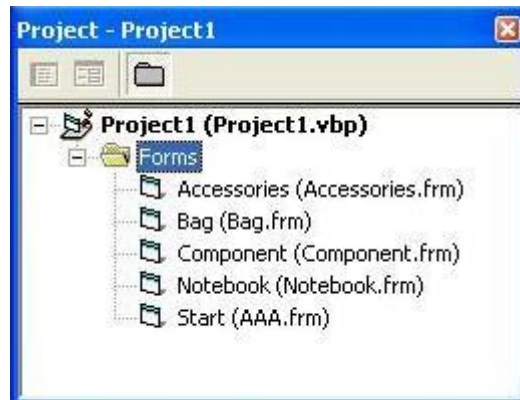
در کلیه جداول فوق به جز فیلد price(Rials) یا همان فیلد قیمت، که از نوع Number تعریف شده، نوع بقیه فیلدها Text است.

## : Relationships

ارتباط بین جداولها به صورتی تعریف شده که فیلد primary key در هر یک از جداول Main با فیلد متناظر در جدول معمولی مربوطه در ارتباط باشد. مثلاً فیلد Brand در جدول MainBag با فیلد Brand در جدول BagTable در ارتباط است.



### ب) توصیف فرم های برنامه



لیست فرم های این پروژه :

۱- Accessories ( فایل Accessories . frm )

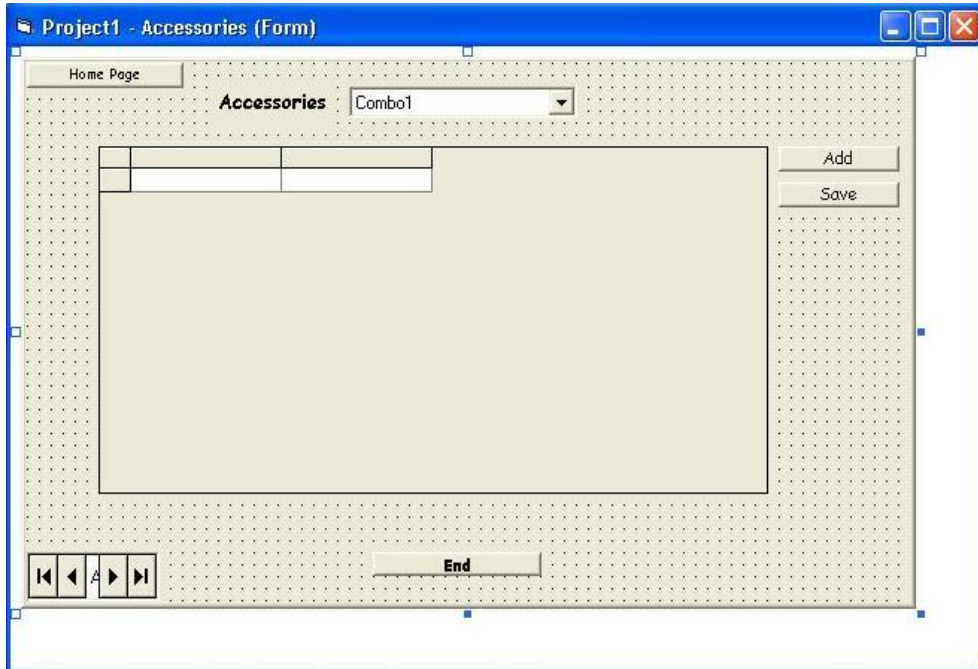
۲- Bag ( فایل Bag . frm )

۳- Component ( فایل Component . frm )

۴- Notebook ( فایل Notebook . frm )

۵- Start ( فایل AAA . frm )

## ۱- فرم Accessories



همان گونه که در شکل فوق ملاحظه می کنید، هشت object مختلف روی این فرم قرار گرفته اند که عبارتند از :

۴ عدد CommandButton ، ۱ عدد Label ، ۱ عدد ComboBox ، ۱ عدد Data Grid و ۱ عدد Adodc ( Ado Data Control )

نکته ۱: DataGrid ، Adodc ، به طور پیش فرض در Toolbox وجود ندارند و به طریق زیر به آن اضافه می شوند :

از منوی project ، زیر منوی components... را انتخاب کنید تا کادر محاوره ای components به نمایش در آید. در این کادر از نوار لغزان به طرف پایین حرکت کرده و گزینه Microsoft ADO Data Control 6.0 را انتخاب کنید. سپس دوباره به طرف پایین حرکت کرده و گزینه Microsoft Data Grid 6.0 را انتخاب کرده و بر روی دکمه OK کلیک نمایید. با بسته شدن این کادر محاوره ای، هر دو کنترل انتخاب شده به پایین Toolbox اضافه می شوند.



نکته ۲ : Adodc1 که روی این فرم قرار داده شده و به صورت جدا از DataGrid1 دیده می شود، در زمان اجرا دیگر به صورت جدا دیده نخواهد شد و به نحوی با Data Grid ترکیب می شود.

Caption در نظر گرفته شده برای label1 , Accessories است.

نام ComboBox قرار گرفته روی این فرم combo1 و نام Adodc قرار گرفته Adodc1 و نام DataGrid 1 مربوطه DataGrid 1 است.

### : Command Button

Home page : Caption با Command 1

End : Caption با Command 2

Add : Caption با index 0 ، Command 3

Save : Caption با index 1 ، Command 3

در پنجره properties مربوط به Data Grid1 تنظیمات زیر صورت گرفته است.

AllowAddNew = False

AllowDelete = True

AllowUpdate = False

DataSource = Adodc 1

**دکمه Home page :** با فشردن این دکمه توسط کاربر صفحه اصلی پروژه ( فرم

start که فرم اصلی برنامه است ) نمایش داده خواهد شد.

**دکمه End :** خاتمه برنامه

**دکمه Add :** اضافه کردن رکورد جدید به جدول Accessories

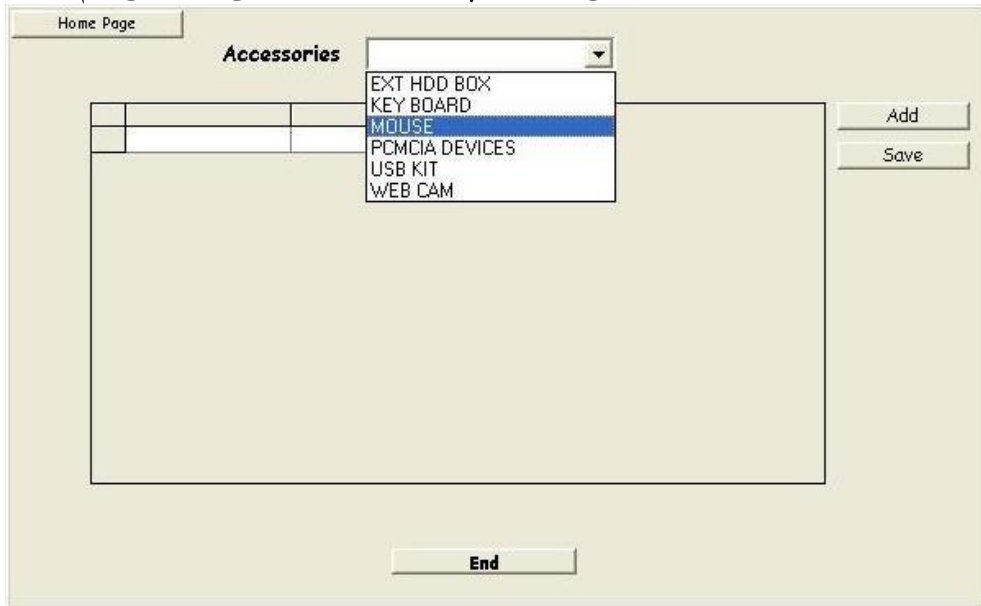
**دکمه Save :** ذخیره کردن اطلاعات Edit شده یا جدید وارد شده داخل جدول

. Accessores

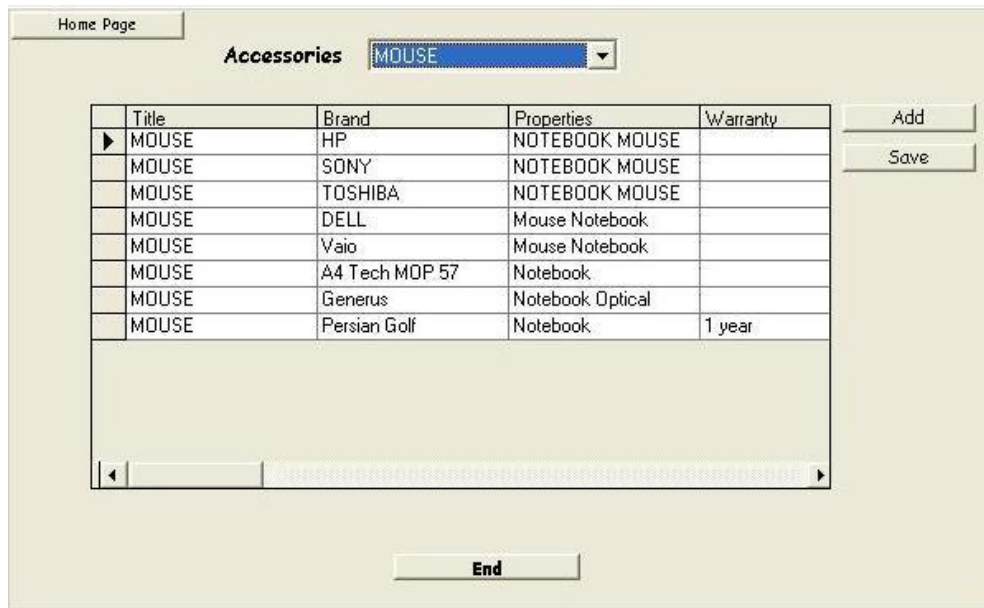
### : Combo1

با کلیک روی این ComboBox امکان انتخاب عنوان لوازم جانبی مورد نظر کاربر

میسر می گردد، به صورتی که در شکل زیر مشاهده می کنید :



کاربر با انتخاب گزینه مورد نظر خود می تواند لیست دلخواه خود را مشاهده کند:



### - شرح کد فرم Accessories ؛

پرینت کد این فرم در ضمیمه ۱ موجود است. (شماره هایی که در سمت چپ دستورات وجود دارد ، تنها به منظور خوانایی بیشتر برنامه و ایجاد آسانی در شرح کد آن ذکر گردیده است و جزء کد اصلی به شمار نمی آید).

برای فشردن یا انتخاب هر یک از object های موجود روی فرم Accessories یک private sub وجود دارد.

مثلاً ( ) private sub combo1\_click یعنی اگر روی ComboBox1 کلیک شد، دستورات زیر انجام شود :

**دستور ۱ تا ۴:** اگر روی Command1 ( دکمه Home page ) کلیک شد، فرم start ( که فرم اصلی و اولیه است ) نمایش داده شود و فرم Accessories پنهان گردد.

**دستور ۵ تا ۷:** اگر روی command 2 ( دکمه End ) کلیک شد، به اجرای برنامه خاتمه داده شود.

**دستور ۸ تا ۱۱:** اگر روی combo 1 ( combo Box ) کلیک شد، محتویات فیلد title از جدول Accessories table به عنوان منبع رکوردهای Adodc1 در نظر گرفته شود و در متن combo1 (لیست ComboBox) درج شود. با هر بار کلیک روی combo1 محتویات Adodc1 بازخوانی گردد. ( دستور ۱۰ ) where , select ( جز دستورات SQL هستند. )

**دستور ۱۲ تا ۲۷:** زمانی که فرم Accessories لود شده و نمایش داده می شود، باید کانکشن بین VB و فایل پایگاه داده های Access ایجاد گردد. برای این منظور به متغیری نیاز داریم که از نوع New ADOB. Connection باشد ( متغیر cn ). تنظیمات این متغیر پس از تعریف، مربوط به connection string است که شامل provider و persist security info و data source است. اولی اسم نرم افزار سازنده این کانکشن و دومی حالت امنیت اطلاعات و سومی اسم فایل پایگاه داده ها می باشد.

سپس از دستور ۱۵ به منظور بازکردن این کانکشن ایجاد شده استفاده می کنیم. علاوه بر متغیر cn به متغیری از نوع New ADOB.Recordset نیاز داریم. در اینجا از اسم rs استفاده کرده ایم. این متغیر وظیفه نقل و انتقال یک رکورد را بین VB و پایگاه داده بر عهده دارد.

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
دستور ۱۷ مشخص می کند که متغیر RS باید از چه جدولی و براساس چه کانکشنی چه  
رکوردهایی را منتقل کند. دستورات ۱۸ تا ۲۲ داده های فیلد Title را که RS از جدول  
MainAcc برداشته بود، به آیتم های combo1 اضافه می کند.  
حال باید داده های منتقل شده در adodc1 هم نمایش داده شوند، سپس با کمک  
دستور ۲۳ ، Adodc1.connectionString را همانند cn تعریف میکنیم.  
منظور از App.path این است که فایل های اصلی پروژه هر کجا که بودند، فایل  
data.mdb ( فایل Access پایگاه داده ) باید در همان آدرس دریافت شود.  
دستور ۲۴ باعث می شود که اطلاعاتی که قرار است در DataGrid نمایش داده شوند،  
مرتب update شوند و آخرین تغییرات سریعاً روی اطلاعات فایل بانک اطلاعاتی ذخیره  
و اعمال شود.  
دستورات ۲۵ و ۲۶ برای فعال کردن دو دکمه Add و save است.  
دستورات ۲۸ تا ۴۲: این دستورات مربوط به Add ، save است. نام هر دوی این  
Commandbutton ها Command3 است و برای هر یک از آنها یک index در  
نظر گرفته شده است.  
Index تعریف شده برای Add صفر و برای save یک می باشد. سپس نیاز به دستور  
select-case برای تشخیص این index و عملیات متناظر با هر کدام می باشد.  
Case 0 یا کلیک روی Add باعث می شود دستورات ۳۰ تا ۳۵ اجرا شوند و یک  
رکورد خالی به انتهای لیست رکوردها در DataGrid اضافه شود تا کاربر اطلاعات  
مربوطه را در آن وارد نماید.  
دستور ۳۱ اشاره گر رکورد جاری جدول را به انتهای جدول می برد و دستور ۳۲ باعث  
ایجاد یک رکورد جدید می شود. دستور ۳۳ باعث اعمال تغییرات داده شده در جدول  
می شود و اطلاعات جدید وارد شده را در جدول ( فایل ) اصلی ذخیره می نماید.  
دستورات ۳۴ و ۳۵ برای این است که دکمه های Add و save بعد از یک بار Add  
کردن غیرفعال نشده و همچنان قابل استفاده و Enable بمانند.

Home Page

Bag

BRAND	Model	PriceF
DICOTA	Start off	48000(
DICOTA	Back Pack (blue)	49000(
DICOTA	Back Pack (brown)	49000(
DICOTA	Back Pack Jump	45000(
DICOTA	Multi Access	44000(
DICOTA	Pep Up Junior	40000(
DICOTA	Ready 2 Go	33000(
DICOTA	Top Traveler Pro	88000(
DICOTA	Ultra Case M.	84000(
DICOTA	Carry 2 Go	35000(
DICOTA	Ultra Case Pro	70000(
DICOTA	Peffect Skin 13.3	15000(
DICOTA	Back Pack Rain Blue	49000(
*		

Add

Save

End

Case 1 یا کلیک روی save باعث می شود، تغییرات داده شده در جدول (موقع Edit ) یا همان اطلاعاتی که بعد از فشردن Add در رکورد جدید وارد کرده ایم، ذخیره شوند. دستورات ۳۷ و ۳۸ باعث فعال ماندن Add , save هر دو می شوند. دستور ۳۹ اشاره گر رکوردهای جدول را به ابتدا و اولین رکورد می برد و دستور ۴۰ باعث بازخوانی اطلاعات DataGrid1 می شود تا اطلاعات نمایش داده شده در آن به روز باشند. دستور ۴۱ برای این است که پس از یک بار زدن دکمه Add و ورود داده ها و ذخیره آن، حالت اضافه کردن یک رکورد به انتهای جدول تا زدن مجدد این دکمه غیرفعال شود.

Home Page

Notebook SONY

BRAND	MODEL	CPU	RAM	HDD
SONY	FE 690 P07	CoreDuo 1.83	1024	100
SONY	S 660 B	Sonoma 1.86	1024	100
SONY	BX 565	Centrino 1.86	512	80
SONY	BX 561 B	Centrino 1.73	512	80
SONY	BX 561 B	Centrino 1.73	512	40
SONY	AR150G	CoreDuo 1.83	512	160
SONY	FS 950	Celeron.M 1.73	256	80
SONY	FS 8900 P9	Sonoma 2.0	1024	80
SONY	BX 560 B11	CoreDuo 2.0	1024	80
SONY	FE 670G	CoreDuo 1.66	1024	100
SONY	SZ 210P/B	CoreDuo 1.66	512	80
SONY	FJ 270 B	Sonoma 1.86	1024	100
SONY	SZ 240 P10	CoreDuo 2.0	512	100
SONY	SZ 240 P11	CoreDuo 2.0	1024	120
SONY	FS 8900 P5	Sonoma 2.0	1024	100
SONY	SZ 260P/Carbon	CoreDuo 1.83	1024	120
SONY	SZ 340 P8	CoreDuo 2.0	1024	100

Add

Save

End

این نکته قابل ذکر است که برای Edit کردن اطلاعات جدول نیاز به زدن دکمه خاصی نیست. تنها کافی است به رکورد مورد نظر خود رفته و روی فیلدی که می خواهیم Edit کنیم، کلیک کنیم، اطلاعات مورد نظر خود را تغییر دهیم و بعد دکمه Save را برای ذخیره نهایی فشاردهیم.

Home Page

Notebook SONY

BRAND	MODEL	CPU	RAM	HDD
SONY	SZ 240 P11	CoreDuo 2.0	1024	120
SONY	FE 690 P07	CoreDuo 1.83	1024	100
SONY	S 660 B	Sonoma 1.86	1024	100
SONY	BX 565	Centrino 1.86	512	80
SONY	BX 561 B	Centrino 1.73	512	80
SONY	BX 561 B	Centrino 1.73	512	40
SONY	AR150G	CoreDuo 1.83	1024	160
SONY	FS 950	Celeron.M 1.73	256	80
SONY	FS 8900 P9	Sonoma 2.0	1024	80
SONY	BX 560 B11	CoreDuo 2.0	1024	80
SONY	FE 670G	CoreDuo 1.66	1024	100
SONY	SZ 210P/B	CoreDuo 1.66	512	80
SONY	FJ 270 B	Sonoma 1.86	1024	100
SONY	SZ 240 P10	CoreDuo 2.0	512	100
SONY	SZ 240 P11	CoreDuo 2.0	1024	120
SONY	FS 8900 P5	Sonoma 2.0	1024	100
SONY	SZ 260P/Carbon	CoreDuo 1.83	1024	120

Add

Save

End

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار در صورتی که بخواهیم رکوردی را به طور کامل حذف کنیم، باید روی مربع خاکستری رنگ سمت چپ کنار رکورد مورد نظر کلیک کنیم تا آن رکورد به طور کامل select شود، سپس دکمه Delete کیبورد را بفشاریم.

Title	Brand	Properties	Warranty	Price(Rials)	Other
RAM	Queen	DDR2 256	Diagram	250000	
RAM	Kingstone	DDR 256	GLC	280000	
RAM	Samsung	256 DDR	---	260000	
RAM	Kingstone	512 DDR	---	490000	
RAM	Vdata	256 DDR	---	260000	
RAM	GIEL	1G DDR2		1100000	
RAM	Kingston	1G DDR		1150000	
RAM	Infineon	256 DDR2		270000	
RAM	Sumsung	256 DDR2		270000	
RAM	Hynix	256 DDR		280000	
RAM	Infineon	512 DDR2		550000	
RAM	GIEL	512 DDR		570000	
RAM	Queen	512 DDR		570000	
RAM	GIEL	512 DDR2		550000	

## فرم Bag

Title	Brand	Properties	Warranty	Price(Rials)	Other
-------	-------	------------	----------	--------------	-------

مشخصات این فرم کاملاً مشابه فرم Accessories است. تنها تفاوت، مربوط به caption در label1 است که در این فرم عبارت "Bag" می باشد.

The screenshot shows a web form titled "Bag" with a "Home Page" button in the top left. A dropdown menu is open, displaying a list of brand names: DELL, DICOTA, SONY, SUMDEX (highlighted), TARGUS, and TOSHIBA. To the right of the dropdown are "Add" and "Save" buttons. Below the dropdown is a large empty rectangular area. At the bottom center is an "End" button.

کاربر با انتخاب گزینه مورد نظر خود می تواند لیست کیفهای نوت بوک را مشاهده کند :

The screenshot shows the "Bag" web form with the dropdown menu set to "SUMDEX". Below the dropdown is a table with the following data:

	BRAND	Model	Price(Rial)
▶	SUMDEX	002	195000
	SUMDEX	854	200000
	SUMDEX	818	520000
	SUMDEX	002 Logo	195000

To the right of the table are "Add" and "Save" buttons. Below the table is a horizontal scrollbar. At the bottom center is an "End" button.



### شرح کد فرم Bag ؛

پرینت کد این فرم در ضمیمه ۲ موجود است. (شماره هایی که در سمت چپ دستورات وجود دارد ، تنها به منظور خوانایی بیشتر برنامه و ایجاد آسانی در شرح کد آن ذکر گردیده است و جزء کد اصلی به شمار نمی آید).

کد این فرم کاملاً مشابه کد فرم Accessories است، با سه تفاوت مهم و تعیین کننده در سه سطر:

۱- در سطر ۹ نام جدولی که قرار است به عنوان منبع اطلاعات رکوردها باشد،

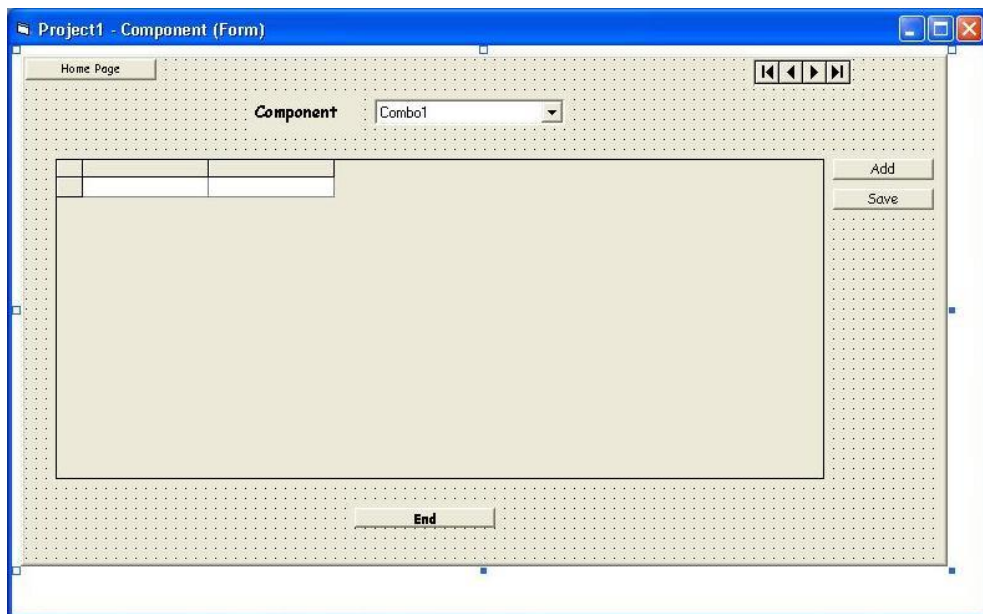
BagTable تعریف شده است و فیلد مورد نظر Brand است نه Title .

۲- در سطر ۱۷ نام جدولی که اطلاعات یکی از فیلدهای آن قرار است در

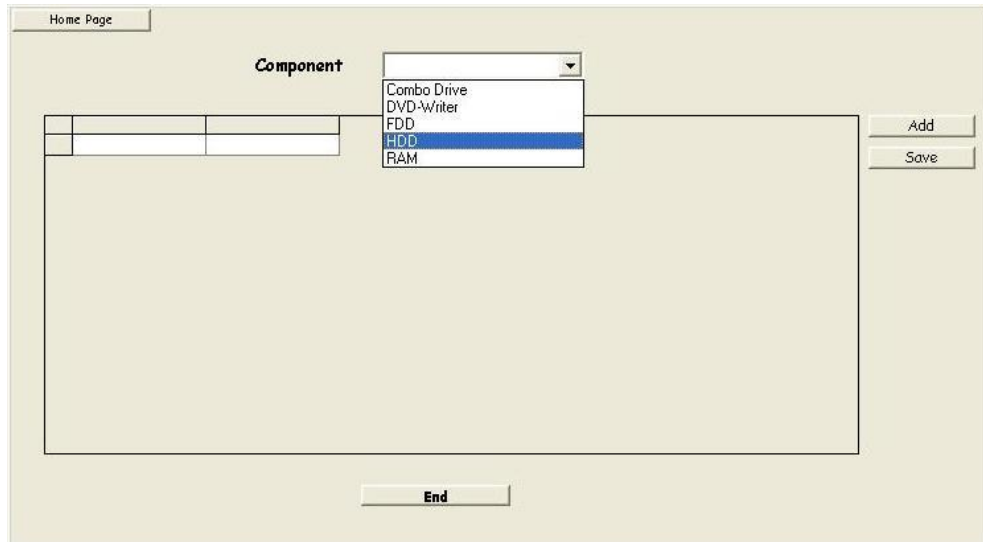
ComboBox قرار بگیرد، MAINBAG تعریف شده است.

۳- اسم فیلد مورد نظر در سطر ۲۰، Brand است نه Title .

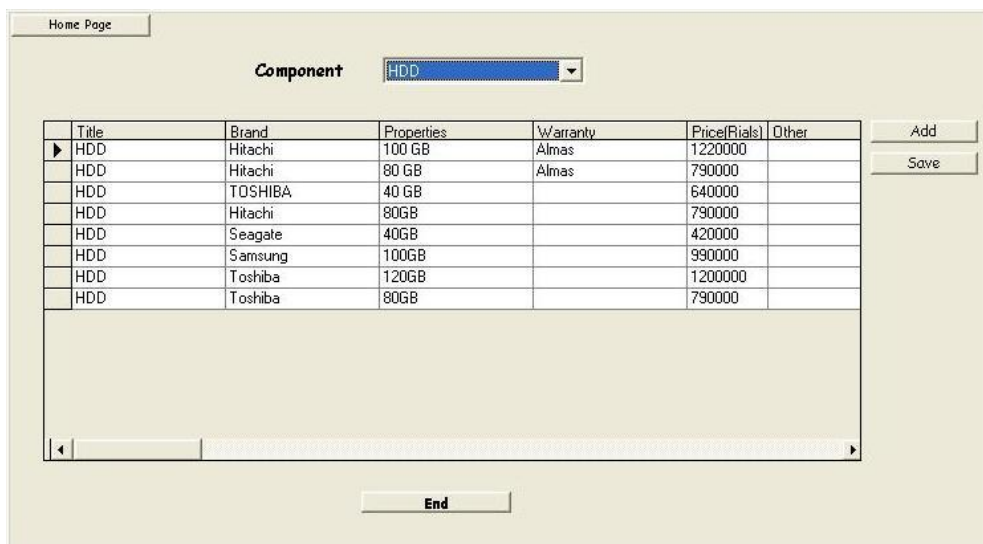
### ۳- فرم component ؛



مشخصات این فرم کاملاً مشابه فرم Accessories است. تنها تفاوت مربوط به caption در label1 است که در این فرم عبارت "component" میباشد.



کاربر با انتخاب گزینه مورد نظر خود می تواند لیست اجزاء اصلی و داخلی موجود در فروشگاه را مشاهده نماید :



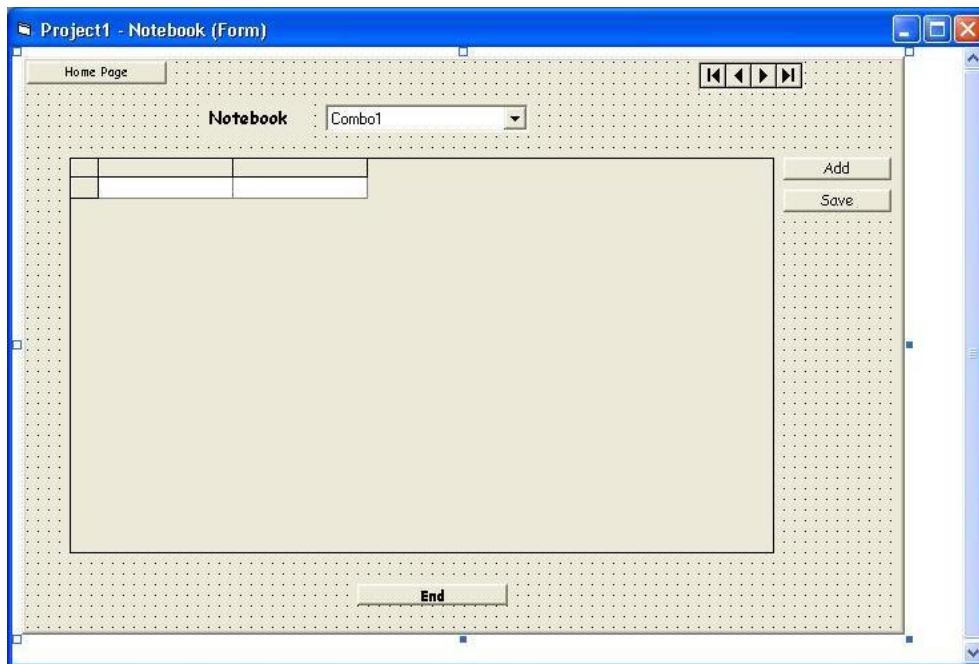
-شرح کد فرم component :

UML SSADM طراحی وبسایت ، پایگاه داده برنامه نویسی ، مهندسی نرم افزار  
پرینت کد این فرم در ضمیمه ۳ موجود است. (شماره هایی که در سمت چپ دستورات  
وجود دارد ، تنها به منظور خوانایی بیشتر برنامه و ایجاد آسانی در شرح کد آن ذکر  
گردیده است و جزء کد اصلی به شمار نمی آید.)  
کد این فرم کاملاً مشابه کد فرم Accessories است. با دو تفاوت مهم و تعیین کننده  
در دو سطر:

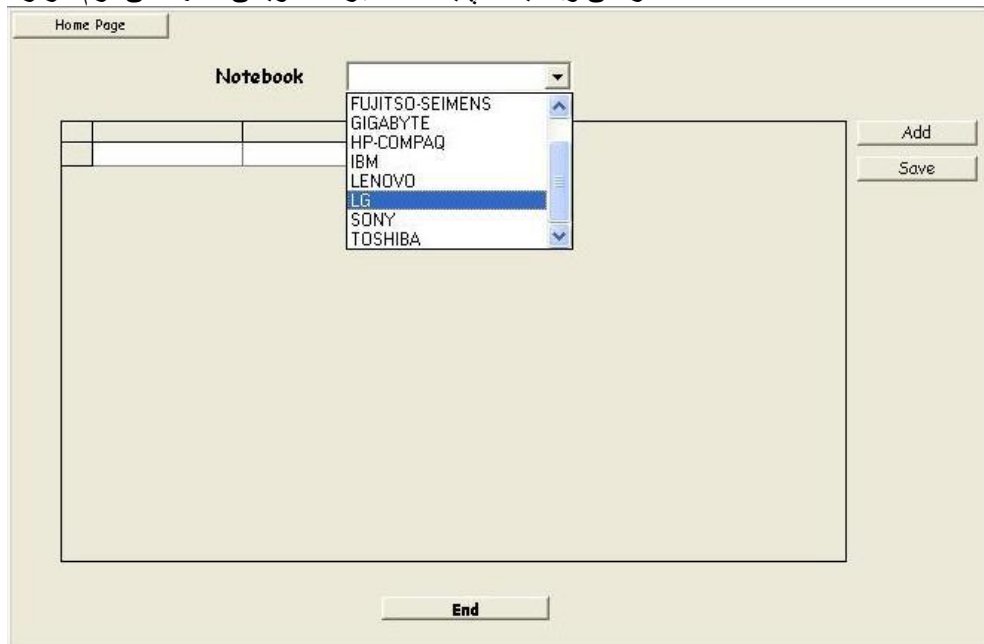
۱- در سطر ۹ نام جدولی که قرار است به عنوان منبع اطلاعات رکوردها باشد،  
componentTable تعریف شده است.

۲- در سطر ۱۷ نام جدولی که اطلاعات یکی از فیلدهای آن قرار است در  
comboBox قرار بگیرد ، MAINCOM تعریف شده است.

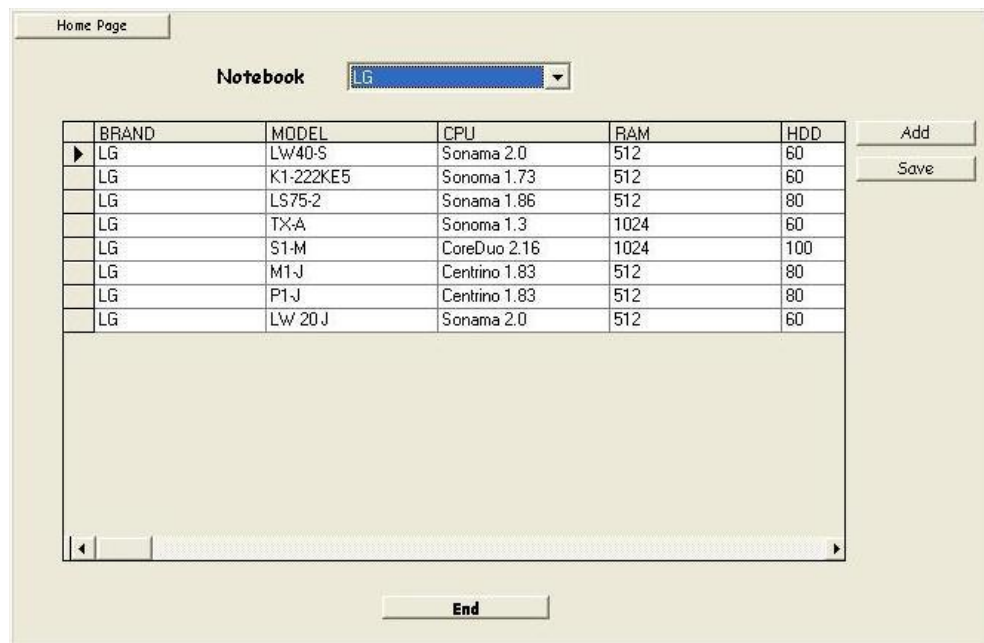
### ۳- فرم Notebook



مشخصات این فرم کاملاً مشابه فرم Accessories است. تنها تفاوت مربوط به  
caption در label1 است که در این فرم عبارت "Notebook" می باشد.



کاربر با انتخاب گزینه مورد نظر خود می تواند لیست نوت بوکهای فروشگاه را مشاهده نماید.



### –شرح کد فرم Note book؛

پرینت کد این فرم در ضمیمه ۴ موجود است. (شماره هایی که در سمت چپ دستورات وجود دارد ، تنها به منظور خوانایی بیشتر برنامه و ایجاد آسانی در شرح کد آن ذکر گردیده است و جزء کد اصلی به شمار نمی آید.)

کد این فرم کاملاً مشابه کد فرم Accessories است، با سه تفاوت مهم و تعیین کننده در سه سطر:

۱- در سطر ۹ نام جدولی که قرار است به عنوان منبع اطلاعات رکوردها باشد،

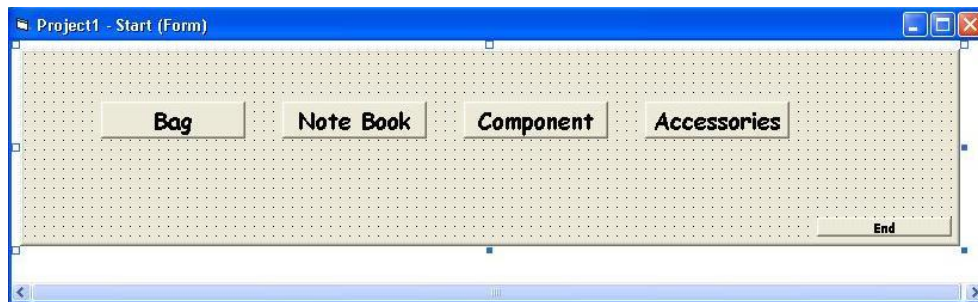
Note book table تعریف شده است و فیلد مورد نظر Brand است نه Title .

۲- در سطر ۱۷ نام جدولی که اطلاعات یکی از فیلدهای آن قرار است در combo

Box قرار بگیرد، MAINNOTE تعریف شده است.

۳- اسم فیلد مورد نظر در سطر ۲۰، Brand است نه Title .

#### ۵- فرم Start (AAA)



همانگونه که در شکل فوق ملاحظه می کنید، پنج object مختلف روی این فرم قرار گرفته اند که همگی از نوع CommandButton هستند.

Command 1 با چهار index :

Bag : index0 -۱

Note Book : index1 -۲

Component : index2 -۳

Accessories : index3 -۴

Command 2 با Caption " End " :

دکمه End برای خروج از برنامه است.

دکمه Bag برای نمایش فرم Bag و دکمه Note Book برای نمایش فرم

Notebook و دکمه Component برای نمایش فرم Component و دکمه

Accessories برای نمایش فرم Accessories است.



کاربر با انتخاب هر یک از دکمه های فوق می تواند به یکی از دسته های کالاهای فروشگاه دسترسی پیدا کند و یا اینکه از پروژه خارج شود.

### - شرح کد فرم Start(AAA) ؛

پرینت کد این فرم در ضمیمه ۵ موجود است. (شماره هایی که در سمت چپ دستورات وجود دارد ، تنها به منظور خوانایی بیشتر برنامه و ایجاد آسانی در شرح کد آن ذکر گردیده است و جزء کد اصلی به شمار نمی آید).

ایندکس command1 به عنوان ورودی تابع command1\_click مشخص کننده فرمی است که قرار است نمایش داده شود.

با یک دستور select-case ساده می توان از مقدار index برای این منظور بهره برد.

اگر مقدار index مساوی صفر باشد دستور سطر ۴ یعنی Bag.show (نمایش فرم Bag) اجرا خواهد شد. اگر مقدار یک باشد، Notebook.show اگر ۲ باشد، component.show و اگر ۳ باشد، Accessories.show اجرا خواهد شد.

در هر صورت هر کدام از دستورات فوق که اجرا شود و هر کدام از فرمها که به نمایش در آید، باید فرم start که همان فرم اصلی برنامه است، پنهان شود با دستور Me. hide .

اگر به جای command1 ,command2 کلیک شود، دستور End (خاتمه دهنده برنامه) اجرا می شود.