

فهرست مطالب

۸	مقدمه ناشر
۹	مقدمه مؤلف
۱۱	فصل اول: تاریخچه و جایگاه پایگاه داده‌ها در ویژوال بیسیک
۱۳	فصل دوم: آشنایی مقدماتی با پایگاه داده‌ها
۱۳	پایگاه داده‌ها چیست؟
۱۳	مفاهیم فیلد و رکورد
۱۴	اشاره‌گر رکورد
۱۵	مفاهیم EOF و BOF
۱۵	مفهوم INDEX
۱۶	ایجاد یک پایگاه داده
۱۶	مراحل طراحی یک پایگاه داده
۱۹	فصل سوم: ایجاد پایگاه داده‌های اکسس از طریق ویژوال بیسیک
۲۹	فصل چهارم: اصول استفاده از پایگاه داده‌ها در ویژوال بیسیک
۲۹	تعریف لیست داده
۲۹	انواع لیست داده
۴۱	فصل پنجم: آشنایی با اشیای پایگاه داده
۴۲	شیء DBEngine
۴۳	شیء Workspace و مجموعه Workspaces
۴۶	شیء Database و مجموعه Database
۴۸	شیء TableDef و مجموعه TableDefs
۵۰	شیء QueryDef و مجموعه QueryDefs
۵۱	شیء Recordset و مجموعه Recordsets
۵۳	شیء Field و مجموعه Fields

فصل ششم: بررسی دقیق تر مشخصات اشیای دسترسی به داده ۵۵

- ۵۷ ایجاد شیء‌های Database
- ۵۸ خصوصیات و متدهای شیء Workspace
- ۵۹ خصوصیات شیء Database
- ۶۰ متدهای اجرایی در شیء Database
- ۶۲ ارتباط با یک پایگاه داده از نوع Jet
- ۶۳ استفاده از مجموعه TableDefs و شیء TableDef
- ۶۵ آشنایی با Flags و ثوابت نشانه‌ای (Symbolic Constant)
- ۶۶ مشخص کردن اعضای اشیای پایگاه داده به وسیله مجموعه TableDefs
- ۶۶ برنامه کاربردی Data Dictionary for Database
- ۶۹ مجموعه فیلدها و ایندکسها
- ۷۱ لیست مقادیر مشخصه Type از شیء Fields
- ۷۶ استفاده از مجموعه QueryDefs و شیء QueryDef
- ۷۹ ایجاد اشیای Recordset و استفاده از آنها
- ۸۰ لیست داده از نوع Dynamic
- ۸۰ لیست داده از نوع Static
- ۸۱ خصوصیات شیء Recordset
- ۸۴ متدهای اجرایی در شیء Recordset و مجموعه آن

فصل هفتم: استفاده از ابزار کنترل داده برای دسترسی به اشیای اطلاعات ۸۷

- ۹۰ متدهای اجرایی در Data Control
- ۹۱ رویداد Validate Data Control
- ۹۲ عملیات مقید سازی
- ۹۳ ایجاد یک فرم اطلاعاتی پیشرفته
- ۹۳ نمایش اطلاعات
- ۹۴ کنترل‌های حرکت روی رکوردها
- ۹۶ رفع خطای No Current Record
- ۹۷ جستجو توسط دستورات Find ...
- ۹۹ حذف، اضافه و تغییر رکوردها
- ۱۰۰ تنظیم کدهای رویداد Validate

۱۰۳	فصل هشتم: زبان پرس و جوی ساختاری (SQL)
۱۰۳	تاریخچه SQL
۱۰۳	سؤال (Query) و (SQL)
۱۰۴	انواع سؤال (Query)
۱۰۴	آشنایی با زبان SQL
۱۰۴	SELECT
۱۰۵	Distinct , All
۱۰۵	FROM
۱۰۵	WHERE
۱۰۵	اوپراتورهای SQL
۱۰۵	ORDER BY
۱۰۶	استفاده از برنامه Visual Data برای آزمایش سؤالها
۱۱۱	اوپراتورها و عبارات در SQL
۱۱۱	اوپراتورها و توابع ریاضی در عملگرهای دو تایی
۱۱۲	ستونهای محاسباتی در سؤال
۱۱۳	ستونهای ترکیبی
۱۱۴	اوپراتورهای تک عملگر
۱۱۵	جداول مرتبط
۱۱۵	ارتباط داخلی INNER JOIN
۱۱۶	ارتباط داخلی چندگانه
۱۱۷	ارتباط خارجی OUTER JOIN
۱۱۸	سؤالهای تو در تو Nested Queries
۱۲۰	زیر سؤال Sub Queries
۱۲۰	شرط به صورت زیر مجموعه Subset Sub Queries
۱۲۰	شرط به صورت مقایسه‌ای Comperison Sub Queries
۱۲۱	اجتماع سؤالها (Union Queris)
۱۲۱	تابعهای جمعی در SQL و دستورات Having , GroupBy
۱۲۵	فصل نهم: زبان پرس و جوی ساختاری (SQL) پیشرفته
۱۲۵	استفاده از سؤال سطر و ستون برای ارائه دادن خلاصه اطلاعات

۱۲۵	جدول زمانی
۱۲۵	جدول طبقه‌بندی
۱۲۶	ایجاد خلاصه اطلاعات توسط دستور Group By
۱۲۹	استفاده از دستورات Pivot , Transform جهت ایجاد یک سؤال سطر و ستون
۱۳۱	ساختن سؤالهای سطر و ستون به صورت طبقه‌بندی شده
۱۳۲	سؤالهای اجرایی
۱۳۲	اجرای یک سؤال اجرایی توسط ویژوال بیسیک
۱۳۳	استفاد از سؤالهای اجرایی
۱۳۳	ساختن یک جدول جدید با استفاده از Make Table Query
۱۳۴	اضافه کردن رکورد به جدول
۱۳۵	تغییر مقادیر اطلاعات در جدولها
۱۳۵	حذف رکوردها از داخل جدول
۱۳۷	فصل دهم: استفاده از کنترل‌های پیشرفته آگاه از داده‌ها
۱۳۷	نمایش و به روز رسانی شیء لیست داده توسط کنترل DBGrid
۱۳۹	استفاده از DBGrid
۱۴۹	استفاده از رویداد Head Click
۱۴۹	تغییر نمایش ستونها در حالت اجرا
۱۵۱	استفاده از رویدادها برای تصدیق به روز رسانی داده‌ها
۱۵۳	انتخاب اقلام به وسیله DB combo , DBList
۱۵۳	استفاده از DB combo برای محدود کردن لیست DB Grid
۱۵۶	اتصال دو کنترل DB combo برای انتخاب رکوردهای مخصوص
۱۵۷	استفاده از DBList
۱۵۹	فصل یازدهم: چاپ گزارش در ویژوال بیسیک
۱۵۹	دسترسی به چاپگر از طریق کد
۱۶۰	متدهای مهم شماره چاپگر
۱۶۱	جدول پارامترهای متد Paint Picture
۱۶۴	چاپ گزارش از لیست داده
۱۶۵	استفاده از کنترل Crystal Report
۱۶۶	ایجاد یک فایل گزارش ساده توسط (Crul32.EXE)

۱۶۷	قسمت اول
۱۶۷	قسمت دوم
۱۶۷	قسمت سوم
۱۶۸	بررسی ظاهری هر گزارش
۱۶۹	اضافه کردن یک پایگاه داده به گزارش
۱۶۹	مرتب کردن یک فیلد در گزارش
۱۷۰	نحوه تعیین قالب برای فیلد متن
۱۷۲	تعیین قالب برای فیلد عددی
۱۷۳	تعیین اندازه کاغذ
۱۷۳	وضعیت نمایش هر قسمت در چاپ
۱۷۳	گروه بندی یک فیلد
۱۷۵	کنترل گزارش در ویژوال بیسیک
۱۷۸	چاپ یک گزارش در ویژوال بیسیک
۱۷۹	چاپ یک گزارش در یک فایل
۱۸۱	ایجاد پنجره پیش نمایش در فرم MDI Child
۱۸۳	فصل دوازدهم: محافظت از پایگاه داده ها در ویژوال بیسیک
۱۸۴	تعریف یک کاربر مشخص در تمام طول کار در برنامه
۱۸۴	استفاده از CreateWorkspace
۱۸۵	شیء Group و مجموعه Group
۱۸۷	شیء User و مجموعه Users
۱۸۹	ایجاد فایل محافظت
۱۹۲	تعریف کاربران و گروه های کاربران
۱۹۵	تخصیص محدودیتها به گروهها و کاربران
۱۹۹	فهرست منابع

مقدمه ناشر

حمد و سپاس ایزد منان را که با الطاف بیکران خود این توفیق را به ما ارزانی داشت تا بتوانیم در راه ارتقای دانش عمومی و فرهنگ این مرز و بوم، در زمینه چاپ و نشر کتاب علمی بخصوص در علوم کامپیوتر و انفورماتیک گامهایی هر چند کوچک برداشته و در انجام رسالتی که بر عهده داریم مؤثر واقع شویم. گستردگی علوم کامپیوتر و توسعه روزافزون آن، شرایطی را به وجود آورده که هر روز شاهد تحولات اساسی چشمگیر در سطح جهان هستیم. این گسترش و توسعه نیاز به منابع مختلف از جمله کتاب را به عنوان قدیمی‌ترین و راحت‌ترین راه دستیابی به اطلاعات و اطلاع‌رسانی، بیش از پیش روشن می‌نماید. در این راستا و برای نیل به این اهداف، واحد انتشارات این مرکز با همکاری جمعی از اساتید، مؤلفان، مترجمان، متخصصان، پژوهشگران، محققان و نیز پرسنل ورزیده و ماهر در زمینه امور نشر درصدد هستند تا با تلاشهای مستمر خود برای رفع کمبودها و نیازهای موجود، منابعی پر بار، معتبر، آسان و با کیفیت مناسب در اختیار علاقه‌مندان قرار دهند. کتابی که در دست دارید با عنوان راهنمای کاربران «پایگاههای داده تحت ویژوال بیسیک» از مجموعه کتابهایی است که تهیه و تدوین آن با همت جمعی از همکاران میسر گشته و شایسته است از یکایک این گرامیان که از مساعدت آنها بهره‌مند بوده‌ایم بخصوص «آقای ایمان مدرس غروی» در زمینه ترجمه این اثر و همچنین سایر پرسنل این مرکز:

امور اجرایی و نشر: مهندس مهدی خانی، خانم بوسه رکنیان

ویراستاری: خانم معصومه فرشیدی

ویرایش کامپیوتری، صفحه‌آرایی و تصویرگذاری: خانم فرحناز امینی

طراحی روی جلد: خانم شیما صدرا

امور چاپ و نشر: آقای حیدر شفیعی

ناظر چاپ: آقای کریم براغ

قدردانی نمائیم، بدیهی است این تلاشها زمانی مفیدتر خواهند بود که هموطنان عزیز و دانش پژوهان گرامی بذل عنایت فرموده، با ارائه پیشنهادها و انتقادهای خود ما را در بهبود کمی و کیفی کارهای انجام شده راهنمایی نمایند تا بتوانیم در آینده کتابهایی با کیفیت بهتر تقدیم حضورشان نماییم.

مقدمه مؤلف

ویژوال بیسیک به عنوان زبانی که پایه زبان برنامه نویسی نرم افزارهای کاربردی میکروسافت نظیر MS Office که در هر بخش آن تخصصهایی نظیر پایگاههای داده ، صفحات گسترده ، ویرایشگر متن و غیره قرار گرفته است ، وظیفه اش را در زمینه ارتباط بین این مقوله های تخصصی به خوبی انجام داده است و همین امر باعث شده که جایگاهش را میان برنامه نویسان حفظ کند.

به وسیله این زبان می توانید برنامه های بنویسید که این تخصصها را یکجا جمع کند. این کتاب به شما کمک می کند تا بتوانید بین ویژوال بیسیک و پایگاههای داده مانند اکسس، فاکس پرو و غیره ارتباط برقرار نمایید و عملیات پایگاه داده را روی آنها انجام دهید. برای استفاده از این کتاب ، آشنایی اولیه با ویژوال بیسیک لازم می باشد اما در صورتی که هیچ آشنایی قبلی با پایگاههای داده ندارید فصول اولیه کتاب شما را با مفاهیم اولیه آن آشنا می سازد.

تاریخچه و جایگاه پایگاه داده‌ها در ویژوال بیسیک

در بین سالهای ۱۹۹۲ و ۱۹۹۳ شرکت مایکروسافت سه محیط جدید پایگاه داده‌ها را تحت ویندوز عرضه کرد که شامل فاکس پرو برای ویندوز، ویژوال بیسیک ۳ و اکسس ۱ بود. این شرکت اعلام کرد که اکسس پایگاه داده مورد استفاده هر فرد می‌باشد و در مدت نود و پنج روز، هفتصد و پنجاه هزار نسخه از آن به فروش رفت. فاکس پرو برای ویندوز با هدف توسعه کار پایگاه داده‌های شرکت بین‌المللی بورلند برای ویندوز به وجود آمد. هم اکسس و هم فاکس پرو در بازار از پاراداکس محصول شرکت بورلند پیشی گرفتند که این برنامه مدتی پس از اکسس ۱ به بازار عرضه شد. تا سال ۱۹۹۶ شرکت مایکروسافت بیش از چهار میلیون نسخه از اکسس، ویرایشهای ۱.۰، ۱.۱ و ۲.۰ را فروخت و تعداد استفاده کنندگان ویژوال بیسیک، ویرایشهای یک، دو و سه را تا دو میلیون نفر بالا برد.

بین اواسط ماههای آگوست و نوامبر در سال ۱۹۹۵، شرکت مایکروسافت نسخه هفتم از اکسس، ویژوال فاکس پرو ۳ و ویژوال بیسیک ۴ را برای نسخه سی و دو بیتی ویندوز یعنی Windows 95 به همراه Ms office 95 عرضه کرد و این درست در زمانی بود که نسخه ۳۲ بیتی ویندوز تحول عظیمی در بازار کاربران بوجود آورده بود و به زبانهایی با ساختار ۳۲ بیتی نیاز شدیدی می‌رفت.

مایکروسافت اکسس، ویژوال فاکس پرو و پاراداکس برای ویندوز در زمره پایگاه داده‌های رومیزی هستند. پایگاه داده‌های رومیزی برنامه‌هایی هستند که به صورت اختصاصی، فایل پایگاه داده را به خدمت می‌گیرند و می‌توانند روی کامپیوترهای شخصی مناسب اجرا شوند. مایکروسافت اعلام کرد که اکسس برای ویندوز به ۱۲ مگا بایت رم احتیاج دارد و برای بهتر اجرا شدن ۱۶ مگا بایت یا بیشتر مقدار مناسبی می‌باشد (البته این برنامه با ۸ مگابایت رم نیز اجرا خواهد شد ولی از بسیاری قابلیت‌ها محروم می‌شود). پایگاه داده‌های رومیزی معمولاً از زبانهای برنامه‌نویسی توکار که برای استفاده از محیط مخصوص همان برنامه‌ها طراحی شده است، تشکیل می‌شوند به این معنی که نوشتن برنامه تحت محیط این پایگاه داده‌ها محدود به آنهاست و فقط در همان محیط پایگاه داده قابل اجرا می‌باشد.

زبان برنامه نویسی VBA (Visual Basic for Application) زبان استاندارد این برنامه‌هاست فقط ویژوال فاکس پرو در همان گروه XBase استوار مانده است. ویژوال بیسیک برای ویندوز یکی از محصولات موفق شرکت مایکروسافت در زبانهای برنامه‌نویسی است و این امر از فروش بیش از دو

میلیونی در بازار جهانی نتیجه می‌شود. هدف اصلی میکروسافت از ویژوال بیسیک این بود که هر فردی حتی اگر هیچ تجربه و یا اطلاعاتی در مورد زبان C نداشته باشد بتواند برنامه‌های خودش را تحت ویندوز به وسیله این برنامه زبان بنویسد و به هدف خود نیز برسد. با ایجاد ویژوال بیسیک خیلی از برنامه‌نویسان با تجربه، زبانهای C، C++ و پاسکال را ترک کردند و به ویژوال بیسیک رو آوردند زیرا اکنون آنها می‌توانستند برنامه‌های تحت ویندوز خود را بسیار سریعتر از زبانهای برنامه‌نویسی مرسوم توسعه ببخشند.

میکروسافت نسخه ۳ از ویژوال بیسیک را با اشیای تغییرپذیر، برنامه‌های واسط چند سندی (MDI) و دیگر افزودنیها غنی‌تر ساخت و یک پایگاه داده محدود به آن اضافه کرد.

در اوایل ۱۹۹۳ شرکت میکروسافت دریافت که ۷۰٪ برنامه‌های ساخته شده توسط ویژوال بیسیک از پایگاه داده‌ها تشکیل شده است، بنابراین ویژوال بیسیک ۳ را با شیء دسترسی به داده‌ها (Date Access Object) DAO معرفی کرد که به شدت با پایگاههای داده همسازي داشت و این روند باعث شد که در اواسط ۱۹۹۵ ویژوال بیسیک عمومی‌ترین ابزار برنامه‌سازی تحت ویندوز برای ساختن پایگاه داده‌های Client / Server شود با وجود اینکه در آن سالها رقابت تبلیغاتی بر سر ابزارهای توسعه یافته مانند Power Builder و بورلند دلفی بود.

یکی از هدفهای اصلی میکروسافت برای تولید ویژوال بیسیک ۴ تأمین محیط ۳۲ بیتی بود که کامل کننده Ms Office تحت نسخه‌های سی و دو بیتی ویندوز شد. این نسخه هدف شرکتهایی را که می‌خواستند سیستمهای اطلاعاتی تجاری خود را تحت محیط شبکه اجرا کنند، تأمین می‌کرد و شامل مدیریت پایگاه داده‌های توزیع شده SQL، Ms Mail 3.0، ارتباط با کامپیوترها ی Main frame (SNA) و مدیریت نرم‌افزار، سخت افزار SMS Server بود.

پایگاههای داده پیشرفته ویژوال بیسیک نسخه‌های ۴ به بعد و خصوصیت اداره اطلاعات آن، این زبان را به صورت یک محیط پیشرفته برنامه‌های پایگاه داده‌ها توصیف می‌کند.

آشنایی مقدماتی با پایگاه داده‌ها

پایگاه داده‌ها چیست؟

به هر مجموعه از داده‌های مربوط به یکدیگر و طراحی شده برای برآوردن خواسته‌های اطلاعاتی یک پایگاه داده‌ها گفته می‌شود. اطلاعات در یک پایگاه داده معمولاً به شکل یک جدول شامل سطرها و ستونها سازماندهی و ذخیره می‌شوند. به عنوان مثال شکل ۱-۲ یک لیست از مشخصات افراد را نشان می‌دهد. هر سطر این لیست شامل نام، نام شرکت، سمت و کد شناسایی است. هر سطر با سطر دیگر در ارتباط می‌باشد زیرا نوع اطلاعات همگی آنها یکسان بوده و چون این لیست بانظم مشخصی سازماندهی شده به آن جدول (Table) گفته می‌شود.

	Name	Company	Title	ID
▶				
*				

شکل ۱-۲.

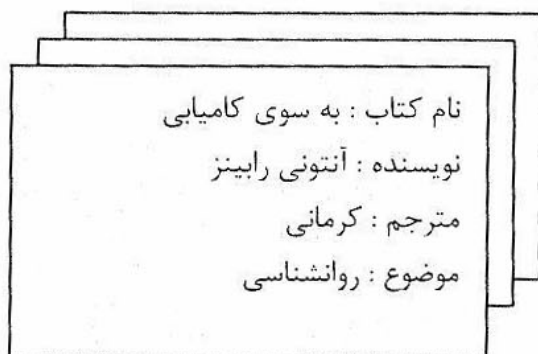
نمونه یک جدول اطلاعاتی.

به مجموعه‌ای که شامل یک یا چند جدول مرتبط یا غیر مرتبط باشند پایگاه داده‌ها (Database) گفته می‌شود. تاکنون با دو اصطلاح پایگاه داده‌ها (Database) و جدول (Table) آشنا شدید. حال شما را با دو مفهوم جدید آشنا می‌کنیم. توجه داشته باشید که این دو اصطلاح جزء مهمترین اصطلاحات این فصل می‌باشند.

مفاهیم فیلد و رکورد

در این قسمت ابتدا از یک مثال شروع می‌کنیم. حتماً در اکثر کتابخانه‌ها با سیستم طبقه‌بندی اطلاعات کتابها روبه‌رو شده‌اید. این سیستم شامل قفسه‌هایی است که در هر کتبی از این قفسه‌ها، مجموعه‌ای از کارتها که بر روی آن اطلاعات هر کتاب نوشته شده قرار دارد. هر کارت شامل نام کتاب،

نویسنده ، مترجم ، موضوع واطلاعات دیگر کتاب است. شکل زیر یک مجموعه از کارتها را با اطلاعات آنها نشان می‌دهد.



شکل ۲-۲.

چند کارت مشخه کتاب.

در هر کتو از برگه دان کتابخانه، صدها برگه ازاین مشخصات وجود دارد. به هریک ازاین برگه‌ها در اصطلاح پایگاه داده‌ها ، یک رکورد (Record) می‌گویند و به هر کتو یک جدول گفته می‌شود که از تعداد زیادی برگه یا رکورد تشکیل شده است. حال روی هر برگه یا رکورد دقیق‌تر نگاه کنید. در مثال بالا هر رکورد شامل مواردی است که از نام کتاب، نویسنده ، مترجم وموضوع تشکیل شده است. به هریک از عناوین این برگه یا رکورد، یک فیلد (Field) می‌گوئیم. پس هر رکورد از چندین فیلد تشکیل شده است. حال همین مثال را روی جدول اطلاعاتی پیاده می‌کنیم. در شکل ۲-۳ یک جدول اطلاعاتی می‌بینید که روی آن فیلدها و رکوردها مشخص شده است. اینجا نیز در هر رکورد موضوع فیلدها تکرار شده است.

موضوع	مترجم	نویسنده	نام کتاب
کامپیوتر	احمد زاده	مارشال کنز	اینترنت
کامپیوتر	محمدی	Robins Tone	Web
کامپیوتر	آیت الله زاده شیرازی	Todd Foley	Windows NT
کامپیوتر	-	مهندس جعفر نژاد قمی	ایجاد بانکهای اطلاعاتی
کامپیوتر	مهندس علیرضا زارع پور	استیون هولز نر	ویژوال بیسیک ۳

شکل ۲-۳

یک جدول اطلاعاتی با داده های آن.

رکوردهای هر جدول می‌تواند شامل هر نوع اطلاعات گروه‌بندی شده باشد. همان طور که در مثال قبل دیدید هر رکورد شامل چهارفیلد طبقه‌بندی شده بود. حال تصور کنید اگر داده‌های شما روی جدول براساس یک فیلد ، مثلاً نام افراد یا نام کتاب مرتب شود ، دسترسی به آن چقدر آسان خواهد بود.

اشاره‌گر رکورد: هر جدول (Table) دارای یک اشاره‌گر می‌باشد که نشان می‌دهد رکورد جاری جدول کدام رکورد است. مثلاً در جدول زیر اشاره‌گر جدول روی پنجمین رکورد اطلاعاتی قرار دارد.

موضوع	مترجم	نویسنده	نام کتاب
کامپیوتر	احمد زاده	مارشال کنز	اینترنت
کامپیوتر	محمدی	Robins Tone	Web
کامپیوتر	آیت الله زاده شیرازی	Todd Foley	Windows NT
کامپیوتر	-	مهندس جعفر نژاد قمی	ایجاد بانکهای اطلاعاتی
کامپیوتر	مهندس علیرضا زارع پور	استیون هولزبر	ویژوال بیسیک ۳

شکل ۴-۲.

اشاره گر رکورد اطلاعاتی.

در مثال بالا رکورد جاری جدول به عنوان رکورد پنجم شناخته می شود و هر عملیاتی را (از قبیل خواندن ، تغییر ، حذف) که روی رکورد این جدول انجام می دهیم، روی رکورد جاری جدول ، یعنی رکورد پنجم اعمال می گردد. برای دسترسی به رکوردهای مختلف بایستی این اشاره گر را حرکت داد تا روی رکورد مورد نظر قرار گیرد و سپس عملیات مورد نظر را بر روی آن رکورد اعمال کرد.

مفاهیم EOF و BOF

در هر جدول اطلاعاتی دو مکان در ابتدا و انتهای جدول قرار دارد که جدا از رکوردهای جدول می باشد. در صورتی که اشاره گر رکوردها روی آخرین رکورد اطلاعاتی قرار داشته باشد و دستور حرکت به جلو برای اشاره گر صادر گردد ، اشاره گر رکورد روی مکان EOF (End Of File) قرار می گیرد و هنگامی که اشاره گر رکوردها روی اولین رکورد اطلاعاتی قرار داشته باشد و دستور حرکت به عقب برای اشاره گر صادر شود، اشاره گر رکورد روی مکان BOF (Begining Of File) قرار می گیرد. باید توجه داشت که مکانهای BOF و EOF حاوی اطلاعات نمی باشند و در صورتی که عملیاتی را روی آنها انجام دهید خطای No Current Record ظاهر می گردد.

مفهوم INDEX

برای آشنایی با مفهوم ایندکس و کاربرد آن از یک مثال استفاده می کنیم. در یک لغت نامه مجموعه زیادی از کلمات همراه با معنی آنها به ترتیب الفبایی آورده شده است. برای هر حرف از حروف الفبا در لغت نامه ها معمولاً از نشانه ای در اول آن بخش استفاده می شود که نشان می دهد آن بخش، ابتدای شروع کلماتی است که با این حرف شروع می شود. این نشانه برای این منظور قرار داده شده است که در هنگام جستجوی یک لغت ، ابتدا با توجه به حرف اول آن به بخش مورد نظر در لغت نامه مراجعه گردد و سپس با توجه به بقیه حروف ، آن لغت پیدا شود.

در پایگاه داده ها نیز می توان از این خاصیت استفاده کرد. فرض کنید جدولی داریم که لغات و معنی آنها را نگهداری می کند. در صورت نامنظم بودن لغات در این جدول برای یافتن یک لغت باید از ابتدای جدول شروع به جستجو کنیم و رکورد به رکورد، فیلد مورد نظر را با کلمه مورد نظر خود

مقایسه نماییم. می‌بینید که این کار وقت بسیار زیادی را می‌گیرد. این جدول همانند لغت نامه‌ای است که کلمات آن هیچ نظم‌ی ندارند و برای یافتن یک لغت در آن باید از ابتدا تا انتهای آن را جستجو کنیم، بنابراین برای منظم کردن داده‌ها جهت جستجوی سریع به نظم‌ی احتیاج داریم که به آن ایندکس می‌گویند.

در مثال جدول کلمات اگر فیلد نام کلمه، ایندکس شده باشد برای جستجوی یک لغت، اشاره‌گر جدول مستقیماً به روی رکوردی منتقل می‌شود که حرف اول آن با حرف اول کلمه مورد نظر یکسان است و همین‌طور این عمل روی حروف بعد انجام می‌گیرد تا رکورد مورد نظر پیدا شود. پس از آشنایی با مقدمات پایگاه داده نحوه طبقه‌بندی اطلاعات را در یک پایگاه داده بررسی می‌کنیم.

ایجاد یک پایگاه داده

به این نکته توجه داشته باشید که در ایجاد یک پایگاه داده، طراحی مناسب نقش بسیار مهمی را ایفا می‌کند بنابراین پیشنهاد می‌کنیم قبل از اینکه پایگاه داده خود را ایجاد کنید مراحل طراحی آن را روی کاغذ انجام دهید و سپس حاصل کار خود را وارد دستگاه نمایید.

مراحل طراحی یک پایگاه داده

۱- ابتدا نیازهای خود را به صورت لیستی تهیه کنید. این لیست می‌تواند شامل تمام خواسته‌های شما از پایگاه داده باشد. مثلاً می‌خواهید:

- اطلاعات مربوط به افراد و سابقه آنها نگهداری گردد.
- این اطلاعات به صورت منظم طبقه‌بندی شود.
- هر فرد در گروه خاصی از افراد مشخص گردد.
- سابقه کاری هر فرد تا مدت ۳ سال بایگانی شود.

در این مرحله، ترتیب در خواسته‌ها اهمیتی ندارد یعنی هر خواسته‌ای که دارید به لیست خود اضافه کنید تا جایی که فکر می‌کنید دیگر کافی است.

۲- در این مرحله، باید خواسته‌ها پالایش شود. این مرحله از اهمیت زیادی برخوردار است پس باید بیشترین دقت را روی این مرحله داشته باشید. کاری که در این مرحله باید انجام دهید مرتب کردن خواسته‌هاست که به سه دسته تقسیم می‌شود:

الف: تجزیه و تحلیل داده‌ها

ب: نوع داده‌ها

ج: یافتن روابط بین داده‌ها

در قسمت الف کار شما روی ورودیهاست یعنی خلاصه کردن ورودیها براساس خواسته‌ها. در مرحله اول مثال فوق ابتدا نیازهای خود را جمع کردیم. حال باید درخواستها دقیق شویم تا اطلاعاتی را که

باید نگهداری شود بیابیم و سپس آنها را به ترتیب لیست کنیم . مثلاً می خواهیم نام اشخاص ، تلفن ، آدرس و کد شناسایی افراد را نگهداری کنیم. پس لیستی به ترتیب زیر تهیه می کنیم.

شماره شناسایی:

نام:

آدرس:

تلفن:

حال باید برای هر فیلد یک نام لاتین انتخاب کنیم که بیانگر نوع داده های این فیلدها باشد. مثلاً برای نام از (Name) ، برای شماره شناسایی از (ID) ، برای آدرس از (Address) و برای تلفن از (Tel) استفاده می نماییم.

حال در قسمت ب باید نوع هر فیلد را مشخص کنیم. مثلاً فیلد ما از نوع عددی است یا متنی و یا تعداد کاراکترهای آن حداکثر چقدر می باشد و دیگر مشخصاتی که متناسب با نوع فیلد تغییر می کند. در قسمت ج باید روابط بین فیلدها و جداول را بیابیم. مثلاً اگر بنا به نیاز پایگاه داده به چند جدول بافیلدهای مرتبط نیاز بود، رابطه بین آنها را مشخص می کنیم.

ایجاد پایگاه داده‌های اکسس از طریق ویژوال بیسیک

پس از اینکه با اصول کلی پایگاه‌های داده آشنا شدیم و مراحل طراحی پایگاه داده خود را گذرانیدیم، کار عملی با پایگاه داده را شروع می‌کنیم. اولین قدم ایجاد این پایگاه داده است. انتخاب بانک اطلاعاتی اکسس به این دلیل می‌باشد که این پایگاه داده، پایگاه داده استاندارد شرکت مایکروسافت بوده و پسوند این فایلها Mdb. است که مخفف Microsoft Database می‌باشد. هر یک از پایگاه‌های داده مانند استانداردهای XBase و یا دیگر پایگاه‌های داده، ویژگیهای بخصوصی دارند که به نحوه طبقه‌بندی اطلاعات، نحوه ذخیره سازی و محافظت و موارد دیگر مربوط می‌شود. فایل‌های Mdb با برنامه پایگاه داده‌های اکسس شناخته شد و همان طور که اشاره شد، یک پایگاه داده رومیزی می‌باشد. با تلفیق دو نرم‌افزار کاربردی اکسس و ویژوال بیسیک ما می‌توانیم برنامه‌های قدرتمند خود را تحت ویندوز با هر یک از قابلیت‌های آنها طراحی کنیم و قدرت اکسس را خارج محیط پایگاه داده رومیزی آن در دست بگیریم.

قبل از اینکه وارد کار عملی با پایگاه‌های داده اکسس شویم ساختار کلی پایگاه‌های داده اکسس را بررسی می‌کنیم. همان طور که گفتیم هر پایگاه داده ساختاری مربوط به خود دارد. مثلاً در پایگاه‌های داده فاکس پرو هر پایگاه داده که با پسوند dbf. مشخص شود، فقط از یک جدول تشکیل شده است، یعنی به ازای هر جدول باید یک فایل dbf ساخته شود. اما در اکسس این امر متفاوت است. در محیط خود مایکروسافت اکسس هر بانک اطلاعاتی از قسمتهای زیر تشکیل شده است.

- Table: شامل هر تعداد جدول مورد استفاده در یک پروژه پایگاه داده است که قابل دسترسی در محیط اکسس و یا خارج از محیط آن می‌باشد.

- Query: شامل مجموعه سؤالاتی که قابل دسترسی در محیط اکسس و یا خارج از محیط آن می‌باشد.

- Form: فرمهای واسط کاربر می‌باشد که فقط در محیط اکسس اجرا می‌شوند.

- Report: گزارشاتی است که می‌توان از اطلاعات جداول گرفت و فقط در محیط اکسس قابل اجرا می‌باشد.

- Macro: مجموعه عملیات و دستورات را می‌توان در یک ماکرو وارد نمود و سپس اجرا کرد. این قسمت نیز فقط در محیط اکسس قابل دسترسی است

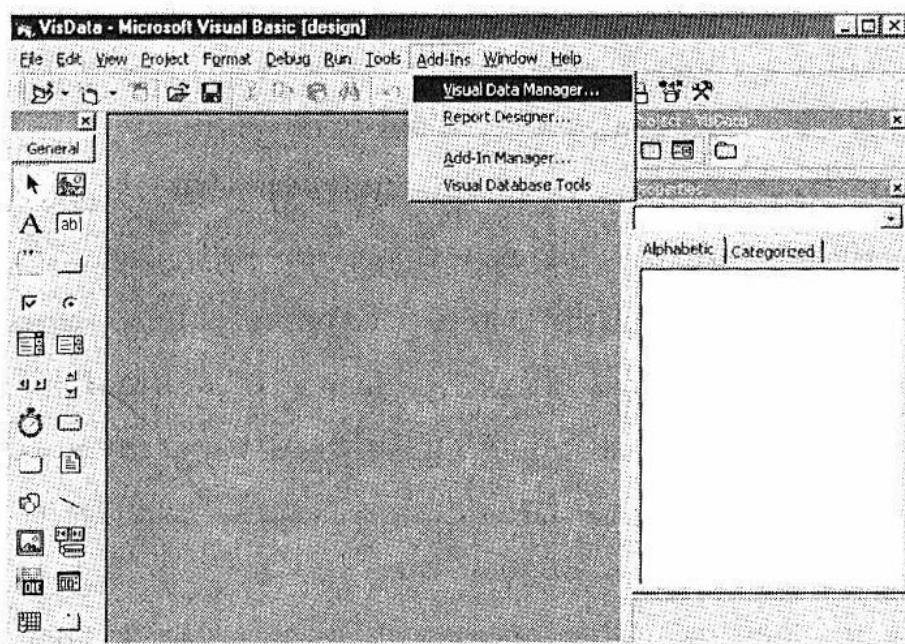
• Module : مجموعه کدهای برنامه نویسی به زبان اکسس است که فقط در محیط اکسس قابل استفاده می باشد.

قسمتهایی از پایگاه داده های اکسس که در ویژوال بیسیک مورد استفاده هستند همان دو قسمت اول می باشد و بقیه قسمتها فقط در خود محیط اکسس قابل دسترسی است. در ادامه ، طریقه ایجاد یک پایگاه داده اکسس وجداول مربوط به پایگاه داده آن را بررسی می کنیم.

برای ایجاد یک پایگاه داده اکسس چند راه وجود دارد. شما می توانید این کار را هم از طریق میکروسافت اکسس و هم از طریق ویژوال بیسیک انجام دهید. در نسخه های مختلف ویژوال بیسیک این کار تقریباً به صورت مشابه صورت می گیرد و شما بدون نیاز به میکروسافت اکسس می توانید آن را ایجاد کنید. در ویژوال بیسیک ۴ از طریق Data Manager و در نسخه های بعدی به وسیله Visual Data Manager می توان از پایگاه های داده استفاده کرد که در این کتاب ایجاد پایگاه داده را به وسیله Visual Data Manager بررسی می کنیم.

ایجاد پایگاه داده به وسیله Visual Data Manager

برای انجام هر گونه عملیات روی پایگاه داده در ویژوال بیسیک می توان از برنامه Visual Data Manager استفاده کرد. برای اجرای این برنامه در ویژوال بیسیک از منوی Add-Ins گزینه Visual Data Manager را انتخاب کنید

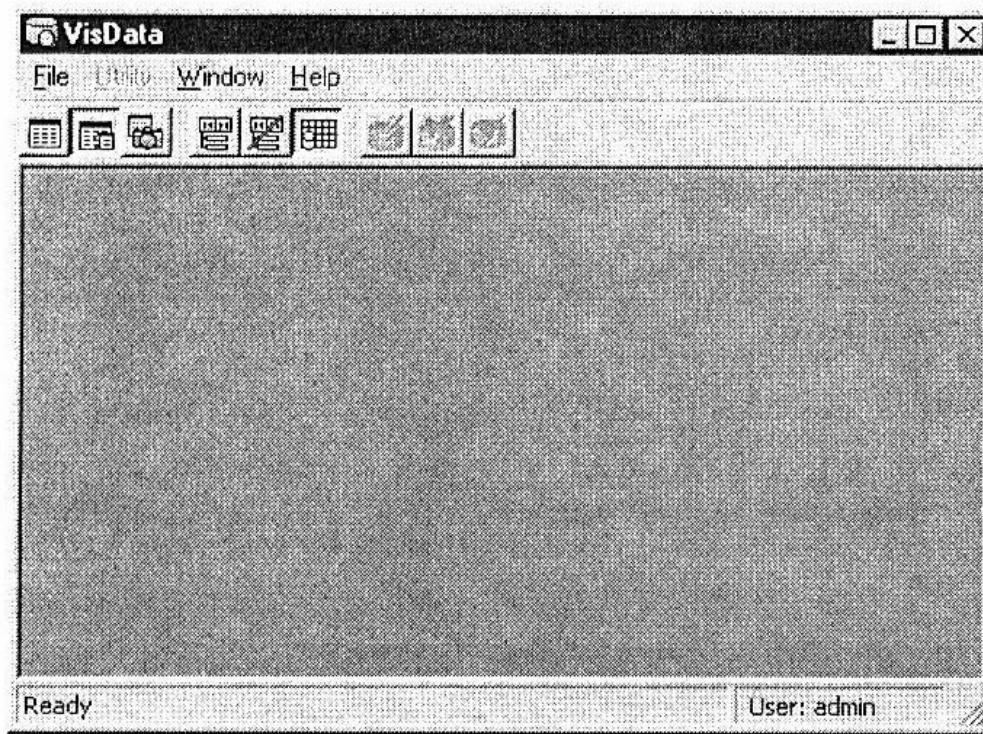


شکل ۱-۳.

انتخاب

Visual Data Manager
از منوی Add-Ins جهت اجرای
برنامه.

در صورتی که برای اولین بار این برنامه را اجرا می کنید پیغامی در مورد فایل حفاظت System.MD دریافت خواهید کرد. برای ادامه دکمه NO را انتخاب کنید تا برنامه Vis Data اجرا شود.



شکل ۲-۳.

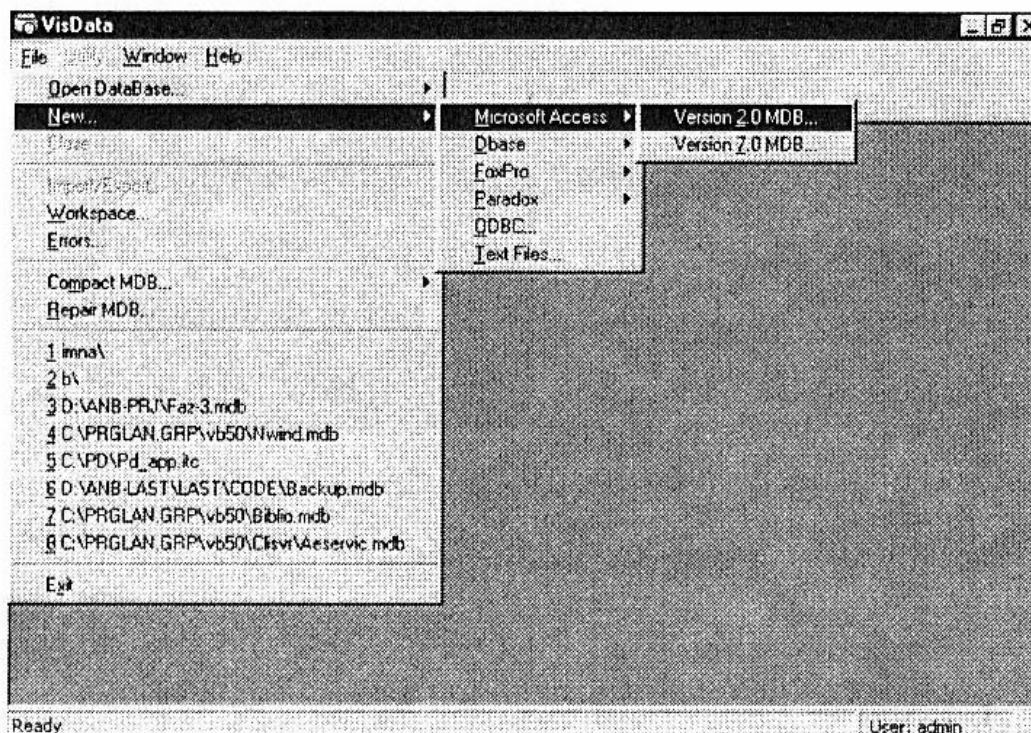
پنجره اصلی برنامه

Visual Data Manager

اکنون می خواهیم یک پایگاه داده ایجاد کنیم. چون در ویژوال بیسیک می توان از انواع پایگاههای داده از قبیل فاکس پرو ، پاراداکس ، درایو ODBC و غیره استفاده کرد ، در برنامه VisData نیز این امکان قرار داده شده است که هر یک از این جدولها ایجاد گردد اما چون پایگاه داده استاندارد میکروسافت اکسس می باشد روش ایجاد پایگاه داده اکسس را توضیح می دهیم.

ایجاد پایگاه داده اکسس در ویژوال بیسیک

۱- همانند شکل ۳-۳ منوی File | New | Microsoft Access | Version 7.0 MDB را انتخاب کنید.



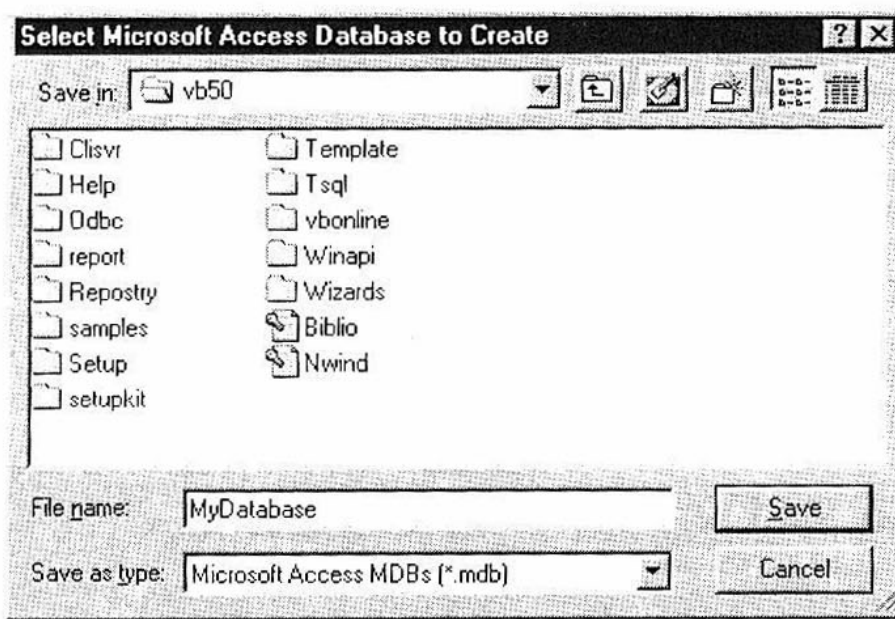
شکل ۳-۳.

ایجاد یک پایگاه جدید

اکسس از طریق منوی File.

در این منو شما می توانید پایگاه داده اکسس را با دو ویرایش 2.0 یا 7.0 ایجاد کنید. توجه داشته باشید که پایگاه داده اکسس با ویرایش 7.0 فقط در سیستم عامل سی و دو بیتی ویندوز قابل دسترسی است ولی از ویرایش 2.0 می توان در ویندوز شانزده بیتی نیز استفاده کرد.

۲- در قسمت بعد باید مسیر و نام پایگاه داده مورد نظر را جهت ذخیره سازی مشخص کنید. در این قسمت برای مثال Mydatabase را در قسمت File Name: تایپ کرده و سپس روی دکمه فرمان Save کلیک می نمایم.



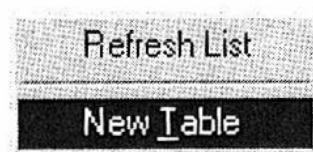
شکل ۳-۴

ذخیره پایگاه داده جدید در مسیر مشخص.

آنگاه برنامه VisData دو پنجره را برای پایگاه داده ای که ایجاد کرده اید در اختیار قرار می دهد که یکی از آنها Database Windows و دیگری SQL Statement می باشد.

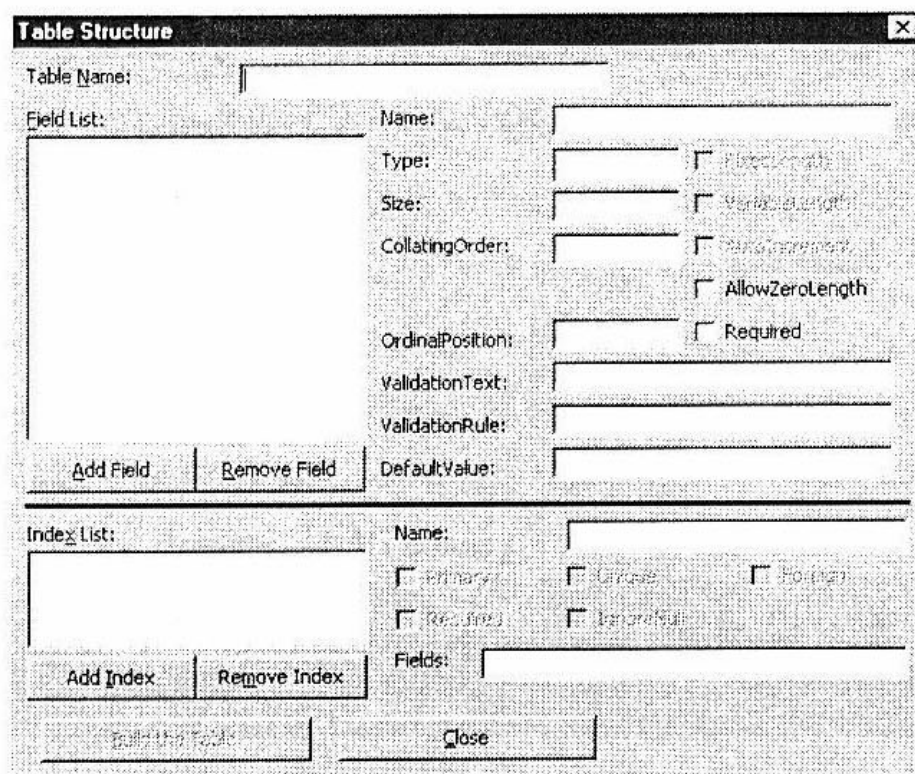
۳- در پنجره Database Window دکمه راست ماوس را کلیک کنید تا منویی روی این پنجره باز شود. از این منو گزینه New Table را برای ایجاد جدول جدید انتخاب نمایید.

شکل ۳-۵



ایجاد جدول جدید.

سپس پنجره Table Structure باز می شود .

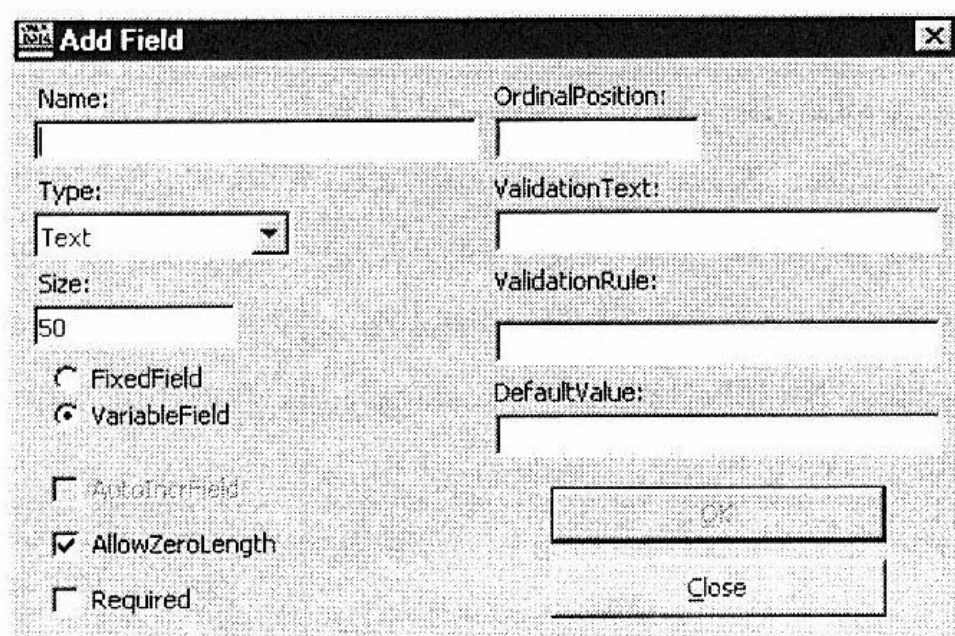


شکل ۳-۶

پنجره Table Structure
برای تعیین ساختار جدول جدید.

۴- پنجره Table Structure شامل دو قسمت می باشد که یکی مربوط به فیلدهای جدول و دیگری مربوط به ایندکسهای آن می باشد.

۵- در قسمت Table Name نام جدول مورد نظر را مانند MyTable وارد می کنیم. برای ایجاد فیلدهای این جدول باید دکمه فرمان Add Field را انتخاب کنیم تا پنجره Add Field باز گردد.



شکل ۳-۷

ایجاد فیلدهای یک جدول
در پنجره Add Field.

۶- حال تک تک مشخصات فیلدهای این جدول را در پنجره Add Field وارد می کنیم. این پنجره از قسمتهای مختلفی تشکیل شده است که در زیر آنها را بررسی می نماییم.

Name: در این قسمت نام فیلد وارد می گردد. نام فیلد را معمولاً طوری انتخاب می کنند که به اطلاعات آن فیلد مربوط باشد. چون ارتباط شما با فیلد از طریق نام فیلد صورت می گیرد، برای سهولت کار در انتخاب نام فیلد دقت کنید.

Type: در این قسمت نوع داده های فیلد مشخص می شود. داده ها به انواع مختلفی تقسیم می گردند که بسته به نوع داده های یک فیلد انتخاب می شوند. مثلا در فیلدی که قرار است نام افراد ذخیره شود، نوع داده فیلد از نوع متنی می باشد و یا اگر قرار است تلفن افراد نگهداری شود، نوع داده فیلد از نوع عددی خواهد بود. انتخاب نوع دقیق در این قسمت از این جهت اهمیت دارد که حجم پایگاه داده بسته به انواع مختلف تغییر می کند زیرا هر نوع ، حافظه مشخصی را اشغال می نماید؛ بنابراین اگر برای فیلد، نوع مناسبی را انتخاب کنیم ، تأثیر زیادی در حجم کمتر حافظه و در نتیجه سرعت دسترسی بیشتر خواهیم داشت. جدول ۱-۳ انواع داده ها را در پایگاه های اکسس نشان می دهد.

جدول ۱-۳: انواع داده ها در پایگاه داده های اکسس

نوع داده	مقدار حافظه مصرفی	موارد استفاده
Text	به ازای هر کاراکتر یک بایت اختصاص داده می شود.	این نوع داده از نوع متنی می باشد و چون بیشترین کاربرد را در فیلدها داراست به صورت پیش فرض انتخاب شده است. در صورتی که از این نوع داده بخواهید استفاده کنید ، باید حداکثر کاراکترهای فیلد آن را نیز وارد کنید.
عددی		این نوع داده ها به چند قسمت تقسیم می شود که در جدول ۲-۳ آنها را مورد بررسی قرار می دهیم.
Memo		یک فیلد توضیحی است و از نظر اندازه نا محدود می باشد و اطلاعات بیشتر، حجم بیشتری از دیسک سخت را اشغال می کند.
Boolean	یک بایت	این نوع داده برای فیلدهایی منطقی می باشد که دارای مقدار Yes/No و یا True/False می باشند. در صورتی که مقدار آن صفر باشد، ارزش منطقی آن برابر False یا No است و اگر مقدار آن غیر صفر باشد ، ارزش منطقی آن True یا Yes می باشد.
Currency		این نوع داده ، برای نگهداری داده با واحد پول استفاده می گردد و یک نوع داده عددی حساب می شود که فرمت ثابتی را داراست و تعداد اعشار آن چهار رقم می باشد. از این نوع داده، معمولاً برای نگهداری مبلغها استفاده می کنند.
Date/Time		این نوع داده برای نگهداری زمان یا تاریخ مورد استفاده قرار می گیرد.

جدول ۲-۳: انواع داده های عددی در پایگاههای داده اکسس

نوع داده	تعداد اعشار	حافظه مصرفی	دامنه ارقام
Integer	ندارد	۲ بایت	-۳۲۷۶۸ تا +۳۲۷۶۸
Long	ندارد	۴ بایت	-۲۱۴۷۴۸۳۶۴۸ تا +۲۱۴۷۴۸۳۶۴۸
Byte	ندارد	۱ بایت	۰ تا ۲۵۵
Single	۷ رقم	۴ بایت	-۳.۴x E ۳۸ تا +۳.۴x E ۳۸
Double	۱۵ رقم	۸ بایت	-۱.۷۹۷x۱۰.E۳۰.۸ تا +۱.۷۹۷x۱۰.E۳۰.۸
Currency	۴ رقم	۴ بایت	-۹۲۲۳۳۷۲.۰۳۶۸۵۴۷۷.۵۸۰.۸ تا +۹۲۲۳۳۷۲.۰۳۶۸۵۴۷۷.۵۸۰.۸

Size : در این قسمت اندازه فیلد مشخص می شود. در صورتی که نوع فیلد متنی باشد این مقدار برابر تعداد کاراکترهای ممکن و در صورتی که فیلد از نوع عددی باشد ، این مقدار برابر با تعداد بایت حافظه مصرفی است.

VariableField و FixedField : در صورتی که نوع داده متنی باشد این قسمت فعال است و تعیین می کند که اندازه فیلد قابل تغییر باشد یا خیر.

AllowZeroLength : اگر این قسمت را انتخاب کنید، باید حداقل یک کاراکتر در فیلد وارد شود. این خصوصیت برای فیلدهای نوع متن قابل استفاده است.

Validation Rule : شرطی است که پس از وارد کردن مقدار در فیلد این شرط ، کنترل می شود و اگر کنترل، نتیجه منفی داشت، خطایی از طرف برنامه داده خواهد شد و اجازه عبور از آن فیلد به فیلد دیگر داده نمی شود.

Validation Text : در صورتی که شرط Validation Rule با داده فیلد صدق نکرد متن داخل این خصوصیت به عنوان خطا ظاهر می شود.

Default Value : مقدار پیش فرضی است که فیلد قبل از مقدار دهی به خود می گیرد.

حال در جدول MyTable می خواهیم فیلد هایی را وارد کنیم که اطلاعات افرادی در آن ذخیره شود. این اطلاعات شامل نام ، تلفن ، آدرس و سن فرد می باشد. برای فیلد نام افراد ، نوع متنی را انتخاب می کنیم و اندازه حداکثر فیلد را همان ۵۰ کاراکتر قرار می دهیم. حال دکمه فرمان OK را انتخاب می کنیم.

Add Field

Name: OrdinalPosition:

Type: ValidationText:

Size: ValidationRule:

FixedField
 VariableField
 AutoIncrField
 AllowZeroLength
 Required

Default Value:

شکل ۸-۳.

خصوصیات فیلد Name.

پس از این کار، فیلد مورد نظر در جدول قرار می گیرد. برای فیلد تلفن، نوع عددی از نوع Long را انتخاب نموده و مقدار اولیه آن را به صفر تنظیم می کنیم.

Add Field

Name: OrdinalPosition:

Type: ValidationText:

Size: ValidationRule:

FixedField
 VariableField
 AutoIncrField
 AllowZeroLength
 Required

Default Value:

شکل ۸-۳.

خصوصیات فیلد Tel.

برای فیلد آدرس باز هم از نوع متنی استفاده کرده و اندازه حداکثر فیلد را ۱۰۰ کاراکتر قرار می دهیم.

Add Field

Name: OrdinalPosition:

Type: ValidationText:

Size: ValidationRule:

FixedField
 VariableField
 AutoIncrField
 AllowZeroLength
 Required

Default Value:

شکل ۹-۳.

خصوصیات فیلد Address.

برای فیلد سن فرد از نوع عددی و (Byte) استفاده می‌کنیم و شرطی را قرار می‌دهیم که سن، باید از ۲۰ سال بیشتر و از ۸۰ سال کمتر باشد.

شکل ۳-۹.
خصوصیات فیلد Age.

۷- در صورتی که بخواهیم برای فیلدها، ایندکسی تعریف کنیم در قسمت پایین پنجره Table Structure دکمه Add Index را انتخاب می‌کنیم تا پنجره مربوط به آن باز گردد.

شکل ۳-۱۰.
پنجره اضافه کردن ایندکس به جدول.

همان طور که می‌بینید، این پنجره شامل لیستی می‌باشد که تمام فیلدها در آن نمایش داده شده است. حال هر فیلدی را که می‌خواهید روی آن ایندکس داشته باشید، انتخاب نموده و در قسمت Name نام ایندکس را وارد می‌کنیم.

شکل ۳-۱۱.
ایجاد یک ایندکس برای جدول MyTable.

در مورد ایندکس در فصل قبل مطالبی را ذکر کردیم. در اکسس دو نوع ایندکس می‌توانیم تعریف کنیم. یکی Duplicated OK. و دیگری No Duplicated که مربوط به تکرار رکوردهای فیلد می‌شود. اگر Duplicated OK باشد، فیلد ایندکس شده می‌تواند در رکوردهای مختلف، مقادیر تکراری را قبول کند و اگر No Duplicated باشد، مقادیر تکراری در فیلد ایندکس شده قابل قبول نخواهد بود. در پنجره Add Index در صورتی که قسمت Unique را انتخاب کنید به معنی No Duplicated است و اگر انتخاب نشده باشد، فیلد ایندکس شده Duplicated OK است.

۸- پس از اینکه تمام تنظیمات را انجام دادید، دکمه فرمان Build The Table را برای ایجاد جدول، انتخاب کنید.

۹- روی این جدول در پنجره Database Window دو بار کلیک کنید تا جدول MyTable نمایش داده شود.

	Name	Tel	Address	Age
*				

شکل ۱۲-۳.

نمایش جدول MyTable

اصلاح جدول ایجاد شده

اگر بخواهیم ساختار یک جدول را تغییر دهیم و یا فیلدها و ایندکسهای آن را اضافه و یا اصلاح کنیم، باید روی نام جدول در Database Window دکمه راست ماوس را کلیک کنیم و از منوی ایجاد شده، گزینه Design را انتخاب نماییم.

آشنایی با ToolsBar برنامه VisData

در ToolsBar برنامه VisData، موارد زیر مشاهده می‌شود.

شکل ۱۳-۳.

نمایش ToolBar در برنامه VisData.



در سه شکل اول، نوع باز کردن لیست داده (RecordSet) مشخص می‌شود. اولین شکل مربوط به باز شدن لیست داده به صورت جدولی و سومین شکل مربوط به باز کردن لیست داده به صورت Dynaset می‌باشد و سومین شکل، مربوط به باز کردن لیست داده به صورت SnapShot می‌باشد.

سه شکل دوم، نوع نمایش اطلاعات را مشخص می‌کند که یا به صورت لیستهای سطر و ستون می‌باشد و یا به صورت فرمهای اطلاعات می‌باشد.

توجه: با پنجره SQL Statement در فصل هشتم آشنا خواهید شد.

اصول استفاده از پایگاه داده‌ها در ویژوال بیسیک

در ویژوال بیسیک با دو روش می‌توان به پایگاه داده دسترسی پیدا کرد. اولین روش استفاده از ابزار داده می‌باشد. با استفاده از این ابزار می‌توان اشیای لازم برای دسترسی به پایگاه داده را ایجاد کرد و از آنها در بقیه ابزارها و یا کدهای برنامه استفاده نمود. روش دوم ساختن تمام اشیاء مورد نیاز برای دسترسی به پایگاه داده‌ها توسط کدهای برنامه نویسی می‌باشد. معمولاً کار با ابزارها در فرم ساده است و به راحتی می‌توان عملیات اطلاعاتی را توسط کد روی آنها انجام داد اما این در صورتی است که جدولها و سؤالها در پایگاه داده قبلاً ایجاد شده باشد و عملیات اطلاعاتی روی رکورد ها شامل خواندن ، اصلاح ، حذف ، جستجو و غیره مورد نظر باشد؛ در غیر این صورت باید از روش دوم استفاده نمود.

در این فصل قصد داریم تا با اصول کلیدی استفاده از پایگاه داده‌ها در ویژوال بیسیک آشنا شویم و در ادامه، ایجاد مرحله به مرحله یک فرم اطلاعاتی ساده را بررسی کنیم.

تعریف لیست داده (RecordSet)

لیست داده‌های موجود در یک جدول (Table) را RecordSet می‌نامند. در واقع لیست داده‌ها به مجموعه رکوردهای یک جدول گفته می‌شود. این لیست، در حالت عادی شامل تمام رکوردهای یک جدول است و می‌تواند از طریق برنامه‌نویسی کنترل و یا محدود گردد. هر عملیاتی که قرار است روی رکوردهای یک جدول انجام شود از طریق یک لیست داده انجام می‌شود که شامل رکوردهای آن جدول باشد. این عملیات شامل اضافه کردن رکورد به جدول ، حذف یک رکورد ، تغییر ، جستجو و غیره می‌باشد.

انواع لیست داده

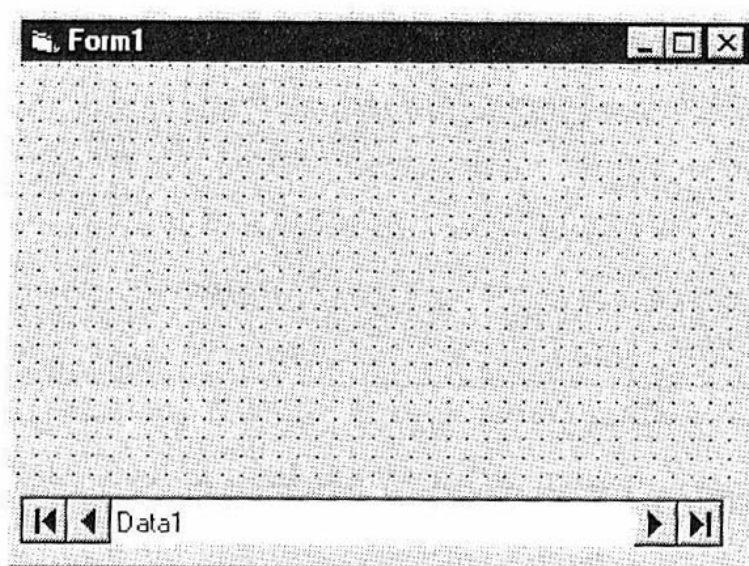
به سه شیوه می‌توان یک لیست داده (RecordSet) را باز کرده و از آن استفاده نمود. اهمیت این روشها در کاربرد هریک می‌باشد که در اینجا به طور کلی فقط از آنها نام می‌بریم و در فصلهای بعد آنها را بیشتر مورد بررسی قرار می‌دهیم.

الف) روش **Table**: از ویژگیهای این روش وجود ایندکس، جستجوی سریع به روش **Seek** و تغییر اطلاعات را می‌توان نام برد.

ب) روش **Dynaset**: دسترسی در این روش به وسیله **SQL** صورت می‌گیرد و برای جستجو در آن از روش **Find** استفاده می‌شود.

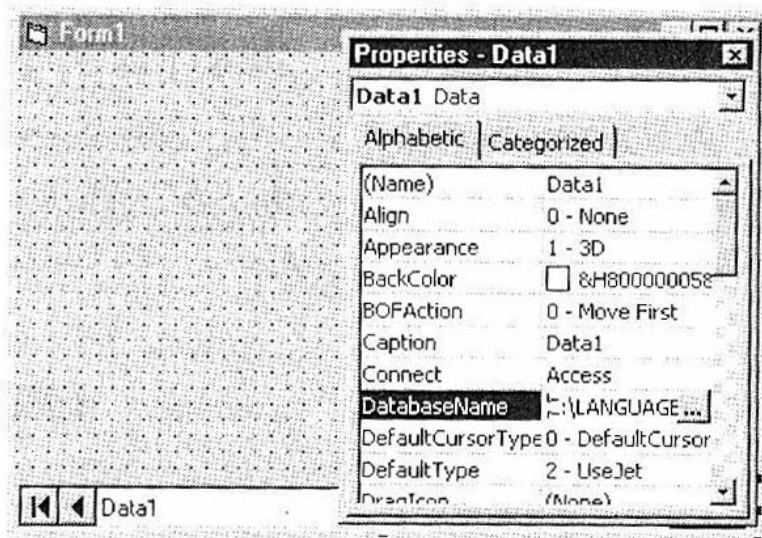
ج) روش **Snapshot**: یک لیست داده فقط خواندنی را در اختیار قرار می‌دهد که در حجم کم اطلاعات، سرعت بالایی را داراست.

در ادامه، شیوه ایجاد یک فرم اطلاعاتی ساده را به صورت مرحله به مرحله، بررسی می‌کنیم.
 ۱- از قسمت ابزارها، ابزار **Data** را انتخاب کرده و روی فرم خود آن را قرار دهید.



شکل ۴-۱.
 استفاده از ابزار **Data**.

۲- روی این ابزار رفته و کلید **[F4]** را برای دیدن خصوصیات آن فشار دهید.



شکل ۴-۲.
 ابزار داده در فرم و خصوصیات آن.

در جدول خصوصیات، خصوصیات متعددی وجود دارد که یک دسته از آنها مانند Visible ، Caption ، Enabled ، Font و غیره، مربوط به شکل ظاهری آن می باشد که این خصوصیات ها با خصوصیات ظاهری بقیه ابزارها یکسان است. دسته دیگر از خصوصیات مربوط به پایگاه داده می شود که در اینجا به توضیح چهار مورد اصلی از این مشخصات می پردازیم.

الف (Connect همان گونه که مشاهده می کنید در این قسمت می توانید نوع پایگاه داده خود را مشخص کنید. از این طریق می توان با پایگاههای داده غیر از اکسس نیز ارتباط داشته و از دادههای آنها نیز استفاده کرد. پیش فرض این خصوصیت، اکسس می باشد.

ب (DatabaseName در این قسمت باید مسیر و نام پایگاه داده مشخص شود. در اینجا برای مثال Biblio.Mdb را از مسیر برنامه ویزوال بیسیک انتخاب کنید.

ج (RecordsetType نوع باز شدن لیست داده باید در این قسمت مشخص شود. برای این مثال نوع Table را انتخاب نمایید.

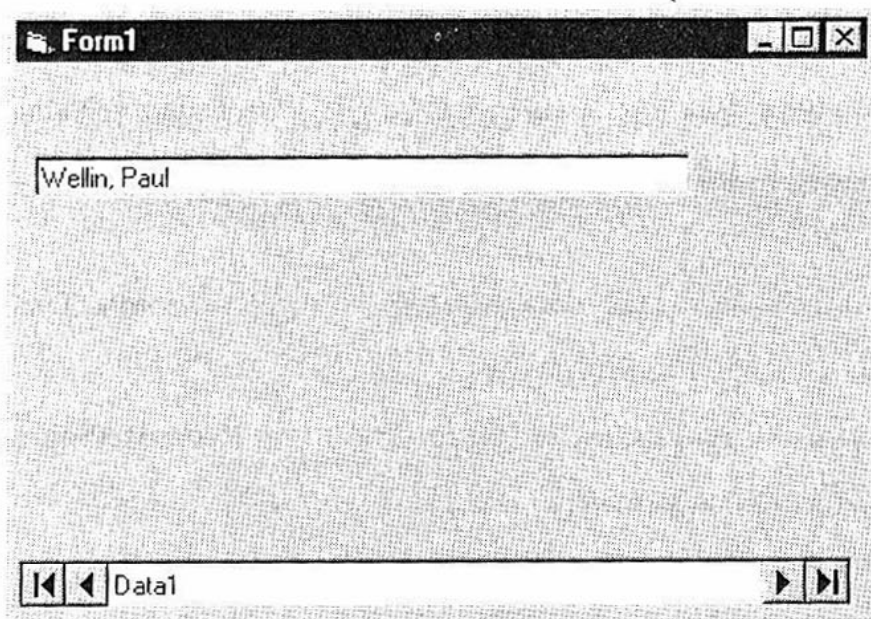
د (RecordsetSource در این خصوصیت منبع لیست داده باید مشخص شود. با باز کردن لیست این خصوصیت ، تمام جداول و سؤالهای پایگاه داده مشخص شده در خصوصیت DatabaseName ظاهر می گردد.

در آینده خواهیم دید که در این قسمت از کدهای SQL نیز می توان برای ایجاد لیست داده استفاده نمود.

برای مثال جدول Authors را از لیست این خصوصیت انتخاب کنید. حال اگر برنامه را اجرا کنید ، چیزی جز همان نوار Data پایین فرم ندارید و اطلاعات را نمی توانید مشاهده نمایید. علت این است که شما فقط جدولی را باز کرده و لیست داده مربوط به آن جدول را مقیم حافظه نموده اید ولی اقدامی برای نمایش اطلاعات نکرده اید؛ بنابراین مراحل بعد را ادامه می دهیم.

۳ - از قسمت ابزارها، یک جعبه متن (Text Box) را انتخاب کرده و آن را در فرم قرار دهید. همان طور که می دانید جعبه متن یک ابزار برای ورود ، اصلاح یا نمایش دادهها می باشد. حال به وسیله کلید [F4] پنجره خصوصیات جعبه متن را فعال نموده و به دو خصوصیت DataSource و DataField توجه کنید. این دو خصوصیت، رابط بین جعبه متن و ابزار Data در فرم است. در خصوصیت DataSource به روی پیکان رو به پایین این خصوصیت کلیک کنید. در این لیست نام تمام ابزارهای Data که در فرم

استفاده شده است، مشخص می‌گردد. چون ابزار Data در این فرم Data1 نام دارد، Data1 در لیست قرار گرفته است. پس مقدار این خصوصیت را همان Data1 قرار دهید. سپس لیست خصوصیت DataField را نگاه کنید. توجه داشته باشید که اگر خصوصیت DataSource را خالی گذاشته باشید، در این لیست هیچ چیز نمایش داده نمی‌شود اما اگر DataSource درست تنظیم شود، می‌بینید که لیست فیلدهای جدولی که در Data1 انتخاب کرده‌اید، به نمایش گذاشته شده است. یکی از این فیلدها را مانند Authors را انتخاب کنید. حال همه چیز برای اجرا آماده است پس کلید [F5] را برای اجرا فشار دهید.

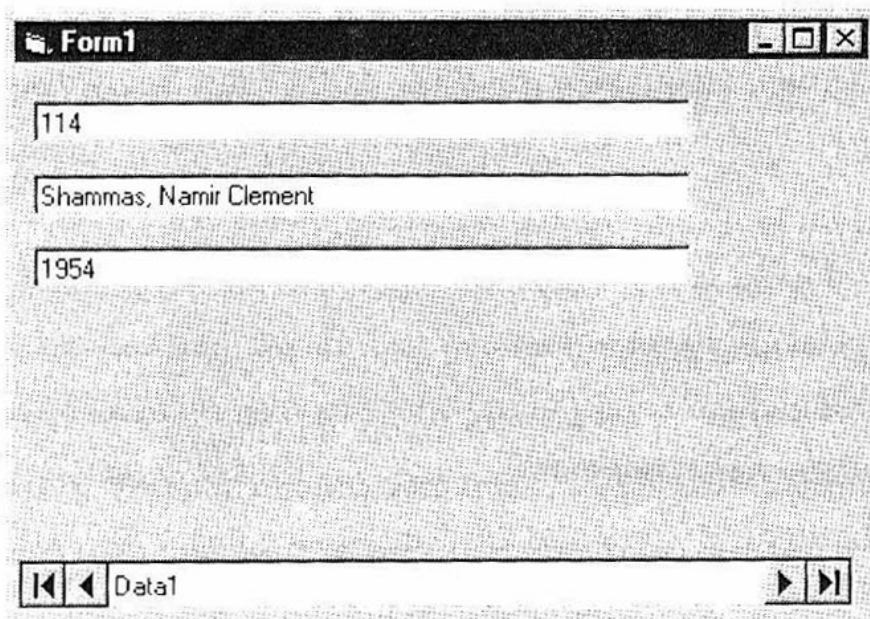


شکل ۳-۴.

مقیدسازی جعبه متن به ابزار داده در فرم

مشاهده می‌کنید که داده‌های درون فیلد انتخاب شده در جعبه متن ظاهر می‌شود. حال با زدن کلیدهای کنترلی روی Data1 می‌توانید روی رکوردها حرکت کنید و اطلاعات را به ترتیب مشاهده نمایید.

حال می‌توانید به تعداد فیلدهای جدول درون Data1 از جعبه متن، برای نشان دادن آن فیلدها استفاده کنید.

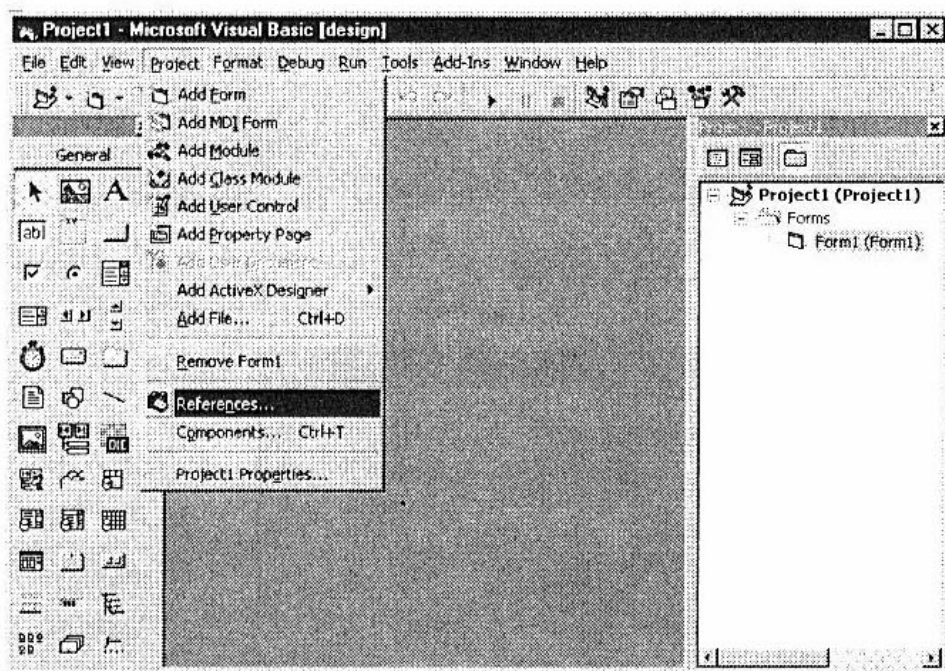


شکل ۴-۴.

مقید سازی جعبه متن به ابزار داده در فرم

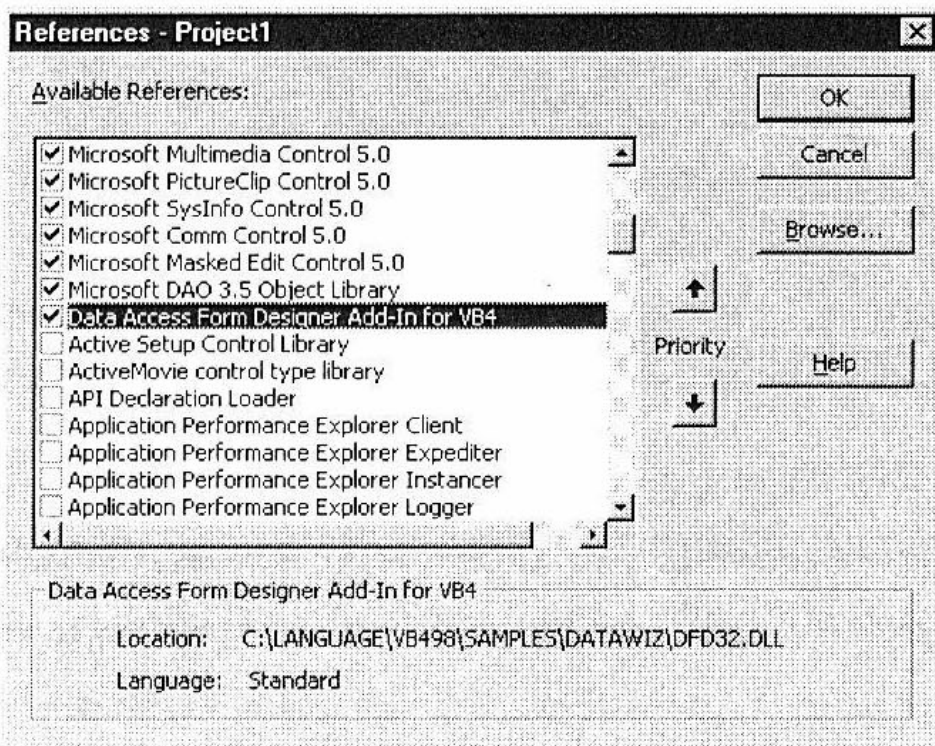
جعبه متن عمومی ترین ابزار برای ورود ، تغییر و نمایش داده ها می باشد. ابزارهای دیگری نیز مانند Table برای نمایش داده ها وجود دارد. در پایان این فصل از روش ساخت فرم شامل پایگاه داده به صورت ویزارد استفاده می کنیم. از نسخه ۴ به بعد ویژوال بیسیک، امکان جدیدی به آن اضافه شده است که طراح فرم اطلاعات نام دارد و به سادگی می توان از آن برای ایجاد یک فرم اطلاعاتی استفاده کرده و از تمام خواص پایگاه داده آن بهره گرفت.

برای فعال کردن این منو در قسمت Add-Ins باید ابتدا مرجع این ابزار را فعال سازیم، پس از منوی Projects قسمت Reference را انتخاب می کنیم و در لیست، Data Access Form Designer Add-Ins for VB را فعال می سازیم.



شکل ۴-۵

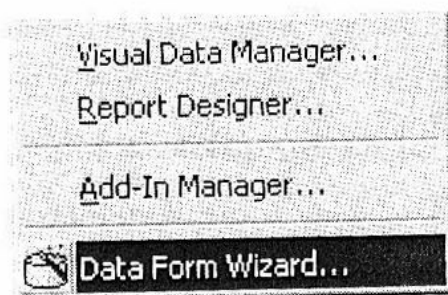
انتخاب Reference از منوی Project



شکل ۴-۶

فعال کردن مرجع برای Data Access Form Designer

حال از منوی Add-Ins گزینه Add-in Manager را انتخاب کرده و Data form را فعال کنید.

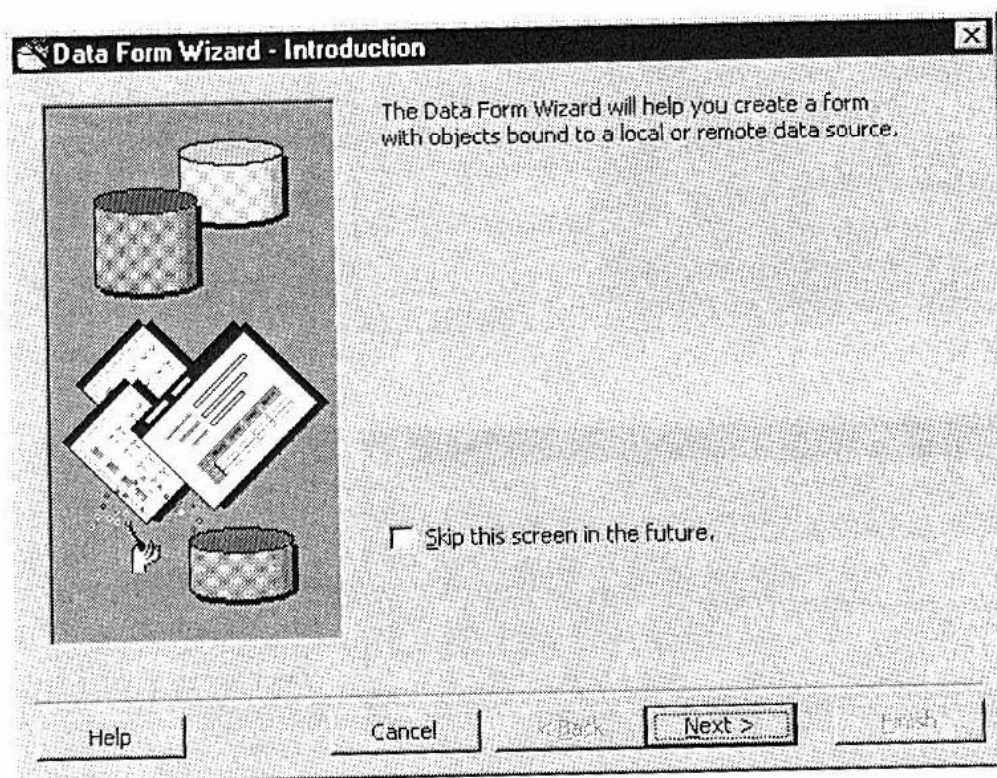


شکل ۴-۷.

فعال شدن منوی
Data Form Wizard...
از منوی
Add-Ins

برنامه Data form Designer یک ویزارد، برای ایجاد فرمهای اطلاعاتی می باشد. در ادامه هر مرحله از این ویزارد را توضیح می دهیم. ابتدا Data form Designer را از منوی Add-in انتخاب کنید و هر مرحله را به ترتیب زیر، دنبال نمایید.

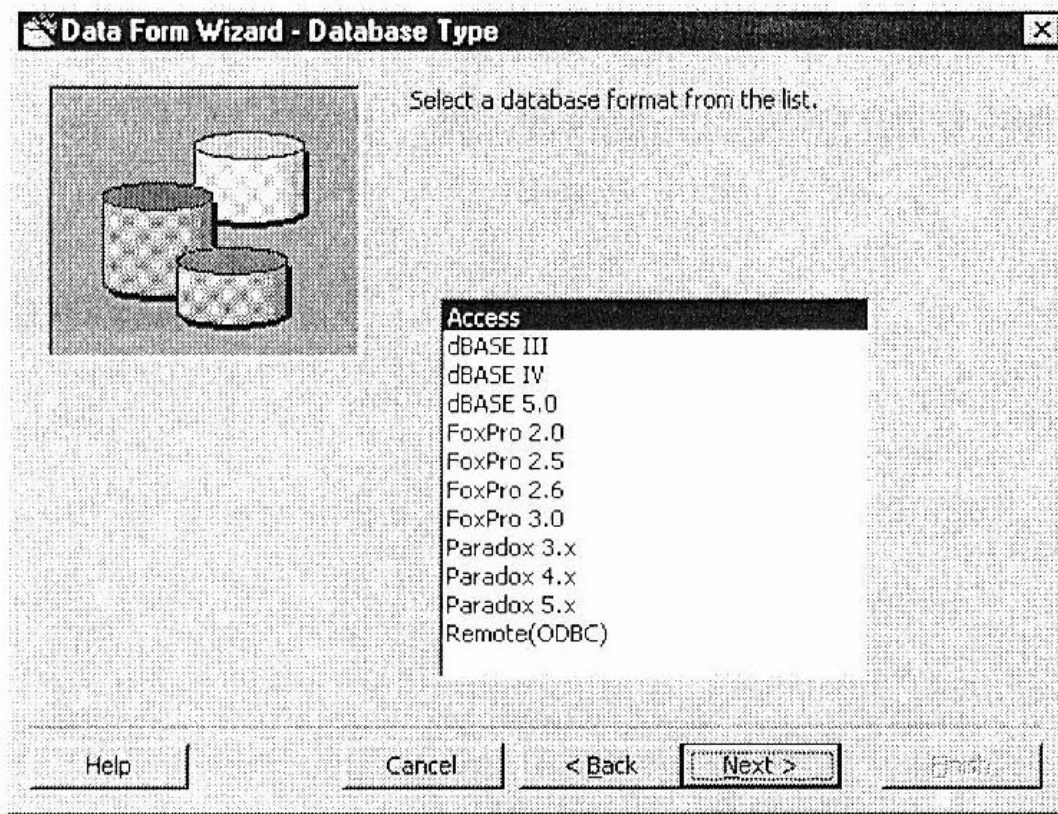
۱ - اولین مرحله در ویزارد طراحی فرم اطلاعات، قسمت معرفی است (شکل ۴-۸) و می توانید با انتخاب گزینه‌ای که در این مرحله قرار دارد از نشان دادن مجدد آن جلوگیری کنید.



شکل ۴-۸.

مرحله اول ویزارد طراحی فرم
اطلاعاتی - قسمت معرفی ویزارد.

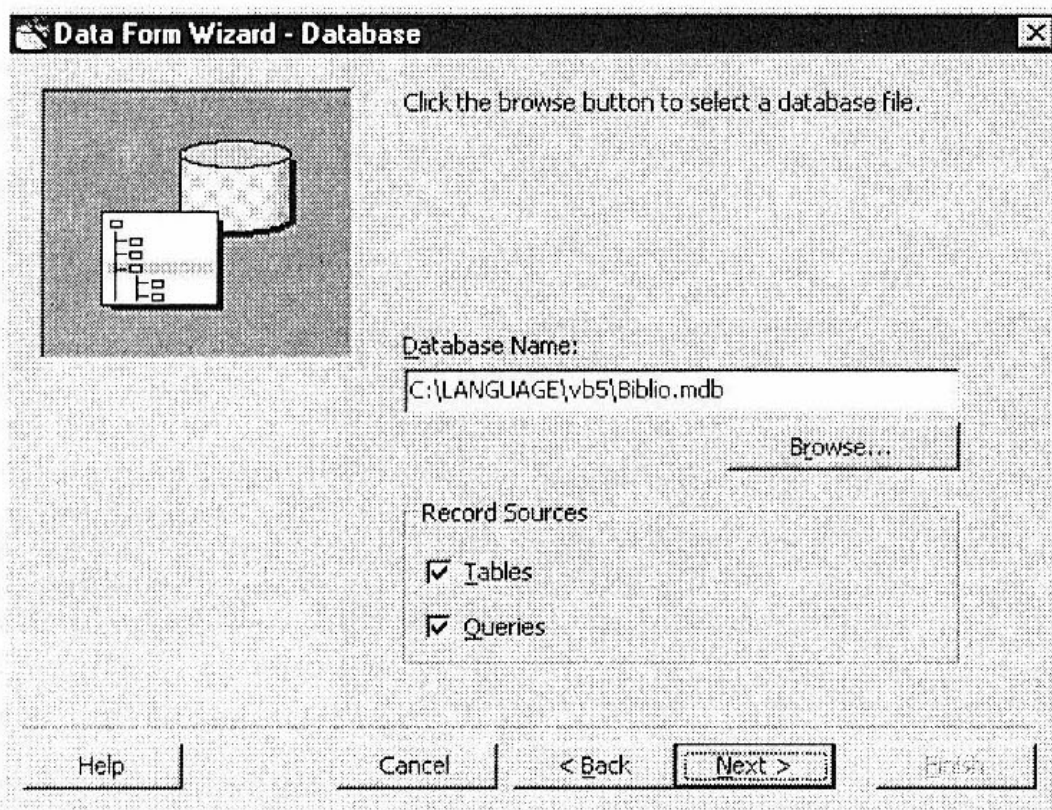
۲- در مرحله دوم که Database Type نام دارد باید نوع پایگاه داده خود را انتخاب کنیم. همان طور که در شکل ۴-۹ می بینید، پایگاه داده شما می تواند انواع مختلفی داشته باشد. حالت پیش فرض برای این قسمت، پایگاه داده اکسس است که در اینجا همین گزینه را انتخاب نموده و روی Next، کلیک می کنیم.



شکل ۹-۴.

مرحله دوم ویزارد طراحی فرم اطلاعاتی - قسمت مشخص کردن نوع پایگاه داده.

۳- مرحله سوم این ویزارد، Database نام دارد و در آن باید نام پایگاه داده و نوع منبع اطلاعات شامل سؤال و جدول را انتخاب کرد. در اینجا برای مثال در قسمت Database Name، پایگاه داده Biblio.MDB را انتخاب می‌نماییم.

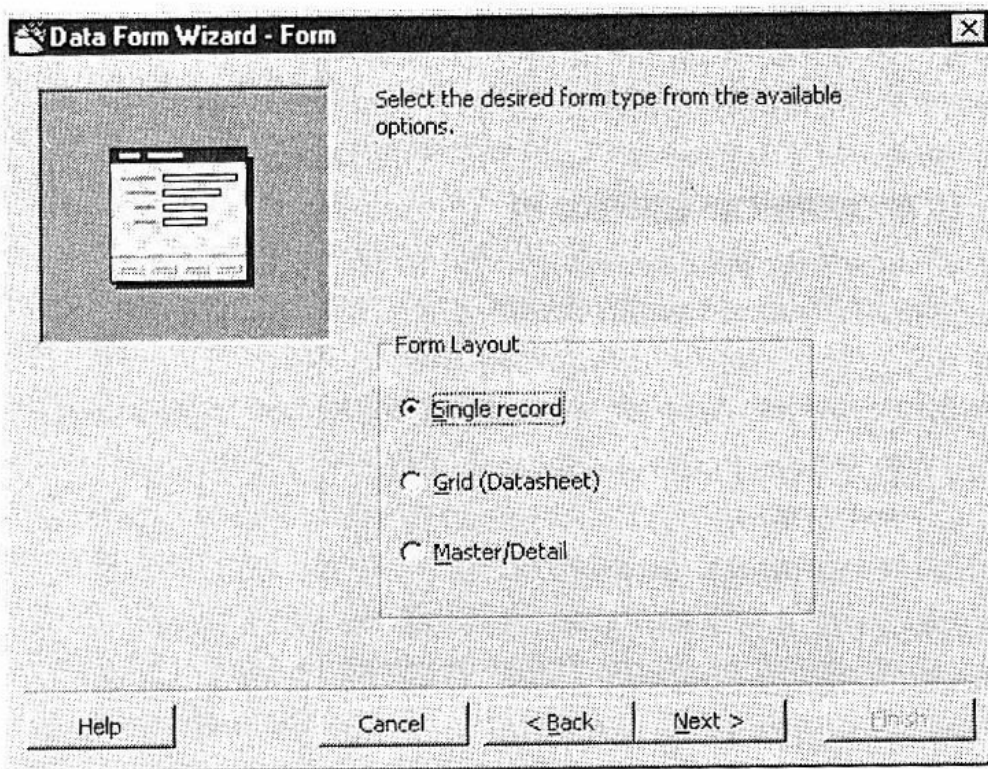


شکل ۱۰-۴.

مرحله سوم ویزارد طراحی فرم اطلاعاتی - قسمت مشخص کردن نام پایگاه داده و منبع داده‌ها.

۴- در این مرحله باید نوع نمایش داده‌ها را در فرم مشخص نمود. سه حالت نمایش داده‌ها در Form Layout مشخص شده است که با انتخاب هر یک، شکل کلی آن در بالای این پنجره نمایش

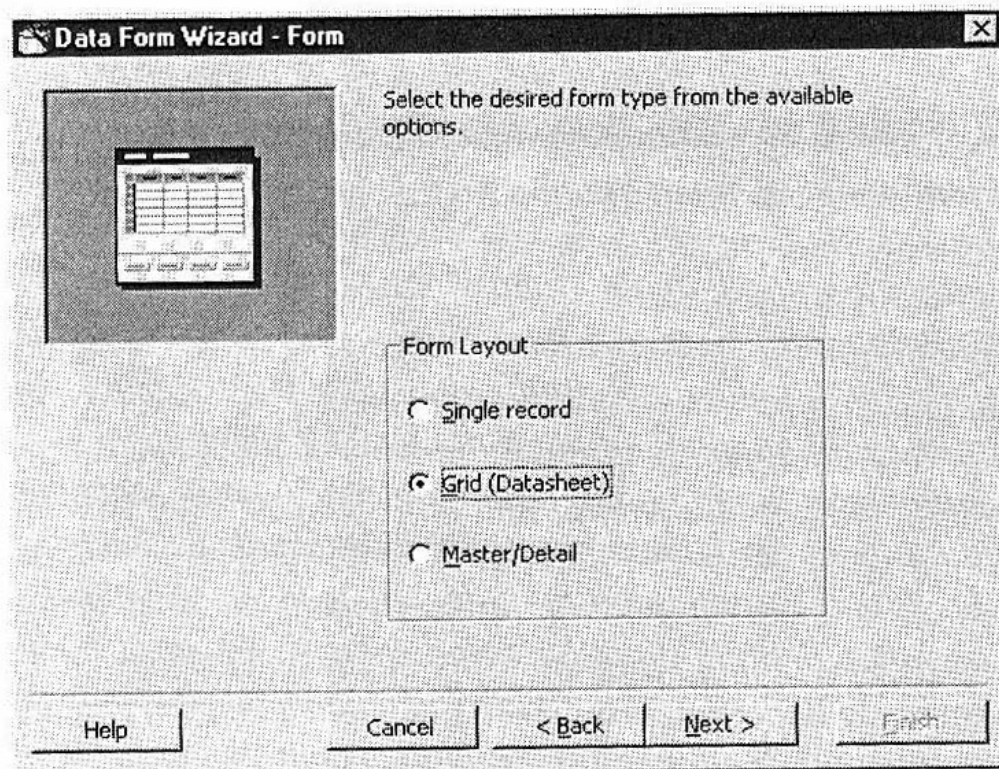
داده می شود. شکل‌های ۴-۱۱، ۴-۱۲ و ۴-۱۳ این سه حالت مختلف را نشان می‌دهد.



The screenshot shows the 'Data Form Wizard - Form' dialog box. On the left, there is a preview window showing a form with several input fields. The main area contains the text 'Select the desired form type from the available options.' Below this, under the heading 'Form Layout', there are three radio button options: 'Single record' (which is selected), 'Grid (Datasheet)', and 'Master/Detail'. At the bottom of the dialog, there are five buttons: 'Help', 'Cancel', '< Back', 'Next >', and 'Finish'.

شکل ۴-۱۱

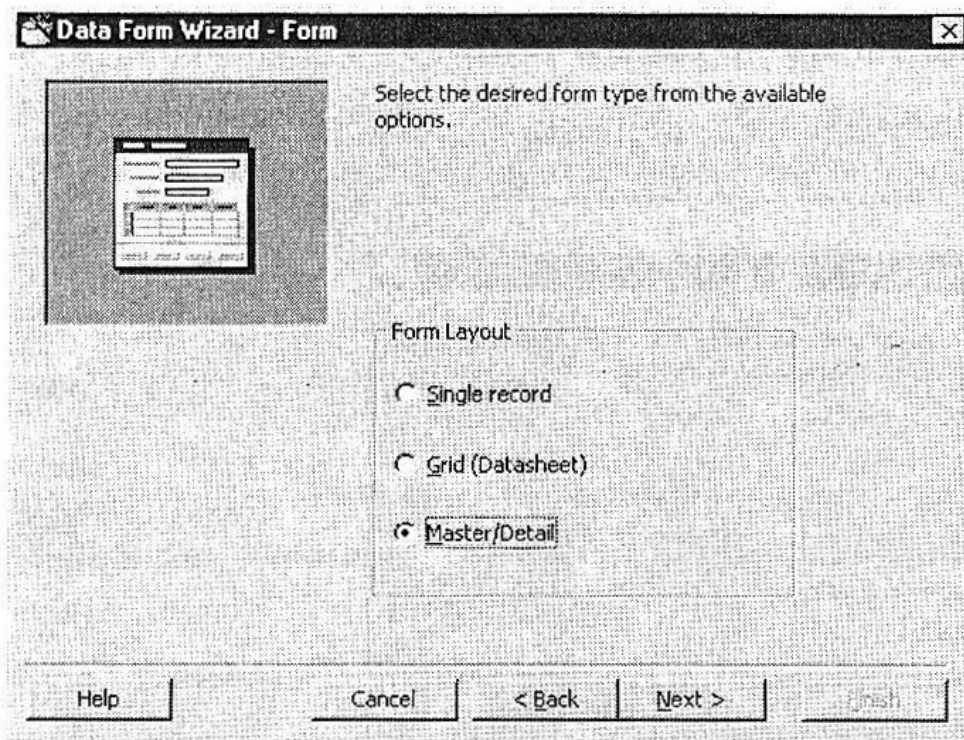
مرحله چهارم ویزارد طراحی فرم اطلاعاتی - انتخاب نوع نمایش داده ها به صورت Single record.



The screenshot shows the 'Data Form Wizard - Form' dialog box. On the left, there is a preview window showing a form with a grid of data. The main area contains the text 'Select the desired form type from the available options.' Below this, under the heading 'Form Layout', there are three radio button options: 'Single record', 'Grid (Datasheet)' (which is selected), and 'Master/Detail'. At the bottom of the dialog, there are five buttons: 'Help', 'Cancel', '< Back', 'Next >', and 'Finish'.

شکل ۴-۱۲

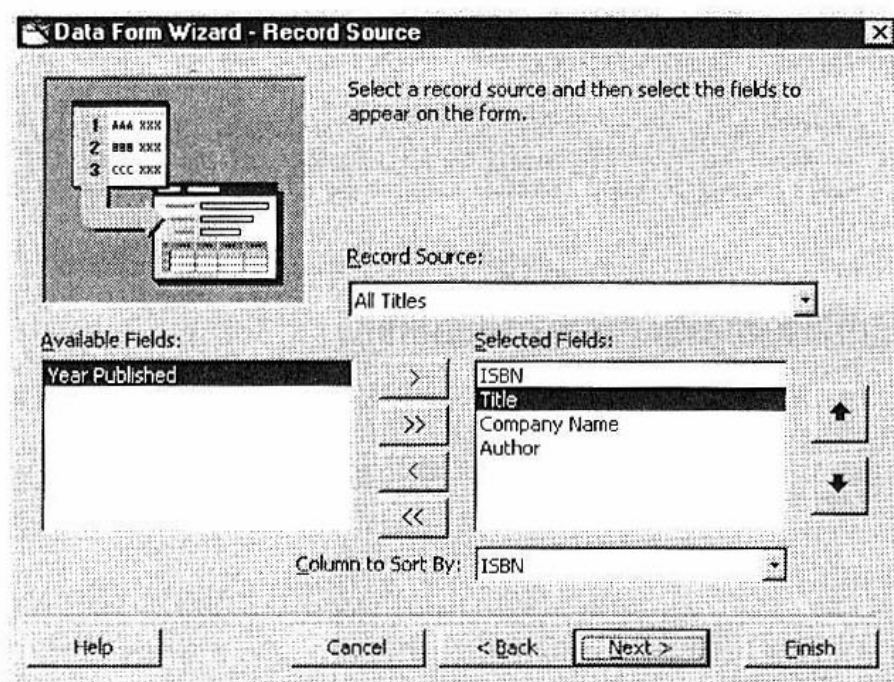
مرحله چهارم ویزارد طراحی فرم اطلاعاتی - انتخاب نوع نمایش داده ها به صورت Grid (Datasheet).



شکل ۱۳-۴.

مرحله چهارم ویزارد طراحی فرم اطلاعاتی - انتخاب نوع نمایش داده‌ها به صورت Master / Detail

۵- در این مرحله پس از انتخاب جدول یا سؤال مناسب، فیلدهای مورد نظر برای نمایش در فرم را توسط دکمه‌های فرمان مربوطه از لیست Available Fields به لیست Selected Field، منتقل کنید و فیلدی را که می‌خواهید لیست داده بر اساس آن مرتب شود در قسمت Column to Sort قرار دهید.



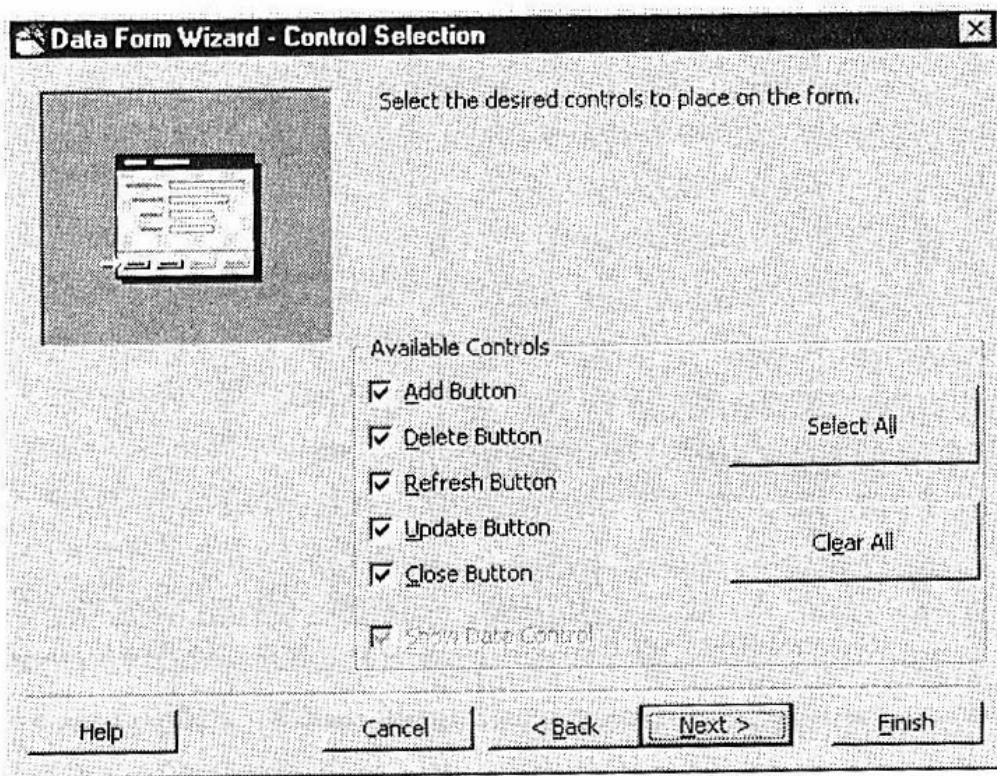
شکل ۱۴-۴.

مرحله پنجم ویزارد طراحی فرم اطلاعاتی - انتخاب فیلدها از جدول یا سؤال برای نمایش در فرم اطلاعاتی.

۶- در مرحله ششم یا Control Selection دکمه‌های فرمان فعال روی رکورد های اطلاعاتی را مشخص می‌کنیم.

دکمه فرمان Add برای اضافه کردن رکورد جدید به لیست داده به کار می‌رود. دکمه فرمان Delete برای حذف کردن رکورد جاری از لیست داده به کار می‌رود.

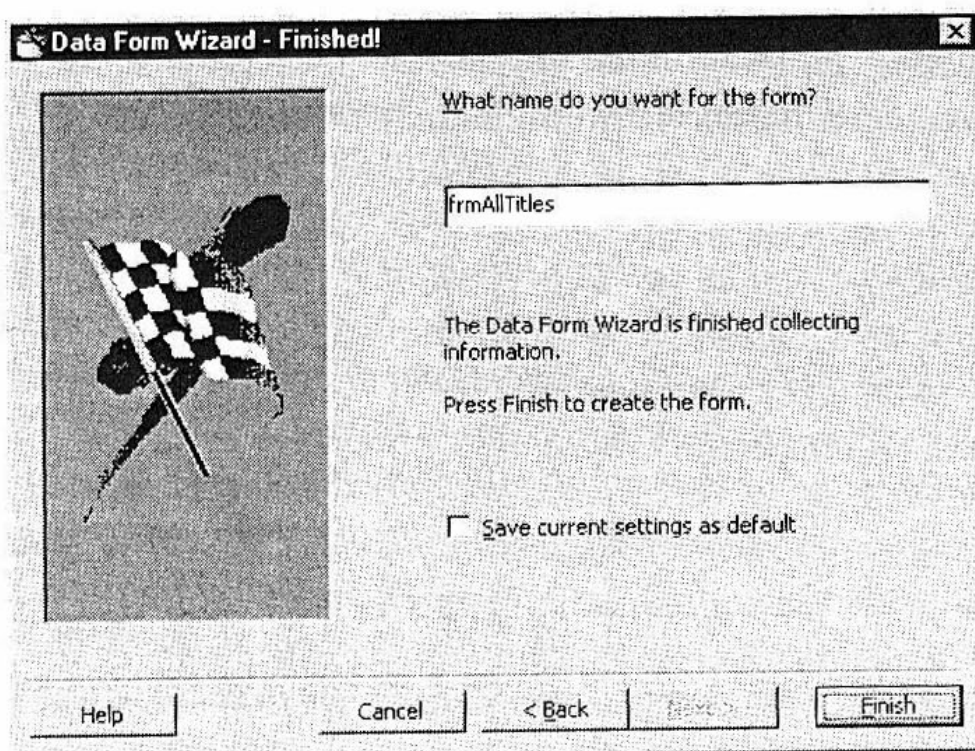
دکمه فرمان Refresh برای نمایش تغییرات ثبت شده در رکورد ها و فیلدها به کار می رود.
 دکمه فرمان Update برای به روز رساندن تغییرات در رکورد ها و فیلدها به کار می رود.
 دکمه فرمان Close برای بسته شدن پنجره فرم به کار می رود.



شکل ۴-۱۵

مرحله ششم ویزارد طراحی فرم
اطلاعاتی - انتخاب دکمه های
فرمان لازم جهت ایجاد در
فرم اطلاعاتی.

۷- در مرحله پایانی این ویزارد ، باید نام فرم را مشخص نمود و با زدن دکمه فرمان Finish کار تمام است. در صورت تمایل می توانید تنظیمات انجام شده روی بخشهای مختلف ویزارد را ذخیره کنید تا در هنگام استفاده مجدد از ویزارد ثابت باشند، گزینه مربوطه را انتخاب نمایید. با انتخاب این گزینه ، تنظیمات اعمال شده حالت پیش فرض در می آیند.



شکل ۴-۱۶

مرحله پایانی ویزارد طراحی فرم
اطلاعاتی - انتخاب نام مناسب برای
فرم اطلاعاتی.

شکل ۴-۱۷ اجرای فرم ساخته شده توسط این ویزارد را نشان می دهد.

شکل ۴-۱۷

اجرای فرم اطلاعاتی طراحی شده توسط ویزارد طراحی فرم اطلاعاتی.

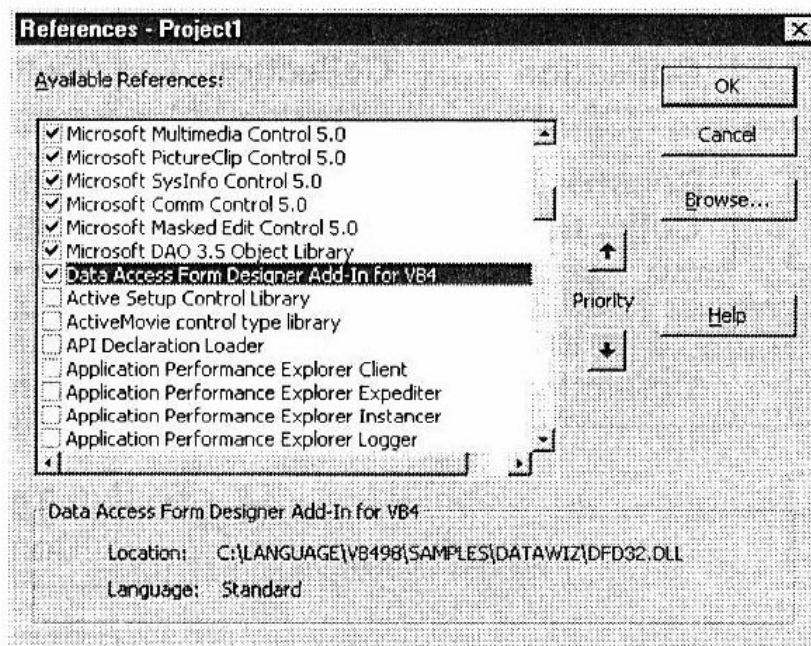
تذکر: اگر می خواهید این فرم در ابتدای پروژه اجرا گردد از منوی Project گزینه Project1 Properties را انتخاب کرده و پس از فعال شدن فرم مربوطه، این فرم را به عنوان فرم اصلی پروژه انتخاب کنید.

شکل ۴-۱۸

آشنایی با اشیای پایگاه داده

تعیین مشخصات شیء Data

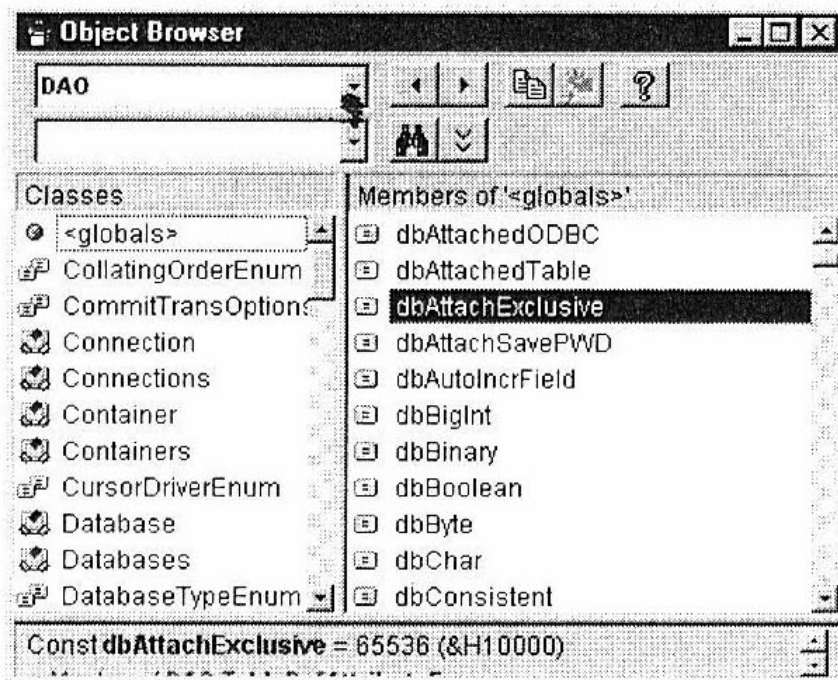
برای استفاده از مشخصات شیء Data ابتدا باید مرجع آن را تعیین کنیم. برای این کار از منوی Projects گزینه References... را انتخاب کرده و از لیست مرجع‌اشیا، Microsoft DAO 3.X Object library را انتخاب می‌کنیم تا کتابخانه آن قابل استفاده گردد.



شکل ۱-۵.

فعال کردن کتابخانه اشیای داده.

حال می‌توان از طریق [F2] Object Browser تک تک اجزای شیء DAO 3.0 و خصوصیات و متدهای آن را مشاهده کرد.



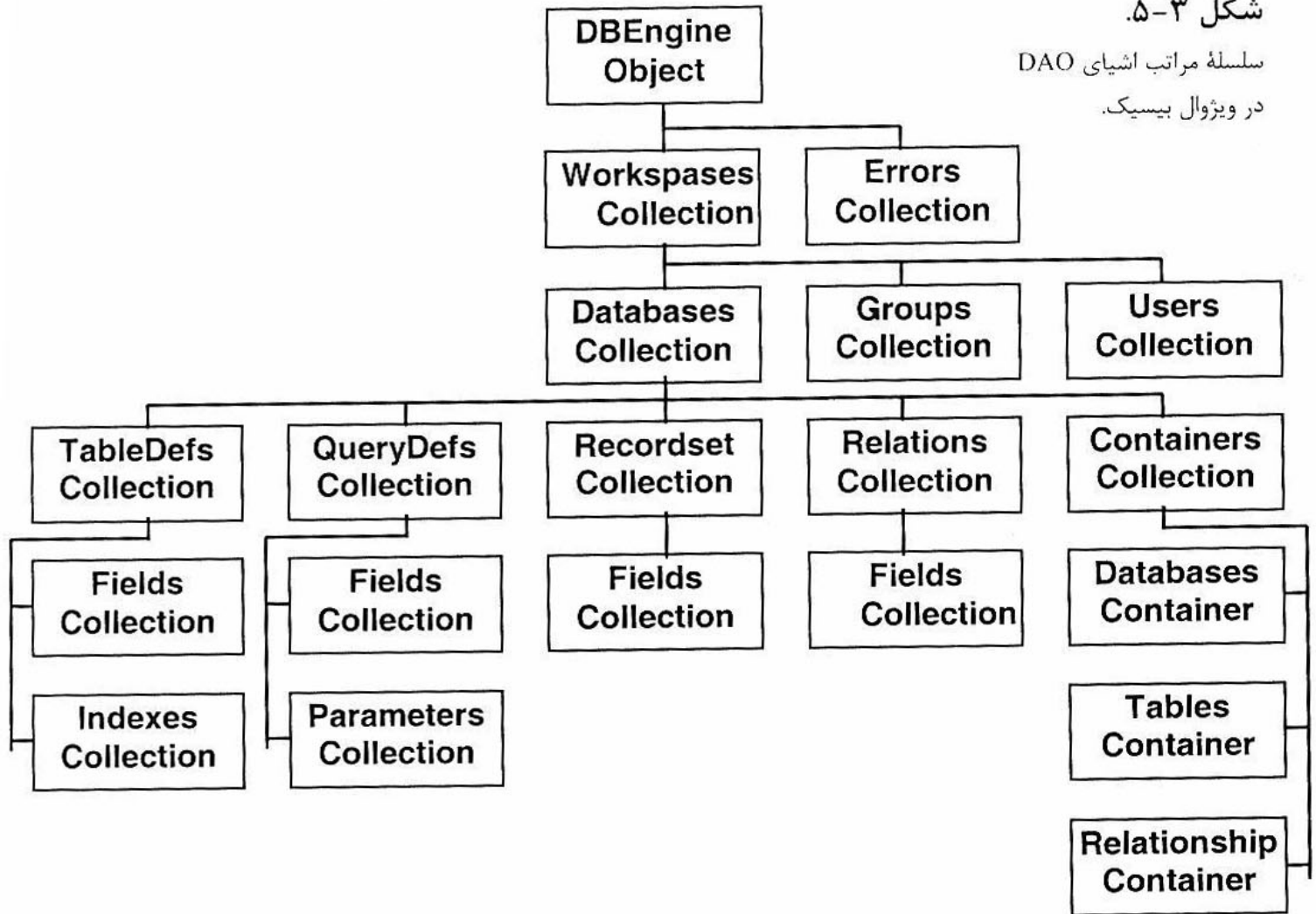
شکل ۲-۵.

نمایش اجزای کتابخانه Object Browser در DAO.

کلاس DAO در ویژوال بیسیک با شیء DBEngine کامل شده و از اشیائی تشکیل گشته است که امکان ایجاد، ارتباط و توزیع انواع پایگاههای داده پشتیبانی شده را در اختیار قرار می دهد. شکل ۳-۵ سلسله مراتب اشیای پایگاه داده را در ویژوال بیسیک نشان می دهد.

شکل ۳-۵.

سلسله مراتب اشیای DAO در ویژوال بیسیک.



حال به بررسی اجمالی هر یک از این اشیای می پردازیم و در فصل بعد هر یک از خصوصیات آنها را بررسی می کنیم.

شیء DBEngine

همان طور که در شکل ۳-۵ مشاهده کردید، شیء DBEngine (Database Engine) اولین شیء، در سلسله مراتب اشیای دسترسی به داده (DAO) می باشد. این شیء منبع و کنترل کننده سلسله مراتب اشیای در DAO می باشد.

از DBEngine می توان برای کنترل کردن Jet database engine، تحلیل خصوصیات آن و انجام عملیات روی اشیای مجازی استفاده نمود. برای مثال شما می توانید:

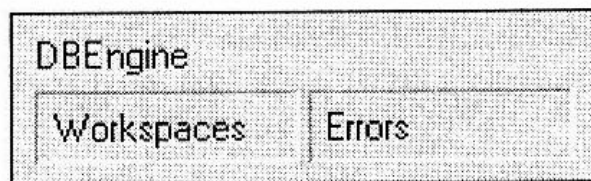
- با استفاده از خصوصیت Version شماره نسخه Jet engine را مشخص کرده و با استفاده از خصوصیت Login Timeout زمان خروج از ODBC را تنظیم و یا به دست آورید و با استفاده از خصوصیت Register Database متدهای اطلاعات ODBC را به Jet engine ارسال کنید.
- با استفاده از متد Idle این امکان را برای Jet engine فراهم می سازید تا عملیات ناتمام را کامل سازد و با استفاده از متدهای Compact Database و Repair Database فایل‌های پایگاه داده را فشرده و بازسازی نمایید.
- با استفاده از CreateWorkspace می توانید یک Session جدید ایجاد کنید (تعریف Session ، در قسمت Workspace آورده شده است).
- با استفاده از مجموعه Errors می توانید جزئیات خطاهای Jet engine را به دست آورید.
- با تنظیم DBEngine به Nothing می توانید آن را برای یک مجموعه عملیات جدید آماده کنید.
- و از شی DBEngine برای کنترل سیستم حفاظتی Jet engine استفاده نمایید.

توجه

DBEngine یک شی از پیش ایجاد شده است و برای استفاده از آن، لزومی به ایجاد آن نمی باشد.

برای اشاره کردن به مجموعه‌ها ، خصوصیتها و متدهای این شی از دستور کلی زیر استفاده می کنیم:
DBEngine. [collection | property | method]

هر شی DBEngine از محیطهای کاری (Workspaces) و خطاها (Errors) تشکیل شده است که در دو قسمت بعد به توضیح آن می پردازیم.



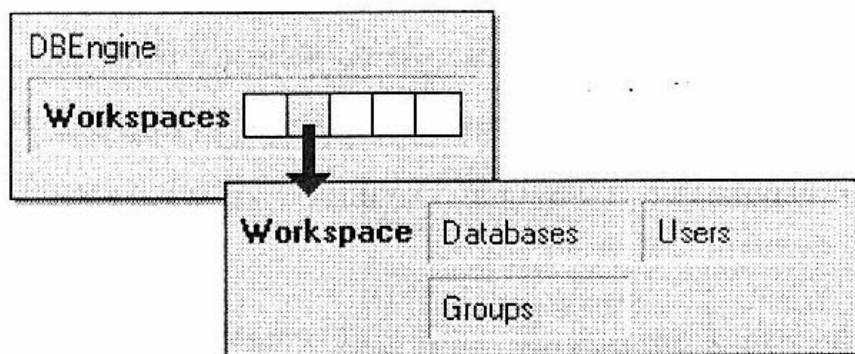
شکل ۴-۵.

اجزای شی DBEngine.

شیء Workspace و مجموعه Workspaces

این شیء برای هر کاربر، یک نشست (Session) تعریف می کند. نشست ، مدت زمانی است که با ورود کاربر به سیستم شروع و با خارج شدن کاربر از سیستم، پایان می پذیرد. تمام اعمالی که توسط کاربر در این زمان صورت می گیرد تحت یک محیط کاری اعمال می شود که از پایگاههای داده باز شده، مکانیزم ارتباطات و تبادل همزمان و سیستم حفاظت این پایگاههای داده تشکیل شده است.

مجموعه Workspaces از تمام Workspace های فعال و غیر پنهان از شیء DBEngine تشکیل شده است.



شکل ۵-۵.

اجزای مجموعه Workspaces

ملاحظات

از شیء Workspace برای مدیریت نشست جاری و یا اضافه کردن یک نشست جدید استفاده می شود. در هر نشست شما می توانید چند پایگاه داده را باز کرده، ارتباط و تبادل آنها را مدیریت کنید و ساختار حفاظت از پایگاه داده را از طریق نام کاربر (User Name) و کلمه رمز (Password) بنا کنید. برای مثال شما می توانید:

- با ایجاد یک محیط کاری (Workspace) و استفاده از مشخصات Name، Password و UserName یک نشست بر اساس نام و کلمه عبور ذکر شده ایجاد کنید.
- با استفاده از متد OpenDatabase یک یا چند پایگاه داده را در محیط کاری فعال نمایید.
- با استفاده از متدهای BeginTrans، CommitTrans و Rollback تبادل اطلاعات به روز رسانی را در یک محیط کاری مدیریت کنید و در صورت نیاز تغییرات را برگردانید.

توجه: در صورتی که فایل SYSTEM.MD? را برای Jet Engine مشخص نکرده باشید، محیطهای کاری، قابلیت حفاظت دادهها را دارا نمی باشند. برای ایجاد سیستم حفاظت دادهها به فصل دوازدهم مراجعه کنید.

نکات

۱. هنگامی که با Jet database engine شروع به کار می کنید به طور خودکار یک محیط کاری به صورت پیش فرض ایجاد می شود [DBEngine.Workspaces(0)]. مشخصات Name و UserName هم به صورت پیش فرض به ترتیب "#Default Workspace#" و "Admin" تنظیم می گردند.
۲. اگر سیستم حفاظتی شما فعال باشد در خصوصیت UserName، نام کاربری باید قرار گیرد که می خواهد به سیستم وارد گردد.
۳. اگر از متد Open Database بدون تنظیم یک محیط کاری استفاده کنید به صورت پیش فرض [DBEngine.Workspaces(0)] مورد استفاده قرار می گیرد.

یک شیء محیط کاری توسط متد CreateWorkspace از شیء DBEngine ایجاد می شود. پس از ایجاد این شیء به مجموعه محیطهای کاری افزوده می شود و می توان به وسیله نام و یا موقعیت آن در مجموعه به آن اشاره کرد. دستورات زیر هر دو شکل را نشان می دهند.

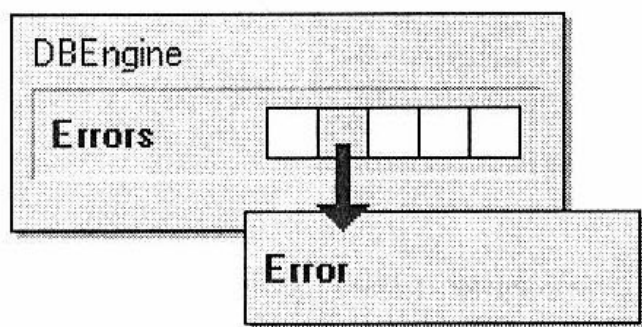
-Workspaces("Name")

-DBEngine.Workspaces(n)

در دستورات کلی بالا Name نام محیط کاری و n شماره آن در مجموعه محیطهای کاری است.

شیء Error و مجموعه Errors :

شیء خطا (Error) شامل جزئیات خطاهای دسترسی به دادههاست و مجموعه خطاها (Errors) شامل اشیای خطایی است که طبقه بندی شده اند.



شکل ۵-۶

اجزای مجموعه Errors

ملاحظات

هر عملی در دسترسی به دادهها می تواند دارای یک یا چند خطا باشد. هر خطایی که اتفاق می افتد، شامل یک یا چند شیء خطا می باشد که در مجموعه خطاها از شیء DBEngine قرار می گیرد. هنگامی که خطای دیگری در DAO اتفاق می افتد، مجموعه خطاها پاک می شود و اشیای جدید خطا در آن قرار می گیرد.

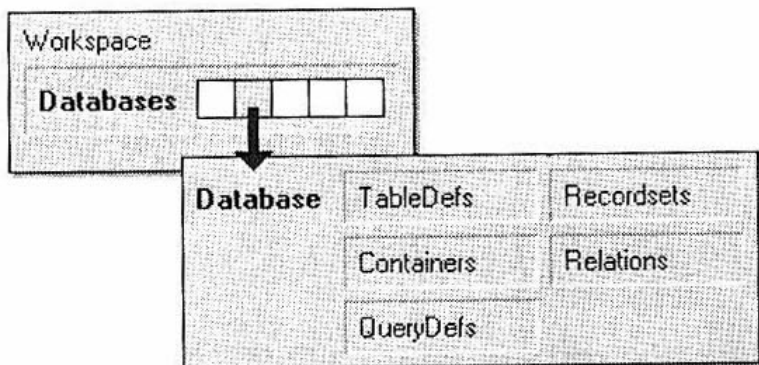
نکات

- اعمالی که برای آنها خطا تعریف نشده باشد، هیچ تأثیری در مجموعه خطاها نخواهد داشت.
- هر دسته از اشیای خطا در داخل مجموعه خطاها نشانگر خطای خاصی است که بنا به اولویت اتفاق در آن مجموعه قرار گرفته است.
- تعداد اشیای خطا در مجموعه خطاها را می توان به وسیله خصوصیت Count به دست آورد.
- هر خطا دارای کدی است که در خصوصیت Number قرار دارد و می توان آن را به دست آورد.
- هر خطا دارای توضیحی است که در خصوصیت Description قرار دارد.
- از طریق خصوصیت Source در خطا، می توان شیئی که خطا برای آن اتفاق افتاده است را به دست آورد.

همان طور که در جدول سلسله مراتب دسترسی به داده ها دیدید هر محیط کاری خود شامل Databases ، Users و Groups می باشد که در قسمت بعد به توضیح Database می پردازیم و دو شیء Users و Group را در فصل دوازدهم توضیح خواهیم داد.

شیء Database و مجموعه Databases

شیء Database نشانگر یک پایگاه داده باز شده می باشد و مجموعه Databases شامل تمام پایگاههای داده ای می باشد که در یک شیء محیط کاری باز و یا ایجاد شده است.



شکل ۷-۵.

اجزای مجموعه Databases

ملاحظات

شما می توانید از طریق شیء Database و متدها و خصوصیات آن ، عملیات اطلاعاتی را روی یک پایگاه داده باز شده انجام دهید. همچنین از طریق این شیء می توان اطلاعات مجموعه های داخل آن مانند جداول ، سؤالاها و ارتباطات را به دست آورد و آنها را ایجاد و تعریف نمود. برای مثال شما می توانید:

- با استفاده از متد Execute ، یک سؤال اجرایی را اجرا و یا یک رشته SQL را به ODBC ارسال کنید.
- با استفاده از متد OpenRecordset ، یک لیست داده جدید را مستقیماً از روی شیء Database ایجاد کنید.
- با استفاده از متد Close یک پایگاه داده باز شده را ببندید.

CreateDatabase این امکان را برای شما به وجود می آورد که یک پایگاه داده جدید را به مجموعه Databases اضافه کرده و آن را بر روی دیسک ذخیره سازید. پس از ایجاد این پایگاه داده باید جداول آن را ایجاد کنید تا بتوانید اطلاعاتتان را در آن نگهداری نمایید. همچنین می توانید ایندکسها ، ارتباطات و کنترل جداول خروجی و خواص محافظتی از قبیل ایجاد محدودیتها ، گروهها و کاربران را نیز به این پایگاه داده اضافه کنید.

برای باز کردن یک پایگاه داده باید از متد OpenDatabase استفاده نمود تا پایگاه داده را به مجموعه Databases اضافه کند. متد OpenDatabase از متدهای محیط کاری می باشد. باز کردن یک پایگاه داده می توان تحت محیط کاری پیش فرض [Workspace(0)] و یا هر محیط کاری دلخواه دیگری صورت گیرد. برای مثال ، برای باز کردن دو پایگاه داده مجزا می توان به صورت زیر عمل کرد:

```
Dim Db1 As Database, Db2 As Database, Ws As Workspace, WS2 As Workspace
Dim MyUserName As String
Set Ws = Workspaces(0)
Set Db1 = Ws.OpenDatabase("MYDB.MDB")
MyUserName = Ws.UserName
Set WS2 = CreateWorkspace("Special", MyUserName, "SpacialID")
Set Db2 = WS2.OpenDatabase("MYDB.MDB")
```

در مثال بالا Ws یک محیط کاری پیش فرض و WS2 یک محیط کاری با نام و نام کاربر و کلمه رمز تعریف شده است. هر پایگاه داده که تحت محیط کاری WS2 ایجاد و یا باز گردد ، با نام و کلمه رمز مخصوص به آن محیط کاری شناخته می شود.

علاوه بر پایگاههای داده Jet engine شما می توانید پایگاههای دیگری نظیر فاکس پرو ، پاراداکس و تمام پایگاههای داده ای که توسط ویژوال بیسیک قابل دسترسی هستند ، ارتباط داشته و از جداول آنها استفاده کنید. برای این کار در فصل بعد به متد OpenDatabase و پارامترهای آن توجه کنید. برای مثال کد زیر یک پایگاه داده فاکس پرو را باز می کند:

```
Dim Db As Database
Set Db = Workspaces(0).OpenDatabase("C:\Foxfiles", False, False, "FoxPro 2.5")
```

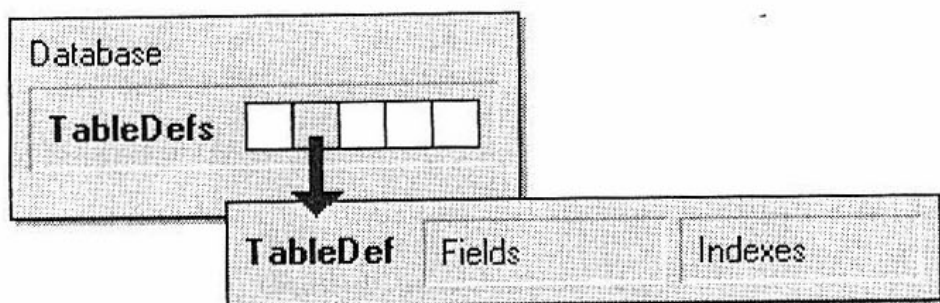
نکته

اگر با پایگاههای داده فاکس پرو آشنا باشید، می دانید که هر پایگاه داده آن فقط شامل یک جدول است و مانند پایگاههای داده اکسس تمام جداول و اجزای آن داخل یک فایل نیست، بنابراین برای باز کردن جداول فاکس پرو باید تمام فایلهای DBF را در یک شاخه ذخیره سازیم و در برنامه (همانند مثال بالا) پایگاه داده را به آن شاخه نسبت دهیم. در این صورت جداول فایلهای DBF در این پایگاه داده قابل دسترسی خواهد بود.

هر شیء Database از اجزایی نظیر TableDefs ، QueryDefs ، Recordsets ، Containers و Relations تشکیل شده است که در ادامه سه عضو اصلی آن را مورد بررسی قرار می دهیم و در فصل بعد به شرح جزئیات هریک از خصوصیات این اشیا می پردازیم.

شیء TableDef و مجموعه TableDefs

این شیء بیانگر یک جدول اصلی از پایگاه داده و یا جدول ارتباط داده شده آن (Linked Table) می باشد. مجموعه TableDefs شامل تمام اشياء TableDef ای می باشد که در پایگاه داده ذخیره شده اند. هر TableDef از مجموعه فیلدها و ایندکسها تشکیل شده است.



شکل ۸-۵.

اجزای مجموعه TableDefs.

ملاحظات

- شما می توانید از طریق شیء TableDef ، متدها و خصوصیات آن ، عملیات اطلاعاتی را روی جدول یک پایگاه داده ، انجام دهید. برای مثال شما می توانید:
 - فیلدها و ساختار ایندکس آنها را در جداول محلی ، جداول ضمیمه شده (Attached Table) و یا حتی یک جدول خارجی نیز مورد بررسی قرار دهید.
 - دو خصوصیت Connect و SourceTableName را که برای تنظیم جداول ضمیمه شده استفاده می شوند ، بخوانید و بنویسید و با استفاده از متد RefreshLink ارتباط با جداول ضمیمه شده را روز رسانی کنید.
 - دو خصوصیت Validation Rule و Validation Text را بخوانید (و یا اگر مشخصه Updatable به True تنظیم شده باشد ، آن را تغییر دهید) تا شرط مورد نظرتان را تعیین و یا تنظیم کنید.
 - با استفاده از متد OpenRecordset یک لیست داده جدید بر اساس جدول مورد نظرتان ایجاد نمایید.
- انتخاب پیش فرض شیء Database ، مجموعه TableDefs و انتخاب پیش فرض شیء TableDef ، مجموعه فیلدهای آن می باشد و انتخاب پیش فرض خصوصیت در شیء TableDef ، خصوصیت Name است. شما می توانید با استفاده از این پیش فرضها کدهای خود را ساده تر و خلاصه تر نمایید. برای مثال دو عبارت زیر ، هر دو عددی را بر می گردانند که نشان دهنده نوع یک فیلد در TableDef ایجاد شده توسط یک ابزار داده می باشد. در این مثال ابزار داده Data1 نام دارد.

```
Print Data1.Database.TableDefs("Publishers").Fields("PubID").Type
```

```
Print Data1.Database("Publishers")("PubID").Type
```

برای جداول اصلی، خصوصیت RecordCount، تعداد رکوردهای جدول را بر می‌گرداند و برای جداول ضمیمه شده این خصوصیت همیشه دارای مقدار ۱- است.

برای دسترسی به یک TableDef که در مجموعه TableDefs قرار دارد باید از دستور کلی زیر استفاده کنیم. در این دستور Name نامی است که در خصوصیت Name از شیء TableDef تنظیم شده است.

```
TableDefs("name ")
```

برای ایجاد یک شیء TableDef باید از متد CreateTableDef استفاده نمود.

برای تغییر در ساختار یک جدول مثل اضافه کردن فیلد به آن باید :

۱- اطمینان پیدا کنید که تمام لیستهای داده که بر اساس آن جدول ایجاد شده است، بسته می‌باشد.

۲- با استفاده از متد CreateField، یک شیء Field ایجاد کنید و خصوصیات آن را تنظیم کنید.

۳- با استفاده از متد Append این شیء فیلد را به مجموعه فیلدهای TableDef اضافه نمایید.

برای ایجاد یک جدول در پایگاه داده که برای رکوردهای جدید آماده باشد باید:

۱- با استفاده از متد CreateTableDef یک شیء TableDef ایجاد کنید.

۲- خصوصیات آن را تنظیم نمایید.

۳- با استفاده از متد CreateField برای هر یک از فیلدهای این جدول یک شیء فیلد ایجاد کنید و

خصوصیات آن را تنظیم نمایید.

۴- با استفاده از متد Append، این فیلدها را به مجموعه فیلدها اضافه کنید.

۵- با استفاده از متد Append، این TableDef را به مجموعه TableDefs در پایگاه داده اضافه

نمایید.

هر جدول ضمیمه شده (Attached Table)، توسط دو خصوصیت Connect و SourceTableName با

پایگاه داده ارتباط داده شده است. برای ضمیمه کردن یک جدول به پایگاه داده باید:

۱- با استفاده از متد CreateTableDef یک شیء TableDef ایجاد کنید.

۲- دو خصوصیت Connect و SourceTableName را در آن تنظیم نمایید.

۳- با استفاده از متد Append، این TableDef را به مجموعه TableDefs در پایگاه داده اضافه

کنید.

شما می توانید یک فیلد را از مجموعه TableDefs حذف کنید. این کار در صورتی انجام می شود که هیچ ایندکسی بر اساس آن ایجاد نشده باشد. توجه داشته باشید که با حذف یک فیلد تمام اطلاعات آن از بین خواهد رفت. برای حذف یک فیلد باید :

- ۱- از داده های جدول و فیلد ، یک پشتیبان تهیه کنید.
- ۲- فیلد مورد نظر خود را از داخل مجموعه فیلدهای آن جدول حذف نمایید.
- ۳- ساختار جدید خود را ایجاد و به TableDef اضافه کنید.
- ۴- داده ها را Refresh نمایید.

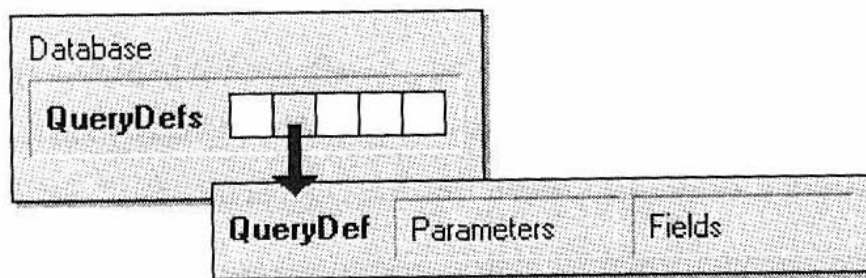
نکته : جستجو روی Table Def با استفاده از روش Find صورت می گیرد.

توجه

چون شیء Fields در TableDef ، QueryDef و Recordset مشترک می باشد، پس از توضیح QueryDef و Recordset آنها را مورد بررسی قرار می دهیم.

شیء QueryDef و مجموعه QueryDefs

شیء QueryDef در Jet Database نشانگر سؤال و مجموعه QueryDefs شامل تمام اشیای QueryDef در پایگاه داده می باشد.



شکل ۹-۵.

اجزای مجموعه Querydefs.

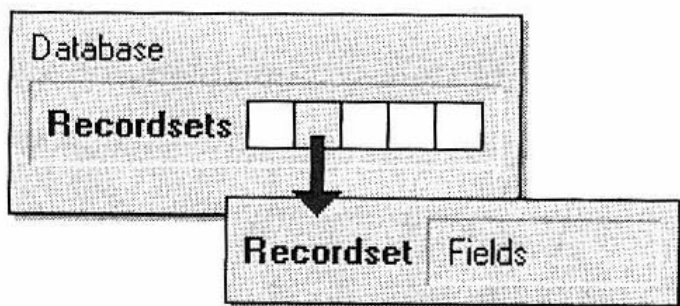
ملاحظات

یک سؤال (Query) مجموعه ای از دستورات SQL می باشد و به دو دسته انتخابی و اجرایی تقسیم می شود. (سؤالها را در فصل هفتم و هشتم بررسی خواهیم کرد). شما می توانید با استفاده از خصوصیات شیء QueryDef یک سؤال ایجاد کنید. برای مثال شما می توانید:

- با استفاده از خصوصیت SQL ، عبارت کد SQL را در یک سؤال به دست آورید و یا آن را برای یک سؤال تنظیم نموده و سپس آن سؤال را اجرا کنید.
 - با استفاده از مجموعه Parameters از شیء QueryDef ، پارامترهای یک سؤال را تنظیم کنید.
 - با استفاده از خصوصیت Type می توانید نوع سؤال را مشخص و یا تنظیم نمایید. انواع سؤالات، شامل Select ، Make Table ، Insert ، Delete و Update می باشد.
 - با استفاده از خصوصیت Connect و ODBC TimeOut ، ارتباط را با درایو ODBC برقرار کنید.
 - با تنظیم خصوصیت ReturnsRecords ، نشان دهید که سؤال، رکوردی را بر می گرداند یا خیر. این خصوصیت مخصوص سؤالهای از نوع Pass-through است.
- سرعت اجرای سؤال در QueryDef بسیار بیشتر از سرعت اجرای آن در متد OpenRecordset است زیرا Jet engine ، قبل از اجرای سؤال دیگر نیاز به کامپایل کردن آن ندارد.
- با استفاده از متد CreateQueryDef می توان یک QueryDef ایجاد کرد.

شیء Recordset و مجموعه Recordsets

شیء لیست داده (Recordset) نشانگر مجموعه ای از رکوردهای یک جدول و یا یک سؤال انتخابی است و مجموعه Recordsets شامل تمام لیستهای داده باز در پایگاه داده می باشد.



شکل ۱۰-۵.

اجزای مجموعه Recordsets

ملاحظات

هنگامی که از اشیای دسترسی به داده استفاده می کنید، فعالیت شما روی داده ها معمولاً توسط اشیای لیست داده صورت می گیرد. هر لیست داده از مجموعه ای رکورد (سطر) و مجموعه ای فیلد (ستون) تشکیل شده است. سه دسته لیست داده وجود دارد:

- ۱- لیست داده از نوع Table: نشان دهنده یک جدول اصلی پایگاه داده می باشد که می توان عملیات حذف، افزودن و تغییر را روی رکوردهای آن انجام داد.
- ۲- لیست داده از نوع Dynaset: نتیجه یک سؤال است که می تواند Updatable باشد و یک لیست داده دینامیکی است و می توان عملیات حذف، افزودن و تغییر را روی رکوردهای جدول و یا جداول آن انجام داد. این نوع لیست داده می تواند شامل فیلدهای یک یا چند جدول از پایگاه داده باشد.

۳- لیست داده از نوع Snapshot : یک نسخه استاتیکی از مجموعه‌ای رکورد است که می‌توانید از آن برای جستجو و یا گزارش، استفاده کنید. این نوع لیست داده می‌تواند شامل فیلدهای یک یا چند جدول از پایگاه داده باشد، اما داده‌های آن فقط خواندنی می‌باشند.

شما می‌توانید نوع لیست داده خود را به وسیله آرگومان *Type* در متد `OpenRecordset` مشخص کنید. در صورتی که این آرگومان تنظیم نشود `Jet Engine` این لیست داده را از نوع `Table` در نظر می‌گیرد. در صورتی که `Jet Engine` نتواند این نوع لیست داده را ایجاد کند، نوع لیست داده `Dynaset` و یا `Snapshot` خواهد بود.

ایجاد یک لیست داده بر اساس یک `TableDef` که ضمیمه نشده است، از نوع `Table` خواهد بود و ایجاد یک لیست داده بر اساس یک `TableDef` ضمیمه شده (`Attached TableDef`) و یا جداول پایگاه‌های داده خارجی مانند `ODBC`، فقط از نوع `Dynaset` و یا `Snapshot` خواهد بود.

خصوصیت `Type` نوع لیست داده را در شیء `Recordset` مشخص می‌کند و خصوصیت `Updatable` مشخص می‌کند که داده‌ها در رکوردهای لیست داده دارای قابلیت به روز رسانی هستند یا خیر.

هنگامی که یک لیست داده جدید را باز می‌کنید، این لیست داده به طور خودکار به مجموعه `Recordsets` اضافه می‌شود و هنگامی که لیست داده را می‌بندید، آن لیست داده از مجموعه `Recordsets` می‌گردد.

اطلاعات ساختار جداول اصلی پایگاه داده مانند نام و نوع فیلدها و ایندکسهای جدول در `TableDef` ذخیره می‌گردد.

شما می‌توانید هر تعداد از متغیرهای شیء لیست داده را که لازم دارید در برنامه‌های خود ایجاد کنید. یک لیست داده می‌تواند به یک یا چند جدول، سؤالاتی منفرد و یا سؤالی که خود از تعدادی لیست داده تشکیل شده‌اند، اشاره داشته باشد. لیست داده می‌تواند از همه یا تعدادی از فیلدها یا رکوردهای یک لیست داده دیگر تشکیل شده باشد.

لیست‌های داده از نوع `Dynaset` و `Snapshot` در حافظه محلی ذخیره می‌شوند و در صورتی که این مقدار حافظه کافی نباشد، اطلاعات را در فضاهای موقتی در هارد نگهداری می‌کند و در صورتی که این فضاها نیز کافی نباشند، خطا رخ خواهد داد.

انتخاب پیش فرض در مجموعه لیست‌های داده، مجموعه فیلدها می‌باشد و خصوصیت پیش فرض در فیلد، خصوصیت `Value` است که مقدار فیلد را بر می‌گرداند. به وسیله این پیش فرضها می‌توان کدها را خلاصه نمود. برای مثال، هر سه عبارات زیر، عملیات تخصیص مقدار را به فیلد `PubID` از رکورد جاری را انجام می‌دهند.


```
MyTable.Fields("PubID").value = 99
```

```
MyTable("PubID") = 99
```

```
MyTable!PubID = 99
```

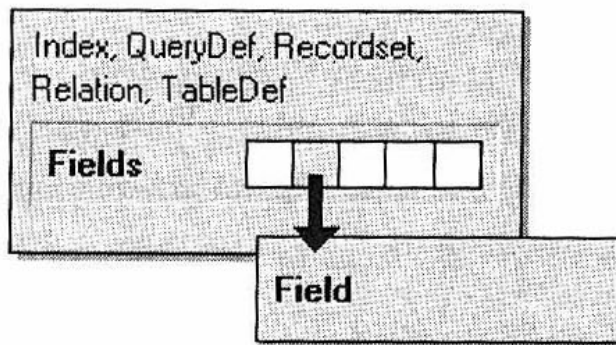
هنگامی که یک لیست داده ایجاد می کنید در صورتی که این لیست داده دارای رکوردی باشد، رکورد جاری روی اولین رکورد این لیست داده قرار می گیرد و در صورتی که لیست داده از رکوردی تشکیل نشده باشد، خصوصیت RecordCount دارای مقدار 0 می شود و خصوصیت های EOF و BOF هر دو ارزش درست (True) دارند.

با استفاده از متدهای MoveNext ، MovePrevious ، MoveFirst ، MoveLast می توان مکان رکورد جاری را تغییر داد و برای لیستهای داده از نوع Dynaset و Snapshot می توان عملیات جستجو را توسط متدهای FindFirst ، FindNext ، FindPrevious ، FindLast انجام داد. در صورتی که یک رکورد مورد جستجو پیدا نشود، خصوصیت NoMatch دارای ارزش True می شود. با استفاده از خصوصیت های EOF و BOF می توان مکان رکورد جاری را در انتها و ابتدای فایل تشخیص داد.

شیء Field و مجموعه Fields

شیء فیلد نشانگر ستونهای داده ها با قالبی مشخص در جداول می باشد و مجموعه Fields

شامل تمام فیلدهای ذخیره شده در اشیای Index ، QueryDef ، Recordset ، Relation ، TableDef می باشد.



شکل ۱۱-۵.

اجزای مجموعه Fields.

ملاحظات

مجموعه فیلدها در شیء لیست داده، نشانگر شیء های فیلد در سطرهای داده ها و یا رکوردها می باشد. با استفاده از این اشیاست است که می توان مقادیر فیلدهای رکورد جاری یک لیست داده را خواند و یا تنظیم نمود.

شما می توانید از طریق شیء Field ، متدها و خصوصیات آن ، عملیات اطلاعاتی را روی فیلدهای یک لیست داده انجام دهید.

برای مثال شما می توانید :

- با خواندن خصوصیت های SourceField و SourceTable منبع اصلی داده های فیلد را به دست آورید.
- با تنظیم خصوصیت ForeignName ارتباط فیلدی را با فیلدهای جداول دیگر برقرار کنید.

- خصوصیات ValidationRule , ValidateOnSet , Required , DefaultValue , AllowZeroLength و یا Validation Text را برای محدود کردن داده های فیلد به کار برید.
 - خصوصیت Value را در یک لیست داده خوانده و یا تنظیم نمایید.
 - با استفاده از متدهای AppendChunk , FieldSize و GetChunk مقادیر فیلدهای از نوع OLE و Memo را خوانده و یا تنظیم نمود.
- شما می توانید به خصوصیت Value در یک فیلد به دو صورت زیر دسترسی داشته باشید:
- استفاده از خصوصیت Name در دستور:

Fields("name ")
Fields!name

نکته

در صورتی که *name* یک نام دو قسمتی باشد برای مشخص کردن آن در دستورات بالا باید به صورت زیر عمل کرد.

Fields("long name ")
Fields![long name]

- استفاده از شماره موقعیت فیلد در مجموعه Fields :

Fields(0)

برای ایجاد یک شیء فیلد جدید در Index , TableDef و Relation از متد CreateField استفاده کنید. در صورتی که شیء فیلد بخشی از شیء لیست داده باشد، داده های فیلدهای رکورد جاری در خصوصیت Value آن نشان داده می شود. برای اشاره کردن به مقدار یک فیلد در لیست داده، مانند آنچه در بالا ذکر شد دیگر نیازی به ذکر مجموعه فیلد نیست و می توان از خاصیت پیش فرض بودن این شیء در لیست داده استفاده کرد و کدها را خلاصه نمود. برای مثال در کد زیر، دسترسی به داده های فیلد موجود در یک لیست داده به صورتهای مختلف نشان داده شده است.

```
Dim Rs As Recordset, Db As Database
Set Db = Workspaces(0).OpenDatabase("MYDB.MDB")
Set Rs = Db.OpenRecordset("Select [Part Name], Size, " _
    & [Part Type], [Part Age] AS Age FROM MyTable")
Debug.Print Rs.Fields(0).Value
Debug.Print Rs(0)
Debug.Print Rs("Part Name")
Debug.Print Rs![Part Type]
Debug.Print Rs!Age
```

بررسی دقیق تر مشخصات اشیای دسترسی به داده

در فصل پیش، با اشیای سلسله مراتب دسترسی به داده‌ها آشنا شدیم و در بعضی موارد به کاربرد و نحوه استفاده از آنها اشاره شد. در این فصل قصد داریم تا با جزئیات هر شیء نظیر متدها، خصوصیتها و روش استفاده از آنها، آشنا شویم. آشنایی هر چه بیشتر با این خصوصیات و متدها، باعث می‌شود تا بتوانیم از امکانات قدرتمند پایگاه داده، درست و مناسب استفاده کنیم.

آشنایی با خصوصیات و متدهای شیء DBEngine

شیء DBEngine دارای خصوصیات و متدهای بسیاری است که در برنامه‌های کاربردی پیشرفته، بسیار مفید می‌باشند. جدول ۱-۶، لیست خصوصیات (Properties) شیء DBEngine را نشان می‌دهد.

هدف	خصوصیت
یک متغیر فقط نوشتنی (Write-Only) از نوع رشته String است که حداکثر کاراکترهای آن ۲۰ می‌باشد و مشخص کننده کاربر پیش فرض، برای تمام نشستهای محیطهای کاری (Workspaces Session) می‌باشد. در صورتی که این خصوصیت برای شیء DBEngine مشخص نشود، کاربر پیش فرض، "Admin" خواهد بود.	DefaultUser
یک متغیر فقط نوشتنی (Write-Only) از نوع رشته String است که حداکثر کاراکترهای آن ۱۴ می‌باشد و مشخص کننده کلمه عبور کاربر پیش فرض، برای تمام نشستهای محیطهای کاری (Workspaces Session) می‌باشد. در صورتی که این خصوصیت برای شیء DBEngine مشخص نشود، کلمه عبور کاربر پیش فرض خواهد بود.	DefaultPassword
از این خصوصیت برای تعیین مکان فایل حفاظت در محیط ۳۲ بیتی استفاده می‌شود.	SystemDB(32-bit)
برای تعیین آدرس فایل INI مورد استفاده در نسخه ۱۶ بیتی به کار برده می‌شود.	InIpath(16-bit)
در این خصوصیت مدت زمانی تنظیم و یا تعیین می‌شود که این زمان مدتی است که برنامه سعی می‌کند تا با منبع داده‌ها در ODBC ارتباط	LoginTimeOut

برقرار کند. این زمان بر حسب ثانیه می باشد و مقدار پیش فرض برای آن ۲۰ ثانیه است.

یک متغیر از نوع رشته و فقط خواندنی (Read-Only) می باشد که نشان دهنده نسخه Jet database engine است.

Version

خصوصیات شیء DBEngine باید قبل از باز شدن شیء محیط کاری تنظیم شود تا مقادیر آن مورد استفاده قرار گیرد. معمولاً عبارت `DBEngine.PropertyName = Value` در رویداد Load در فرم (Form_Load) قرار می گیرد.

جدول ۶-۲ لیست متدهای قابل اجرا را در شیء DBEngine نشان می دهد.

جدول ۶-۲: متدهای شیء DBEngine

هدف	متد
درب برنامه های کاربردی چند کاربره در زمانی که فعالیت پردازش اصلی در حال انجام است، به Jet engine این اجازه را می دهد که به طور همزمان مشغول عملیات پردازش پس زمینه شود.	Idle
یک پایگاه داده از نوع Jet را که در هیچ برنامه ای باز نیست، فشرده می سازد تا فضای کمتری را اشغال کند و یا یک نسخه از Jet به نسخه دیگر آن تبدیل می کند. توجه داشته باشید که از این متد برای تبدیل نسخه پایگاههای داده MDB که در برنامه های میکروسافت اکسس مورد استفاده هستند، استفاده نکنید.	CompactDatabase
این متد در صورت امکان، یک پایگاه صدمه دیده را درست می کند. روش استفاده از این متد به صورت زیر است:	RepairDatabase
<code>dbEngine.RepairDatabase(strPathFile)</code> مسیر و نام پایگاه داده می باشد.	
یک محیط کاری جدید، ایجاد می کند که دارای یک نام کاربر (User ID) مخصوص و کلمه عبور (Password) مخصوص می باشد.	CreateWorkspace
این متد نام منبع داده ODBC را ثبت می کند. در صورتی که نسخه شما ۳۲ بیتی باشد، عملیات ثبت در Registry ویندوز انجام می شود و در صورتی که نسخه شما ۱۶ بیتی باشد، این ثبت در فایل ODBC.INI صورت می گیرد.	RegisterDatabase

دستور کلی استفاده از CompactDatabase به صورت زیر است :

`DBEngine.CompactDataBase(strOldPathFile, StrNewPathfile, [intOption])`

اگر شما در هر دو قسمت `strOldPathFile` و `StrNewPathfile` یک مسیر و یک نام فایل را وارد کنید ، فایلی که فشرده می شود ، جایگزین فایل قبلی می گردد.

`intOption` آرگومانی است که می توان در آن ویرایش فایل خود را مشخص کنید و یا آنها را رمزدار کرده یا رمزگشایی نمود.

در جدول ۳-۶ حالت های مختلف این آرگومان ، آورده شده است.

جدول ۳-۶: حالت های مختلف در آرگومان `intOptions` از متد `CompactDatabase`.

ثابت	ارزش	هدف
dbVersion10	1	فایل موجود را به فایل Mdb اکسس 1.0 فشرده می کند.
dbVersion11	8	فایل موجود را به فایل Mdb اکسس 1.1 فشرده می کند.
dbVersion20	16	فایل موجود را به فایل Jet 2.5 اکسس 2.0 فشرده می کند.
dbVersion30	32	فایل موجود را به فایل Jet 3.0 اکسس ۹۵ فشرده می کند.
dbEncrypt	2	فایل فشرده شده را پنهان می کند.
dbDecrypt	4	فایل فشرده شده را آشکار می کند.

دستور کلی استفاده از `RepairDatabase` به صورت زیر می باشد:

`DBEngine.RepairDataBase(strPathfile)`

که در `strPathfile` مسیر و نام فایل MDB ذکر می گردد.

ایجاد شیء های Database

قبل از اینکه بتوانیم خصوصیات یک شیء نظیر شیء `Database` را تنظیم کنیم و یا از متدهای اجرایی آن استفاده کنیم ، باید با یک متغیر از نوع آن شیء ایجاد نماییم. در مورد شیء `Database` می توان هم آن را به صورت متغیر تعریف کرد و هم می توان از ابزار کنترل داده (`Data Control`) استفاده نمود. در اینجا ایجاد شیء را به وسیله تعریف متغیر انجام می دهیم. در مثال زیر متغیر `dbName` از نوع `Database` تعریف می شود و پس از آن به وسیله دستور `Set` ، این متغیر را به یک پایگاه نسبت می دهیم :

`Dim dbName As DataBase`

`Set dbName = OpenDatabase(ConnectArguments)`

پس از آنکه به این متغیر یک پایگاه داده نسبت داده شد، می توان خصوصیات آن را تنظیم کرد و یا متدهای آن را اجرا نمود. دستور کلی استفاده از این خصوصیات به صورت زیر می باشد:

`dbName.[PropertyName] = Value`

خصوصیات و منتهای شیء Workspace

شیء Workspace یک نشست (Session) از شیء DBEngine ایجاد می‌کند. این نشست، از زمان ورود کاربری با نام و کلمه عبور مخصوص و در حالت پیش فرض نام Admin و کلمه عبور خالی، ایجاد می‌شود و تا پایان خروج کاربر از سیستم، در DBEngine فعال می‌باشد و پس از آن از بین می‌رود. هر عملیاتی که کاربر در این زمان انجام دهد، محدود به محدودیتهای مخصوص آن کاربر می‌باشد که در سیستم حفاظت DBEngine تعریف شده است (در مورد حفاظت پایگاه داده به فصل دوازدهم مراجعه کنید).

دستورات OpenDatabase و CreateDatabase از منتهای شیء Workspace می‌باشند و برای اجرای آنها ابتدا باید محیط کاری مناسب را ایجاد کنید و سپس پایگاه داده خود را تحت آن محیط کاری ایجاد کرده و باز نمایید. در قسمت قبل دیدید که این کار بدون تعریف محیط کاری، مستقیماً بر روی متغیر از نوع پایگاه داده صورت گرفت که به معنی تعریف پایگاه داده تحت محیط کاری پیش فرض است؛ بنابراین تعریف پایگاه داده با رعایت اصول شیء گرایی به صورت زیر است.

```
Dim wsName AS Workspace
Dim dbName AS Database
Set wsName = DBEngine.Workspaces(0)
Set dbName = wsName.OpenDatabase(ConnectArguments)
```

جدول ۴-۶ لیست منتهای شیء Workspace را نشان می‌دهد.

جدول ۴-۶: منتهای شیء Workspace.

نام متد	هدف
OpenDatabase	یک پایگاه داده از پیش ایجاد شده و تحت پشتیبانی Jet را باز می‌کند.
CreateDatabase	یک پایگاه داده از نوع Jet Database را تحت نسخه‌ای که در آرگومان iniOption مشخص می‌شود، ایجاد می‌کند.
CreateGroup	یک شیء جدید از نوع Group را برای محیط کاری ایجاد و به مجموعه Groups اضافه می‌کند.
CreateUser	یک شیء جدید از نوع User را برای محیط کاری ایجاد می‌کند و به مجموعه Users اضافه می‌نماید.
RollBack	در صورتی که پایگاه داده از Transaction پشتیبانی کند (dbName.Transaction=True)، به روز رسانی که در لیستهای داده‌ای و یا جداول صورت می‌گیرد، نادیده گرفته می‌شود.
Close	تحت این محیط کاری باز باشند با اجرای این متد، آنها

نیز محیط کاری مشخص شده را می بندد و اگر پایگاه‌های داده بسته خواهند شد.

خصوصیات شیء Database

شیء Database دارای یازده خصوصیت می باشد که اکثر این خصوصیات فقط خواندنی (Read-Only) می باشند و به وسیله ارزش آنها می توان مشخصه‌های این شیء را تعیین کرد. جدول ۵-۶ این خصوصیات را نشان می دهد.

جدول ۵-۶: خصوصیات شیء Database.

هدف	خصوصیت
یک متغیر رشته ای از مسیر و نام پایگاه داده را برمی گرداند.	Name
نوع پایگاه داده ارتباط داده شده را در یک متغیر رشته ای برمی گرداند. در صورتی که این پایگاه داده از نوع اکسس باشد ، مقدار برگشتی این خصوصیت تهی است.	Connect
نوع باز شدن پایگاه داده را از جهت خواندن / نوشتن (Read/Write) و یا فقط خواندنی (Read-Only) مشخص می کند. ارزش این خصوصیت به صورت منطقی می باشد و در حالت اول ، مقدار آن True و در حالت دوم مقدار آن False است.	Updatable
مشخص می کند که این پایگاه داده دارای امکان RollBack باشد یا خیر.	Transaction
مدت زمانی است که برای اجرای یک سؤال در پایگاه های داده Client/Server تحت درایور ODBC سعی می شود. این زمان بر حسب ثانیه می باشد و مقدار پیش فرض آن ۶۰ ثانیه است.	QueryTimeOut
برای مقایسه فیلدها، زبان مرتب شدن پایگاه داده را مشخص می کند و حالت پیش فرض آن به صورت زیر است :	CollatingOrder
English and most western European Language	
نگارش نسخه پایگاه داده را مشخص می کند و یک خصوصیت فقط خواندنی است.	Version
وقتی که این خصوصیت True باشد ، یک پایگاه داده با قابلیت تکثیر سازی ایجاد می شود که می توان هر نسخه از آن را در اختیار یکی از کاربران ، قرار داد. وقتی این خصوصیت True باشد ، خصوصیت Replica نیز مقدار True می گیرد و نمی توان آن را False کرد.	Replicable
یک کد عمومی مجزا (GUID) است که نشان می دهد کدام کاربر از روی پایگاه داده قابل تکثیر سازی ، نسخه دومی تهیه کرده است.	DesignMasterID
یک کد عمومی مجزا (GUID) است که به وسیله آن می توان نسخه المثنی را تشخیص داد.	ReplicaID
در صورتی که این خصوصیت True باشد ، طول رشته های صفر ("") در اکسس نسخه یک ، تبدیل به مقادیر Null می شود.	V1xNullBehavior

خصوصیتهای Replicable ، ReplicaID ، DesignMasterID در جدول ۵-۶ مخصوص نسخه Jet 3.0 (Access 95) به بعد می باشد و جزء خصوصیات عمومی شیء Database نمی باشد.

از ارزش خصوصیت CollatingOrder ، همان طور که کد زیر نشان می دهد می توان در آرگومان intCompare از توابع Instr() و StrComp() استفاده نمود. راه دیگر برای مشخص کردن نوع مقایسه بین فیلدها ، استفاده از دستور Option در ابتدای قسمت تعریف رویه است.

```
Dim intCompar As Integer
Dim intInStr As Integer
Dim varStrComp As Variant
```

```
intCompare = dbName.CollatingOrder
intInStr = Instr(StrSource, StrTest, intCompare)
varStrComp = StrComp(Str1, Str2, intCompare)
```

متدهای اجرایی در شیء Database

تعداد متدهای اجرایی شیء Database از خصوصیات آن بیشتر است اما بسیاری از این متدها که در ویژوال بیسیک ۳ جزء شیء Database بود و اکنون جزء شیء Workspace می باشد. دو متد OpenRecordset و CreateQueryDef جزء متدهای شیء Database می باشد زیرا شیء داده می تواند عملیات خود را تحت یک پایگاه داده روی بیش از یک جدول انجام دهد. متدهایی را که می توان تحت شیء Database اجرا کرد در جدول ۶-۶ نشان داده شده است.

هدف	نام متد
این متد با استفاده از دستور Set باعث ایجاد یک شیء TableDef می شود و این شیء پس از تنظیمات لازم به یک جدول از پایگاه داده تبدیل می گردد.	CreateTableDef
این متد با استفاده از دستور Set، ارتباط بین فیلدهای کلیدی جداول اصلی و فیلدهای خارجی جداول ارتباط داده شده را برقرار می کند.	CreateRelation
این متد با استفاده از دستور Set، یک شیء لیست داده جدید (New Recordset) از نوع Table، Dynaset و یا Snapshot ایجاد می کند. این متد بیشترین کاربرد را در بین متدهای شیء Database داراست.	OpenRecordset
مجموعه اشیا موجود در شیء Database مانند TableDef و QueryDef را به روز رسانی (Update) می کند.	Refresh
این متد با استفاده از دستور Set باعث ایجاد یک شیء QueryDef بر مبنای دستورات SQL می شود. در فصل هشتم با SQL آشنا خواهید شد.	CreateQueryDef
این متد باعث اجرای QueryDef هایی می شود که مجموعه رکوردی را بر نمی گردانند و اصطلاحاً سؤالهای اجرایی نام دارند.	Execute
یک خصوصیت جدید ایجاد می کند تا کاربر به طور دلخواه از این مشخصات به طور خاص استفاده نماید.	CreateProperty
این متد باعث ایجاد یک نسخه بدل از پایگاه داده برای کاربر می شود و در صورتی قابل اجراست که پایگاه داده دارای این خصوصیت باشد. به وسیله آرگومانهای این متد می توان مسیر و نام این پایگاه داده را مشخص می کرد و یا تعیین کرد که این پایگاه داده فقط خواندنی باشد و یا قابلیت به روز رسانی را نیز داشته باشد.	MakeReplica
به وسیله این متد می توان شیء پایگاه داده اصلی را با نسخه بدلی که مسیر و نام آن مشخص می شود، در تغییرات، مطابقت نمود. به وسیله آرگومان intExchange در این متد می توان مشخص کرد که تغییرات دریافت و یا ارسال شود و در حالت پیش فرض به روز رسانی را دو طرفه انجام داد.	Synchronize
شیء Database را می بندد و مراجع مصرف آن را نیز آزاد می سازد.	Close

ارتباط با یک پایگاه داده از نوع Jet

دستور کامل متد () OpenDatabase برای ارتباط با یک پایگاه داده موجود به صورت زیر می باشد.

```
Set dbName = [WorkspaceName.]OpenDatabase(strDBPath["\" & _  
strDBFile.ext][, fExclusive [, fReadOnly[, strConnect]])
```

جدول ۶-۷ آرگومانهای متد () OpenDatabase را توضیح می دهد.

جدول ۶-۷: آرگومانهای متد OpenDatabase.

آرگومان	هدف
strDBPath	در این قسمت مسیر کامل شامل مسیر درایو و شاخه‌ای که فایل پایگاه داده در آن وجود دارد، قرار می‌گیرد که منبع ODBC شامل این قسمت نیست. اگر پایگاه داده و جدولهای آن و ایندکسها از نوع فاکس پرو، پاراداکس و یا Xbase باشد دادن مسیر و شاخه ای که این فایلها در آن قرار گرفته، کافی است. تمام فایل‌های این نوع پایگاه‌های داده باید در یک شاخه قرار بگیرد و از طریق این دستور شاخه به عنوان یک Database باز می‌شود و فایل‌های DBF و ایندکسها به عنوان عناصر این Database شناخته می‌شوند. شما می‌توانید به روش Uniform Naming convention (UNC) آدرس پایگاه‌های داده موجود در شبکه را در این قسمت وارد کنید. برای مثال شما می‌توانید به طریق زیر عمل کنید:
strDBFile	//Servername { \folder \Sharename } برای اتصال به فایل‌های پایگاه داده در مسیر ODBC API، از نام مسیر اطلاعات (DSN) که در \WINDOWS\ODBC.INI ثبت شده است استفاده کنید. اگر پایگاه داده از نوع Jet MDB باشد، در قسمت strDBPath و strDBFile آدرس و نام فایل را وارد نمایید.
fExclusive	یک Flag از نوع Integer می‌باشد که نشان می‌دهد که پایگاه داده به صورت Exclusive (True) باز شده است یا به صورت Shared (False) حالت Exclusive باز کردن پایگاه داده به صورت منفرد می‌باشد و حالت Shared برای استفاده از پایگاه داده در محیط‌های چند کاربره است. حالت پیش فرض در این آرگومان حالت Shared (False) است.
fReadOnly	یک Flag از نوع Integer است که نشان می‌دهد پایگاه داده به صورت فقط خواندنی (True) باز شود و یا خواندنی و نوشتنی (False). حالت پیش فرض این آرگومان False است.
strConnect	یک رشته بوده و شامل نوع پایگاه داده ای است که قرار است توسط متد OpenDatabase باز شود. این دستور برای پایگاه‌های داده Jet (اکسس) نمی‌باشد و برای دیگر پایگاه‌های داده باید تنظیم شود.

مثال ۱: کد زیر باعث می‌شود که پایگاه داده Bilbli.mdb با دسترسی فقط خواندنی (Read-only) و تک کاربره باز شود.


```
Dim dbBilio As DataBase
```

```
Set dbBilio = DBEngine.Workspaces(0).OpenDatabase("Biblio.Mdb",True,false)
```

مثال ۲: کد زیر یک پایگاه داده جدید از نوع Jet 3.0 (اکسس ۹۵) در شاخه جاری می‌سازد.

```
Dim dbnew As DataBase
```

```
Set dbnew = DBEngine.Workspaces(0). _
```

```
CreateDatabase("New.Mdb", dbLangGeneral,dbVersion30)
```

برای اینکه بتوانیم از یک پایگاه داده جدید استفاده کنیم باید به قسمت TableDef پایگاه داده ، جداول را اضافه کرده و سپس فیلدها و ایندکسها را به آنها بیافزاییم. در قسمت بعد مشخصات شیء TableDef را بررسی می‌کنیم.

استفاده از مجموعه TableDefs و شیء TableDef

تمام جداول یک پایگاه داده با شیء Table Def شناخته می‌شوند و این شیء ، یکی از اعضای مجموعه TableDefs می‌باشد. جدول ۸-۶ خصوصیات شیء TableDef را نشان می‌دهد.

جدول ۸-۶: خصوصیات شیء TableDef.

هدف	خصوصیت
نام جدول در این خصوصیت قرار می‌گیرد.	Name
برای ضمیمه کردن (Attach) جداول Jet database مورد استفاده قرار می‌گیرد. شما فقط می‌توانید جداول MDB را ضمیمه کنید.	Connect
نام جدول ضمیمه شده در این خصوصیت قرار می‌گیرد. شما می‌توانید توسط خصوصیت Name یک نام دیگری برای جدول ضمیمه شده خود انتخاب کنید.	SourceTableName
یک شرط برای محدود کردن رکوردهای جدول می‌باشد که همانند دستور WHERE در کدهای SQL عمل می‌کند. برای مثال در صورتی که از شرط InvoiceDate <= OrderDate در این قسمت استفاده کنیم رکوردهایی از جدول انتخاب می‌شوند که فیلد InvoiceDate در آنها از فیلد OrderDate کوچکتر یا مساوی باشند. اگر خواسته باشید از شرط‌های بیشتری استفاده کنید می‌توانید از اوپراتورهای ترکیبی نظیر And استفاده کنید.	ValidationRule
در این خصوصیت یک متن قرار می‌گیرد و در صورتی که به جدول	ValidationText

رکوردهایی اضافه کرده باشید که در قانون و شرط ValidationRule صدق نکند در زمان به روز رسانی جدول، این متن به صورت پیغام ظاهر می شود.

نام جدول جانبی ای را برمی گرداند که در زمان عملیات، با آن کار مطابقت می کند. این خصوصیت از قابلیت های Jet 3.0 به بعد می باشد. تاریخ و زمان ساخت جدول را برمی گرداند و این خصوصیت فقط خواندنی (Read-only) است.

تاریخ و زمان آخرین تغییر را برمی گرداند و این خصوصیت فقط خواندنی (Read-only) می باشد.

مقدار این خصوصیت در صورتی که دسترسی به آن به صورت خواندن و نوشتن Read/Write باشد False است و در صورتی که فقط خواندنی (Read-Only) باشد True است.

یک مقدار از نوع Long Integer است و انواع مقادیر آن در جدول ۶-۹ ذکر شده است.

ConflictTable

DateCreate

LastUpdated

Updatable

Attributes

جدول ۶-۹ لیست مقادیر خصوصیت Attributes را از شیء TableDef نشان می دهد.

جدول ۶-۹: لیست مقادیر خصوصیت Attribute از شیء TableDef.

توضیح	ارزش	Attribute
بیان می کند که جدول از نوع جداول سیستمی است.	-2147483648 &H 80000002	DbSystemObject
نشان می دهد که جدول پنهان به صورت موقتی می باشد که Jet برای استفاده از آن، آن را درست کرده است.	1 &H1	DbHiddenObject
نشان می دهد جدول از نوع رومیزی است (به صورت ODBC نمی باشد) و به یک پایگاه داده ای اکسس ضمیمه شده است. در صورتی که جداول ضمیمه شده از نوع جداول اکسس باشند، به صورت Shared باز خواهند شد.	1073741824 &H40000000	dbAttachedTable
نشان می دهد که جدول ضمیمه شده به صورت تک کاربره (Exclusive) باز شده است.	65536 &H10000	dbAttachedExclusive

نشان می‌دهد جدول ، یک جدول ODBC است که به پایگاه داده اکسس ضمیمه شده است.

536870912
&H20000000

dbAttachedODBC

نشان می‌دهد که نام کاربر و کلمه عبور برای پایگاه داده‌ای که جداول ضمیمه شده از آن انتخاب شده‌اند، در خصوصیت Connect قرار گرفته است. این بدان معنی است که دیگر کاربر برنامه در زمان ارتباط با منبع جداول ضمیمه شده، نیازی به وارد کردن نام کاربر و کلمه عبور برای دسترسی به آنها ندارد.

131072
&H20000

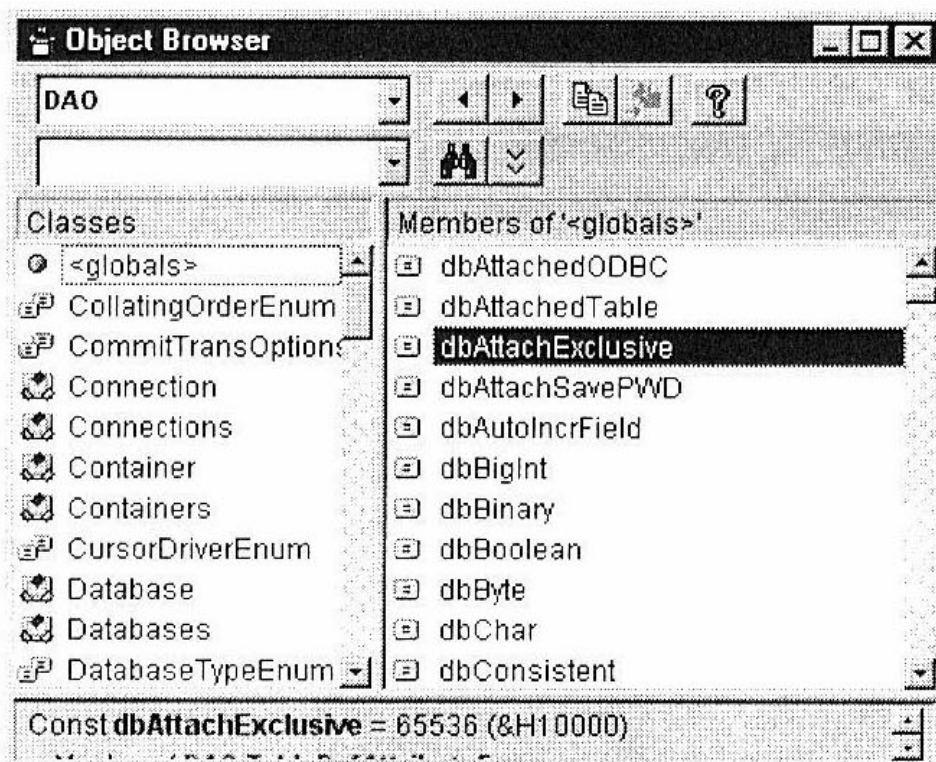
dbAttachSavePWD

آشنایی با Flags و ثوابت نشانه‌ای (Symbolic Constant)

در بیشتر برنامه‌های تحت ویندوز معمولاً Flag ها دارای ثوابتی عمومی هستند که به طور نشانه‌ای می‌توانند به جای کد مخصوص آن Flag ها به کار روند. این ثوابت Symbol نامیده می‌شوند. در ویژوال بیسیک می‌توان ثوابتی را به طور دلخواه تعریف کرد و به آن مقادیری را اختصاص داد. مکان تعریف این ثوابت در قسمت تعاریفات ماژول می‌باشد. برای این کار می‌توان از دستور زیر استفاده نمود.

Public Const Symbol = Value [As DataType]

ثوابت نشانه‌ای برای شیء دسترسی به داده (Data Access Object) در ویژوال بیسیک در مجموعه Constants این شیء قرار گرفته‌اند و همگی با db آغاز می‌شوند. استفاده از این ثوابت در کدهای برنامه بسیار مفید می‌باشد زیرا باعث خواناتر شدن آن می‌شود. شکل ۱-۶ کتابخانه شیء DAO را در مرورگر شیء نشان می‌دهد.



شکل ۱-۶.

کتابخانه شیء DAO در مرورگر شیء.

در صورتی که بخواهید بیشتر از یک Attribute در جدول استفاده کنید، باید از اوپراتور Or برای ثوابت و + برای مقادیر آنها استفاده نمایید. برای مثال در صورتی که بخواهید جدول ضمیمه شده از کلمه عبور پایگاه داده‌ای که در خصوصیت Connect استفاده کند (&H20000) و نیز این جدول به صورت Exclusive باز شود (&H10000) باید در خصوصیت Attribute از مقدار (&H30000) استفاده کرد. این مقدار توسط دستور زیر نیز تنظیم می‌گردد.

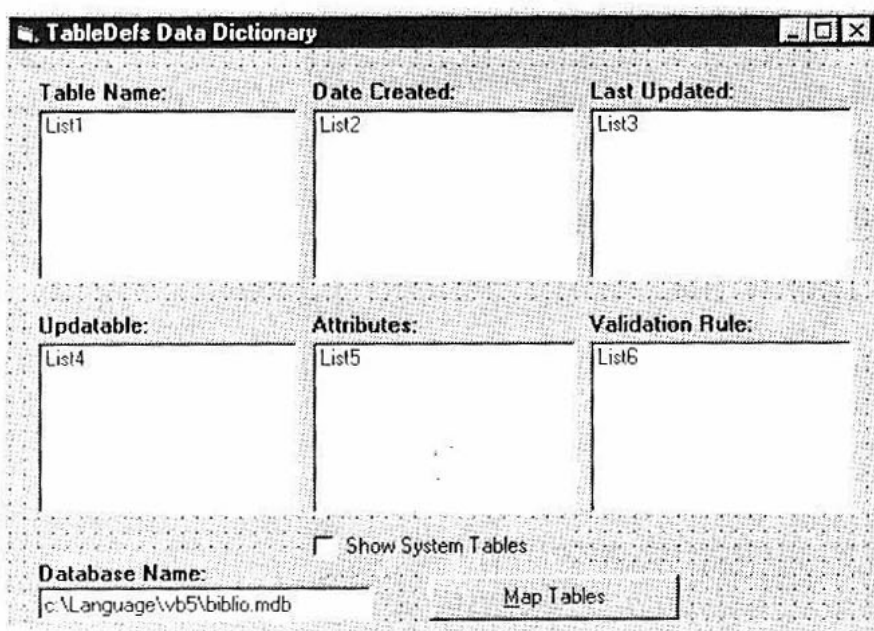
dbName.Attributes = dbAttachedSavePWD Or dbAttachExclusive

مشخص کردن اعضای اشیای پایگاه داده به وسیله مجموعه TableDefs

با استفاده از مجموعه TableDefs و اشیای زیر مجموعه آن می‌توان خصوصیات و اجزای جداول را از یک پایگاه داده به دست آورد. به دست آوردن جداول و خصوصیات آن اولین مرحله در ایجاد برنامه Data Dictionary for Database است. با استفاده از این برنامه می‌توانید هر پایگاه داده‌ای که توسط ویژوال بیسیک پشتیبانی می‌شود باز کرده و خصوصیات شیء TableDef را مشاهده نمایید. در قسمت بعد شروع به بررسی این برنامه می‌کنیم.

برنامه کاربردی Data Dictionary for Database

مجموعه TableDefs همانند سایر اشیای زیر مجموعه شیء Database دارای خصوصیتی به نام Count است. این خصوصیت تعداد اعضای مجموعه TableDefs را که جداول می‌باشند، بر می‌گرداند. با استفاده از این عدد می‌توان خانه های آرایه ای را که در هر یک از آنها شیء TableDef قرار دارد، کنترل نمود. در کد زیر با استفاده از این خصوصیت، یک حلقه For..Next تشکیل داده ایم که به تعداد اعضای مجموعه TableDefs تکرار می‌شود و نام هر یک از جداول و مشخصات آنها را در چند ListBox که در فرمی به نام frmTableDefs قرار دارد، وارد می‌کند. شکل ۲-۶ فرم ابتدایی این برنامه را در قسمت طراحی و لیست ۱-۶ کد قسمت اول این برنامه را نشان می‌دهد.



شکل ۲-۶.

فرم ابتدایی برنامه کاربردی
Data Dictionary

Option Explicit

'Form-level variables

```
Dim strTableName As String 'Name of selected table
Dim flsFields As Boolean 'Flag, True for Fields
Dim flsIndexes As Boolean 'Flag, True for Indexes
Dim dbMap As Database 'Declare a database object variable
Dim intCtr As Integer 'declare a general - purpose counter
```

```
Private Sub Command1_Click()
```

```
    Set dbMap = DBEngine.Workspaces(0).OpenDatabase(Text1.Text)
    Call TableProc
```

```
End Sub
```

```
Private Sub TableProc()
```

```
    On Error GoTo TDError 'Set error handler
```

```
    flsFields = False
    flsIndexes = False
```

```
    Label1.Caption = "Table Name: "
```

```
    Label2.Caption = "Date Created: "
```

```
    Label3.Caption = "Last Updated: "
```

```
    Label4.Caption = "Updatable: "
```

```
    Label5.Caption = "Attributes: "
```

```
    Label6.Caption = "Validation Rule: "
```

```
    frmTableDefs.Caption = " TableDefs Data Dictionary
```

```
Call ClearList
```

```
    Fill the listBoxes , selectively display system objects
```

```
For intCtr = 0 To dbMap.TableDefs.Count - 1
```

```
    If (dbMap.TableDefs(intCtr).Attributes And_
```

```
        dbSystemObject) = 0 Or Check1.Value Then
```

```

List1.AddItem dbMap.TableDefs(intCtr).Name
List2.AddItem dbMap.TableDefs(intCtr).DateCreated
List3.AddItem dbMap.TableDefs(intCtr).LastUpdated
List4.AddItem dbMap.TableDefs(intCtr).Updatable
List5.AddItem "&H" & Hex$(dbMap.TableDefs(intCtr).Attributes)
List6.AddItem dbMap.TableDefs(intCtr).ValidationRule

```

```

End If
Next intCtr
Exit Sub

```

TDError:

```

' provide an standard error handler
MsgBox Error$, , "Tabledefs Error"
Exit Sub
End Sub

```

Private Sub ClearList()

```

List1.Clear
List2.Clear
List3.Clear
List4.Clear
List5.Clear
List6.Clear

```

End Sub

هنگامی که مسیر و نام یک پایگاه داده را در جعبه متن (Text1) وارد کنید، با زدن دکمه فرمان Map Tables شش خصوصیت از شیء Tabledef در جعبه‌های لیست وارد می‌شوند و با انتخاب هر جدول از جعبه لیست اول، دیگر مشخصات آن جدول در بقیه جعبه‌های لیست مشخص می‌گردند. شکل ۳-۶ فرم frmTableDefs را در زمان اجرا روی پایگاه داده Biblio.MDB نشان می‌دهد. جعبه لیست Validation Rule هیچ چیز را نشان نمی‌دهد زیرا این خصوصیت در هیچ یک از جداول وارد نشده است.

Table Name:	Date Created:	Last Updated:
Authors	7/25/96 11:38:02 AM	7/25/96 11:40:18 AM
Publishers	7/25/96 11:38:09 AM	7/25/96 12:06:21 PM
Title Author	7/25/96 11:38:10 AM	7/25/96 11:38:36 AM
Titles	7/25/96 11:38:25 AM	7/25/96 12:09:22 PM

Updatable:	Attributes:	Validation Rule:
True	&H0	
True	&H0	
True	&H0	
True	&H0	

Show System Tables

Database Name: c:\Language\vb5\biblio.mdb Map Tables

شکل ۳-۶.

برنامه کاربردی

Data Dictionary در زمان اجرا.

در صورتی که گزینه Show System Tables را در این فرم انتخاب کنید و آنگاه دکمه فرمان Map Tables را فشار دهید، تعدادی جدول به لیست جداول شما افزوده می شود که همگی آنها با Msys... شروع می شوند (شکل ۴-۶). این جداول، جداول سیستمی می باشند که در خود مایکروسافت اکسس مورد استفاده است. برای مثال جدول MSysMacros مربوط به نگهداری ماکروها در پایگاه داده MDB می باشد که دسترسی به این ماکروها فقط از طریق مایکروسافت اکسس امکان پذیر است.

Table Name:	Date Created:	Last Updated:
MSysModules	7/25/96 11:10:20 AM	7/25/96 11:10:25 AM
MSysObjects	12:00:00 AM	12:00:00 AM
MSysQueries	12:00:00 AM	12:00:00 AM
MSysRelationships	12:00:00 AM	12:00:00 AM
MSysToolbars	7/25/96 10:53:12 AM	7/25/96 10:53:12 AM
Publishers	7/25/96 11:38:09 AM	7/25/96 12:06:21 PM
Title Author	7/25/96 11:38:10 AM	7/25/96 11:38:36 AM

Updatable:	Attributes:	Validation Rule:
True	&H2	
True	&H80000000	
True	&H80000000	
True	&H80000000	
True	&H2	
True	&H0	
True	&H0	

Show System Tables

Database Name: c:\Language\vb5\biblio.mdb Map Tables

شکل ۴-۶.
نمایش جداول سیستمی.

در ادامه با خصوصیات فیلدها و ایندکسها آشنا می شویم و سپس از آنها در برنامه Data Dictionary برای مشخص کردن فیلدها و ایندکسهای یک جدول استفاده خواهیم کرد.

مجموعه فیلدها و ایندکسها

مجموعه Recordser, QueryDef, GtableDef همگی دارای زیر مجموعه Field می باشند. جدول

۱۰-۶ لیست مشخصات شیء Field را نشان می دهد.

توضیح	خصوصیت
اگر در حالت True است این فیلد می‌تواند در جدول با طول رشته صفر باشد و اگر False باشد حداقل یک کاراکتر باید در آن وارد شود.	AllowZeroLength
مجموعه‌ای Flag می‌باشد که مشخصه‌های فیلد را تعیین می‌کند. لیست این Flag ها در جدول ۱۲-۶ آورده شده است.	Attributes
نوع Sort فیلدهای متنی را مشخص می‌نماید.	CollatingOrder
اگر True باشد، فیلد می‌تواند به روز رسانی شود. در غیر این صورت این قابلیت را ندارد.	DataUpdatable
یک مقدار ثابت است که در هنگام ایجاد رکورد جدید به طور خودکار در فیلد قرار می‌گیرد و پیش فرض آن می‌شود.	DefaultValue
در هنگام ارتباط با یک جدول دیگر، این نام به فیلد ارتباطی اختصاص داده می‌شود.	ForeignName
نام فیلد را برمی‌گرداند. در مورد فیلدهایی که در یک QueryDef استفاده شده‌اند، در این خصوصیت نام پس از دستور As در کد SQL قرار می‌گیرد.	Name
شماره فیلد جاری را در جدول نشان می‌دهد و از شماره ۱ شروع می‌شود. اگر این خصوصیت True باشد این فیلد نباید دارای مقدار تهی (Null) باشد.	OrdinalPosition
اندازه فیلد را به Byte بر می‌گرداند.	Required
نکته: برای فیلدهای از نوع Memo و OLE مقدار این خصوصیت صفر می‌باشد.	Size
در صورتی که فیلد از جدول ضمیمه شده انتخاب شده باشد، نام اصلی فیلد در این مکان قرار می‌گیرد.	SourceField
نام جدول فیلد ضمیمه شده را بر می‌گرداند.	SourceTable
نوع فیلد را مشخص می‌کند. جدول ۱۱-۶ مقادیر این خصوصیت را نشان می‌دهد.	Type
اگر True باشد شرط ValidationRule را در حین ورود چک می‌کند در غیر این صورت پس از خارج شدن از فیلد این شرط چک می‌شود.	ValidateOnSet

ValidationRule	شرطی است که فیلد باید مطابق آن مقدار دهی شود و عملاً مانند دستور Whrer در SQL عمل می کند.
ValidationText	پیغامی است که در صورت صادق نبودن شرط ValidationRule ظاهر می شود.

جدول ۱۱-۶

جدول ۱۱-۶: مقادیر خصوصیت Attributes در شیء Fields.

Attributes	ارزش	توضیح
dbFixedfield	&H1	نشان می دهد که طول فیلد ثابت است و فیلد Date/Time.Text, Binary, Memo و یا Long نمی باشد.
dbVariableField	&H2	نشان می دهد طول فیلد تغییرپذیر است.
dbAutoIncrField	&H10	نشان می دهد که فیلد در حالت Conter قرار دارد.
dbUpdatablefield	&H20	نشان می دهد که ساختار فیلد قابلیت به هنگام شدن را داراست.

لیست مقادیر مشخصه Type از شیء فیلد

جدول ۱۲-۶: مقادیر خصوصیت Type در شیء Fields.

نام نوع داده	ارزش
dbBoolean	1
dbByte	2
dbInteger	3
dblong	4
dbCurrency	5
dbSingle	6
dbDouble	7
dbDate	8
dbBinary	9
dbText	10
dbLongBinary	11
dbMemo	12

مجموعه ایندکسها تنها جزء اعضای مجموعه TableDef می باشد. جدول ۱۳-۶ لیست خصوصیات شیء Index را نشان می دهد.

جدول ۱۳-۶: خصوصیات شیء Index.

نام خصوصیت	توضیح	نوع داده
Name	نام ایندکس جدول را نشان می دهد.	String
fields	نام فیلدهای یک ایندکس را نشان می دهد.	String
Foreign	اگر ایندکس دارای فیلد کلیدی خارجی باشد این خصوصیت True است.	boolean
Required	اگر ایندکس روی فیلدی است که مشخصه Required آن True باشد مقدار این خصوصیت True می گردد.	boolean
Primary	True است اگر که ایندکس روی فیلد کلیدی اصلی باشد.	boolean

لیست ۲-۶ ادامه برنامه Data Dictionary را نشان می دهد. در این کد دو رویه (FieldsProc) و IndexesProc() را به برنامه اضافه می کنیم تا مشخصه های فیلدها و ایندکسها را نیز در فرم خود به نمایش در آوریم.

لیست ۲-۶: ادامه برنامه Data Dictionary.

```
Private Sub FieldsProc()
' Display the properties of the selected field
Dim fldMap As Fields
On Error GoTo FieldsError 'Set error handler

strTableName = List1.List(List1.ListIndex)

Call ClearList

Label1.Caption = "Field Name: "
Label2.Caption = "Data Type: "
Label3.Caption = "Maximum Size: "
Label4.Caption = "Collating Order: "
Label5.Caption = "Attributes: "
Label6.Caption = "Position: "
frmTableDefs.Caption = "Fields Collection Data Dictionary for " & strTableName
Set fldMap = dbMap.TableDefs(strTableName).Fields
For intCtr = 0 To fldMap.Count - 1 'Fill the list Boxes
List1.AddItem fldMap(intCtr).Name
```



```

List2.AddItem strFieldType(CVar(fldMap(intCtr).Type))
List3.AddItem fldMap(intCtr).Size
List4.AddItem fldMap(intCtr).CollatingOrder
List5.AddItem "&H" & Hex$(fldMap(intCtr).Attributes)
List6.AddItem fldMap(intCtr).OrdinalPosition
Next intCtr
Exit Sub
FieldsError:
' Provide an standard error handler
MsgBox Error$, , "Field Error"
Exit Sub
End Sub

```

```

Private Sub IndexesProc()

```

```

' Display the properties of the selected index
Dim ixsMap As Indexes

On Error GoTo IndexesError 'set error handler
Call ClearList
' Use the Indexes Collection
frmTableDefs.Caption = "Indexes Collection Data Dictionary for " & strTableName
Label1.Caption = "Index Name: "
Label2.Caption = "Fields: "
Label3.Caption = "Primary: "
Label4.Caption = "Unique: "
Label5.Caption = "Ignore Nulls: "
Label6.Caption = "Foreign: "

Set ixsMap = dbMap.TableDefs(strTableName).Indexes
For intCtr = 0 To ixsMap.Count - 1 'Fill the list Boxes
List1.AddItem ixsMap(intCtr).Name
List2.AddItem ixsMap(intCtr).Fields
List3.AddItem ixsMap(intCtr).Primary
List4.AddItem ixsMap(intCtr).Unique
List5.AddItem ixsMap(intCtr).IgnoreNulls
List6.AddItem ixsMap(intCtr).Foreign
Next intCtr

Exit Sub

IndexesError:
' Provide an standard error handler
MsgBox Error$, , "Indexes Error"

```

```
Exit Sub
End Sub
```

```
Private Sub List1_Click()
    List2.ListIndex = List1.ListIndex
    List3.ListIndex = List1.ListIndex
    List4.ListIndex = List1.ListIndex
    List5.ListIndex = List1.ListIndex
    List6.ListIndex = List1.ListIndex
End Sub
```

```
Private Sub List1_DblClick()
' Choose the procedure to call based on the data
' presently displays
If flsFields Then
    Call IndexesProc
    flsIndexes = True
    flsFields = False
Else
    If flsIndexes Then
        Call TableProc
    Else
        Call FieldsProc
        flsIndexes = False
        flsFields = True
    End If
End If
End Sub
```

```
Public Function strFieldType(IngType As Variant)
' return the string value of field data type
Select Case IngType
    Case dbBoolean
        strFieldType = "Boolean (Yes/No)"
    Case dbByte
        strFieldType = "Byte (tinyint)"
    Case dbInteger
        strFieldType = "Integer (smallint)"
    Case dbLong
        strFieldType = "Long Integer (int) "
    Case dbCurrency
        strFieldType = "Currency"
```

```

Case dbSingle
    strFieldType = "Single-precision"
Case dbDouble
    strFieldType = "Double-precision"
Case dbDate
    strFieldType = "Date/Time"
Case dbText
    strFieldType = "Text (string) "
Case dbLongBinary
    strFieldType = "Long Binary"
Case dbMemo
    strFieldType = "Memo"
End Select
End Function

```

هنگامی که روی نام یک جدول دوبار کلیک کنید رویه () FieldsProc اجرا می‌شود و فیلدهای آن جدول به همراه خصوصیت‌های آنها نشان داده می‌شود (شکل ۵-۶) و در صورتی که روی نام فیلد دوبار کلیک نمایید رویه () IndexesProc اجرا می‌شود و ایندکس‌های آن فیلد به همراه مشخصه‌های آنها نشان داده می‌شود (شکل ۶-۶).

Field Name:	Data Type:	Maximum Size:
Au_ID	Long Integer (int)	4
Author	Text (string)	50
Year Born	Integer (smallint)	2

Collating Order:	Attributes:	Position:
1033	&H1	0
1033	&H2	1
1033	&H1	2

Database Name: c:\Language\vb5\biblio.mdb

شکل ۵-۶.

نمایش مجموعه فیلدها همراه با خصوصیات آنها در برنامه Data Dictionary

Index Name:	Fields:	Primary:
AuthorsAu_ID	+Au_ID	False
PrimaryKey	+Au_ID	True

Unique:	Ignore Nulls:	Foreign:
False	False	False
True	False	False

Database Name: c:\Language\vb5\biblio.mdb

شکل ۶-۶.

نمایش مجموعه ایندکسها همراه با خصوصیات آنها در برنامه Data Dictionary

استفاده از مجموعه QueryDefs و شیء QueryDef

شما می‌توانید از طریق دستور زیر یک شیء QueryDef بسازید:

```
Dim qdName As QueryDef
```

```
Set qdName = dbName.CreateQueryDef([StrName][, StrSQL]
```

اگر StrName را از دستور فوق حذف کنیم یا از ("") استفاده نمائیم، QueryDef ذخیره نمی‌شود و یک QueryDef موقتی خواهد بود و اگر StrSQL را حذف کنیم، یک Query Def با نامی مخصوص ذخیره کرده‌ایم که باید قبل از استفاده آن خصوصیت SQL آن را تنظیم کنیم. در مثال زیر یک QueryDef با نام qryPubs ساخته می‌شود و در زمان اجرا، تمام رکوردهای جدول Publishers را از Biblio.mdb فعال می‌سازد.

```
Dim qdfPubs As QueryDef
```

```
Set qdfPubs = dbBiblio.CreateQueryDef("qdfPubs", "SELECT * From Publishers")
```

بسیاری از خصوصیات اشیای QueryDef با Tabele Def مشترک می‌باشد. جدول ۱۴-۶ خصوصیات شیء QueryDef را نشان می‌دهد. دو خصوصیت Name و SQL در اول این جدول آمده است زیرا بیشترین کاربرد را در این شیء دارا هستند. QueryDef هم مانند TableDef شامل مجموعه Field می‌باشد که مشخصات اشیای این مجموعه را در قسمتهای قبل آوردیم. بجز چند مورد بقیه خصوصیات شیء Field در QueryDef با TableDef یکسان است و آن موارد شامل SourceTable SourceField Name می‌باشد که در اینجا Name نام فیلدی است که توسط SQL داده می‌شود و SourceField نام فیلد اصلی و SourceTable نام جدول اصلی می‌باشد که در SQL استفاده شده است.

جدول ۱۴-۶: خصوصیات شیء QueryDef.

توضیح	خصوصیت
نام سؤال را تنظیم می‌کند و یا بر می‌گرداند. حالت پیش فرض برای این خصوصیت تهی است.	Name
کد SQL ای می‌باشد که در سؤال اجرا می‌گردد.	SQL
فقط در موارد ایجاد سؤال از نوع Passthrough استفاده می‌شود. در ابتدای رشته این خصوصیت، وجود الزامی است.	Connect
تاریخ ایجاد سؤال را مشخص می‌کند.	DataCreated
تاریخ آخرین به روز رسانی را در سؤال مشخص می‌نماید.	LastUpdated
اگر True باشد، یک جدول در پایگاه داده ایجاد می‌شود و پیغامهای ODBC درون آن ذخیره می‌گردد.	LogMessages

RecordsAffected

تعداد رکوردهای تحت تأثیر اجرای یک سؤال اجرایی همانند UPDATE , APPEND یا DELETE را مشخص می کند. برای مثال اگر در اثر اجرای یک سؤال اجرایی، پنج رکورد یک جدول حذف شود، مقدار این خصوصیت ۵ خواهد بود.

Type

به وسیله یک مقدار از نوع Integer نوع سؤال را مشخص می کند (جدول ۱۵-۶).

Updatable

اگر True باشد نشان می دهد که QueryDef دارای قابلیت به روز رسانی است.

جدول ۱۵-۶: مقادیر خصوصیت Type در شیء QueryDef.

نوع سؤال	مقادیر خصوصیت	ثابت
Select (SELECT)	0	dbQSelect
.. PIVOT).Crosstab (TRANSFORM	16	dbQCrosstab
Delete (DELETE)	32	dbQDelete
Update (UPDATE)	48	dbQUpdate
Append (INSERT)	64	dbQAppend
Make - table (INSERT INTO)	80	dbQMakeTable
..).Data - definition (Create	96	dbQDDL
SQL Passthrough	112	dbSQLPassThrough
Union	128	dbSetOperation
سؤالهای اجرایی (DELETE, UPDATE, INSERT, INSERT INTO, CREATE, INSERT INTO)	240	dbQAction

توجه داشته باشید که مقادیر جدول ۱۵-۶ همگی بر مبنای ۱۰ می باشند و قابل ترکیب نیستند. جدول ۱۶-۶ متدهایی را که در شیء QueryDef کاربرد بیشتری دارند نشان می دهد.

جدول ۱۶-۶: متدهای کاربردی در شیء QueryDef.

نام متد	هدف
OpenRecordset	این متد باعث باز کردن یک لیست داده روی نتیجه دستور SELECT در سؤال می شود.
Execute	باعث اجرای سؤالهای اجرایی می گردد.
CreateProperty	اجازه اضافه کردن خصوصیت LogMessages در شیء QueryDef را به شما می دهد.

هر سؤال که ایجاد شد، برای استفاده از آن از طریق متدها عمل می‌نماییم. همان طور که دیدید دو نوع سؤال داشتیم که یکی از آنها از نوعی بود که مجموعه‌ای از رکوردها را از طریق SELECT در اختیار قرار می‌داد و دیگری از نوع اجرایی بود که شامل عملیات INSERT, APPEND و غیره می‌شد. برای اجرای سؤالهایی که لیست رکوردها را در اختیار ما قرار می‌دهند از متد OpenRecordset استفاده می‌کنیم و دستور کلی آن به صورت زیر است.

```
Dim rsName AS Recordset
Set rsName = qdfName.OpenRecordset([intType [, Options]])
```

مقدار intType می‌تواند به دو صورت (2) dbOpenDynaset و یا (4) dbOpenSnapshot باشد و به صورت Table نمی‌توان روی نتیجه QueryDef، لیست داده باز کرد. جدول ۱۷-۶ لیست مقادیر intOption را مشخص می‌کند که قابلیت استفاده از آنها در متدهای OpenRecordset(RS) و Execute(EX) مشخص شده است.

مثال زیر یک لیست داده از نوع Snapshot و با قابلیت فقط پیشروی به جلو باز می‌کند.

```
Dim rssPubs AS Recordset
Set rssPubs = qdfPubs.OpenRecordset(dbOpenSnapshot , dbforwordOnly)
```

دستور کلی استفاده از متد Execute به صورت زیر است :

```
qdfName.Execute ([intoptions])
```

جدول ۱۷-۶: مقادیر آرگومان intOption در متدهای Execute و OpenRecordset.

هدف	ارزش	EX	RS	ثابتهای
اجازه نوشتن را به کاربران دیگر در هنگام اجرای سؤال یا باز کردن لیست داده، نمی‌دهد.	&H1	X	X	DbDenyWnt
اجازه خواندن را به کاربران دیگر در هنگام اجرای سؤال یا باز کردن لیست داده، نمی‌دهد.	&H2		X	DbDenyRead
یک لیست داده از نوع فقط خواندنی می‌سازد.	&H4		X	DbReadOnly
اجازه اجرا را فقط به سؤالهای از نوع INSERT می‌دهد.	&H8		X	DbAppendOnly
لیست داده را با روشی باز می‌کند که حرکت روی رکوردها فقط به طرف جلو باشد. با این روش سرعت حرکت روی رکوردها به جلو از طریق MoveNext بسیار سریعتر انجام می‌شود و برای استفاده از این تنظیم حتماً باید لیست داده به	&H100		X	dbForwardOnly

صورت Snapshot باز شود.

این تنظیم باعث می شود در زمان به روز رسانی

داده باشد، خطایی رخ ندهد. سؤال، در صورتی که کاربر دیگری داده‌ها را تغییر داده باشد، خطایی رخ ندهد.

جدول ۱۸-۶ لیست متدهای اجرایی را در مجموعه‌های Fields, Indexes, TableDefs و QueryDefs نشان می‌دهد.

تذکر: فقط شیء TableDef دارای مجموعه Indexes می‌باشد.

جدول ۱۸-۶: متدهای قابل اجرا در Fields(FLD), QueryDefs(QDF), TableDefs(TDF) و Indexes(IDX).

نام متد	TDF	QDF	FLD	IDX	هدف
Refresh	X	X	X	X	ساختار مجموعه مشخص شده از مقادیر را در هر یک از گروهها به روز رسانی می‌کند.
Append	X	X	X	X	یک عنصر جدید را مانند Table, Query, Field و یا یک Index جدید را به پایگاه داده می‌افزاید.
Delete	X	X		X	یک شیء مجموعه را از داخل مجموعه حذف می‌کند.

ایجاد اشیای Recordset و استفاده از آنها

قبل از اینکه طریقه ساخت یک لیست داده و مشخصات آن را شرح دهیم، مقداری در مورد خود لیست داده و انواع دسترسی به آن توضیح می‌دهیم.

قبلاً همان طور که اشاره کردیم Recordset یک لیست داده همانند یک جدول می‌باشد که داده‌های آن ممکن است شامل قسمتی از رکوردهای یک یا چند جدول و یا همه رکوردهای یک جدول تشکیل شده باشد. از نظر ساختاری لیست داده یک جدول مجازی است که از کپی قسمتی از یک جدول و یا همه آن از طریق اجرای یک کد SQL توسط سؤال تشکیل شده در RAM ذخیره می‌شود و ظاهری فیزیکی ندارد و نمی‌توان یک شیء لیست داده را مستقیماً در دسک ذخیره کرد. اگر این کپی مجازی جدول از مقدار حافظه آزاد RAM بیشتر شد مقدار اضافی به صورت موقتی روی هارد دیسک ذخیره می‌گردد که معمولاً در شاخه \WINDOWS\TEMP قرار می‌گیرد. این لیست داده فقط در مدت فعال بودن متغیر مخصوص لیست داده که اشاره بر جدول مجازی دارد باقی می‌ماند و سپس از RAM پاک می‌شود و اگر فایلی به صورت موقتی نیز ایجاد شده بود از بین می‌رود. شما

می‌توانید بین ساختار Dynamic (غیر ثابت) و یا Static (ساکن) لیست داده انتخاب داشته باشید که هر یک در زیر به اختصار توضیح داده می‌شود.

لیست داده از نوع Dynamic

دو نوع لیست داده از نوع Dynamic وجود دارد که هر دو بر پایه جدول مجازی می‌باشند :

- اولین نوع لیست داده که به صورت پیش فرض به این صورت ایجاد می‌شود، نوع Table می‌باشد که از اطلاعات یک جدول منفرد تشکیل شده است. این نوع لیست داده برای جستجو از متد Seek بر روی فیلدهای Index شده استفاده می‌کند که سرعت آن از هر نوع جستجویی بیشتر می‌باشد.

نکته

در این شیوه دسترسی به لیست داده نمی‌توان از دستورات Find برای جستجو استفاده کرد.

- دومین نوع دسترسی، دسترسی به شیوه Dynaset است که از اطلاعات یک جدول منفرد و یا نتیجه اجرای یک SQL از نوع SELECT که بر یک یا چند جدول ساخته شده است، شیوه دسترسی به روش Dynaset در ویژوال بیسیک معمولترین راه دسترسی به لیست داده است. از طریق این شیوه می‌توان کدهای SQL از نوع SELECT را مستقیماً اجرا نموده و لیست داده دلخواه را انتخاب کرد. مزیت استفاده از روش Dynaset این است که اطلاعات زیاد را به زیر مجموعه‌هایی تقسیم می‌کند و دسته دسته از آنها استفاده می‌نماید. برای مثال وقتی یک لیست داده حجیم را به شیوه Dynaset باز می‌کنید. در ابتدا فقط ۱۰۰ رکورد اول لیست داده را در اختیار قرار می‌دهد و وقتی از دستور Move... استفاده می‌نمایید به مقدار لازم به رکوردهای مقیم، می‌افزاید. هنگامی که MoveLast را روی لیست داده اجرا می‌کنید همه رکوردها را شامل می‌شود و این شیوه در حجم عظیم رکوردها بسیار مفید می‌باشد. لازم به ذکر است که در این روش، جستجو توسط Find انجام می‌شود که البته نسبت به Seek از سرعت کمتری برخوردار می‌باشد.

لیست داده از نوع Static

روش دسترسی به لیست داده از نوع Static، Snapshot نام دارد که به صورت یک تصویر ثابت از روی لیست داده، اطلاعات را فراخوانی می‌کند. شیوه دسترسی Snapshot به صورت فقط خواندنی (Read - Only) می‌باشد و قابلیت اضافه کردن و یا تغییر اطلاعات به کاربران داده نمی‌شود. خاصیت این شیوه، دسترسی سریع به اطلاعات کم حجم در حدود ۱۰۰ رکورد است.

پس کلاً سه روش دسترسی داریم که خواص برجسته آنها به صورت زیر است :

۱- Table : جستجو با Seek روی Index.

۲- Dynaset جستجو با Find.

۳- Snapshot فقط خواندنی

دستور عمومی ساختن یک لیست داده به صورت زیر است :

```
Dim rstName As Recordset
```

```
Set rstName = dbName.OpenRecordset(StrTableName,(intType [, intOption])
```

[intType] نوع لیست داده را مشخص می کند که شامل موارد زیر می باشد :

dbOpenTable : دسترسی به شیوه Table.

dbOpenDynaset : دسترسی به شیوه Dynaset.

dbOpenSnapshot : دسترسی به شیوه Snapshot.

در صورتی که int Type را مشخص نکنید Jet لیست داده را به شیوه Table باز خواهد کرد.

برای باز کردن یک لیست داده بر پایه SQL به صورت زیر عمل می نماید.

```
Dim rstName As Recordset
```

```
Set rstName = dbName.OpenRecordset(strSQL, [intType [, intOption]
```

در صورتی که intType را مشخص نکنید jet این لیست داده را به روش Dynaset باز می کند چون از

کد SQL استفاده می نماید.

نکته : دسترسی به روش Table روی جداول ضمیمه شده صورت نمی گیرد و برای باز کردن این جداول

باید از طریق کد ابتدا پایگاه داده مرجع را باز کرد و جدول را از داخل آن به این روش باز نمود و

استفاده کرد. در ادامه خصوصیات شیء لیست داده را بررسی می کنیم.

خصوصیات شیء Recordset

جدول ۱۹-۶ خصوصیات شیء لیست داده را نشان می دهد. قابلیت استفاده از هر یک از این

خصوصیتها را در هر یک از شیوه های دسترسی به لیست داده نظیر Table , Dynaset , Snapshot با

نشانه های TBL , DS , SS و علامت X در هر یک از موارد نشان داده شده است.

جدول ۱۹-۶ : لیست خصوصیت های شیء Recordset.

توضیح	SS	DS	TBL	خصوصیت
نام لیست داده را مشخص می کند. این				
خصوصیت به صورت فقط خواندنی (Read - only) می باشد.	X	X	X	Name
داده های فیلد را از رکورد جاری برمی گرداند.	X	X	X	Field!Value
نوع باز شدن لیست داده را مشخص می کند.				
انواع لیست داده شامل Table , Dynaset و Snapshot می باشد.	X	X	X	Type
تاریخ و زمان ساخت جدول را مشخص می کند. این				
خصوصیت به صورت فقط خواندنی (Read - only) می باشد.			X	DateCreated

توضیح	SS	DS	TBL	خصوصیت
تاریخ و زمان آخرین به روز رسانی جدول را مشخص می‌کند. این خصوصیت به صورت فقط خواندنی (Read - only) می‌باشد.			X	LastUpdated
اگر True باشد، دسترسی به داده‌ها به صورت خواندن/نوشتن (Read-Write) می‌باشد و اگر False باشد، دسترسی به داده‌ها به صورت فقط خواندنی (Read - only) می‌باشد.	X	X		Updatable
اگر True باشد، لیست داده از قابلیت Requery پشتیبانی کند و در غیر این صورت False است.	X		X	Restartable
اگر True باشد، قابلیت نشانه‌گذاری را در جدول دارا است و در غیر این صورت False می‌باشد.	X	X	X	Bookmarkable
یک مقدار Binary است که به یک رکورد مخصوص اشاره دارد و در مواقع نیاز می‌توان آن را روی یک رکورد قرار داد و بعد آن را بازیافت کرد.	X	X	X	Bookmark
به مقدار Bakmark مخصوصی اشاره دارد که آخرین بار تنظیم شده است.		X	X	Last Modified
(Read-Only) مربوط به فیلدهایی می‌باشد که Index آنها به Yes تنظیم شده باشد و از این طریق Index را فعال نموده تا عملیات Seek را انجام دهیم.			X	Index
برای مرتب کردن لیست داده بر اساس یک یا چند فیلد به کار می‌رود.		X	X	Sort
تعداد رکوردهای لیست داده را مشخص می‌کند. اگر مقدار این مشخصه صفر باشد نشان می‌دهد که هیچ رکوردی در لیست داده نمی‌باشد و در صورت ۱ بودن نشان می‌دهد که رکوردهایی در جدول موجود است.	X	X	X	RecordCount
برای به دست آوردن تعداد رکوردها ابتدا از متد MoveLast استفاده می‌کنیم تا رکورد جاری، روی آخرین رکورد قرار گرفته و سپس تعداد آن را توسط				

توضیح	SS	DS	TBL	خصوصیت
این خصوصیت تعیین می‌کنیم. یک متغیر Long می‌باشد که نشان می‌دهد شماره رکورد جاری چه مقدار است (این شمارش از صفر شروع می‌شود و اولین رکورد دارای شماره صفر است) اگر True باشد نشان‌دهنده این است که نشانگر جدول روی اولین رکورد قرار دارد و در غیر این صورت، این خصوصیت دارای مقدار False می‌باشد.	X	X		AbsolutePosition
اگر True باشد، نشان‌دهنده این است که نشانگر جدول روی آخرین رکورد قرار دارد و در غیر این صورت، این خصوصیت دارای مقدار False می‌باشد.	X	X	X	BOF (Beginning Of File)
این خصوصیت، حاصل جستجویی را که به وسیله دستور Seek و یا Find صورت گرفته است، نشان می‌دهد. اگر مقدار این خصوصیت True باشد، جستجو ناموفق بوده و رکورد مورد نظر پیدا نشده و اگر False باشد رکورد مورد نظر پیدا شده است.	X	X	X	EOF (End Of File)
یک شرط می‌باشد که مشخص می‌کند رکوردهای مخصوصی در لیست داده قرار گیرند. این خصوصیت همانند دستور WHERE در SQL عمل می‌کند.	X	X	X	Nomach
یک شرط برای محدود کردن رکوردها از روی فیلدها می‌باشد. در این عبارت شرط می‌توان از دستورات ترکیبی مانند And، Or و غیره نیز استفاده نمود.	X	X	X	Filter
پیغامی است که در صورت صادق نبودن شرط ValidationRule ظاهر می‌شود.	X	X	X	Validation Rull
حالت‌های ورود اطلاعات را به لیست داده مشخص می‌کند و شامل موارد زیر است: dbEditNone: فقط اطلاعات خوانده می‌شوند و نمی‌توان آنها را تغییر داد. DbEditInProgress: هم می‌توان اطلاعات را خواند و	X	X	X	Validation Text
			X	Editmode

هم می توان آنها را تصحیح کرد و یا تغییر داد.

dbEditAdd : اطلاعات فقط اضافه می شوند و اطلاعات

قبلی نمایش داده نمی شوند.

نحوه خواندن خصوصیت های مربوط به شیء لیست داده به طریق زیر می باشد :

```
varPropertyValue = rsname.PropertyName
```

varPropertyValue متغیر دلخواه ، rsname نام شیء لیست داده و PropertyName خصوصیت مربوطه می باشد.

به روش زیر نیز می توان خصوصیت هایی که قابلیت تنظیم را دارند (Read-Only نیستند) را تنظیم کرد.

```
rsName.PropertyName = varPropertyValue
```

متدهای اجرایی در شیء Recordset و مجموعه آن

جدول ۶-۲۰ لیست متدهای اجرایی شیء لیست داده را نشان می دهد.

جدول ۶-۲۰ : لیست متدهای اجرایی در شیء Recordset.

نام متد	TBL	DS	SS	توضیح
Clone	X	X	X	یک لیست داده جدید با نشانگر مستقل می سازد.
OpenRecordset	X	X	X	یک لیست داده را بر اساس یک جدول، سؤال و یا یک کد SQL باز می کند.
CopyQueryDef	X	X	X	این متد با استفاده از ایجاد یک لیست داده، یک کپی از QueryDef ایجاد می کند.
Requery	X	X	X	این خاصیت اطلاعات لیست داده را دوباره سازی می کند. مثلاً فرض کنید یک لیست داده ساخته اید که Filter آن وابسته به متغیر باشد. در صورتی که آن متغیر، تغییر کند لیست داده تغییری را نشان نمی دهد و برای نشان دادن داده های تغییر داده شده از این متد استفاده می کنیم.
Close	X	X	X	لیست داده را می بندد.
Edit	X	X	X	فیلدهای رکورد جاری لیست داده را آماده تغییر می کند. اگر بخواهیم مقادیر فیلدها را عوض کنیم، پس از قرار گرفتن روی هر رکورد باید ابتدا از این متد استفاده نماییم تا اجازه تغییر فیلد به برنامه داده شود.

توضیح	SS	DS	TBL	نام متد
یک رکورد جدید به آخر لیست داده از نوع Table یا Dynaset اضافه می‌کند که فیلدهای این رکورد برای تغییر آماده است.		X	X	AddNew
رکورد جاری در لیست داده از نوع Table یا Dynaset را حذف می‌کند.		X	X	Delete
پس از اینکه عملیات ویرایش، حذف و یا اضافه کردن رکورد پایان یافت برای ثبت کردن تغییرات از این متد باید استفاده نمود.		X	X	Update
پس از اعمال تغییرات، توسط این متد می‌توان داده‌ها را به حالت قبلی تبدیل کرد. در واقع این متد باعث لغو عملیات به روز رسانی می‌شود.		X	X	CancelUpdate
با تنظیم این خصوصیت به یک عدد، نشانه‌گر جدول روی رکوردی قرار می‌گیرد که شماره ردیف آن با این شماره برابر است.	X	X	X	Move
نشانه‌گر لیست داده را روی اولین رکورد قرار می‌دهد	X	X	X	MoveFirst
نشانه‌گر لیست داده را روی رکورد بعدی قرار می‌دهد.	X	X	X	MoveNext
نشانه‌گر لیست داده را در رکورد قبلی قرار می‌دهد.	X	X	X	MovePrevious
نشانه‌گر لیست داده را در آخرین رکورد قرار می‌دهد.	X	X	X	MoveLast
روی Index مشخص شده، عملیات جستجو را انجام می‌دهد.			X	Seek
هر فیلد را یکی یکی با شرط مربوطه مقایسه می‌کند و بدین ترتیب جستجو را انجام می‌دهد و روی اولین رکورد که با شرط مربوطه صدق مطابقت داشته باشد قرار می‌گیرد.	X	X		FindFirst
عملیات مقایسه را ادامه می‌دهد و روی رکورد بعدی که با شرط مربوطه صدق مطابقت داشته باشد قرار می‌گیرد.	X	X		FindNext

همانند FindNext عمل می کند، با این تفاوت که عملیات مقایسه در جستجو را برای رکوردهای بالاتر از رکورد جاری انجام می دهد.	X	X	FindPrevious
همانند FindFirst عمل می نماید، با این تفاوت که روی آخرین رکوردی قرار می گیرد که در شرط صدق می کند.	X	X	FindLast

دستور کلی استفاده از متدهای بالا به صورت زیر می باشد:

`rsName.MethodName [Argument1 [, Argument2 [, Argument#]]]`

بعضی از متدها مانند `Update`، دارای آرگومان نمی باشند و بعضی دیگر دارای چند آرگومان هستند که در فصلهای بعدی آنها را بررسی می کنیم.

فصل هفتم

استفاده از ابزار کنترل داده برای دسترسی به اشیای اطلاعات

در فصل قبل با نحوه عملکرد اشیای پایگاه داده و خصوصیت‌های آنها و نحوه استفاده آنها در کدهای برنامه‌نویسی ویژوال بیسیک آشنا شدیم. در این فصل با یک ابزار جدید آشنا می‌شویم و نحوه استفاده تخصصی از آن را فرا می‌گیریم. Data Control ابزاری است که عملیات دسترسی به پایگاه داده را بسیار آسان کرده است. همان طور که در فصل چهارم دیدید با استفاده از این ابزار بدون نیاز به کد به سادگی توانستیم در یک فرم ساده فایل داده را باز کرده و اطلاعات آن را نمایش دهیم.

برای استفاده بهینه از این ابزار نیاز داریم تا با مشخصات دقیق آن آشنا شویم و نیز نحوه کنترل آن را از طریق کد فرا بگیریم و سپس فرمهای قدرتمند خود را روی پایگاههای داده فعال سازیم.

در واقع Data Control ابزاری است که بسیاری از مراحل کدنویسی را خلاصه می‌کند و به سادگی یک لیست داده ایجاد می‌نماید و پس از این مرحله می‌توان این لیست داده را از طریق ابزارهای دیگر نمایش داد و عملیات اطلاعاتی را از طریق آن ابزارها انجام داد.

در ابتدای این فصل به بررسی خصوصیات Data Control می‌پردازیم. جدول ۱-۷ لیست خصوصیت‌های این ابزار را نشان می‌دهد. این ابزار نیز همانند بسیاری از ابزارهای ویژوال بیسیک دارای خصوصیت‌هایی است که مربوط به ظاهر کار می‌باشد. این خصوصیتها در همه ابزارها مشترکند و دارای یک وظیفه خاص می‌باشند. در این جدول از آوردن این خصوصیتها خودداری شده است.

جدول ۱-۷: لیست خصوصیات ابزار Data Control.

توضیح	خصوصیت
این خصوصیت، مرجع شیء پایگاه داده برای Data Control می‌باشد.	Database
در این خصوصیت مسیر و نام فایل پایگاه داده قرار می‌گیرد. مثلاً اگر می‌خواهید پایگاه داده Biblio.Mdb را که از نوع اکسس است، باز کنید در این خصوصیت مقدار زیر را وارد کنید:	DatabaseName
c:\VB\Biblio.Mdb	Connect
در این قسمت نوع فایلی را که قرار است با آن ارتباط برقرار کنید مشخص می‌گردد. فایل مورد نظرمی‌تواند از نوع فایل‌های پایگاههای داده نظیر اکسس، پاراداکس، Xbase، فاکس پرو و غیره و یا فایل‌های صفحه گسترده نظیر اکسل	

توضیح	خصوصیت
<p>و یا Lotos 123 و غیره و یا فایل متنی باشد. در صورتی که بخواهید از منبع ODBC استفاده کنید، در این قسمت موارد زیر را وارد نمایید:</p> <p>ODBC; DATABASG = DefaultDatabase; DSN = DataSourceName ; UID=UserID; PWS=Password;</p>	Record Source
<p>در این خصوصیت مرجع لیست داده مشخص می‌شود. پس از اینکه نوع پایگاه داده را در Connect انتخاب کردید و در خصوصیت DatabaseName، نام و آدرس پایگاه داده را وارد کردید در این خصوصیت جداول و سؤالها (Query) باز می‌شود و می‌توانید هریک را به عنوان مرجع لیست داده باز کنید و نیز می‌توانید مستقیماً از این قسمت کد SQL از نوع SELECT را وارد کنید و از آن به عنوان اجرا کننده SQL استفاده کنید.</p>	Recordset
<p>خصوصیتی است که از طریق کد، می‌توانید لیست داده این ابزار را در دست‌گیرید. به این معنی که این ابزار دقیقاً همانند یک لیست داده که توسط کد باز می‌شود عمل می‌کند و می‌توان از آن استفاده نمود.</p>	RecordsetType
<p>شیوه دسترسی به جدول یا سؤال را مشخص می‌کند که می‌تواند به سه صورت Table، Dynaset و یا Snapshot باشد.</p>	EOFAction
<p>این خصوصیت تعیین می‌کند که اگر اشاره‌گر رکورد روی آخرین رکورد قرار گرفته باشد و دستور انتقال به رکورد بعدی صادر گردد چه حالتی رخ می‌دهد.</p>	
<p>۱ - (VbEOFActionMoveLast,0) که حالت پیش‌فرض است، اشاره‌گر رکورد را روی رکورد آخر باقی می‌گذارد و دستور انتقال به رکورد بعد را نادیده می‌گیرد.</p>	
<p>۲ - (VbEOFActionEOF,1) که اشاره‌گر رکورد را به روی مکان Eof انتقال می‌دهد و در صورتی که جدول با قابلیت افزایش رکورد باشد یک رکورد خالی در انتهای جدول باز می‌کند تا اطلاعات جدید درون آن وارد گردد.</p>	

توضیح

خصوصیت

این خصوصیت تعیین می‌کند که اگر اشاره‌گر رکورد روی اولین رکورد قرار گرفته باشد و دستور انتقال به رکورد قبل صادر گردد چه حالتی رخ دهد.
 ۱- (VbBOFActionMoveFirst,0) که حالت پیش‌فرض است، اشاره‌گر رکورد را روی اولین رکورد باقی می‌گذارد و دستور انتقال به رکورد قبل را نادیده می‌گیرد.

۲- (VbBOFActionBOF, 1) که اشاره‌گر رکورد را به مکان BOF انتقال می‌دهد.

نکته: در مورد مفاهیم BOF و EOF در فصل دوم توضیح داده شد.

یک Flag از نوع Long می‌باشد که جدول ۲-۷ حالت‌های مختلف آن را نشان می‌دهد. حالت پیش‌فرض این خصوصیت صفر می‌باشد.

اگر True باشد، پایگاه داده به صورت تک کاربره (Single User) باز خواهد شد و در صورتی که False باشد، دسترسی به پایگاه داده به صورت چندکاربره (Multi User) خواهد بود.

این خصوصیت تعیین می‌کند که اطلاعات این لیست داده فقط خواندنی باشد (True) و یا قابلیت خواندن و نوشتن را داشته باشد (False).

Options

Exclusive

ReadOnly

نکته: در صورتی که مقادیر خصوصیت‌های Database , RecordSource , Options , Exclusive یا Readonly توسط کدهای ویژوال بیسیک در هنگام اجرای برنامه تغییر کنند، برای اعمال این تغییرات باید از متد اجرایی Refresh استفاده کرد.

جدول ۲-۷: لیست مقادیر خصوصیت Oprions در ابزار کنترل داده.

هدف

ارزش

نام

اجازه تغییر اطلاعات جدولی را که توسط برنامه شما باز شده است را به برنامه‌های دیگر نمی‌دهد.	1	vbDataDenyWrite
اجازه خواندن اطلاعات تشکیل‌دهنده را که توسط برنامه شما باز شده است به برنامه‌های دیگر نمی‌دهد.	2	vbDataDenyRead
یک لیست داده ایجاد می‌کند که توسط هیچ یک از کاربران قابلیت تغییر و افزایش ندارد و فقط خواندنی (Read-only) می‌باشد.	4	vbDataReadOnly
به لیست داده برنامه شما فقط قابلیت افزایش رکورد را می‌دهد.	8	vbDataAppendonly

متدهای اجرایی در Data Control

از ویژگی‌های بیسیک نسخه چهار به بعد همه متدهای اجرایی لیست داده از نوع Dynaset در شیء لیست داده از Data Control قابل اجرا می‌باشند و فقط دو متد اجرایی در شیء لیست داده از Data Control وجود دارد که در جدول متدهای اجرایی شیء لیست داده (جدول ۲۰-۶) آورده نشده است زیرا این متدها مخصوص Data Control می‌باشند؛ بنابراین در جدول ۳-۷ آنها را به همراه متد Refresh به دلیل استفاده خاص Data Control از این متد، توضیح می‌دهیم.

جدول ۳-۷: لیست تکمیلی متدهای Data Control.

نام متد	توضیح
UpdateRecord	این متد باعث می‌شود که تمام تغییرات داده‌ها که Data Control روی آنها لیست داده تشکیل داده است، ثبت گردد. از این متد می‌توان هنگامی استفاده کرد که کاربر پس از انجام تغییرات روی دکمه فرمان OK ، Save و یا Update کلیک کند.
UpdateControls	عملی عکس UpdateRecord انجام می‌دهد و باعث می‌شود که تغییرات انجام شده روی رکوردها ثبت نگردد. این متد را می‌توان در فرم به عنوان Undo یا Cancel اجرا کرد.
Refresh	این متد باعث می‌شود پایگاه داده جدید باز گردد (و یا در صورت باز بودن آن، بار دیگر مجدداً باز نماید) و از روی آن یک لیست داده جدید برای Data Control ایجاد نماید. از این متد معمولاً هنگامی استفاده می‌شود که خواسته باشیم تغییراتی را که روی خصوصیت‌های Data Control به وسیله کد ایجاد کرده‌ایم اعمال گردد. مثلاً از طریق کد، یک SQL جدید برای خصوصیت RecordSource تعریف می‌کنیم و برای اعمال این تغییرات باید از Refresh استفاده کرد والا Data Control هیچ تغییری در لیست داده نخواهد داد.

مکان استفاده از عباراتی که شامل متدهای UpdateRecord و Updatecontrols هستند فقط در رویداد Validate می‌باشد. مثال استفاده از این متدها را در همین فصل بررسی خواهیم کرد.

رویداد Validate در Data Control

در سه حالت این رویداد (Event) اتفاق می افتد :

- وقتی که از دکمه‌های ابزار Data Control استفاده می‌شود و یا فرامین حرکت روی رکوردها Move... صورت گیرد. این رویداد قبل از اینکه اشاره‌گر به رکورد دیگری انتقال یابد اتفاق می‌افتد.

- وقتی که از متد Update روی Data Control استفاده می‌شود. (این رویداد در هنگام استفاده از متد UpdateRecord اجرا نمی‌گردد).

- در هنگام استفاده از متدهای Delete یا Close یا هنگام خروج از فرم این رویداد اتفاق می‌افتد. این رویداد پارامترهایی دارد که در صورت اتفاق افتادن رویداد، دارای مقداری می‌گردند و می‌توان به وسیله این پارامترها کنترل‌های لازم را انجام داد. دستور کلی این رویداد به صورت زیر می‌باشد :

```
Sub DataControl_ValiDate ([iaintIndex As Integer,] intAction As Integer, fsave As Integer)
```

...

End Sub

اگر Data Control یکی از اعضای یک آرایه باشد، پارامتر iaintIndex مقدار Index آرایه را مشخص می‌کند.

جدول ۴-۷ لیست مقادیر intAction را مشخص می‌کند. پارامتر fSave نشان می‌دهد که آیا تغییری در رکورد صورت گرفته است یا خیر. True نشان دهنده تغییر در رکورد می‌باشد و False نشانگر این است که هیچ تغییری روی رکورد جاری صورت نگرفته است.

جدول ۴-۷ : لیست مقادیر intAction.

نام عبارت	ارزش	زمان اعمال
vbDataActionCancel	0	به روز رسانی Cancel شده است.
vbDataActionMoveFirst	1	MoveFirst
vbDataActionMoveProvisions	2	MovePrevious
vbDataActionMoveNext	3	MoveNext
vbDataActionMoveLost	4	MoveLast
vbDataActionAddnew	5	AddNew
vbDataActionUpdate	6	Update
vbDataActionDelete	7	Delete
vbDataActionFind	8	یکی از متدهای Find... اجرا شده است.
vbDataActionBookmark	9	خصوصیت Bookmark تنظیم شده است.
vbDataActionclose	10	Close
vbDataActionUnload	11	Unload

عملیات مقید سازی

مقید سازی به ایجاد هر گونه ارتباط بین Data Control و سایر کنترل‌های فرم گفته می‌شود. به این صورت که چند خصوصیت در کنترل‌های فرم نظیر Text Box ، Lable و غیره وجود دارد که توسط آنها می‌توان روی داده‌های Data Control ، عملیات مقید سازی را انجام داد. جدول ۷-۵ خصوصیات مرتبط با داده را در کنترل‌های مختلف نشان می‌دهد.

جدول ۷-۵ : خصوصیات کنترل‌های مرتبط با Data Control.

نام خصوصیت	کنترل‌ها	توضیح
DataSource	تمام کنترل‌های مقیدسازی	در این خصوصیت نام Data Control ای که روی داده‌های آن مقیدسازی انجام می‌شود، مشخص می‌گردد.
DataField	تمام کنترل‌های مقید سازی	در این خصوصیت نام فیلد مورد نظر از لیست داده Data Control برای مقید سازی، مشخص می‌شود.
DataChange	تمام کنترل‌های مقید سازی	اگر True باشد، کاربر یا برنامه می‌تواند داده‌های Data Control دهد و اگر False باشد این امکان برای کاربر و یا برنامه وجود ندارد.
Text	Text Box	مقادیر داده‌های مقید شده تحت این کنترل را بر می‌گرداند. اگر این مقدار (0) باشد، انتخاب نشده است و اگر (1) باشد
Value	Check Box	انتخاب شده است و اگر (2) باشد، این کنترل غیر فعال می‌باشد.
Caption	Label	در صورتی که از این کنترل برای مقید سازی داده‌ها استفاده شود، در این خصوصیت داده مقید شده قرار می‌گیرد.

لیست ابزارهایی که می‌توانند با Data Control در ارتباط باشند به صورت زیر است :

TextBox , Label , Image , Picture Box , CheckBox , List Box , Combo Box , OLE Container

Masked Box , 3d Panel , 3d Check Box , DB List , DB Combo , DB Grid

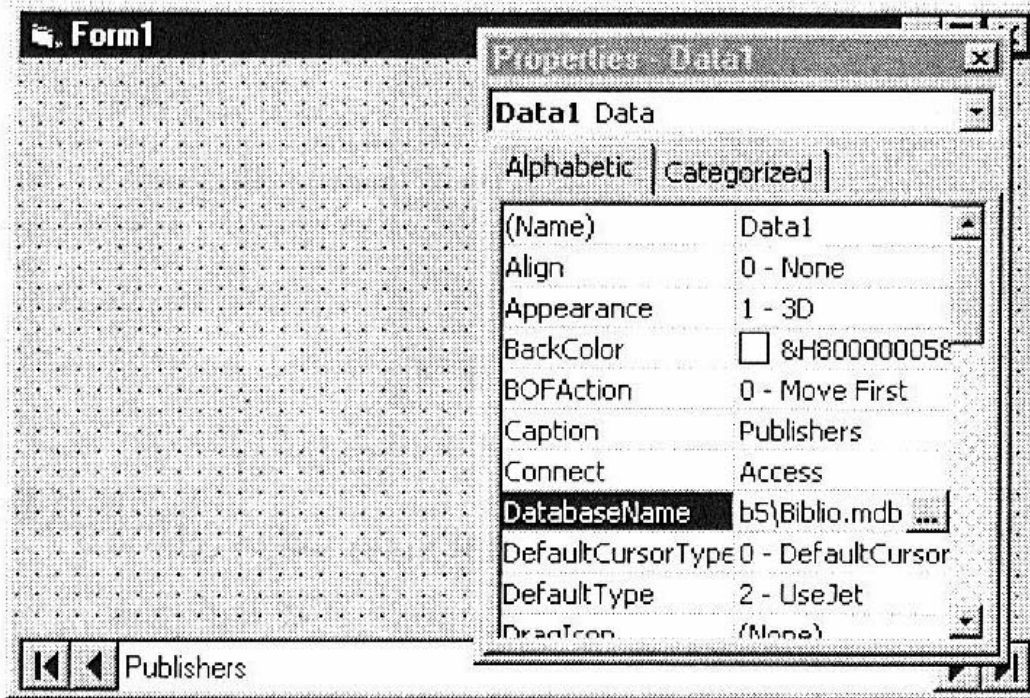
بجز چند مورد نحوه ارتباط همگی این کنترل‌ها با Data Control توسط خصوصیت‌های جدول

۷-۵ صورت می‌گیرد و روش دسترسی یکسان است که در فصل دهم نحوه ارتباط کنترل‌های مهمتر این مجموعه را بررسی می‌کنیم.

ایجاد یک فرم اطلاعاتی پیشرفته

الف) نمایش اطلاعات

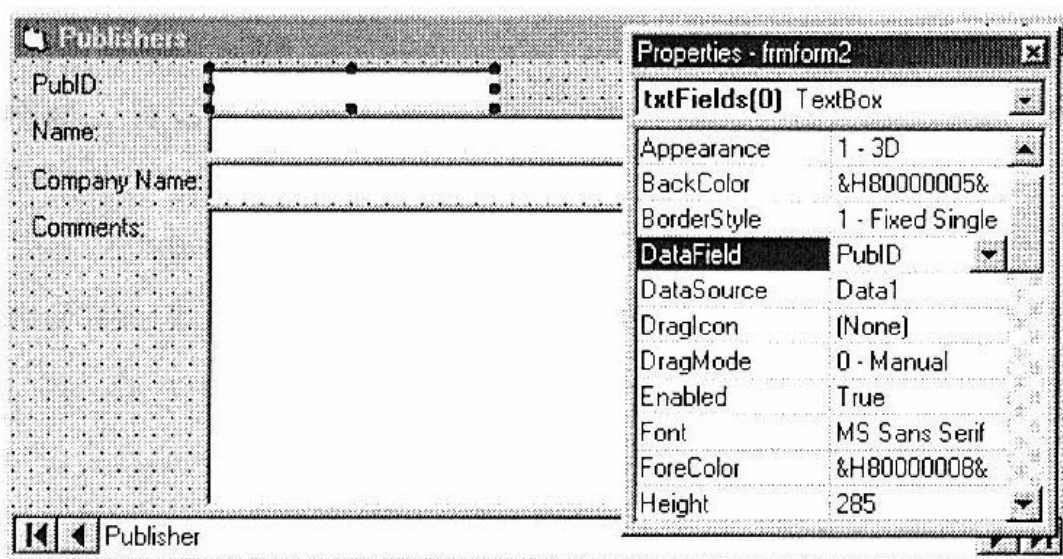
۱ - پس از ایجاد فرم ، ابزار Data Control را به فرم می‌افزاییم و خصوصیت‌های DatabaseName ، RecordSource و RecordsetType آن را تنظیم می‌نماییم.



شکل ۱-۷.

اضافه کردن ابزار Data Control به فرم نمایش داده‌ها.

۲ - برای نشان دادن اطلاعات لیست داده، از جعبه متن استفاده می‌کنیم. پس از قرار دادن این ابزار در جای مناسب روی فرم و انتخاب برچسب مناسب، دو خصوصیت Data Field و Data Source را برای هر یک تنظیم می‌کنیم که در خصوصیت DataSource جعبه متن نام ابزار Data Control و در Data Field نام فیلد مورد نظر از جدول لیست داده قرار می‌گیرد.

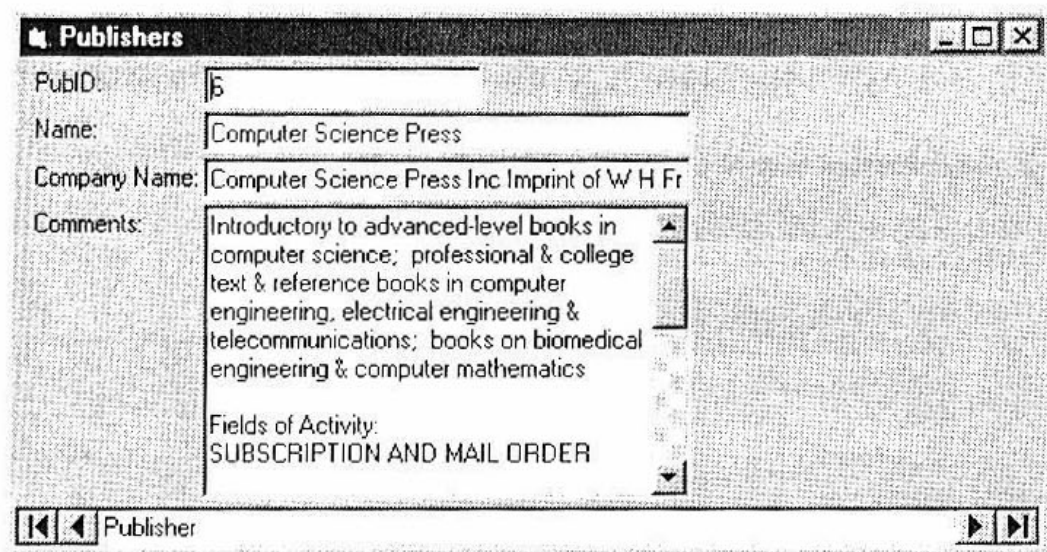


شکل ۲-۷.

اضافه کردن جعبه های متن مقید به ابزار

Data Control

نکته: برای فیلدهایی که اطلاعات آن از یک خط بیشتر است مانند Comments، از خصوصیت Multiline و Scroll Bars استفاده می‌کنیم. حال فرم نمایش اطلاعات آماده است.



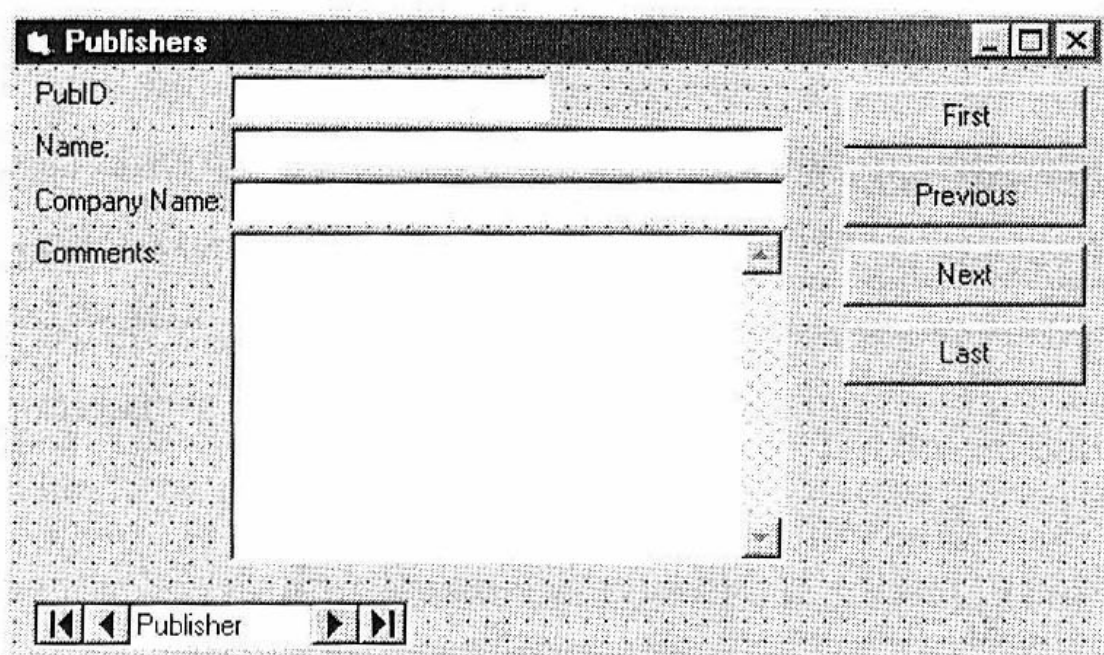
شکل ۳-۷.

تنظیم خصوصیت MultiLine برای جعبه های متن چند خطی.

ب) کنترل‌های حرکت روی رکوردها

همان گونه که در شکل‌های بالا مشاهده کردید، هر Data Control دارای چهار دکمه کنترلی است که عملیات حرکت به جلو و عقب و انتقال به اولین و آخرین رکورد را برعهده دارند. در صورتی که خواسته باشید خود این کنترل‌ها را به صورت دلخواه تنظیم کنید به ترتیب زیر عمل نمایید.

۱ - چهار دکمه فرمان به فرم اضافه کرده و خصوصیت Caption آنها را به ترتیب First، Previous، Next و Last قرار دهید و نام آنها را به صورت Cmd_First، Cmd_Pre، Cmd_Next، Cmd_Last تنظیم کنید.

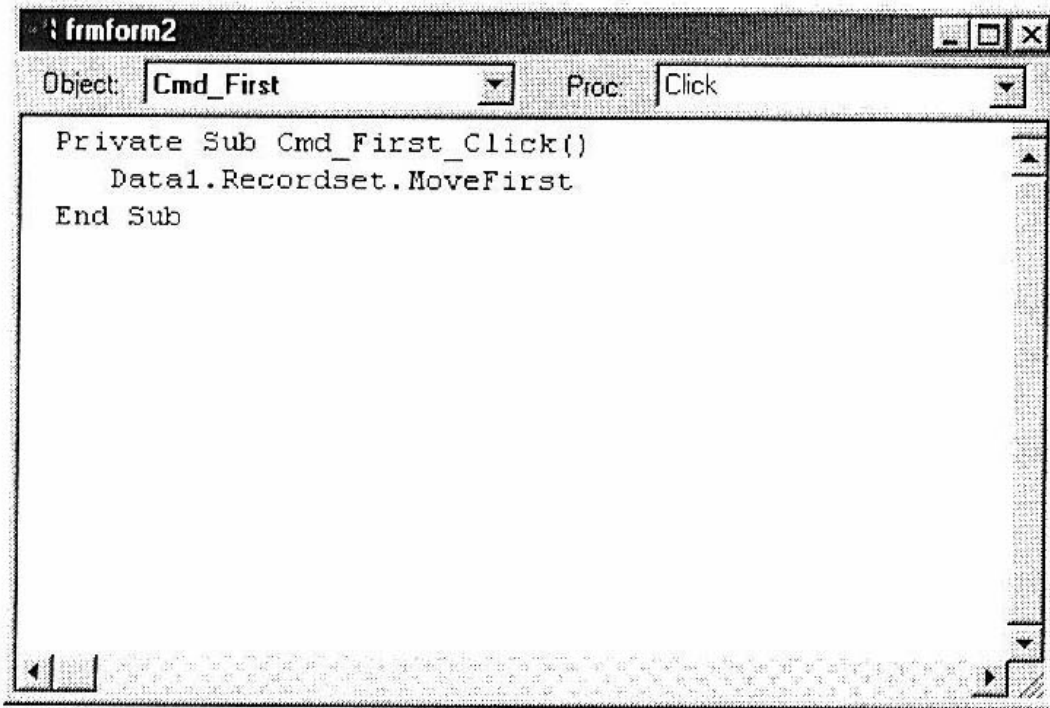


شکل ۴-۷.

کنترل حرکت روی رکوردها به وسیله دکمه های فرمان.

۲ - روی اولین دکمه فرمان دوبار کلیک کنید تا پنجره کد نمایش یابد. کد زیر را درون رویداد Click وارد کنید.

```
Private Sub Cmd_First_Click()  
    Data1.Recordset.MoveFirst  
End Sub
```



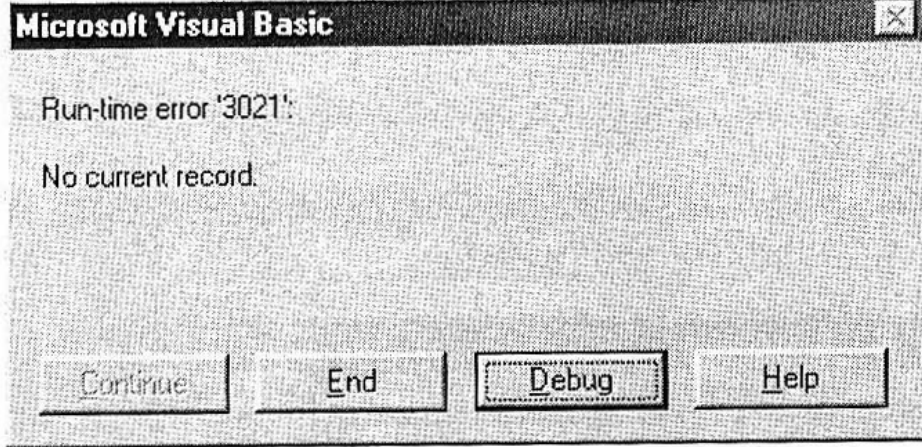
شکل ۵-۷.

کد مربوط به دکمه فرمان First.

به همین ترتیب در دکمه‌های فرمان، کدهای مربوطه را قرار دهید.

```
Private Sub Cmd_Pre_Click()  
    Data1.Recordset.MovePrevious  
End Sub  
Private Sub Cmd_Next_Click()  
    Data1.Recordset.MoveNext  
End Sub  
Private Sub Cmd_Last_Click()  
    Data1.Recordset.MoveLast  
END Sub
```

حال فرم را اجرا کنید و به ترتیب دکمه‌های فرمان را آزمایش کنید. روی دکمه Last و سپس دوبار دکمه Next را کلیک نمایید. پیغام خطایی به صورت شکل ۶-۷ ظاهر می‌گردد، زیرا خواسته‌اید که اشاره‌گر رکورد را به مکان بعد از EOF (End Of File) انتقال دهید. خطا در یک حالت دیگر نیز رخ می‌دهد و آن زمانی است که دکمه First را کلیک کرده و دوبار دکمه Previous را کلیک نمایید. در این حالت نیز خطا صورت می‌گیرد زیرا خواسته‌اید اشاره‌گر را به مکان قبل از BOF (Beginning Of File) انتقال دهید.



شکل ۶-۷.

بروز خطای زمان اجرا.

ج (رفع خطای No Current Record

چند راه برای جلوگیری از خطای No Current Record وجود دارد. این موارد به صورت زیر می‌باشند :

- شما می‌توانید از رویداد Validate استفاده کنید. به این ترتیب که در آنجا دو پارامتر EOF و BOF را امتحان کرده و در صورت وقوع آنها را کنترل نمایید. برای مثال می‌توانید از کد زیر در این رویداد استفاده کنید.

```
Private Sub Data1_Validate(Action As Integer, Save As Integer)
    Select Case Action
        Case VbDataActionMoveNext
            If Data1.Recordset.EOF Then
                Data1.Recordset.MoveLast
            End IF
        Case VbDataActionMovePrevious
            If Data1.Recordset.Bof Then
                Data1.Recordset.MoveFirst
            End IF
    End Select
End Sub
```

- شما می‌توانید از طریق Bookmark این کار را انجام دهید. به این صورت که اولین رکورد و آخرین رکورد را در دو متغیر StrFirst و StrLast قرار دهید و در هنگام حرکت روی رکوردها این پارامترها را چک کنید و از وقوع خطا جلوگیری نمایید.

- رکورد بعد از مکان EOF و قبل از مکان Bof دارای مقادیر Null هستند. بدین وسیله می‌توان فیلدها را امتحان کرد و فهمید که در مکان Bof یا EOF قرار داریم. این متد برای بانکهای اطلاعاتی اکسس و در بانکهای اطلاعاتی تحت معماری Client / Server صدق می‌کند و در مورد فایل‌های DBF و DB. کراکتری که باید امتحان شود در این بانکهای اطلاعاتی مقدار 0 می‌باشد.

- راه دیگر، استفاده از خصوصیت‌های Bof و EOF از لیست داده است. هنگامی که از دستورات MovePrevious و MoveNext استفاده می‌کنید، پس از این دستورات به وسیله خصوصیت EOF, of این خطا را کنترل کنید. برای مثال در این روش می‌توانید از کد زیر استفاده کنید:


```

Private Sub Cmd_Next_Click()
    Data1.Recordset.MoveNext
    If Data1.Recordset.Eof Then
        Beep
        Data1.Recordset.MoveLast
    End IF
End Sub

Private Sub Cmd_Per_Click()
    Data1.Recordset.MovePrevious
    If Data1.Recordset.Bof Then
        Beep
        Data1.Recordset.MoveFirst
    End IF
End Sub

```

د) جستجو توسط دستورات Find...:

از روش Find برای جستجو روی رکوردهای جدولی استفاده می‌شود که تعداد رکوردهای آن کم باشد. مثلاً در پایگاه داده Biblio. mdb جدول Published دارای ۶۰ رکورد می‌باشد که برای جستجوی رکوردی خاص، روش Find روش مناسبی می‌باشد. نکته قابل توجه در جستجو به این روش این است که متد Find روی لیست داده‌ای عمل می‌کند که به شیوه‌های Dynaset و یا Snapshot باز شده باشند و برای جداولی که به روش Table باز شده‌اند، نمی‌توان از Find برای جستجو استفاده کرد، بلکه باید از Seek استفاده گردد. دستور کلی استفاده از متدهای Find به صورت زیر می‌باشد:

```

rsRecordset.FindFirst StrCriteria
rsRecordset.FindNext StrCriteria
rsRecordset.FindPrevious StrCriteria
rsRecordset.FindLast StrCriteria

```

rsRecordset نام لیست داده‌ای است که به روش Dynaset و یا Snapshot باز شده باشد و StrCriteria عبارت شرطی است که می‌خواهید برای جستجو اعمال شود. برای مثال می‌خواهید اولین ناشر را در لیست داده پیدا کنید که در California واقع شده است. کد زیر این جستجو را انجام می‌دهد:

```

Dim StrCriterion As String
StrCriterion = "State='CA'"
Data1.Recordset.FindFirst StrCriterion

```

همان طور که دیدید در شرط باید ابتدا نام فیلد مورد نظر را آورده و سپس مقدار دلخواه را به آن نسبت داد و عملیات جستجو از ابتدای جدول شروع می‌شود و اشاره‌گر رکوردها روی اولین رکوردی قرار می‌گیرد که فیلد قید شده، دارای شرط مذکور باشد. در مثال بالا اشاره‌گر رکوردها روی اولین رکوردی قرار می‌گیرد که فیلد State آن دارای مقدار CA باشد.

باید توجه داشت که برای جستجو در فیلدهایی که از نوع متنی می‌باشد، باید حتماً از علامت (^) استفاده کرد و نمی‌توان از (") استفاده کرد، زیرا با علامت مقدار دهی به متغیر، اشتباه خواهد شد. برای گرفتن مقدار شرط از کاربر می‌توان از تابع Input Box استفاده کرد و شرط مورد نظر را تشکیل داد. راه دیگر استفاده از فرمی است که در آن فرم مقادیر از کاربر گرفته شود و شرط تشکیل گردد. در اینجا ما از تابع Input Box برای گرفتن مقدار شرط استفاده می‌کنیم.

۱- دو دکمه فرمان به فرم اضافه می‌کنید و Caption آنها را به ترتیب FindFirst و FindNext قرار دهید و نام آنها را به صورت Cmd_Find_First و Cmd_Find_Pre تنظیم نمایید.

۲- روی هر یک از دکمه‌های فرمان دوبار کلیک نمایید و کدهای لیست ۱-۷ را در آنها وارد کنید.

۳- حال برنامه را اجرا کنید و روی دکمه FindFirst کلیک کنید. سپس در جعبه متن ca را تایپ کنید و دکمه فرمان OK را برای جستجوی اولین رکوردی که فیلد State آن دارای مقدار California باشد، انتخاب کنید.

لیست ۱-۷: کدهای دکمه‌های فرمان جستجو

```
Private Sub cmdFindFirst_Click()
    Dim strPrompt As String
    Dim strTitle As String
    Dim strEntry As String

    strPrompt = "Enter the toe-Character postal abbreviation " & _
        "for the state you want to find."
    strTitle = "Find First Occurrence of State"
    strEntry = InputBox(prompt:=strPrompt, Title:=strTitle)
    If Len(Trim(strEntry)) = 2 Then
        strBookMark = Data1.Recordset.Bookmark
        strCriterion = "State = " & UCase(strEntry) & ""
        Data1.Recordset.FindFirst strCriterion
        If Data1.Recordset.NoMatch Then
            MsgBox prompt:="No records match " & UCase(strEntry) & ".",
                Title:="Record Not Found"
            Data1.Recordset.Bookmark = strBookMark
        Else
            Text7.SetFocus
        End If
    End If
End Sub

Private Sub cmdFindNext_Click()
    If Len(strCriterion) > 0 Then
        Data1.Recordset.FindNext strCriterion
        If Data1.Recordset.NoMatch Then
            MsgBox prompt:="No more records match " & strCriterion & ".", Title:="Record
            Not Found"
```

```

    Data1.Recordset.Bookmark = strBookMark
Else
    Text7.SetFocus
End If
Else
    MsgBox prompt:="You need to enter a criterion with the " & _
        "Find First button first ", _
        Title:="Record Not Found"
End If
End Sub

```

ه (حذف ، اضافه و تغییر رکوردها

دکمه‌های فرمانی که در این مرحله به این فرم اضافه شده‌اند شامل Delete , Mark , Update , Cancel , Add , می‌باشد که Update کار ثبت تغییرات یک رکورد را برعهده دارد. لیست ۲-۷ کد دکمه‌های فرمان اضافه شده به این فرم را نشان می‌دهد. لیست ۲-۷: کدهای دکمه‌های فرمان برای حذف ، اضافه و به هنگام سازی.

```

Private Sub cmdUpdate_Click()
    Data1.Recordset.Update
End Sub

```

```

Private Sub cmdCancel_Click()
    Data1.Refresh
    If IsNull(text1.Text) Or text1.Text = "" Then
        If Len(strBookmark) > 0 Then
            Data1.Recordset.Bookmark = strBookmark
        Else
            Data1.Recordset.MoveLast
        End If
    End If
End Sub

```

```

Private Sub cmdAddNew_Click()
    Data1.Recordset.AddNew
End Sub

```

```

Private Sub cmdDelete_Click()
    If MsgBox (propt:="Are you sure you want to delete this record " , & _
        button:=vbQuestion + vbYesNo , _
        Title:= " Delete Record") = vbYes Then
        If (Len(strBookmark) > 0) And
            (strBookmark = CStr(Data1.Recordset.Bookmark)) Then
            strBookmark = ""

```

```

End If
Data1.Recordset.Delete
Data1.Refresh
If Len(strBookmark) > 0 Then
    Data1.Recordset.Bookmark = strBookmark
Else
    Data1.Recordset.MoveLast
End If
End If
End Sub

```

```

Private Sub cmdMark_Click()
    StrBookmark = Data1.Recordset.Bookmark
End Sub

```

```

Private Sub cmdReturn_Click()
    If Len(strBookmark) > 0 Then
        Data1.Recordset.Bookmark = strBookmark
    Else
        Beep
    End If
End Sub

```

و (تنظیم کدهای رویداد Validate

هنگامی که یک رکورد جدید برای ورود اطلاعات ایجاد می‌شود، بعضی از فیلدها نیاز دارند که حتما مقدار دهی شوند. مثلاً در جدول Publisher هر رکورد برای ثبت شدن نیاز به Name و PubID دارد پس باید در هنگام ثبت رکورد جدید این موارد را چک کنیم. برای این کار از رویداد Validate استفاده می‌کنیم.

لیست ۳-۷ کد مربوط به این رویداد را نشان می‌دهد. در صورت کلیک کردن روی AddNew این رویداد فعال می‌شود و پیغام می‌دهد که کدام فیلدها نیاز به مقدار دهی دارند.

لیست ۳-۷: کد مربوط به رویداد Validate و AddNew.

```

Private Sub Data1_Validate(Action As Integer, Save As Integer)
    Dim strMsg As String
    If (intAction = vbDataActionAddNew) Or (intAction = vbDataActionMoveNext)
    Then
        If (text1.Text = "") Or (text2.Text = "") Then
            If text1.Text = "" Then
                strMsg = "A Publisher ID number(PubID) is required. "
            Else
                strMsg = "An imprint (Name) is required. "
            End If
        End If
    End If

```

```
MsgBox prompt:=strMsg, Title:="Required Field Data Missing"  
intAction = vbDataActionCancel  
fSave = False  
End If  
End If  
End Sub
```

```
Private Sub cmdAddNew1_Click()  
Dim lngPubID As Long  
Data1.Refresh  
Data1.Recordset.MoveLast  
lngPubID = Data1.Recordset!PubId  
If text2.Text <> "" Then  
    lngPubID = lngPubID + 1  
End If  
Data1.Recordset.AddNew  
text1.Text = lngPubID  
End Sub
```

نکته: Refresh هنگامی استفاده می شود که بخواهیم تغییراتی را که روی لیست اعمال شده است را فعال کنیم. مثلاً هنگامی که یک رکورد را توسط Delete حذف می کنید. این تغییرات انجام می گردد، اما نمایش داده نمی شود، یعنی در لیست داده تغییری دیده نمی شود. در صورتی که از Refresh استفاده کنیم در لیست داده جدید تغییرات صورت گرفته است.

زبان پرس و جوی ساختاری (SQL)

تاریخچه SQL

در اوایل ۱۹۷۰، IBM، زبانی را ایجاد کرد که (Structured English Query Language) SEQUEL نام گرفت. این زبان برای جستجو در سیستمهای پایگاه داده رابطه‌ای ساخته شد. بعد از آن این زبان به SEQUEL/2 ارتقا یافت و در پایان به SQL یا زبان پرس و جوی ساختار یافته تبدیل شد. بقیه شرکتها نیز به نظریه پایگاه داده رابطه‌های علاقه‌مند شدند و رابطه‌های SQL را ساختند. نرم افزار رابطه‌ای، محصول پایگاه داده رابطه ای را ایجاد کرد که در سال ۱۹۷۹ به آن اوراکل گفته شد. بعد از اوراکل شرکت IBM اولین پایگاه داده رابطه‌ای را در سال ۱۹۸۱ عرضه کرد که SQL/DS (SQL Data System) نام گرفت. در سال ۱۹۸۲ مؤسسه استاندارد آمریکا (ANSI) اهمیت مدل رابطه‌ای را دریافت و روی زبان پایگاه داده رابطه‌ای (RDL) شروع به کار کرد. در سال ۱۹۸۴ سازندگان محصولاتی نظیر Oracle، SQL/DS، DB2، تصمیم گرفتند که استاندارد برای زبان پایگاه داده رابطه ای (RDL) ایجاد کنند. اولین محصول این استاندارد در سال ۱۹۸۶ تحت نام SQL-86 عرضه شد که دارای استاندارد ANSI و ISO بود و بعد از آن در سال ۱۹۸۹ نیز نسخه بعدی این زبان ایجاد گردید. آخرین استاندارد عرضه شده که SQL2 یا SQL-92 نامیده شد یک کار جامع از اعضای استاندارد بین المللی بود که هم زبان پیشرفته‌ای بود و هم فاقد مشخصه‌های مبهم استاندارد اصلی یعنی SQL-86 بود. در ویژوال بیسیک از یک نوع SQL استاندارد مایکروسافت استفاده می شود که Jet SQL نام دارد و در برنامه‌های مایکروسافت مورد استفاده قرار می گیرد و بر پایه استاندارد SQL ساخته شده است. در این کتاب این نوع SQL را بررسی می‌کنیم.

سؤال (Query) و SQL

سؤال یا Query یکی از عناصر پایگاههای داده می‌باشد که اساس ساختار آن SQL است و در کنار بقیه عناصر پایگاههای داده نقش مهمی را ایفا می‌کند. از طریق سؤالها می‌توان بسیاری از عملیات پایگاههای داده را با سرعتی بسیار بالا انجام داد و از تواناییهای آن در برنامه‌ها استفاده نمود. همان طور که اشاره شد در ویژوال بیسیک و بانک اطلاعاتی مایکروسافت اکسس از Jet SQL برای ساخت سؤال استفاده می‌شود.

انواع سؤال (Query)

سؤال یا Query به دو دسته انتخابی و اجرایی تقسیم بندی می شود. نتیجه حاصل از اجرای یک سؤال از نوع انتخابی یک لیست داده می باشد که همانند یک جدول می توان از آن استفاده کرد و می توان به وسیله این روش اعمال دلخواه را روی جداول انجام داد. برای مثال می توانید جداول مختلف را ترکیب کرده و لیست داده ای را به وجود آورید که از فیلدهای این جداول تشکیل شده باشد و یا در سؤال خود از شرط استفاده کنید. مثلاً لیست داده ای از چند جدول به وجود آورید که فیلد خاصی از این لیست داده، دارای مقدار مخصوصی باشد و بسیاری از امکانات دیگر که در این فصل با مهمترین آنها آشنا خواهید شد.

نوع دیگر، سؤال از نوع اجرایی است که نتیجه حاصل از اجرای آن یک لیست داده نمی باشد، بلکه با اجرای آن عملیاتی روی داده ها و یا ساختار جداول انجام می گردد. مثلاً رکوردهای خاصی را در یک جدول حذف می کند، اطلاعات فیلدی را تغییر می دهد و یا یک جدول را با ساختاری مشخص ایجاد می کند و...

برای ایجاد یک سؤال و یا استفاده از آن باید مستقیماً از کدهای SQL کمک بگیریم. بدین معنی که باید با نوشتن یک کد SQL، سؤال مورد نظر را اعمال کنیم و یا آن را در یک QueryDef ذخیره سازیم. در ادامه با این زبان بیشتر آشنا خواهید شد و کاربرد عملی آن را در برنامه ها خواهید دید.

آشنایی با زبان SQL

الف) دستور عمومی یک SQL انتخابی ساده: دستور عمومی این کد به صورت زیر می باشد:

```
SELECT [ALL|DISTINCT] select_List
FROM table_names
[WHERE {search_criteria|join_criteria}
  {{AND|OR search_criteria}}]
ORDER BY {field_list} [ASC|DESC]
```

دستور فوق کلی ترین دستور در SQL می باشد و بیشترین کاربرد را در برنامه ها خواهد داشت. در ادامه پارامترهای این دستور را بررسی می کنیم و در قسمت بعد مثال این فرمان را فرا خواهید گرفت.

• **SELECT**: این کلمه نشان می دهد که سؤال از نوع انتخابی است و نتیجه اجرای فرمان، یک لیست داده می باشد که از پایگاه داده انتخاب شده است. عنصر *select_List* نشان دهنده نام فیلدهای یک و یا چند جدول است که می خواهید در لیست داده حاصل از نتیجه اجرای دستور قرار گیرد. در صورتی که بیش از یک فیلد در این قسمت قرار بگیرد نام هر فیلد با کاما (،) از نام فیلد دیگر جدا می گردد. اگر در این قسمت از (*) استفاده کنید به این معنی است که می خواهید همه فیلدهای یک

جدول انتخاب شود. در صورتی که بیش از یک جدول در سؤال وجود داشته باشد، برای هر فیلد باید قبل از نام فیلد، نام جدول آن نیز ذکر گردد و حالت کلی آن به صورت زیر است:

table_name.field_name

که (.) نام جدول را از نام فیلد جدا می‌سازد.

• **ALL و DISTINCT**: این دستور، بیانگر این است که می‌خواهید تمام رکوردها در لیست داده شما قرار گیرد و نسبت به رکوردهای تکراری بی‌اعتنا باشد و **DISTINCT** فقط رکوردهای غیر تکراری را بر می‌گرداند و برای اجرا، زمان بیشتری را می‌گیرد. این مشخصه در جداول مرتبط اهمیت می‌یابد.

• **FROM**: بعد از این دستور، نام جداولی آورده می‌شود که فیلدی از آنها در قسمت *select-field* آورده شده است. اگر در قسمت فیلدها، از جداول مختلف، فیلد انتخاب شده باشد باید نام جداول با (.) از هم جدا گردد.

• **WHERE**: از این دستور در **Ansi SQL** برای دو منظور استفاده می‌شود. یکی ارتباط بین فیلدهای جداول مرتبط و دیگری شرطی که محدودکننده رکوردهای لیست داده می‌باشد. بعد از **WHERE** باید یک اوپراتور و دو مقدار باشد که اولی حتماً باید یک فیلد باشد (این فیلد لازم نیست حتماً در *select-field* انتخاب شده باشد، اما باید از جدولی انتخاب شود که نام آن در *table-names* قید شده باشد).

• **اوپراتورهای SQL**: شامل **IN, { IS Null | IS NotNull }, Like** و اوپراتورهای ریاضی شامل **=, <=, >, <, >, <** می‌باشد. اگر از اوپراتور " = " مقادیر *Table-Name.Field-Name* در طرفین آن بعد از دستور **WHERE** استفاده کنید یک ارتباط داخلی (**inner join**) بین دو جدول ایجاد می‌شود. ارتباط بین دو جدول به صورت زیر است:

۱ - **داخلی (inner)**: در این نوع ارتباط فقط رکوردهایی از جدول اول انتخاب می‌شود که رکورد مشترک در جدول دوم داشته باشد.

۲ - **خارجی (Outer)**: که شامل خارجی چپ و خارجی راست می‌باشد و در ارتباط خارجی چپ تمام رکوردهای جدول اول چه در جدول دوم رکورد مشترک داشته و یا نداشته باشند، لیست می‌شوند و در ارتباط خارجی راست تمام رکوردهای جدول دوم چه در جدول اول باشند یا نباشند لیست می‌شود.

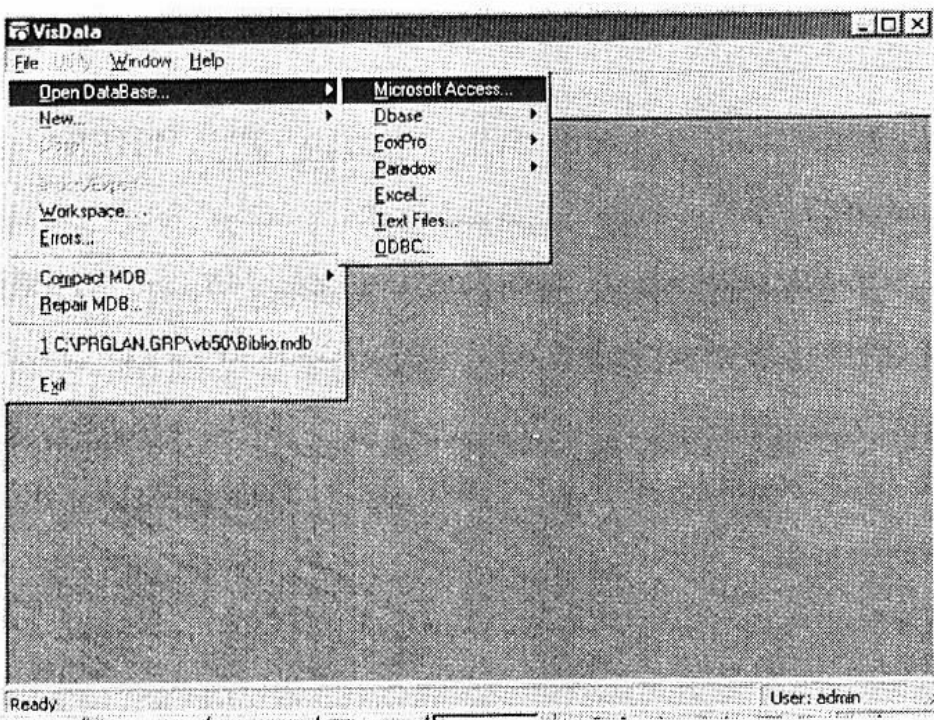
• **ORDER BY**: دستوری است که باعث می‌شود عملیات مرتب‌سازی **Sort** یا روی یک یا چند فیلد صورت گیرد و در صورتی که بیش از یک جدول در *table-names* لیست شده باشد باید قبل از فیلد، نام جدول نیز آورده شود و با (.) از یکدیگر جدا شوند *table-name. field name*. دو پارامتر **ASC** و **DESC** نوع مرتب‌سازی را مشخص می‌کند که **ASC** مرتب‌سازی از کم به زیاد و **DESC** مرتب‌سازی از زیاد به کم است.

استفاده از برنامه Visual Data برای آزمایش سؤاها

برنامه Visual Data یک مثال ضمیمه شده به ویژوال بیسیک می باشد که از نسخه پنج به بعد در ویژوال بیسیک به عنوان مدیر پایگاه داده در منوی Add - Ins قرار گرفت. در این قسمت قصد داریم تا مثالهای SQL را از طریق این برنامه به نمایش درآوریم و مباحث قبل را کاملتر سازیم. شما می توانید با استفاده از Visual Data کدهای SQL خود را طبق مراحل زیر امتحان کنید.

۱ - از منوی Add - Ins گزینه Visual Data Manager را انتخاب نمایید.

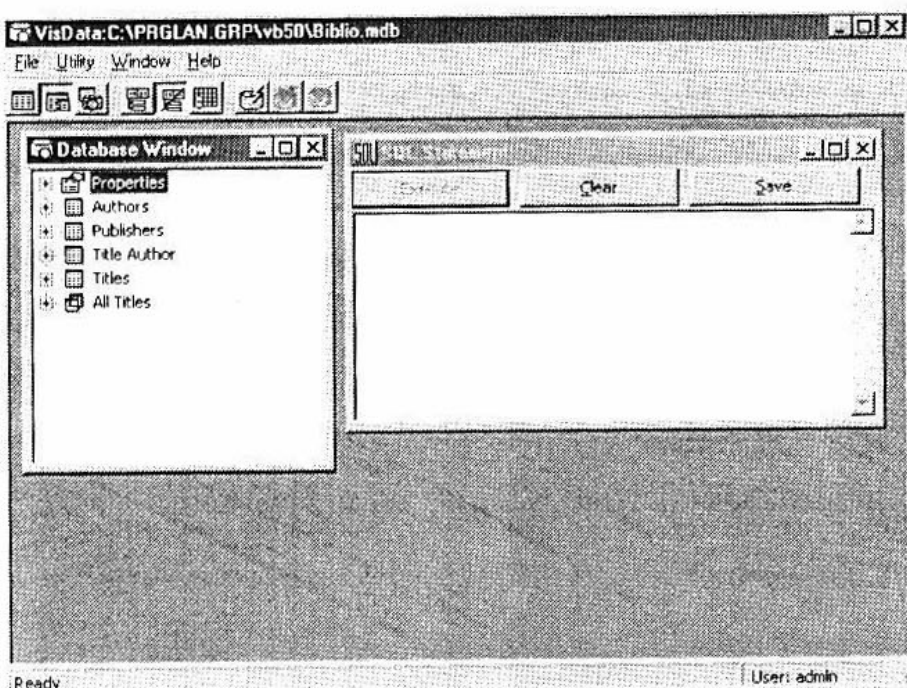
۲ - از منوی File در Vis Data گزینه Open Database | Microsoft Access را انتخاب کنید تا کادر تبدالی جهت باز کردن فایل MDB نشان داده شود.



شکل ۱-۸.

باز کردن پایگاه داده.

۳ - در شاخه برنامه ویژوال بیسیک فایل Biblio. MDB را انتخاب کنید. جداول و سؤالهای این پایگاه داده در Database Window لیست می شود.



شکل ۲-۸.

باز شدن پایگاه داده Biblio.mdb و لیست اشیاء آن در Visual Data.

۴- کد زیر را در پنجره SQL Statment وارد کنید.

```
SELECT * FROM Authors
```

حال در قسمت Toolbar شکل جدول را انتخاب نمایید تا سؤال شما به صورت جدول نمایش داده شود.

شکل ۳-۸.



Toolbar در برنامه Visual Data.

۵- دکمه Execute را در پنجره SQL Statement کلیک کنید.

در صورتی که پیغامی مبنی بر نوع SQL Passthrough داده شد No را انتخاب نمایید.

نتیجه اجرای SQL به صورت جدول در پنجره **Select * From Authars** نشان داده می شود.

	Au_ID	Author	Year Born
▶	1	Jacobs, Russell	
	2	Metzger, Philip W.	
	3	Boddie, John	
	4	Sydow, Dan Parks	
	6	Lloyd, John	
	8	Thiel, James R.	
	10	Ingham, Kenneth	
	12	Wellin, Paul	
	13	Kamin, Sam	
	14	Gaylord, Richard	
	15	Curry, Dave	
	17	Gardner, Juanita Me	
	19	Knuth, Donald E.	
	21	Hakim, Jack	
	22	Winchell, Jeff	
	24	Clark, Claudia	

شکل ۴-۸.

نتیجه اجرای SQL.

همان طور که در شکل ۴-۸ می بینید نتیجه SQL به صورت لیستی از اطلاعات نمایش داده شده است، بنابراین SQL و سؤال از نوع انتخابی است. این کد فرمان می دهد که تمام فیلدهای جدول Authors انتخاب شود.

۶- در صورتی که بخواهیم فیلدهای خاصی در لیست داده انتخاب شود به جای (*) نام فیلدها را قرار می دهیم، برای مثال کد SQL زیر را اجرا کنید.

```
SELECT Au_ID, Author FROM Authors
```


شکل ۵-۸. نتیجه اجرای SQL روی فیلدهای مشخص.

au_id	author
1	Jacobs, Russell
2	Metzger, Philip W.
3	Boddie, John
4	Sydow, Dan Parks
6	Lloyd, John
8	Thiel, James R.
10	Ingham, Kenneth
12	Wellin, Paul
13	Kamin, Sam
14	Gaylord, Richard
15	Curry, Dave
17	Gardner, Juanita Me
19	Knuth, Donald E.
21	Hakim, Jack
22	Winchell, Jeff
24	Clark, Claudia

دستور فوق به این معنی است که دو فیلد Au-Id , Author از جدول Authors انتخاب شوند.

نکته :

در صورتی که نام فیلد از دو قسمت تشکیل شده باشد باید نام فیلد را بین [] قرار داد. برای مثال فیلد Company Name را در SQL نشان می دهیم.

SELECT [Company Name] FROM Publishers

۷ - برای انتخاب لیست داده‌ای که تعدادی فیلد از یک جدول و تعدادی فیلد از جدول دیگری را داراست باید در قسمت نام فیلدها ابتدا نام جدول مورد نظر و سپس نام فیلد را قرار دهیم و این دو را با (.) از یکدیگر جدا کنیم. در کد زیر فیلد Author از جدول Authors همراه با فیلد Name از جدول Publishers انتخاب می گردد.

SELECT Authors.Author,Publishers.Name FORM Authors,Publishers

شکل ۶-۸. نتیجه اجرای SQL روی فیلدهای چند جدول.

author	name
Jacobs, Russell	SAMS
Jacobs, Russell	PRENTICE HALL
Jacobs, Russell	M & T
Jacobs, Russell	MIT
Jacobs, Russell	MACMILLAN COMPU
Jacobs, Russell	HIGHTEXT PUBNS
Jacobs, Russell	SPRINGER VERLAG
Jacobs, Russell	O'REILLY & ASSOC
Jacobs, Russell	ADDISON-WESLEY
Jacobs, Russell	JOHN WILEY & SONS
Jacobs, Russell	SINGULAR
Jacobs, Russell	Duke Press
Jacobs, Russell	Oxford University
Jacobs, Russell	Mit Press
Jacobs, Russell	CAMBRIDGE UNIV
Jacobs, Russell	Q E D

۸ - در پنجره SQL Statment روی دکمه Clear کلیک کنید. حال می‌خواهیم لیست داده را توسط دستور WHERE محدود سازیم. کد SQL زیر را وارد کنید:

```
SELECT * FROM Publishers WHERE State="NY"
```

در قسمت اول کد SQL، تمام فیلدها از جدول Publishers انتخاب می‌شود و در قسمت دوم دستور WHERE را اضافه کرده‌ایم. این دستور باعث می‌شود رکوردهایی از این لیست داده نمایش داده شوند که فیلد State در آنها برابر مقدار NY باشد. با اجرای این کد تمام رکوردهایی انتخاب می‌شوند که فیلد State در آنها دارای مقداری برابر NY است.

Address	City	State	Zip
4 Gentry St.	Baldwinsville	NY	13027
15 Columbus Cir.	New York	NY	10023
15 Columbus Cir.	New York	NY	10023
11 W. 42nd St., 3rd	New York	NY	10036
15 Columbus Cir.	New York	NY	10023
State University Plaz	Albany	NY	12246-0
4 Gentry St.	Baldwinsville	NY	13027
605 Third Ave	New York	NY	10158
11 W. 42nd St., 3rd	New York	NY	10036
State University Plaz	Albany	NY	12246-0
29 W. 35th St.	New York	NY	10001-2
State University Plaz	Albany	NY	12246-0
1221 Ave of the Am	New York	NY	10020
15 Columbus Cir.	New York	NY	10023
4 Gentry St.	Baldwinsville	NY	13027

شکل ۷-۸.

نتیجه اجرای SQL با استفاده از دستور Where.

به همین ترتیب می‌توان شرطهای مختلف را روی رکوردهای یک یا چند جدول ایجاد و نتیجه را به صورت یک لیست داده در اختیار داشت. در صورتی که بخواهید شرط مورد استفاده در دستور Where از بیشتر از یک شرط تشکیل شده باشد، باید از دستورات ترکیبی مانند AND و OR در ساخت این ترکیبات استفاده کنید.

۹ - پنجره نتایج سؤال خود را ببینید و در پنجره SQL Statement همان کد قبلی را وارد کنید و به انتهای آن کد زیر را اضافه کنید :

```
ORDER BY Zip
```

دستور فوق اعلام می‌کند که همان لیست داده محدود شده توسط دستور Where تشکیل شود، با این تفاوت که جدول براساس فیلد Zip مرتب گردد .

State	Zip	Telephone	Fax
NY	10001-2291	212-244-3336	
NY	10019-4323		
NY	10020	800-822-8138	
NY	10020	800-822-8138	
NY	10020	800-822-8138	
NY	10020	800-822-8138	
NY	10020	800-822-8138	
NY	10020	800-227-0900	
NY	10020	212-512-2000	
NY	10023	800-922-0579	
NY	10023	212-373-8500	
NY	10023	800-922-0579	
NY	10023	800-922-0579	
NY	10023	800-922-0579	
NY	10023	800-922-0579	

شکل ۸-۸.

لیست مرتب شده توسط دستور

. ORDER BY

بنابراین همان طور که در شکل ۸-۸ می بینید رکوردها به ترتیب فیلد Zip مرتب شده اند و این ترتیب از بالا به پایین می باشد. این نوع مرتب شدن به صورت پیش فرض برای دستور ORDER BY تعریف شده است (ASC). در صورتی که بخواهیم نوع مرتب شدن از پایین به بالا باشد (DESC) پس از ذکر نام فیلد یا فیلدها پس از دستور ORDER BY نوع ترتیب را انتخاب می کنیم (ASC | DESC). در مثال فوق اگر بخواهیم ترتیب قرار گرفتن رکوردها بر اساس فیلد Zip از پایین به بالا باشد کد زیر را وارد می کنیم:

```
SELECT * FROM Publishers
WHERE State = "NY"
ORDER BY Zip DESC
```

State	Zip	Telephone	Fax
NY	14610	716-482-7700	
NY	14610	716-482-7700	
NY	13027	315-638-0175	
NY	13027	315-638-0175	
NY	13027	315-638-0175	
NY	13027	315-638-0175	
NY	13027	315-638-0175	
NY	13027	315-638-0175	
NY	12246-0001		
NY	12246-0001		
NY	12246-0001		
NY	12246-0001		
NY	11791	516-496-7121	
NY	10573		
NY	10158	212-850-6000	212-850

شکل ۸-۹.

ترتیب لیست داده به صوت Desc.

اوپراتورها و عبارات در SQL

همان طور که در ابتدای این فصل اشاره کردیم، SQL از اوپراتورهای ریاضی مانند = ، < ، > ، <= ، >= ، < > پشتیبانی می‌کند و همچنین برای ترکیب عطفی داده‌هایی از نوع متن دارای اوپراتورهای مانند Like و IN و همچنین مقادیر تهی (Null) دارای اوپراتورهایی مانند Is Not و IsNull می‌باشد. همچنین Jet Engin بسیاری از رشته‌های محلی و تابعهای عددی VBA را در SQL می‌تواند به کارگیرد. به طور کلی اوپراتورها در SQL به دو دسته عملگرهای دو تایی و تک عملگرها تقسیم می‌شوند. تمام توابع ریاضی و دستور Between جزء گروه عملگرهای دو تایی می‌باشد و اوپراتورهایی از قبیل Like , In , Is Null , Is Not Null جزو تک عملگرها می‌باشند.

نام گذاری این اوپراتورها به این علت است که عبارت مورد استفاده در عملگرهای دو تایی از دو بخش و عبارت مورد استفاده در تک عملگرها از یک بخش تشکیل شده است. در ادامه با این عملگرها بیشتر آشنا خواهیم شد.

اوپراتورها و توابع ریاضی در عملگرهای دو تایی

استفاده از اوپراتورهای ریاضی در SQL و زبانهای مختلف تفاوتی نمی‌کند؛ بنابراین در زیر تنها به نکات مهم استفاده از این عملگرها و توابع ریاضی در SQL (مخصوصا در شرط Where) می‌پردازیم .

- اوپراتورهای مقایسه شامل = ، < ، > ، هم برای مقایسه داده‌های عددی و هم برای مقایسه داده‌های متنی کاربرد دارند .
- اوپراتورهای مقایسه عددی شامل > ، <= ، < و >= در مقایسه داده‌های عددی به کار برده می‌شوند .
- بقیه اوپراتورهای ریاضی شامل + ، - ، / ، * و ^ و ** اوپراتورهای مقایسه‌ای نیستند بلکه در محاسبه مقدار فیلد سؤال مورد استفاده قرار می‌گیرند . در قسمت بعد ، این مورد را بیشتر بررسی خواهیم کرد.
- برای مقایسه داده‌های متنی که عدد را نشان می‌دهد می‌توان از تابع Val() استفاده کرد . مثلا در جدول Publishers فیلد Zip از نوع Text تعریف شده است ، اما در آن یک عدد نشان داده می‌شود . بنابراین می‌توان از تابع Val برای مقایسه مقدار عددی این فیلد استفاده کرد .

```
SELECT * FROM Publishers  
WHERE Val(Zip) < 12000
```


پس از اجرای کد فوق خطائی از طرف Vis Data داده می شود مبنی براینکه این شرط قابل اجرا نیست. این خطا به این علت رخ می دهد که بعضی از مقادیر فیلد Zip، دارای مقدار تهی یا Null هستند و تابع Val() نمی تواند مقدار آنها را به عدد برگرداند بنابراین به شرط خود دستوری را اضافه می کنیم تا مقادیر Null در لیست انتخاب نشوند.

```
WHERE Zip IS NOT NULL AND Val(Zip) < 12000
```

• اوپراتور Between برای انتخاب داده ها، بین یک محدوده تعریف شده است. نوع داده ها می تواند هم عددی، هم تاریخ و زمان باشد. دستور کلی استفاده از آن به صورت زیر است.

```
field_name Between Value1 And Value2
```

برای مثال در شرط کد SQL مثال قبل می توان از Between استفاده کرد.

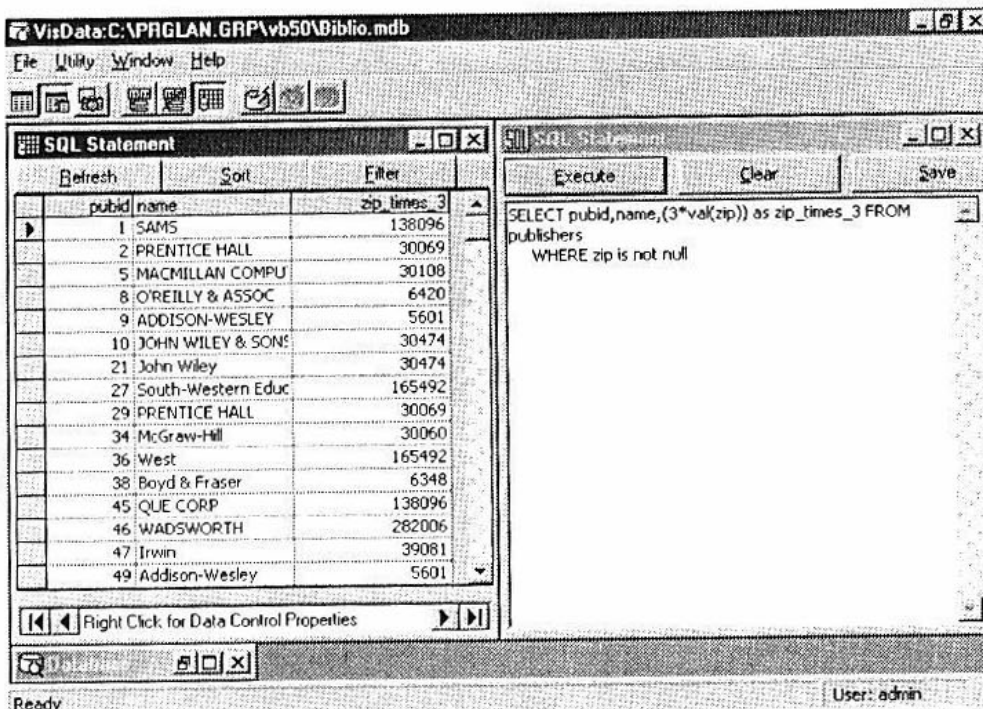
```
WHERE Val(Zip) Between 2000 AND 3000
```

ستونهای محاسباتی در سؤال

از جمله امکانات SQL در تشکیل لیست داده، ایجاد ستونهایی است که براساس یک فرمول تشکیل شده است. در این فرمول می توان از اوپراتورهای ریاضی SQL استفاده کرد و یا اگر در برنامه خود از آن استفاده کنید، می توانید از توابعی که خود تعریف کرده اید جهت تشکیل این فرمول استفاده نمایید. برای مثال کد زیر را در پنجره SQL Statement وارد کنید.

```
SELECT PubID,Name, 3*Val(Zip) AS Zip_Times_3  
FROM Publishers  
WHERE Zip is Not Null
```

این کدهای SQL فیلدهای PubID, Name و یک فیلد محاسباتی از ضرب یک عدد در مقدار ارزش عددی فیلد Zip ایجاد می کند و در جدول نام Zip-Times-3 را به آن اختصاص می دهد.



شکل ۱۰-۸.

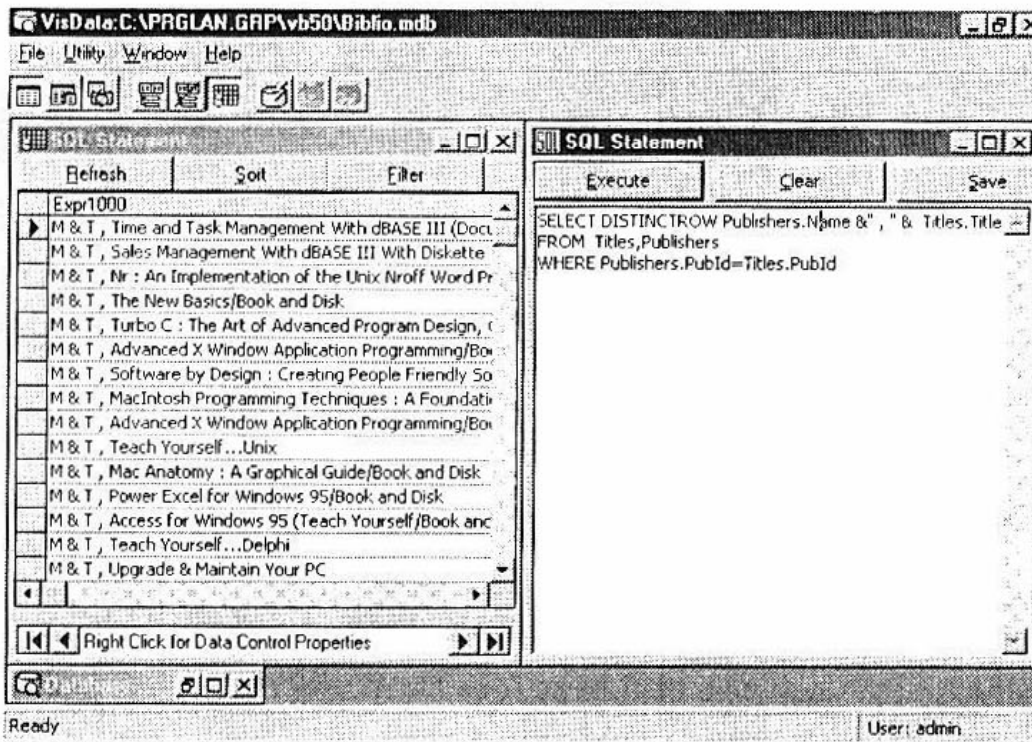
ایجاد فیلد محاسباتی
در SQL.

نقش AS پس از فرمول ، اختصاص یک نام برای فیلد محاسبه شده می باشد . از این دستور می توان پس از هر فیلدی استفاده کرد و نام آن را تغییر داد . در صورتی که پس از فیلد محاسباتی توسط AS نامی به آن اختصاص نگردد این فیلد با نام Expr n تشکیل می شود و اگر از فیلدهای محاسباتی دیگر استفاده کنید n به ترتیب زیاد می گردد .

ستونهای ترکیبی

ستونهای ترکیبی یک حالت از ستونهای محاسباتی است که می توان در آنها چند فیلد را در یک ستون نمایش داد. فرمول این ستونها توسط اپراتورهای اتصال متن و نام فیلدها تشکیل می شود . در مثال زیر جدولی تشکیل می شود که یک ستون آن را فیلد Title از جدول Titles و Name از جدول Publishers تشکیل می دهد و این دو جدول توسط فیلد PubID به هم ارتباط داده شده اند . شکل (۸-۱۱)

```
SELECT DISTINCTROW Publishers.Name & ", " & Titles.Title
FROM Publishers, Titles
WHERE Publishers.PubID = Titles.PubID
```



شکل ۸-۱۱

ایجاد فیلد ترکیبی در SQL .

همان طور که در شکل مشاهده می کنید ، ستونی به نام Exp 1000 ایجاد شده است که هر دو فیلد را نمایش می دهد. اپراتورهای ترکیب متن شامل & و + می باشند که هر دو یک عمل را انجام می دهند . کار این اپراتورها اتصال دو متغیر از نوع متن است و در مثال بالا این عمل روی دو فیلد از جداول مرتبط صورت گرفت .

توجه : در قسمتهای بعد، ارتباط بین جداول بررسی خواهد شد .

اوپراتورهای تک عملگر

یکی از پرکاربردترین اوپراتورها برای شرط دستور Where روی فیلدهای متنی در SQL ، اوپراتور Like می باشد . توسط این دستور شما می توانید روی تک تک کاراکترهای هر فیلد جستجو کنید و لیستی را به وجود آورید که رکوردهای تشکیل دهنده آن در این شرط صدق کند. جدول ۸-۱ استفاده های مختلف اوپراتور Like را نشان می دهد.

جدول ۸-۱ : کاربرد اوپراتور Like در SQL .

هدف	دستور Jet SQL	موارد انتخاب شده
پیدا کردن تمام متنهایی که درون آنها کاراکترهای مخصوصی باشند .	“Like“	ram , rams Damset , amnesty
پیدا کردن تمام متنهایی که ابتدای آنها با کاراکترهای خاصی شروع شود.	“Like John *“	johnson , johnson
پیدا کردن متنهایی که انتهای آنها کاراکترهای خاصی باشد .	“Like ? am“	jackson , johnson Anderson
پیدا کردن متنهایی که یک یا چند فیلد آن نامشخص است.	Like “Glen?“ Like “? am“	Glenn , Glens dam , pam , Ram

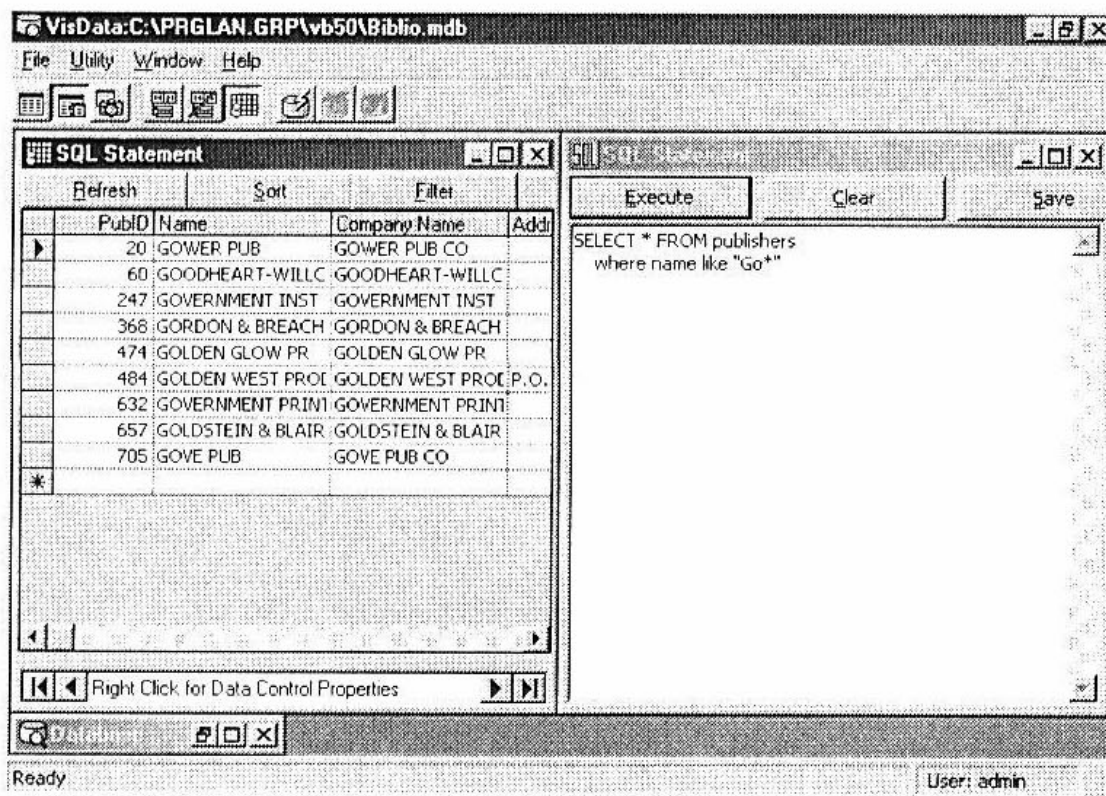
همان طور که در جدول ۸-۱ نشان داده شد ، با استفاده از علایم خاصی می توانیم انواع شرطها را به وجود آوریم . در جدول ۸-۲ این علامات توضیح داده شده است .

جدول ۸-۲ : علامات مورد استفاده در دستور Like .

هدف	موارد انتخاب شده
* این علامت نشانگر تعداد کاراکترهای نامعلوم است . برای مثال " * Like شامل تمام مقادیر می باشد و یا " * G Like شامل تمام مقادیری است که ابتدای آن با G شروع می شود و تعداد کاراکترهای بعد از آن مشخص نیست و یا " * I * N Like بیانگر تمام عباراتی است که ابتدا و انتهای آن کاراکترهای مشخص I و N است مانند IN , IRAN , ...	
? این علامت نشانگر یک کاراکتر نامشخص است . مثلا: " ??? " Like تمام عبارات سه حرفی را انتخاب می کند و یا " X ? " Like تمام عباراتی را انتخاب می کند که ابتدای آن X و حرف بعد از آن نامشخص است .	

در مثال زیر تمام رکوردهایی از جدول Publishers انتخاب می‌شوند که فیلد Name در آنها با دو کلمه GO شروع شود.

SELECT * FROM Publishers WHERE Name Like "GO*"



شکل ۱۲-۸.

استفاده از علامات ویژه در SQL.

جداول مرتبط

همان طور که در ابتدای فصل اشاره کردیم یکی دیگر از کاربردهای دستور Where ارتباط دادن بین جداول می‌باشد و ارتباطهای بیشتر به وسیلهٔ اوپراتور AND بعد از Where برقرار می‌شوند.

ارتباط داخلی INNER JOIN

این ارتباط به دو روش صورت می‌گیرد:

الف) توسط دستور Where که دستور کلی آن به صورت زیر می‌باشد:

```
SELECT Table1.field_name,... Table2.field_name ...
FROM Table1,Table2
WHERE Table1.field_name = Table2.field_name
```

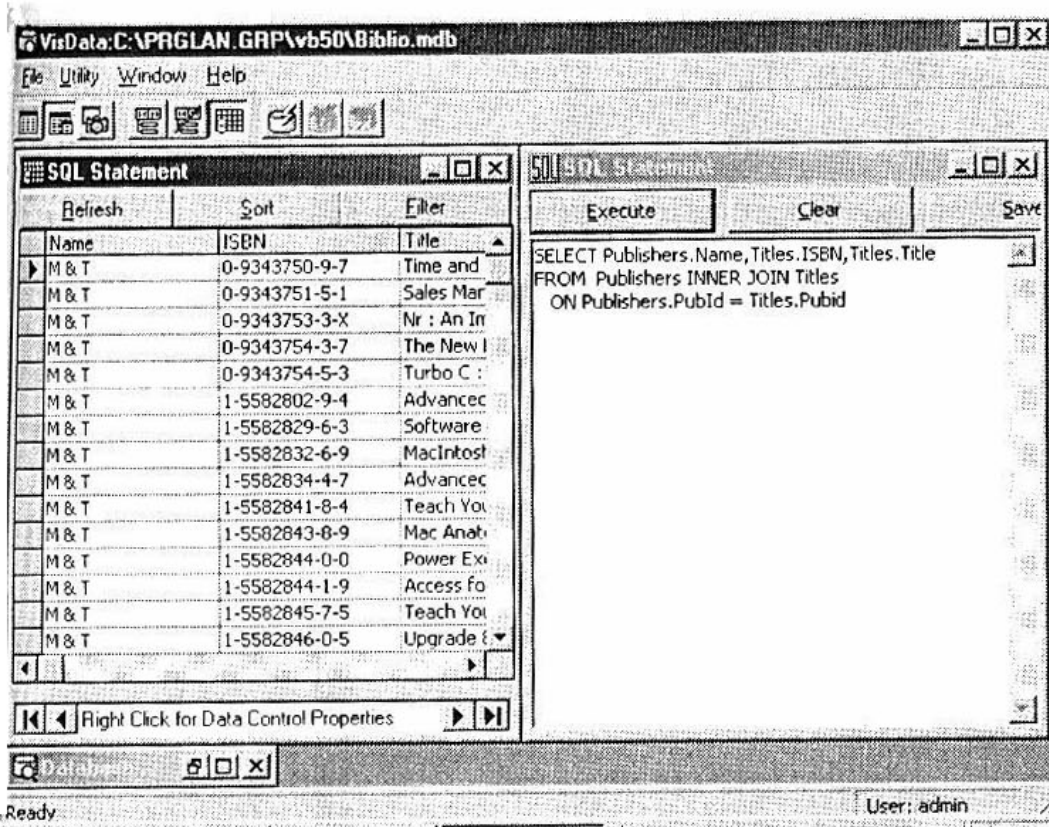
ب) توسط دستور Join که دستور کلی آن به صورت زیر می‌باشد:

```
SELECT Table1.field-name,... Table2.field_name
FROM Table1 INNER JOIN Table2
On Table1.field-name = Table2.field_name
```

مثال زیر یک ارتباط داخلی روی فیلد PubID از جدول Publishers و فیلد PubID از جدول Authors برقرار می‌کند.

```
SELECT Publishers.Name, Titles.ISBN, Titles.Title
FROM Publishers INNER JOIN Titles
On Publishers.PubID = Titles.PubID
```

نتیجه اجرای این کد در شکل ۱۳-۸ آورده شده است.



شکل ۱۳-۸.

ایجاد ارتباط داخلی روی یک فیلد در SQL.

اگر بخواهیم همین ارتباط را توسط دستور Where انجام دهیم از کد زیر استفاده می‌کنیم:

```
SELECT Publishers.Name, Titles.ISBN, Titles.Title
FROM Publishers, Titles
WHERE Publishers.PubID = Titles.PubID
```

ارتباط داخلی چندگانه

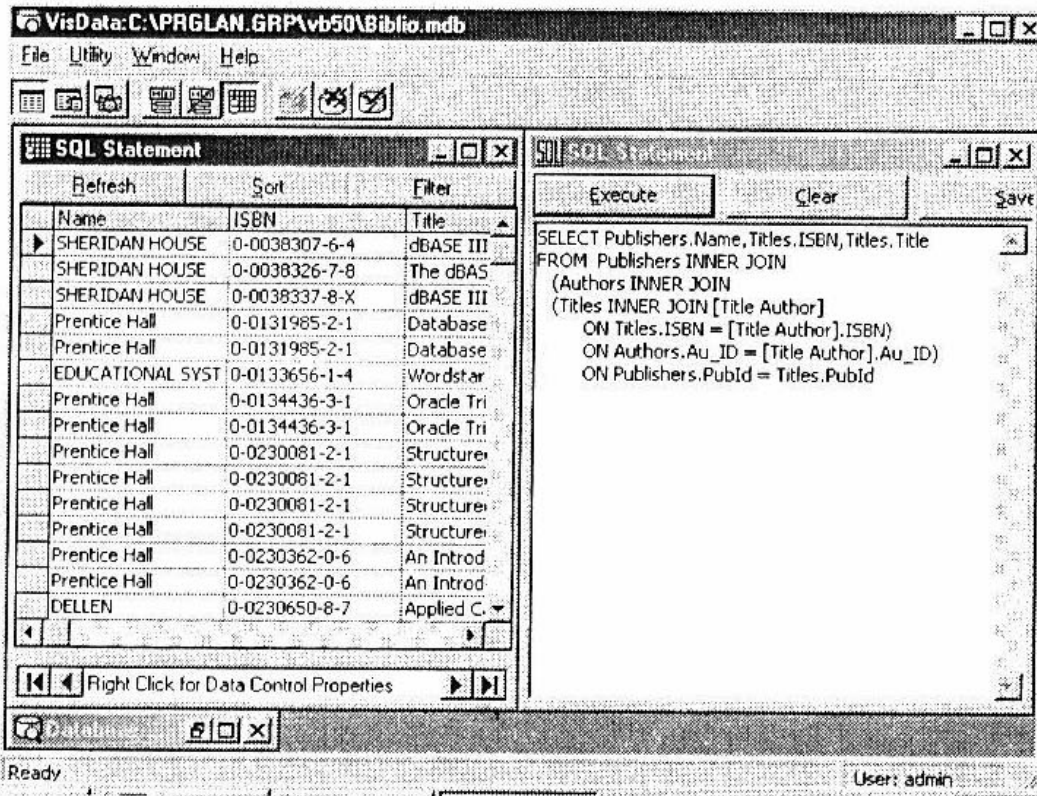
شما می‌توانید یک ارتباط داخلی را بین چند جدول برقرار کنید. برای مثال شما می‌توانید بین جداول Publishers, Titles, Authors, Title از پایگاه داده Biblio. MDB توسط کد SQL زیر ارتباط برقرار کنید.

```
SELECT DISTINCTROW Titles.Title, Publishers.Name, Titles.ISBN, Authors.Author
FROM Publishers INNER JOIN
(Authors INNER JOIN
(Titles INNER JOIN [Title Author]
On Titles.ISBN = [Title Author].ISBN)
On Authors.Au_ID = [Title Author].Au_ID)
On Publishers.PubID = Titles.PubID)
```

دستور DistinctRow باعث می‌شود که رکوردهای تکراری نمایش داده نشوند و نتیجه سؤال قابل ویرایش می‌گردد. این کد را می‌توان توسط دستور Where به صورت زیر نوشت:

```
SELECT Titles.Title, Publishers.Name, Titles.ISBN, Authors.Author
FROM Publishers, Titles, Authors, [Title Author]
WHERE Titles.ISBN = [Title Author].ISBN
AND Authors.Au_ID = [Title Author].Au_ID
AND Publishers.PubID = Titles.PubID
```


شکل ۱۴-۸ نتیجه اجرای سؤال را نمایش می دهد.



شکل ۱۴-۸.
ایجاد ارتباط داخلی
چندگانه
در SQL.

ارتباط خارجی OUTER JOIN

از یک ارتباط داخلی فقط سطرهایی نتیجه می شود که نظیر مقدار فیلد در جدول مرتبط وجود داشت. اما در یک ارتباط داخلی تمام سطرها از یک جدول و تمام سطرها با مقادیر نظیر شده از جدول دیگر انتخاب می شوند. بنابراین دونوع ارتباط خارجی خواهیم داشت:

ارتباط خارجی چپ: سطرهایی را انتخاب می کند که شامل تمام سطرهای جدول قرار گرفته در طرف چپ دستور LEFT OUTER JOIN و سطرهای نظیر شده از جدول طرف دیگر می باشد. در دستور WHERE این ارتباط توسط اوپراتور = * مشخص می شود

ارتباط خارجی راست: سطرهایی را انتخاب می کند که شامل تمام سطرهای جدول قرار گرفته در طرف راست دستور RIGHT OUTER JOIN و سطرهای نظیر شده از جدول طرف دیگر می باشد. در دستور WHERE این ارتباط توسط اوپراتور = * مشخص می شود.

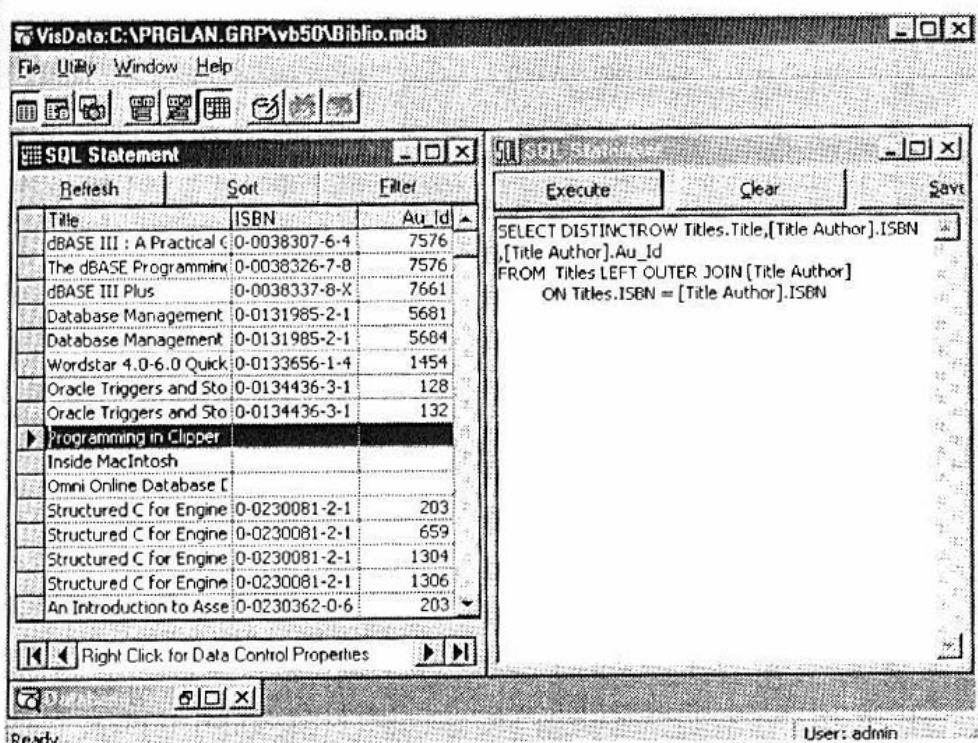
با اجرای مثال زیر تمام رکوردها از جدول Titles همراه با رکوردهای نظیر شده از جدول Authors انتخاب می شوند.

```
SELECT DISTINCTROW Titles.title,[Title Author].ISBN,[Title Author].Au_ID
FROM Titles LEFT OUTER JOIN [Title Author]
ON Titles.ISBN = [Title Author].ISBN
```


نکته: توجه داشته باشید که در Jet SQL برای ارتباط خارجی نمی‌توان از Where استفاده کرد و اوپراتورهای *= و =* تحت پشتیبانی SQL-92 می‌باشد.

نتیجه اجرای سؤال مثال بالا از پایگاه داده Biblio. MDB در شکل ۸-۱۵ نشان داده شده است.

اولین سطری که نظیرش در [Title Author] یافت نشده است و از جدول سمت چپ دستور JOIN می‌باشد در شکل نشان داده شده است.



شکل ۸-۱۵

ایجاد ارتباط خارجی در SQL

سؤالهای تودرتو Nested Queries

همان طور که در فصلهای قبل توضیح دادیم، QueryDef یک شیء از اجزای پایگاه داده Jet می‌باشد و می‌توانستید توسط دستور Create QueryDef آن را ایجاد و ذخیره نمایید. پایه ساخت QueryDef یک کد SQL می‌باشد که نتیجه اجرای آن به صورت سؤال خواهد بود و اگر این سؤال از نوع انتخابی باشد و در یک QueryDef ذخیره شود، برخورد Jet با آن همانند یک جدول خواهد بود. بنابراین اگر کد SQL را به صورت QueryDef ذخیره سازیم می‌توانیم از آن در سؤالهای دیگر استفاده کنیم. به این نوع سؤالها که از سؤالهای دیگر درونشان استفاده می‌کنند سؤالهای تودرتو گفته می‌شود. همان طور که اشاره کردیم نحوه استفاده از QueryDef همانند یک جدول می‌باشد یعنی در قسمت field_list از کد SQL به صورت کلی زیر می‌توانیم فیلد مورد نظر را مشخص کنیم.

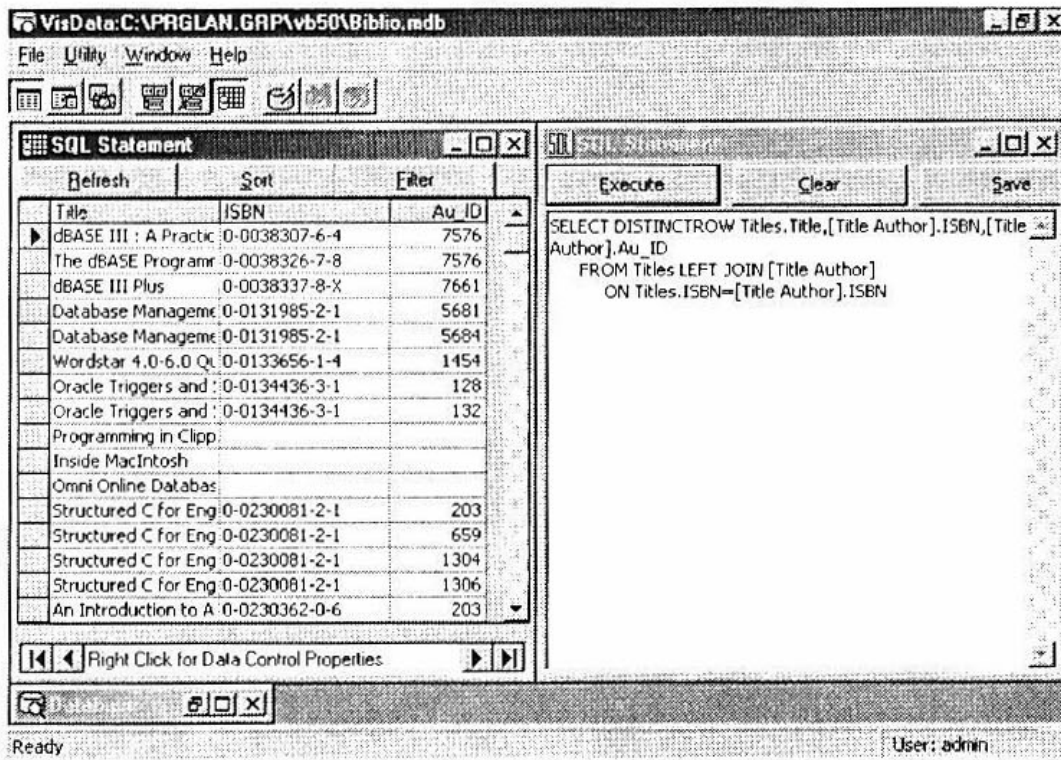
QueryDefName.field_name

برای مثال برای اجرای یک سؤال تودرتو مراحل زیر را انجام دهید.

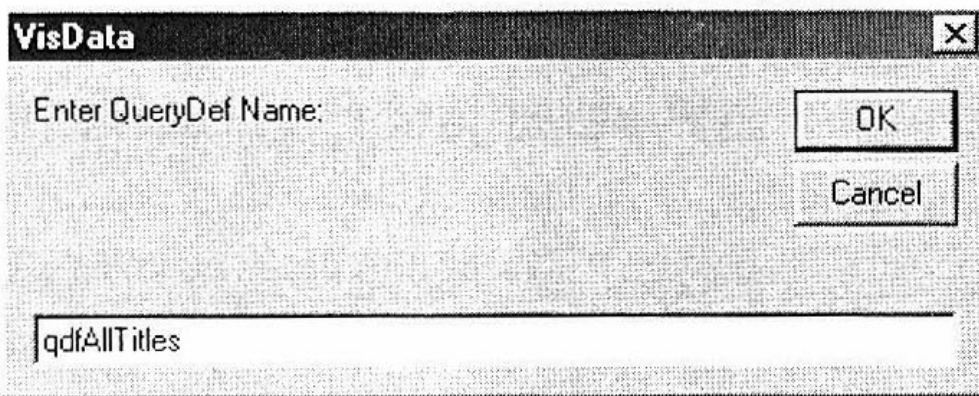
۱- در پنجره کد SQL در VisData روی دکمه فرمان Clear کلیک نموده و کد زیر را وارد کنید.

```
SELECT DISTINCTROW Titles.Title,[Title Author].ISBN,[Title Author].Au_ID
FROM Titles LEFT OUTER JOIN [Title Author]
ON Titles.ISBN = [Title Author].ISBN
```

۲- روی دکمه فرمان Save کلیک کنید تا پنجره گرفتن پارامتر Vis Data برای نام QueryDef فعال گردد (شکل ۱۷-۸) و سپس qdfAllTitle را در آن تایپ کنید و دکمه فرمان OK را برای ذخیره شدن QueryDef کلیک نمایید.



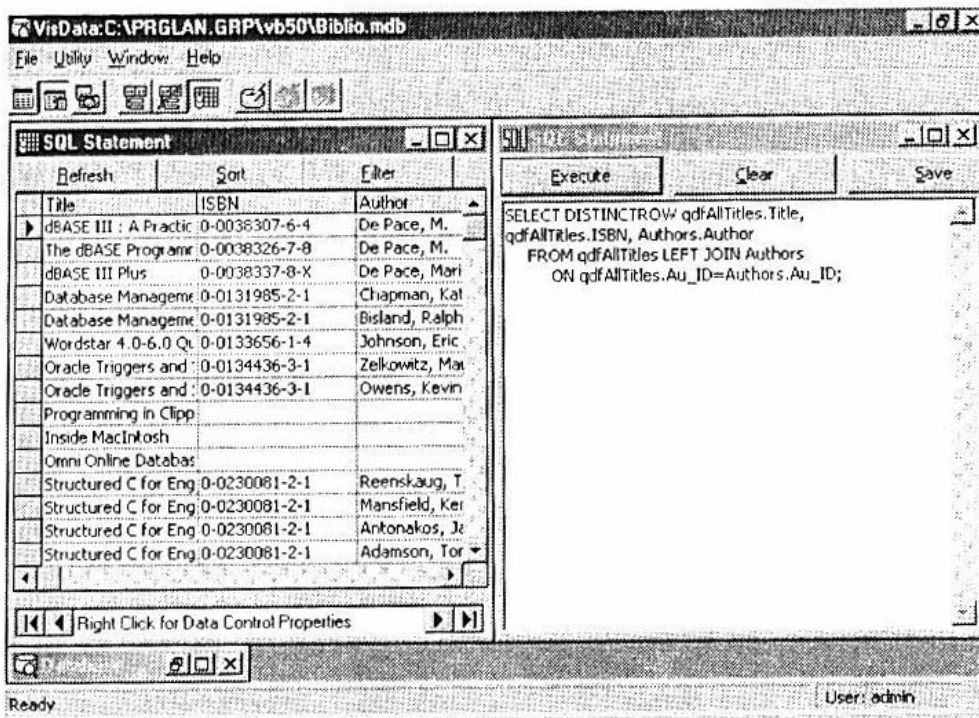
شکل ۱۶-۸.
نتیجه اجرای SQL.



شکل ۱۷-۸.
گرفتن پارامتر نام برای نام سؤال.

۳- روی دکمه فرمان در Clear پنجره SQL کلیک کنید و کد زیر را در آن تایپ نمایید.

```
SELECTDISTINCTROW qdfAllTitle.Title,qdfAllTitle.ISBN,Authors.Author
FROM qddfAllTitle.LEFT OUTER JOIN Authors
ON qdfAllTitle.Au-ID = Authors.Au-ID
```



شکل ۱۸-۸
نتیجه اجرای SQL

زیر سؤال Sub Queries

سؤالهای تودرتو و زیر سؤالها دارای مفاهیم بسیار نزدیکی هستند. همان طور که دیدید یک سؤال تودرتو از دو کد مجزا تشکیل شده بود، یعنی ابتدا یک سؤال به صورت شیء QueryDef ذخیره می شود و سپس در سؤال دیگر همانند جدول، مورد استفاده قرار می گرفت، اما در زیر سؤال، دیگر نیازی به ذخیره سؤال به صورت شیء QueryDef نیست بلکه کد زیرسؤال، مستقیماً در داخل کد اصلی وارد می گردد و تعداد ستونهای آن نمی تواند از یک ستون بیشتر باشد. این امکان توسط اکسس ۲ به Jet 2.0 اضافه شد و در نسخه های بعد نیز قرار گرفت.

از زیرسؤال می توان در قسمت field-list از دستور SELECT و یا در شرط دستور WHERE استفاده کرد که نحوه استفاده از آن به صورت زیر می باشد.

– شرط به صورت زیر مجموعه ای Subset Sub Queries: این زیر سؤال توسط دستور SELECT ایجاد می شود و رکوردهای سؤال اصلی را محدود می کند. دستور کلی آن به صورت زیر است.

```
SELECT field-list
FROM table-list
WHERE field-name [NOT] IN
(SELECT field-name FROM table-name WHERE Criteria)
```

دستور IN باعث می شود یک ارتباط بین سؤال و زیر سؤال به وجود آید و مانند اپراتور = عمل می کند و [NOT] قبل از دستور IN انتخابی است و نتیجه استفاده از آن مانند اپراتور < > می باشد.

– شرط به صورت مقایسه ای Comparison Sub Queries: گاه فیلدهای عددی را ارزیابی می کند و رکوردهای سؤال اصلی را محدود می سازد. دستور کلی آن به صورت زیر است:

```
SELECT field_list
FROM table_list
```

WHERE field_name {<|<|=|=|>} { Any | Some | All }

(SELECT field_name FROM table_name WHERE Search_Criteria)

دستور Some یا Any رکوردهایی را برمی گرداند که عمل مقایسه با هر رکورد زیرسؤال، نتیجه True را داشته باشد و دستور ALL رکوردهای اصلی ای را برمی گرداند که عمل مقایسه با تمام رکوردهای زیرسؤال، نتیجه True داشته باشد .

اجتماع سؤالها (Union Queries)

این نوع سؤالها بعد از Jet 2.0 شناخته شد و از ترکیب دو سؤال مجزا به وجود می آید. دستور کلی آن به صورت زیر است.

```
{TABLE table_name1|SELECT field_list1 FROM table_list1 }
```

```
Union [ALL]
```

```
{TABLE table_name2|SELECT field_list2 FROM table_list2 }
```

نکته مهم در اجرای این نوع سؤال این است که تعداد فیلدها در table_name1 باید با تعداد فیلدها در table_name2 و یا field-list برابر باشند ، اما یکسان بودن نوع داده های فیلد لزومی ندارند.

سطرهای تکراری تا زمانی که از ALL استفاده نکرده اید نشان داده نمی شود. در مثال زیر تمام

مشتریان و خریدارانی از پایگاه داده Nwind. MDB انتخاب می شوند که در کشور برزیل می باشند .

```
SELECT CompanyName, City, SupplierID AS ID
```

```
FROM Suppliers
```

```
WHERE Country = Brazil , Customer ID As ID
```

```
UNION SELSET CompanyName, City, CustomerID
```

```
FROM Customers
```

```
WHERE Country = "Brazil"
```

```
ORDER BY City
```

تابع های جمعی در SQL و دستورات HAVING , GROUP BY

زبان ANSI SQL دارای توابعی است که به آنها توابع جمعی گفته می شود و روی دسته ای از

رکوردها اعمال می گردد. در SQL-92 این توابع به شرح زیر می باشد:

– **COUNT (field_name)** : تعداد سطرهایی را که دارای مقدار Null نیستند برمی گرداند .

– **MAX (field_name)** : بیشترین مقدار را از field -name بر می گرداند.

– **MIN (field_name)** : کمترین مقدار را در field -name بر می گرداند.

– **SUM (field_name)** : جمع مقادیر field -name را بر می گرداند.

– **AVG (field_name)** : میانگین ریاضی مقادیر field_name را بر می گرداند.

از تابع های جمعی می توان در قسمت انتخاب فیلد سؤال استفاده کرد. به مثال زیر توجه کنید.

```
SELECT Count(*) AS Count,
```

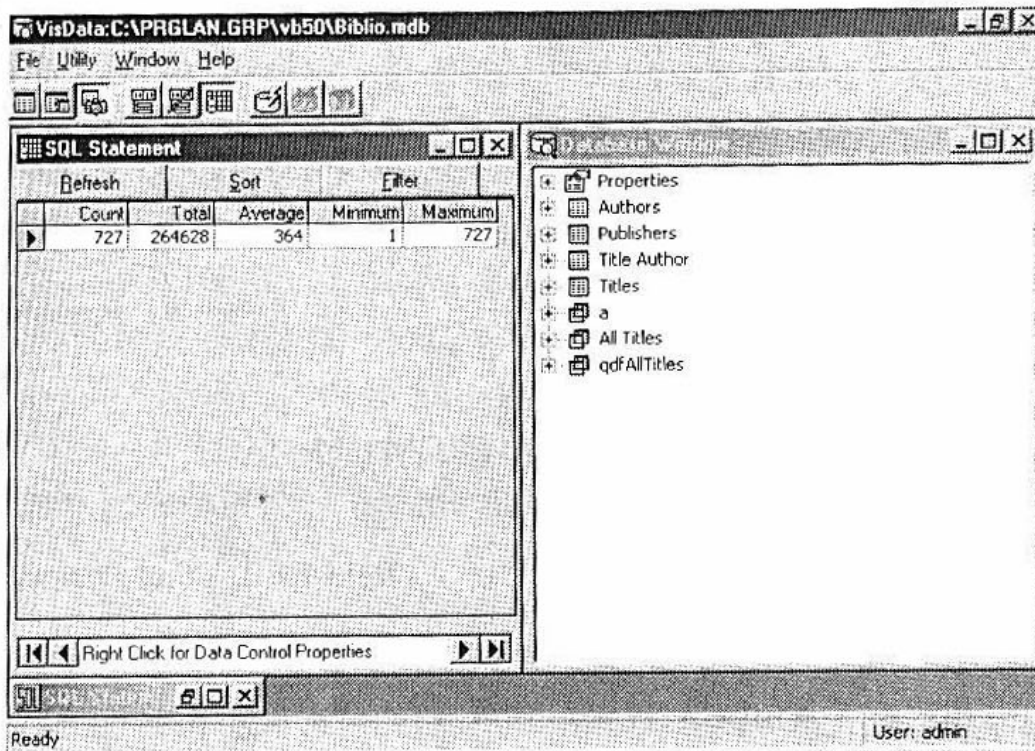
```
Sum(PubID) As Total,
```

```
Avg(PubId) As Average,
```

```
Min(PubId) As Minimum,
```


Max(PubId) As Maximum FROM Publishers

نتیجه اجرای SQL بالا در شکل ۸-۱۹ نشان داده شده است .



شکل ۸-۱۹

نتیجه اجرای SQL

هنگامی که از این توابع استفاده می کنیم ، عملیات محاسبه روی تمام رکوردها صورت می گیرد، اما با استفاده از دستور Group By این عملیات را مجزا انجام می دهیم. هنگامی که روی یک فیلد از دستور Group By استفاده می کنید مقادیر مساوی این فیلد در گروهها ، دسته دسته می شوند و این توابع به طور جداگانه روی این دستهها اعمال می گردد .

برای مثال فرض کنید لیستی از سفارشات جداگانه داریم و نیاز به جدولی است که جمع مبلغ سفارشات را برای هر کالا به طور جداگانه حساب کند. برای این کار روی فیلد کد کالا از Group By استفاده می کنیم و توابع ذکر شده را روی مبلغ سفارشات اعمال می نماییم. کد زیر عملیات مشابهی را روی فیلدهای جدول Products از پایگاه داده Nwind. MDB انجام می دهد .

```
SELECT CategoryID As Catagory,Count(ProductID) As Items,  
Format(Avg(UnitPrice),”$#,###0.00”) As Avg_Up,  
Sum(UnitsInStock) As Sum_Stock,  
Sum(UnitsOnOrder) As Sum_Orderd  
FROM Products  
Group By CategoryID
```

نتیجه اجرای کد فوق در شکل ۸-۲۰ مشخص شده است :

Category	Items	avg_up	Sum_stock	Sum_Ordered
1	12	\$37.98	559	60
2	12	\$23.06	507	170
3	13	\$25.16	366	180
4	10	\$28.73	393	140
5	7	\$20.25	308	90
6	6	\$54.01	165	0
7	5	\$32.37	100	20
8	12	\$20.68	701	120

اگر بخواهید رکوردهای فیلد گروه شده را محدود سازید و روی آن شرط قرار دهید باید از دستور HAVING استفاده کنید. ممکن است تصور نمایید که این شرط را می‌توان توسط Where نیز اعمال کرد، اما دستور WHERE روی جداول ایجاد شده عمل می‌کند و کنترلی روی جداول مجازی ندارد. دستور Having همانند WHERE می‌باشد و برای فیلدهای گروه شده استفاده می‌گردد. مثال زیر لیست مثال قبل را محدود می‌کند و فقط رکوردهایی را انتخاب می‌نماید که فیلد Category ID آن ۱ یا ۲ باشد.

```
SELECT CatagoryID As Catagory,Count(ProductID) As Items,
Format(Avg(UnitPrice),"$#,##0.00") As Avg_Up,
Sum(UnitsInStock) As Sum_Stock,
Sum(UnitsOnOrder) As Sum_Orderd
FROM Products
Group By CategoryID
HAVING CategoryID IN (1,2)
```

Category	Items	avg_up	Sum_stock	Sum_Ordered
1	12	\$37.98	559	60
2	12	\$23.06	507	170

زبان پرس و جوی ساختاری (SQL) پیشرفته

تا اینجا با عمومی‌ترین نوع SQL یعنی سؤال انتخابی آشنا شدید. در این فصل با پنج نوع سؤال دیگر نیز آشنا خواهید شد که شامل سؤالهای سطر و ستون، افزایش، حذف، به روز رسانی و ساخت جدول می‌باشد.

سؤال سطر و ستون یک نوع سؤال انتخابی است که اطلاعات را به طور خلاصه در یک جدول نمایش می‌دهد و برای تهیه گزارشات، نمودارها و آمارها به کار برده می‌شود.

استفاده از سؤال سطر و ستون برای ارائه دادن خلاصه اطلاعات

در گذشته، قبل از زمانی که برنامه‌های کامپیوتری و الکترونیکی صفحه گسترده وجود داشته باشد، برای مدیران بسیار معمول بود که اطلاعاتشان را در قالب جداول محاسباتی به صورت سطر و ستون خلاصه کنند. این جداول معمولاً به دو دسته تقسیم می‌شوند:

الف) جداول زمانی

این جداول به صورتی بودند که تیتروستونهای آن را فاصله‌های زمانی مانند سال، ماه، فصل، روز یا ساعت تشکیل می‌داد و تیتروسطرهای آن، دسته‌های اطلاعاتی که می‌خواستند مقایسه شوند تشکیل داده می‌شد. اطلاعات جزئی‌تر در قالب گروه‌های دوره‌ای و دسته‌ای تقسیم می‌شدند. این جداول معمولاً برای تهیه نمودارهای میله‌ای که در یک دستگاه دو بعدی $Y.X$ رسم می‌شد، مورد استفاده قرار می‌گرفت.

ب) جداول طبقه‌بندی

در این جدولها تیتروستونها از نامهای افراد، محلها، سازمانها و غیره تشکیل می‌شود و دسته‌های داده‌ها در سطرها قرار می‌گیرد. این جدولها در دوره‌های زمانی از پیش تعیین شده و منفرد می‌باشد. معمولاً جداولی که به این ترتیب تشکیل می‌شوند، به صورت نمودار کیکی نمایش داده می‌شوند. امروزه برنامه‌های صفحه گسترده مانند لوتوس ۱-۲-۳ و میکروسافت اکسل با عملیات خودکار و پیشرفته، جایگزین جداول پیشین شده‌اند. چون اکثر اطلاعات جزئی برای مدیران نیاز به خلاصه شدن دارد، بنابراین، این امر صورت نمی‌گیرد، مگر اینکه داده‌ها به صورت فیلد در جدولهای مرتبط پایگاه داده محاسبه شوند و سپس مورد استفاده قرار گیرد. در ابتدای این فصل طریقه خلاصه کردن اطلاعات و تبدیل آن به جداول سطر و ستون را در SQL فرا می‌گیریم.

ایجاد خلاصه اطلاعات توسط دستور GROUPBY

شکل ۹-۶ نتیجه اجرای یک سؤال انتخابی را نشان می‌دهد که توسط تابع جمعی (Sum) اطلاعات را برای ساختن یک جدول با ساختار زمانی آماده می‌سازد. جداول مورد استفاده در این سؤال در پنجره SQL در طرف راست شکل مشخص شده و شامل جداول , Products , Order Details , Categories و Orders می‌باشد که از پایگاه داده Nwind انتخاب شده‌اند. در جدول Categories دسته‌های کالا نگهداری می‌شود.

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees,
2	Condiments	Sweet and savory sa
3	Confections	Desserts, candies, a
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pa
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean
8	Seafood	Seaweed and fish

شکل ۹-۱

اطلاعات جدول Categories

در جدول Orders اطلاعات کلی یک سفارش نگهداری می‌شود. از جمله فیلدهای این جدول ، کد سفارش ، کد خریدار ، کد فروشنده و تاریخ سفارش می‌باشد.

OrderID	CustomerID	EmployeeID	OrderDate	Freight
10330	LILAS	3	11/16/94	12.75
10331	BONAP	9	11/16/94	10.19
10332	MEREP	3	11/17/94	52.84
10333	WARTH	5	11/18/94	0.59
10334	VICTE	8	11/21/94	8.56
10335	HUNGO	7	11/22/94	42.11
10336	PRINI	7	11/23/94	15.51
10337	FRANK	4	11/24/94	108.26
10338	OLDWO	4	11/25/94	84.21
10339	MEREP	2	11/28/94	15.66
10340	BONAP	1	11/29/94	166.31
10341	SIMOB	7	11/29/94	26.78
10342	FRANK	4	11/30/94	54.83
10343	LEHMS	4	12/1/94	110.37
10344	WHITC	4	12/2/94	23.29

شکل ۹-۲

اطلاعات جدول Orders

در جدول Order Details اطلاعات جزئی هر سفارش نگهداری می‌شود و با فیلد کد سفارش با جدول Order می‌تواند در ارتباط باشد؛ بنابراین در این جدول کد سفارش ، دارای چند رکورد است که اقلام جزئی را در سفارش نگهداری می‌کند. از جمله فیلدهای این جدول کد سفارش، کد کالا ، قیمت واحد ، تعداد کالای سفارش شده می‌باشد.

شکل ۳-۹.

اطلاعات جدول Order Details.

OrderID	ProductID	UnitPrice	Quantity	Discount
10248	11	14	12	0
10248	42	9.8	10	0
10248	72	34.8	5	0
10249	14	18.6	9	0
10249	51	42.4	40	0
10250	41	7.7	10	0
10250	51	42.4	35	0.15
10250	65	16.8	15	0.15
10251	22	16.8	6	0.05
10251	57	15.6	15	0.05
10251	65	16.8	20	0
10252	20	64.8	40	0.05
10252	33	2	25	0.05
10252	60	27.2	40	0
10253	31	10	20	0
10253	39	14.4	42	0

در جدول Product نیز اطلاعات کالا نگهداری می‌شود. از جمله فیلدهای این جدول کد کالا، نام کالا، قیمت واحد و گروه کالا می‌باشد.

شکل ۴-۹.

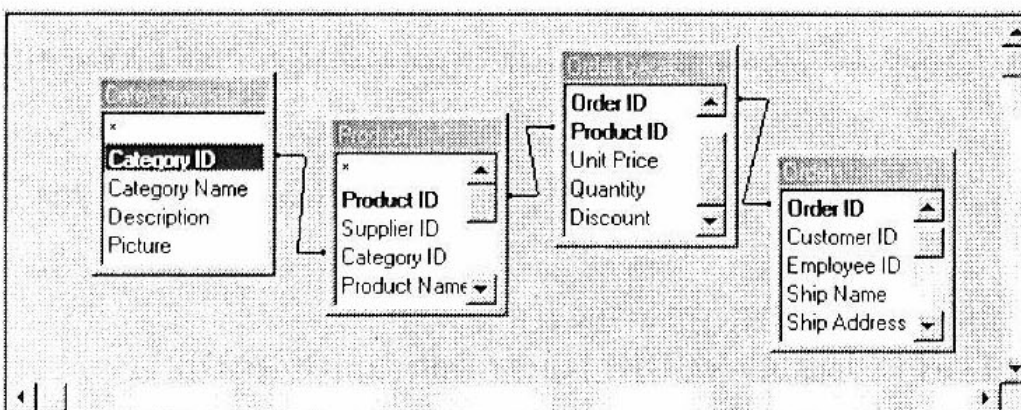
اطلاعات جدول Product.

ProductID	ProductName	CategoryID	QuantityPerUnit
1	Chai	1	10 boxes x 20 bags
2	Chang	1	24 - 12 oz bottles
3	Aniseed Syrup	2	12 - 550 ml bottles
4	Chef Anton's Cajun	2	48 - 6 oz jars
5	Chef Anton's Gummy	2	36 boxes
6	Grandma's Boysen	2	12 - 8 oz jars
7	Uncle Bob's Organic	7	12 - 1 lb pkgs.
8	Northwoods Cranberry	2	12 - 12 oz jars
9	Mishi Kobe Niku	6	18 - 500 g pkgs.
10	Ikura	8	12 - 200 ml jars
11	Queso Cabrales	4	1 kg pkg.
12	Queso Manchego Large	4	10 - 500 g pkgs.
13	Konbu	8	2 kg box
14	Tofu	7	40 - 100 g pkgs.
15	Genen Shouyu	2	24 - 250 ml bottles

رابطه بین این جداول در شکل ۵-۹ مشخص شده است.

شکل ۵-۹.

روابط بین جداول.



نتیجه اجرای سؤال ، دادن لیست داده‌های است که سفارشات را از کالاهای Categories بین یک بازه زمانی بر می گرداند.

The screenshot shows a window titled 'VisData: C:\PRGLAN.GRP\vb50\Nwind.mdb'. It contains two panes. The left pane, 'SQL Statement', displays a table with columns 'Cat', 'Period', and 'Sales'. The right pane shows the SQL query used to generate this data.

Cat	Period	Sales
1	1994Q3	\$8,474.40
1	1994Q4	\$33,636.80
2	1994Q3	\$4,511.60
2	1994Q4	\$13,359.50
3	1994Q3	\$11,485.80
3	1994Q4	\$13,475.40
4	1994Q3	\$12,304.00
4	1994Q4	\$20,146.40
5	1994Q3	\$1,544.20
5	1994Q4	\$5,920.20
6	1994Q3	\$6,529.40
6	1994Q4	\$14,339.80
7	1994Q3	\$4,976.20
7	1994Q4	\$6,476.20
8	1994Q3	\$6,975.90
8	1994Q4	\$11,189.30

شکل ۶-۹.

نتیجه اجرای سؤال.

در لیست زیر تک تک قسمت‌های کد SQL اجرا شده در شکل ۶-۹ را توضیح خواهیم داد :

- در قسمت ابتدایی کد از دستور AS برای تخصیص نام فیلد سؤال استفاده شده است که فیلد Categories را با نام Cat و دو فیلد بعد را به ترتیب با نام Priod و Sales در لیست نمایش می‌دهد که توسط تابع Format قالبی برای نمایش این فیلدها تعیین شده است.

```
SELECT Categories.[CategoryID] AS Cat,
       Format([Order Date], "yyyy\Qq") As Priod, Format([Order Detlis].[Unit Price]*
       [Order Datalis].[Quantity], "$# , ##0,00 ) As Sales
```

- اولین تابع Format استفاده شده در کد، برای نشان دادن فیلد Order Date به صورتی است که ابتدا سال (yyyy) در فیلد قرار بگیرد و بعد از آن کاراکتر " Q (Q) و سپس شماره فصل (q) در فیلد نشان داده شود.

- تابع جمع (Sum () در کد مجموع حاصل ضرب فیلد تعداد کالا را در فیلد قیمت آن از جدول اطلاعات جزئی سفارشات محاسبه می‌کند.

- تابع (Format () در فیلد سوم ، قیمت محاسبه شده توسط تابع (Sum () را در قالب واحد پول و مبلغ در می‌آورد.

- در قسمت بعد جدولها و رابطه بین آنها تعیین می‌گردد که این ارتباط در شکل ۵-۹ مشخص شده است :

```
FROM Categoris INNER JOIN (Products INNER JOIN (Orders
INNER JOIN [Order Details] ON Orders.[OrderID] = [Order Details].[OrderID])
```

ON Products.[ProductID] = [Order Details].[ProductID])

ON Categories.[CategoryID] = Products.[CategoryID]

• در قسمت بعد، شرط WHERE مجموع قیمت‌های فروش را در سال 1994 محدود می‌کند.

WHERE Orders.[Order Date] Between #1/1/94# And #12/31/94 #

• دستور

GROUPBY Categories.[Categorie ID],Format([Order Date],“yyyy\Qq”)

دو فیلد Order Dete , Categorie ID را گروه‌بندی می‌کند و گروه جدول تشکیل شده یک کاربرگ

از نوع زمانی است که اطلاعات به طور خلاصه‌شده در دوره‌های زمانی مشخص شده است.

استفاده از دستورات PIVOT و TRANSFORM جهت ایجاد یک سؤال سطر و ستون

خصوصیت سؤال‌های سطر و ستون در Jet SQL مشکلات ترتیب مجدد سلول‌های یک کاربرگ را حل

کرده است. دو دستور اصلی PIVOT و TRANSFORM در SQL این عملیات را روی داده‌ها انجام

می‌دهند. اجزای تشکیل‌دهنده یک سؤال سطر و ستون به شرح زیر می‌باشد:

- دستور TRANSFORM مقادیر سلول‌های داده را مشخص می‌کند و حتماً باید از توابع جمعی

نظیر Sum() یا Avg () برای مقدار دادن استفاده کرد. برای تعیین ظاهر داده‌ها در این قسمت

می‌توانید از تابع Format () نیز استفاده کنید.

- در قسمت بعد دستور SELECT را داریم که تیترا سطرها را طراحی می‌کند. فیلدی را که شما

برای این قسمت انتخاب می‌کنید حتماً باید با فیلد گروه‌بندی‌شده یا GROUP BY یکسان باشد.

- در قسمت بعد، از دستور FROM برای تعیین جداول مورد استفاده در سؤال استفاده می‌کنیم و

اگر بیش از یک جدول بود رابطه آنها را با INNER JOIN مشخص می‌کنیم و توجه داشته باشید که از

() برای جدا کردن جدولها استفاده نکنید و حتماً رابطه داخلی را بین آنها تشکیل دهید.

- دستور GROUP BY داده‌های سطرها را گروه‌بندی می‌کند. در سؤالهایی که از نوع سطر و ستون

می‌باشد فقط یک فیلد باید در این قسمت گروه‌بندی شود.

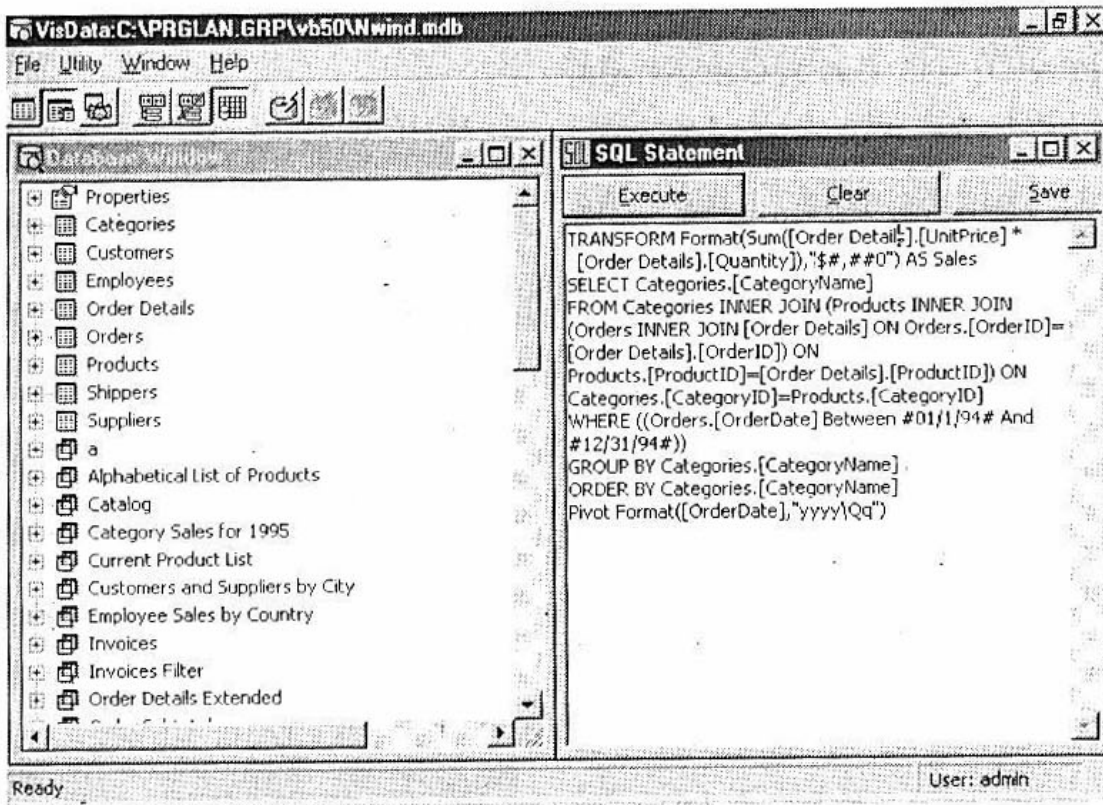
- دستور دلخواه ORDER BY داده‌های یک یا چند فیلد را مرتب می‌کند.

- دستور PIVOT عملیات گروه‌بندی ستون‌ها و تعیین نام تیترا آنها را انجام می‌دهد. عملیات

گروه‌بندی همانند Group By می‌باشد ، با این تفاوت که این گروه‌بندی روی ستونها انجام می‌گیرد.

شکل ۷-۹ یک نمونه سؤال از نوع سطر و ستون را در پنجره کد SQL نشان می‌دهد که نتیجه اجرای

آن در شکل ۸-۹ مشخص شده است.



شکل ۷-۹.

نمونه ای از یک سؤال
به صورت سطر و ستون.

Refresh	Sort	Filter
CategoryName	1994Q3	1994Q4
Beverages	\$8,474	\$33,637
Condiments	\$4,512	\$13,360
Confections	\$11,486	\$13,475
Dairy Products	\$12,304	\$20,146
Grains/Cereals	\$1,544	\$5,920
Meat/Poultry	\$6,529	\$14,340
Produce	\$4,976	\$6,476
Seafood	\$6,976	\$11,189

شکل ۸-۹.

نتیجه اجرای سؤال شکل ۸-۹.

دستورات INNER JOIN به صورت تودرتو همانند آنچه در شکل ۷-۹ دیده می شود، معمولاً مشکل می باشد. برای آسانتر کردن آن می توان از دستور WHERE استفاده کرد. کد مثال بالا با دستور WHERE به صورت زیر می باشد:

```
TRANSFORM Format(Sum([Order Details].[UnitPrice] *
[Order Details].[Quantity]), '$#,##0') AS Sales
SELECT Categories.[CategoryName]
FROM Categories,[Order Details],Orders,Products
WHERE Products.[CategoryID] = Categories.[CategoryID]
AND [Order Details].[OrderID] = Orders.[OrderID]
AND [Order Details].[ProductID] = Products.[ProductID]
```


AND Orders.[OrderDate] Between #01/1/94# And #12/31/94#
 GROUP BY Categories.[CategoryName]
 ORDER BY Categories.[

ساختن سؤالیهای سطر و ستون به صورت طبقه‌بندی شده

با یک تغییر ساده و اضافه کردن یک مورد به سؤال قبل می‌توانیم یک سؤال سطر و ستون ایجاد کنیم که در قالب طبقه‌بندی می‌باشد و مبالغ فروش در دسته‌های کالا را برای هر فروشنده درباره زمانی نشان می‌دهد.

در کد زیر جدول Employees شامل اطلاعات فروشنده را به مثال قبل اضافه می‌کنیم و نام خانوادگی هر فروشنده را در تیتراستونها قرار می‌دهیم تا مبلغ فروش هر یک از فروشندگان در دسته‌های کالا مشخص گردد.

```

TRANSFORM Format(Sum([Order Details].[UnitPrice] *
  [Order Details].[Quantity]), "$#,### ") AS Sales
SELECT Categories.[CategoryName]
FROM Categories,[Order Details],Orders,Products,Employees
WHERE Products.[CategoryID] = Categories.[CategoryID]
AND [Order Details].[OrderID] = Orders.[OrderID]
AND [Order Details].[ProductID] = Products.[ProductID]
AND Orders.[OrderDate] Between #01/1/94# And #12/31/94#
GROUP BY Categories.[CategoryName]
ORDER BY Categories.[
Pivot Employees.[LastName]
  
```

شکل ۹-۹ نتیجه اجرای کد بالا را نشان می‌دهد.

The screenshot shows a PivotTable in VisData software. The table has 10 columns representing categories and 10 rows representing employees. The data is summarized as follows:

CategoryName	Buchanan	Callahan	Davolio	Dodsworth	Fuller	King	Leverling	Peacock	Suyama
Beverages	\$42,111	\$42,111	\$42,111	\$42,111	\$42,111	\$42,111	\$42,111	\$42,111	\$42,111
Condiments	\$17,871	\$17,871	\$17,871	\$17,871	\$17,871	\$17,871	\$17,871	\$17,871	\$17,871
Confections	\$24,961	\$24,961	\$24,961	\$24,961	\$24,961	\$24,961	\$24,961	\$24,961	\$24,961
Dairy Products	\$32,450	\$32,450	\$32,450	\$32,450	\$32,450	\$32,450	\$32,450	\$32,450	\$32,450
Grains/Cereals	\$7,464	\$7,464	\$7,464	\$7,464	\$7,464	\$7,464	\$7,464	\$7,464	\$7,464
Meat/Poultry	\$20,869	\$20,869	\$20,869	\$20,869	\$20,869	\$20,869	\$20,869	\$20,869	\$20,869
Produce	\$11,452	\$11,452	\$11,452	\$11,452	\$11,452	\$11,452	\$11,452	\$11,452	\$11,452
Seafood	\$18,165	\$18,165	\$18,165	\$18,165	\$18,165	\$18,165	\$18,165	\$18,165	\$18,165

شکل ۹-۹

طبقه‌بندی سؤالیهای
 سطر و ستون.

سؤالهای اجرایی

سؤالهای اجرایی ابزاری هستند که با استفاده از آنها می توان اشیای پایگاه داده را ایجاد و یا اصلاح نمود. این سؤالها برای نیازهای حذف ، اضافه یا تغییر رکوردها در حجم زیاد بسیار مؤثر می باشند این اعمال را می توان از این طریق با دستورات ساده ای انجام داد. لیست زیر به طور مختصر انواع سؤالهای اجرایی را توضیح می دهد :

Append Query : با استفاده از دستور `INSERT INTO Table-Name` که پس از آن از دستور `SELECT` استفاده می شود، رکوردهای انتخاب شده را به جدول `Table-Name` اضافه می کند.

DELETE Query : با استفاده از دستور `DELETE FROM table-name WHERE {Criteria}` همه رکوردهای انتخاب شده محدود شده توسط دستور حذف می شود.

Update Query : با استفاده از دستور `UPDATE table-name SET field-name = {عبارت}` اطلاعات فیلد را تغییر می دهد.

Make-table Query : با استفاده از دستور

`SELECT field-names INTO dest-Table FROM Source-table` یک جدول جدید با فیلدهای مشخص شده در `field - names` ، ساخته می شود.

در ادامه فصل با مثالهای این سؤالها آشنا خواهیم شد.

اجرای یک سؤال اجرایی توسط ویژوال بیسیک

همان طور که گفتیم سؤالهای اجرایی گروهی از رکوردها را انتخاب نمی کنند، بنابراین نتیجه آن به صورت یک لیست داده (Recordset) نخواهد بود. پس برای اجرای یک سؤال اجرایی از دستور زیر استفاده می کنیم :

```
dbName.Execute {qdfName|StrSQL}
```

در قسمت بعد از دستور `Execute` شما می توانید به دو صورت عمل کنید ، یکی آنکه یک سؤال اجرایی را که در قالب شیء `QueryDef` می باشد اجرا کنید و یا اینکه مستقیماً کد `SQL` خود را پس از دستور `Execute` وارد نمایید که نتیجه آن اجرای کد خواهد بود.

شما می توانید از طریق خصوصیت `Records Affected` تعداد رکوردهای عمل شده توسط سؤال را به دست آورید.

توجه : در صورتی که از دستور `Execute` برای سؤالهای غیر اجرایی استفاده کنید ، پیغام خطا ظاهر خواهد شد.

هریک از انواع سؤالهای اجرایی دارای کد نظیری در ویژوال بیسیک می باشند. جدول ۹-۱ لیست انواع سؤالهای اجرایی را همراه با کدهای نظیر آن در SQL و ویژوال بیسیک نشان می دهد. لیست داده (Recordset) از نوع Dynaset و یا Table است که قابلیت تغییر اطلاعات را داراست (Updatable).

جدول ۹-۱: کاربرد اوپراتور Like در SQL.

کد ویژوال بیسیک	دستور Jet SQL	سؤال اجرایی
rsName.Update	INSERT INTO	Append
rsName.DELETE	DELET...FROM	Delete
rsName.Edit rsName.Update	Update... Set	Update
Dim tdfNewtable As New TableDef Dim fldNewField As New Field tdfNewTable.Append fldNewField	SELECT... INTO	Make-Table

استفاده از سؤالهای اجرایی

در این قسمت از انواع سؤالهای اجرایی مثالی را می آوریم که همگی در برنامه Vis Data روی پایگاه داده Nwind.mdb اجرا می شود.

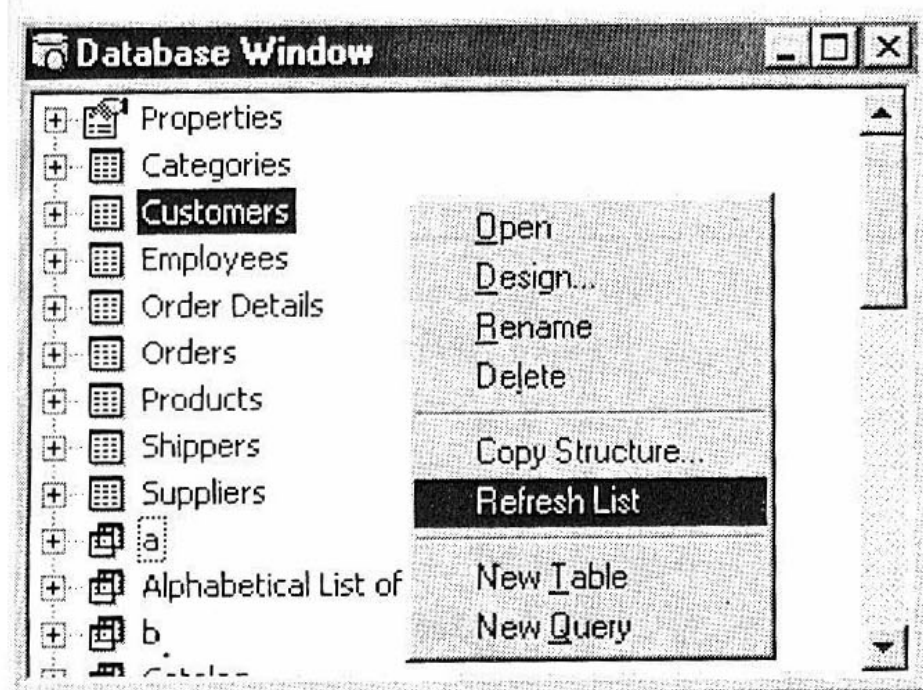
ساختن یک جدول جدید با استفاده از Make-Table Query

در مثال زیر جدولی به نام tbl Orders ساخته می شود که فیلدهای آن همان فیلدهای جدول Orders می باشد و رکوردهای آن از تاریخ 7/1/94 به قبل می باشد. این کد را در پنجره SQL Statment در برنامه Visdata وارد نمایید.

```
SELECT * [OrderDate]
      INTO tblOrders
      FROM Orders
      WHERE [OrderDate] < #7/1/94#
```

وجود فیلد [Order Date] پس از دستور SELECT الزامی است زیرا شما می خواهید شرطی را روی این فیلد اعمال نمایید.

توجه: جدول ایجاد شده توسط سؤال اجرایی در لیست پنجره Database ظاهر نمی‌شود تا زمانی که دکمه راست ماوس داره را روی این پنجره کلیک کنید و از لیست گزینه Refresh List را انتخاب کنید. آنگاه این جدول در لیست ظاهر می‌شود.



شکل ۹-۱۰

به روز رسانی لیست Database Window

دستور کلی سؤال از نوع ساخت جدول به صورت زیر است :

```
SELECT [ALL|DISTINCT|DISTINCTROW] select_list INTO testTable [IN
databasename[Connect_Str]
FROM Source_Table
WHERE {Criteria}
```

شما می‌توانید جداول خود را در یک پایگاه دیگر نیز ذخیره کنید در مثال زیر این عمل صورت گرفته است.

```
SELECT *
INTO tblOrders
IN C:\msoffice\access\sample\Nwind.mdb
FROM Orders
```

اضافه کردن رکورد به جدول

شما می‌توانید رکوردهایی را از جدول Orders که OrderDate آن در سال 1994 می‌باشد به جدولی که در قسمت قبل ساختید به وسیله کد زیر اضافه کنید :

```
INSERT INTO tblorders ([OrderID],[CustomerID],[EmployeeID],
[OrderDate],[ ShippedDate])
SELECT [OrderID],[CustomerID],[Employeeed],[OrderDate],[Shipped Date]
FROM Orders
WHERE [ORDERDate] Between #7/1/64 #AND #12/31/94#
```

پس از اجرای این سؤال خواهید دید که رکوردهای محدود شده توسط سؤال در جدول tbl orders اضافه می‌گردد.

OrderDate	OrderID	CustomerID	EmployeeID
11/16/94	10330	LILAS	3
11/16/94	10331	BONAP	9
11/17/94	10332	MEREP	3
11/18/94	10333	WARTH	5
11/21/94	10334	VICTE	8
11/22/94	10335	HUNGO	7
11/23/94	10336	PRINI	7
11/24/94	10337	FRANK	4
11/25/94	10338	OLDWO	4
11/28/94	10339	MEREP	2
11/29/94	10340	BONAP	1
11/29/94	10341	SIMOB	7
11/30/94	10342	FRANK	4
12/1/94	10343	LEHMS	4
12/2/94	10344	WHITC	4

شکل ۱۱-۹.

حذف رکوردها با شرط مخصوص در SQL.

در این نوع سؤال حتما باید نام فیلدها در هر دو قسمت INSERT و SELECT قید شود و نام فیلدها با یکدیگر برابر باشند.

تغییر مقادیر اطلاعات در جدولها

برای تغییر داده‌ها در فیلد از دستور UPDATE استفاده می‌شود. SET باعث جایگزین شدن عبارت جدید در فیلد می‌گردد. مثال زیر به تمام مقادیر فیلد [Shipped Date] که در تاریخ آپریل 1994 می‌باشد دو واحد اضافه می‌کند.

```
UPDATE tblOrders
SET [ShippedDate] = [ShippedDate] + 2
WHERE [OrderDate] Between #4/1/94# And #4/30/94#
```

در صورتی که بخواهید تعداد بیشتری فیلد را مقداردهی کنید از (،) برای جدا کردن آنها در قسمت Set استفاده نمایید.

حذف رکوردها از داخل جدول

آسانترین دستور در سؤالهای اجرایی دستور حذف رکوردها می‌باشد در مثال زیر رکوردهایی که فیلد [Order Date] آن بین یک بازه زمانی می‌باشد حذف می‌گردد:

```
DELETE FROM TblOrders
WHERE [OrderDate] Between #7/1/94# AND #12/31/94#
```


فصل دهم

استفاده از کنترل‌های پیشرفته آگاه از داده‌ها

در فصل هفتم با طریقه استفاده از Data Control آشنا شدید و نیز دیدید که چگونه می‌توان از جعبه متن به عنوان ابزار مقیدسازی داده‌ها استفاده نمود. نسخه استاندارد ویژوال بیسیک ۳ شامل ابزارهایی بود که عمل مقیدسازی اطلاعات توسط آنها صورت می‌گرفت و شامل کنترل‌های از قبیل Image , Label , Picture Box , Text Box و Check Box بود. در نسخه حرفه‌ای آن نیز کنترل‌های به آن اضافه شد که شامل : Masked Box , 3Dsscheck , Sspanel بودند و همگی از نوع کنترل‌های VBX بودند. VBX ها این قابلیت‌ها را داشتند که به برنامه‌ها اضافه شوند، مثلاً اگر شما به ابزارهای بیشتری برای مقیدسازی داده‌ها احتیاج داشتید، می‌توانستید آنها را از شرکت‌های مختلفی که این کنترل‌ها را تولید می‌کردند، تهیه کنید. نسخه حرفه‌ای و جامع ویژوال بیسیک ۴ شامل مجموعه کاملی از ابزارهای پیشرفته مقیدسازی بود که به صورت کنترل‌های سی و دوبیتی و شانزده بیتی از نوع OCX عرضه شدند که کنترل‌هایی از قبیل DBGrid , DBCombo , DB شامل این دسته بودند و در نسخه‌های بعدی نیز به تعداد این ابزارها افزوده شد.

در این فصل کنترل‌های DB , DBCombo , DBList را در برنامه‌ها بررسی می‌کنیم.

نمایش و به روز رسانی شیء لیست داده توسط کنترل DBGrid

ظاهر گرافیکی کنترل DBGrid (DBGrid16.OCX , DBGrid 32.OCX) بسیار شبیه به لیست‌های اطلاعاتی جداول زیر فرم در اکسس ۹۵ می‌باشد و استفاده از این کنترل در برنامه‌های کاربردی بسیار مفید می‌باشد، شکل ۱-۱۰ کنترل DB را نشان می‌دهد که به یک Data Control مقید شده است و اطلاعات جدول Customers را از پایگاه داده Nwind. mdb (که از مثال‌های اکسس ۹۵ می‌باشد) نشان می‌دهد.

شکل ۲-۱۰ یک لیست اطلاعات جدول را که به صورت زیر فرم در اکسس ۹۵ طراحی شده را نشان می‌دهد که به جدول Customers مقید شده است.

شکل ۱-۱۰.

CustomerID	CompanyName	ContactName	ContactTitle	Address
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Ob...
ANATR	Ana Trujillo Empared	Ana Trujillo	Owner	Avd...
ANTON	Antonio Moreno Taq	Antonio Moreno	Owner	Ma...
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120
BERGS	Berglunds snabbk,p	Christina Berglund	Order Administrator	Ber...
BLAUS	Blauer See Delikates	Hanna Moos	Sales Representative	For...
BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manager	24,
BOLID	Bólido Comidas prepa	Martín Sommer	Owner	C/A
BONAP	Bon app'	Laurence Lebihan	Owner	12,
BOTTM	Bottom-Dollar Market	Elizabeth Lincoln	Accounting Manager	23
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fau...
CACTU	Cactus Comidas para	Patricio Simpson	Sales Agent	Cer...
CENTC	Centro comercial Mo	Francisco Chang	Marketing Manager	Sie...
CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hau...

Record: 13 of 91

اطلاعات جدول Customers در کنترل DB Grid

شکل ۲-۱۰.

Custom	CompanyName	ContactName	ContactTitle	Address
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Represent: Obe	
ANATR	Ana Trujillo Emparedac	Ana Trujillo	Owner	Avd:
ANTON	Antonio Moreno Taquer	Antonio Moreno	Owner	Mat:
AROUT	Around the Horn	Thomas Hardy	Sales Represent: 120	
BERGS	Berglunds snabbköp	Christina Berglund	Order Administra: Berg	
BLAUS	Blauer See Delikatess	Hanna Moos	Sales Represent: Fore	
BLONP	Blondel père et fils	Frédérique Citeaux	Marketing Manag	24, 1
BOLID	Bólido Comidas prepar	Martín Sommer	Owner	C/A
BONAP	Bon app'	Laurence Lebihan	Owner	12, 1
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manag	23 T
BSBEV	B's Beverages	Victoria Ashworth	Sales Represent: Fau	
CACTU	Cactus Comidas para l	Patricio Simpson	Sales Agent	Cerr
CENTC	Centro comercial Moct	Francisco Chang	Marketing Manag	Sier
CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hau

Record: 14 of 91

اطلاعات جدول Customers در زیر فرم Access

برای اضافه کردن تعداد کل رکوردهای DataControl در یک Label و نشان دادن شماره رکورد در تیترا DataControl کافی است کد زیر را به برنامه اضافه کنید :

```

DIM fLoaded As Boolean
Private Sub dtcCustomers_Reposition()
    If fLoaded Then
        dtcCustomers.Caption = dtcCustomers.Recordset.AbsolutePosition
    End If
End Sub

```

```
Private Sub Form_Activate()
```

```
    dtcCustomers.Recordset.MoveLast
```

```
    lblRecCount.Caption = "Of" & dtcCustomers.Recordset.RecordCount
```

```
    dtcCustomers.Caption = 1
```

```
    dtcCustomers.Recordset.MoveFirst
```

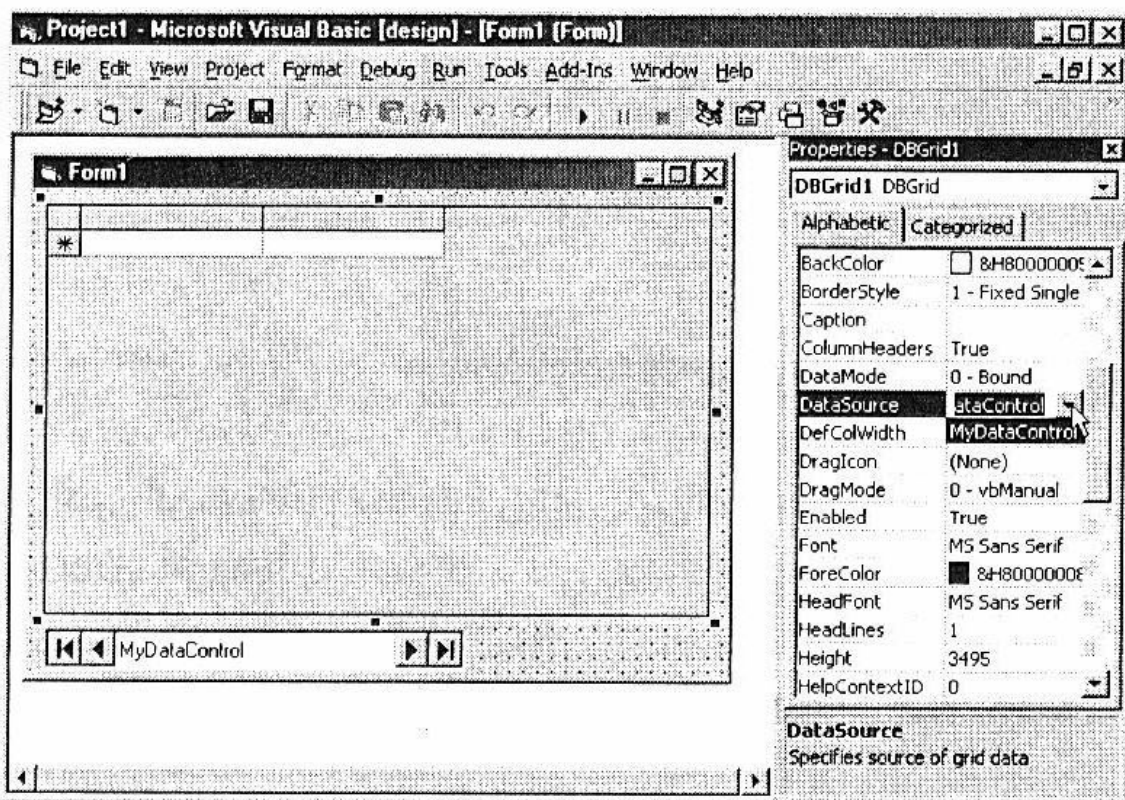
```
    fLoaded = True
```

```
End Sub
```

برای اینکه یک کنترل DBGrid را به DataControl مقید کنید باید در خصوصیت

DataSource آن نام DataControl (نامی که در مشخصه Name از Data Control وارد کرده‌اید) را

مشخص کنید.



شکل ۳-۱۰.

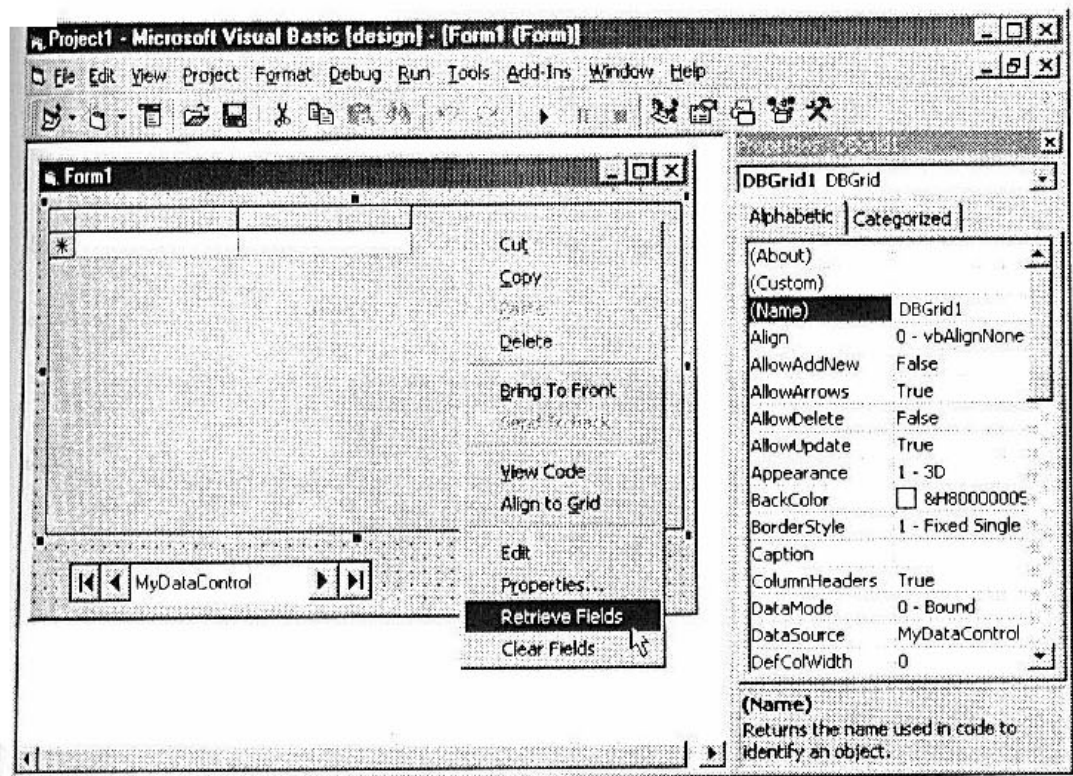
در تمام کنترل‌های مقیدسازی، مشخصه DataSource در هنگام اجرای برنامه فقط خواندنی می‌باشد. بنابراین اگر بخواهیم منبع اطلاعات این کنترلها را تغییر دهیم، باید اطلاعات ابزار کنترل داده (DataControl) را توسط خصوصیت‌های RecordSource و یا در صورت لزوم خصوصیت Database Name را تغییر دهیم. همانند آن چیزی که در اکسس داریم، لیستهای داده از نوع Updateable به شما اجازه اضافه کردن و یا حذف رکوردها را به وسیله DBGrid می‌دهند. در ادامه استفاده از کنترل DBGrid را در یک برنامه ویژوال بیسیک توضیح خواهیم داد.

استفاده از DBGrid

هنگامی که یک کنترل DBGrid را به فرم، اضافه می‌کنیم، در حالت پیش فرض یک جدول شامل دو سطر و دو ستون ایجاد می‌شود و پس از اینکه خصوصیت DataSource آن را مقدار دهی کردیم، هیچ تغییری در ساختار این جدول به وجود نمی‌آید.

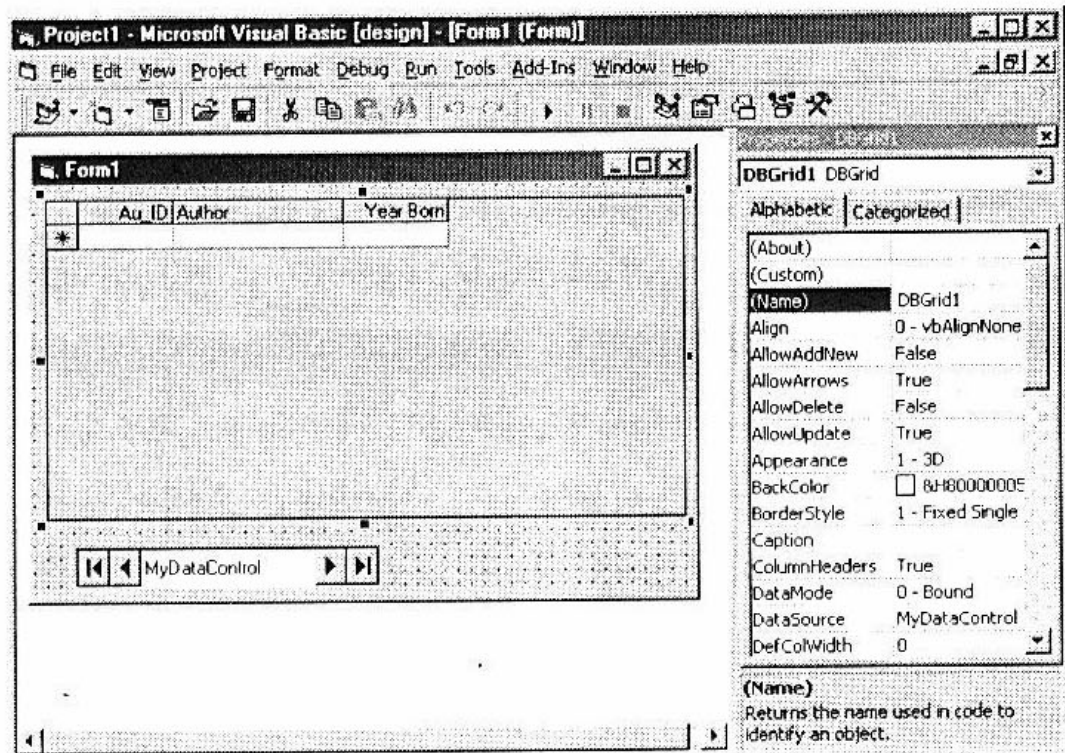
برای تعیین ساختار برای DBGrid باید روی این کنترل کلید سمت راست ماوس را کلیک کنیم تا منوئی مانند شکل ۴-۱۰ باز گردد.

شکل ۴-۱۰

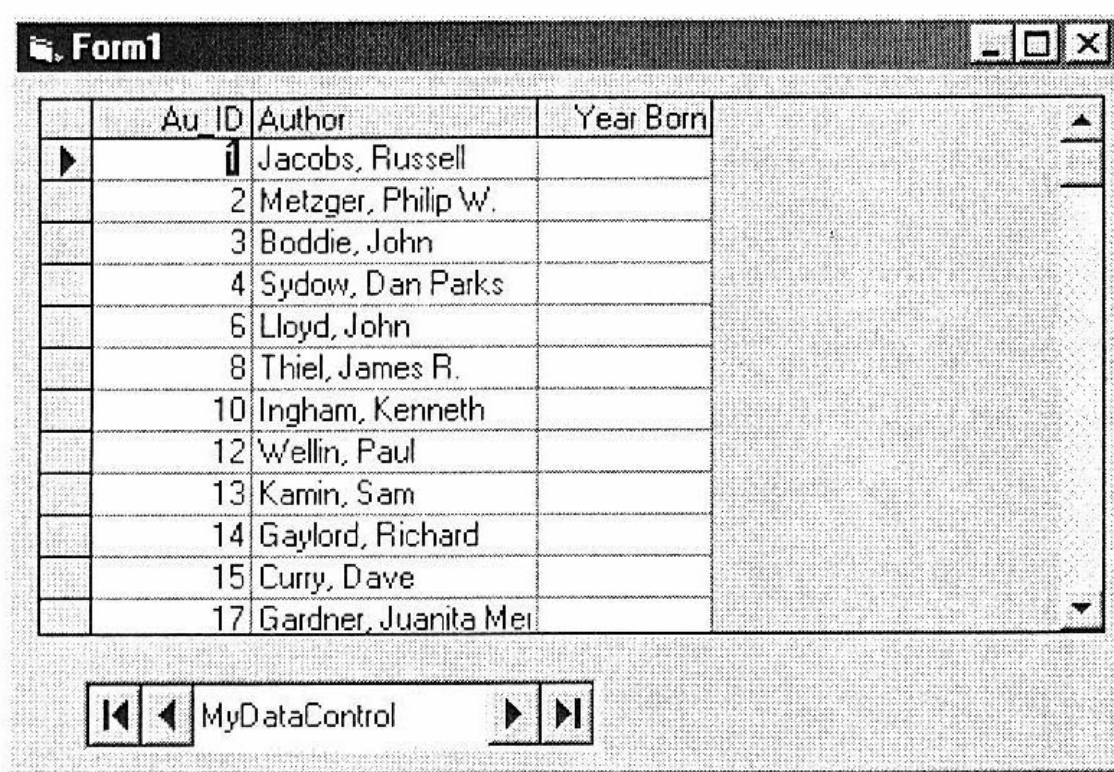


اکنون گزینه Retrieve Fields را انتخاب می کنیم. همان طور که در شکل ۵-۱۰ می بینید جدول DBGrid ساختار جدولی را می گیرد که در ابزار کنترل داده تعیین شده است و در تیت هر ستون نام فیلد جدول قرار گرفته است.

شکل ۵-۱۰



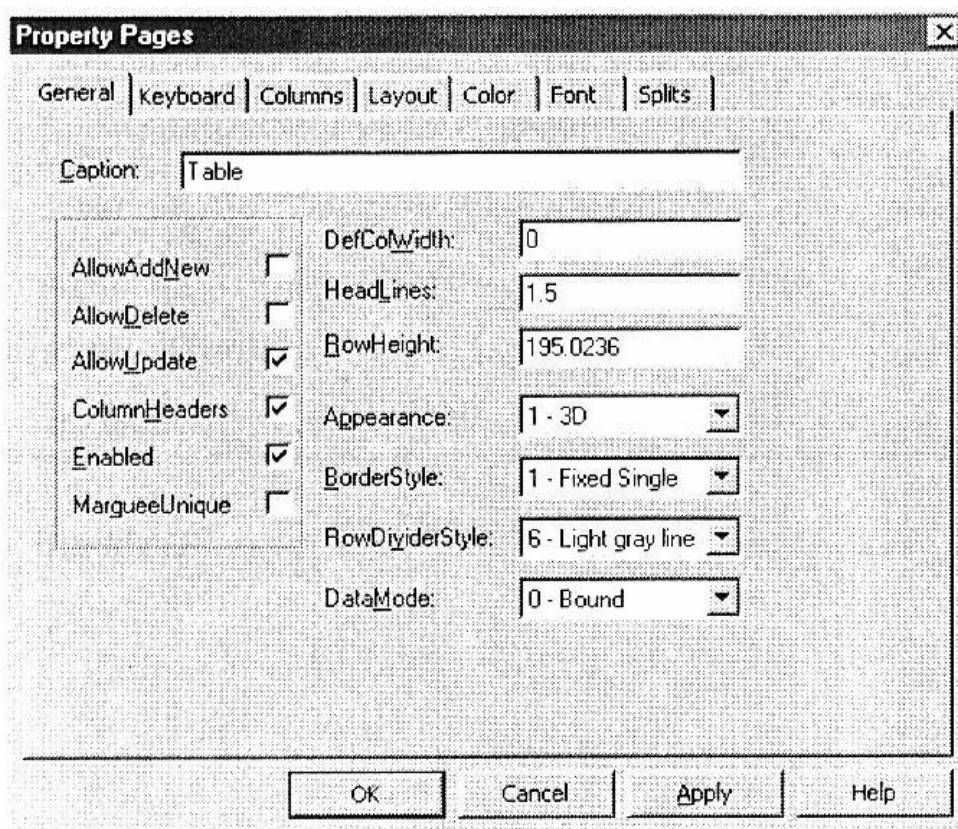
هنگامی که برنامه را اجرا می کنید، لیست داده در DBGrid نشان داده می شود.



شکل ۶-۱۰.

در صورتی که بخواهید تنظیمات بیشتری را نظیر ساختار ستونها تغییر فونت داده‌ها و غیره انجام دهید در پنجره خصوصیتها گزینه (Custom) را انتخاب کنید و یا می‌توانید با انتخاب دکمه سمت راست ماوس روی این ابزار کلیک نمایید و از منوی باز شده، گزینه Properties را انتخاب کنید. پنجره Page باز می‌شود که از قسمت‌های زیر تشکیل شده است:

• **General**: در این صفحه همان طور که در شکل ۷-۱۰ می‌بینید می‌توانید موارد کلی این کنترل را تنظیم کنید. جدول ۱-۱۰ قسمت‌های مختلف این قسمت را نشان می‌دهد.

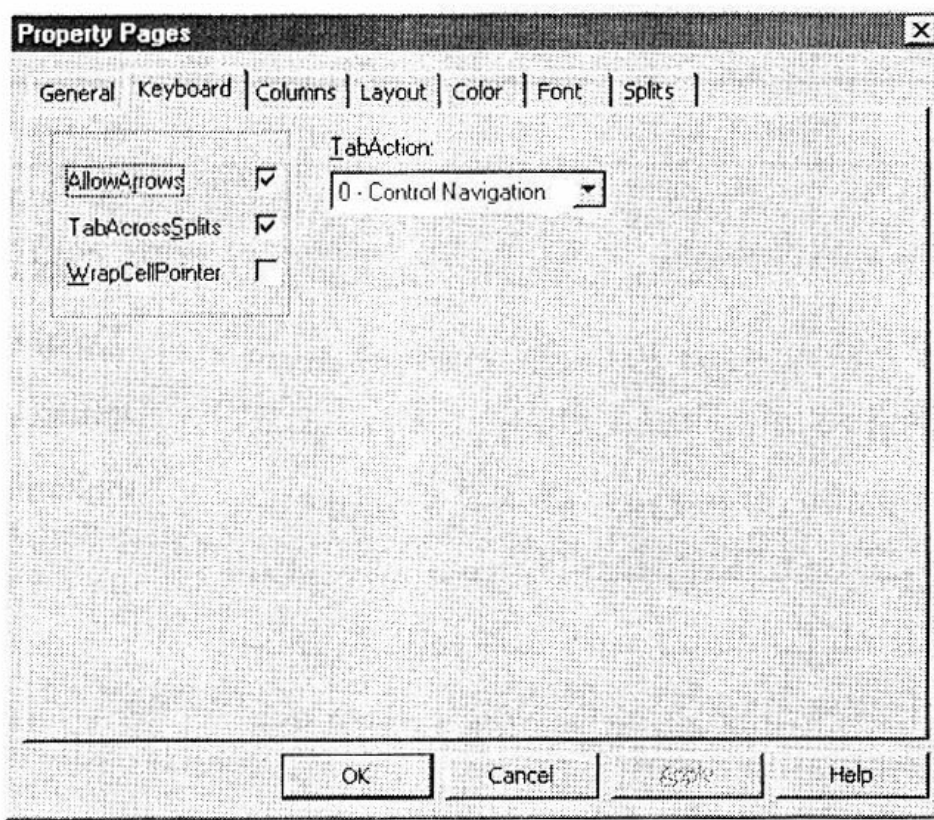


شکل ۷-۱۰.

نام قسمت	توضیح
Caption	تیتتر کلی جدول در این قسمت مشخص می شود.
AllowAddNew	در صورتی که True باشد، در حالت اجرا، یک رکورد خالی در انتهای DBGrid قرار می گیرد و در صورتی که در آن داده‌ای وارد کنید، آن رکورد به لیست داده شما اضافه می شود. در صورتی که ارزش این خصوصیت False باشد، دیگر نمی توان به لیست داده رکوردی اضافه نمود.
AllowDelete	در صورتی که True باشد می توانید با کلیک کردن روی مکان نمای هر سطر آن را انتخاب کرده و با زدن دکمه Delete آن رکورد را حذف نمایید. در صورتی که این خصوصیت را False کنید، این عمل نمی تواند صورت گیرد.
AllowUpdate	در صورتی که True باشد، در هنگام اجرای برنامه، تغییراتی که در داده‌ها DBGrid داده می شود ثبت می گردد و در غیر این صورت تغییرات در لیست داده ثبت نمی شود.
ColumnHeaders Enable	تعیین می کند که ستونهای فیلدها تیتتر داشته باشند یا خیر. در صورتی که True باشد، DBGrid فعال است و در صورتی که False باشد DBGrid غیر فعال می شود.
MarqueeStyle	مشخص می کند که نوع انتخاب شده Marquee در قسمت Splite فعال باشد یا خیر.
HeadLines	اندازه سطر تیتتر را مشخص می نماید.
RowHeight	اندازه عرض سطرها را تعیین می کند.
Appearance	نمایش جدول را به صورت 3D یا معمولی مشخص می کند.
BorderStyle	تعیین می کند که دور جدول محدود شود یا خیر.
RowDividerStyle	نوع خطهای بین رکوردها، سطرها و ستونها را تعیین می کند.
DataMode	تعیین می کند که جدول مقید به داده‌های کنترل داده باشد یا خیر.

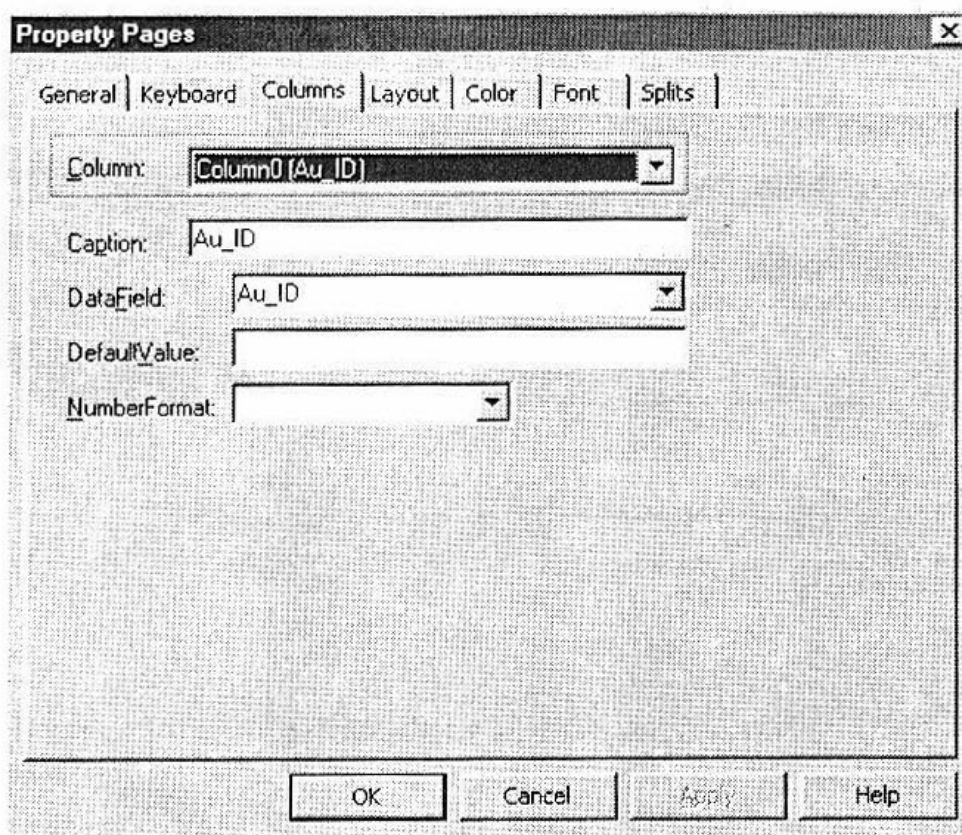
• **Keyboard**: در این قسمت کلیدهای فعال روی جدول مشخص می شوند.

شکل ۸-۱۰.
برگه Keyboard



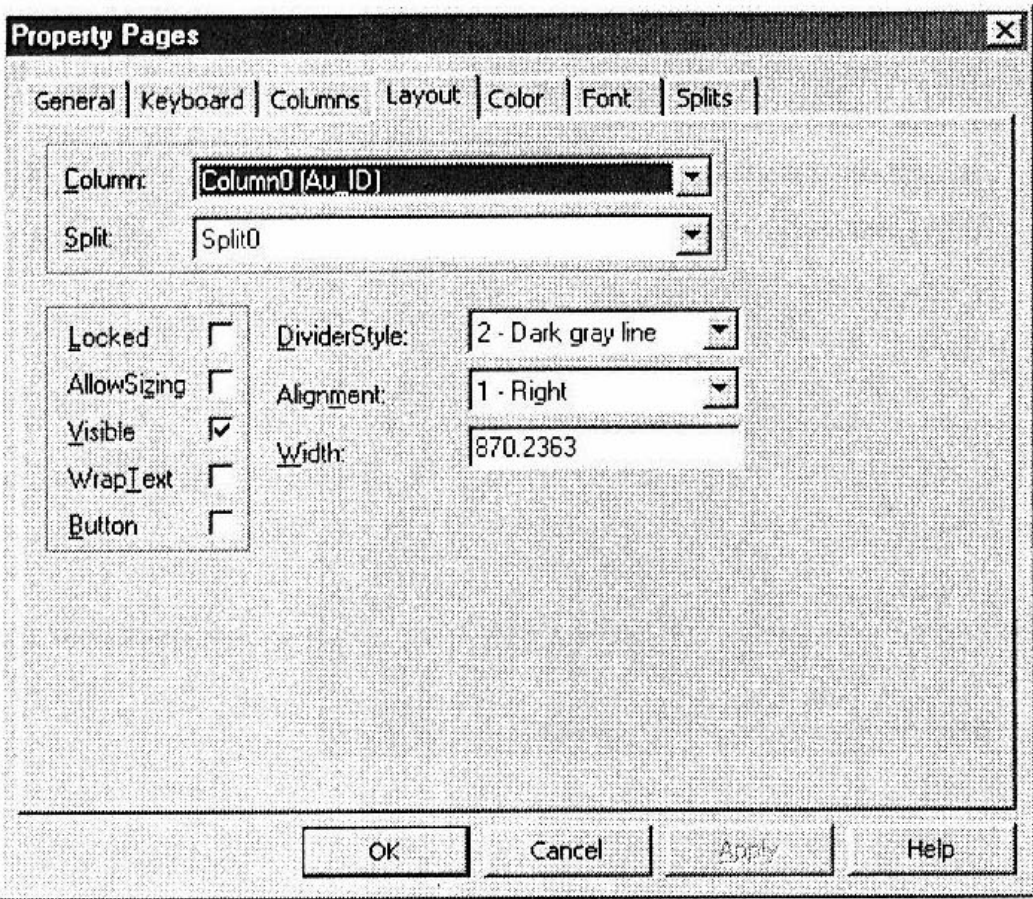
• **Columns**: در این قسمت می‌توانید فیلد مربوط به هر ستون و تیترا آن را مشخص کنید. در جدول ۱۰-۲ این خصوصیات نشان داده شده است.

شکل ۹-۱۰



نام قسمت	توضیح
Column	ستون جدول را مشخص می‌کند.
Caption	تیتر هر ستون را مشخص می‌نماید.
DataField	در این قسمت می‌توانید فیلد مربوط به هر ستون را تعیین کنید.
DefaultValue	مقدار پیش‌فرض را برای فیلدهای رکوردهای جدید جدول تعیین می‌کند.
NumberFormat	در صورتی که فیلد شما از نوع عددی باشد از طریق این قسمت می‌توانید به آن قالب خاصی را اختصاص دهید. در صورتی که بخواهید از قالب خاصی استفاده کنید که در لیست نمی‌باشد، در این قسمت می‌توانید قالب دلخواه را وارد نمایید. برای مثال اگر از (###, # ریال) استفاده کنید. مقدار فیلد به همراه متن ریال در قالب نمایش مبلغ در فیلد قرار می‌گیرد.

• **Layout**: در این قسمت می‌توانید برای هر ستون به طور مجزا خصوصیات را اختصاص دهید. در جدول ۳-۱۰ این خصوصیات را نشان می‌دهد.



شکل ۹-۱۰.
برگه Layout

توضیح

نام قسمت

در صورتی که برای هر ستون این گزینه را انتخاب کنید اطلاعات آن ستون غیر قابل تغییر می‌شود و در صورتی که روی اطلاعات آن تاپ کنید هیچ عملی صورت نخواهد گرفت.

Locked

تعیین می‌کند که اندازه ستون قابل تغییر باشد یا خیر.

AllowSizing

تعیین می‌کند که اطلاعات این ستون نمایش داده شود یا خیر.

Visible

یک دکمه در کنار فیلد قرار می‌دهد که می‌توانید در رویداد Button_Click آن را کنترل کنید.

Button

خط‌های بین ستون فیلد را در حالت‌های مختلف تعیین می‌کند. در صورتی که این خصوصیت به No Dividers تنظیم شود بین ستون این فیلد، خطی دیده نمی‌شود.

DividerStyle

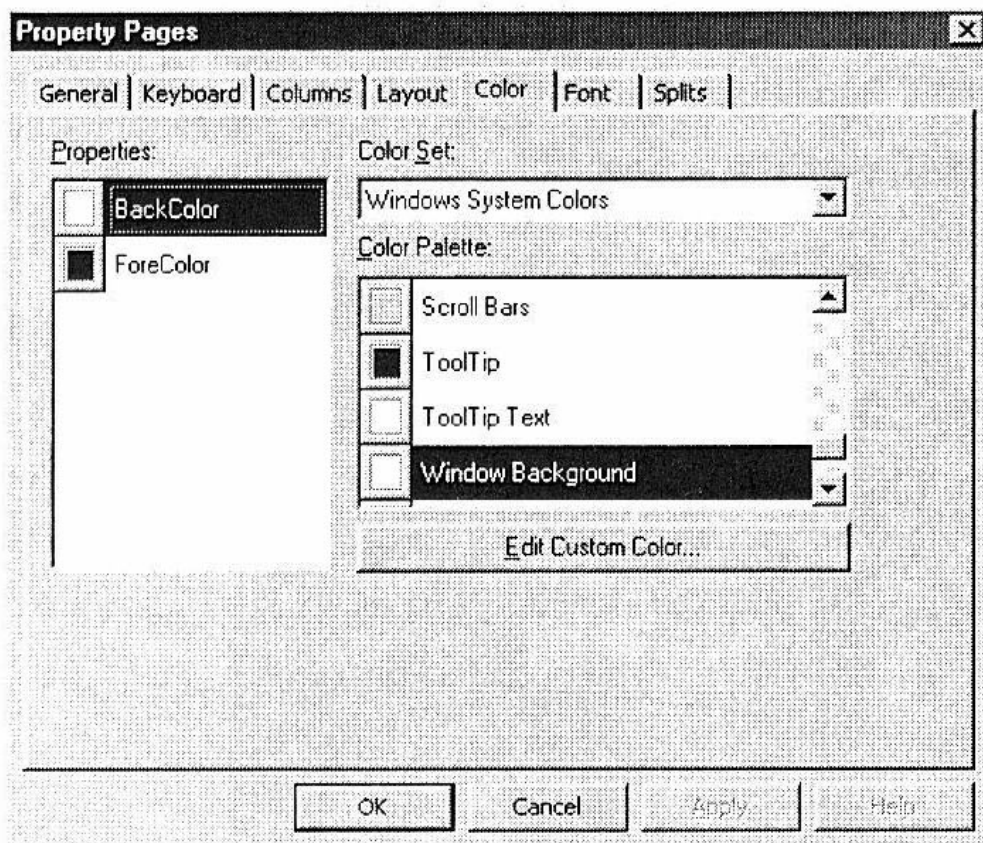
مکان قرار گرفتن متن را درون ستون تعیین می‌کند که می‌تواند راست‌چین، چپ‌چین و یا وسط‌چین باشد و یا در حالت عمومی قرار گیرد.

Alignment

پهنای ستون را مشخص می‌کند.

Width

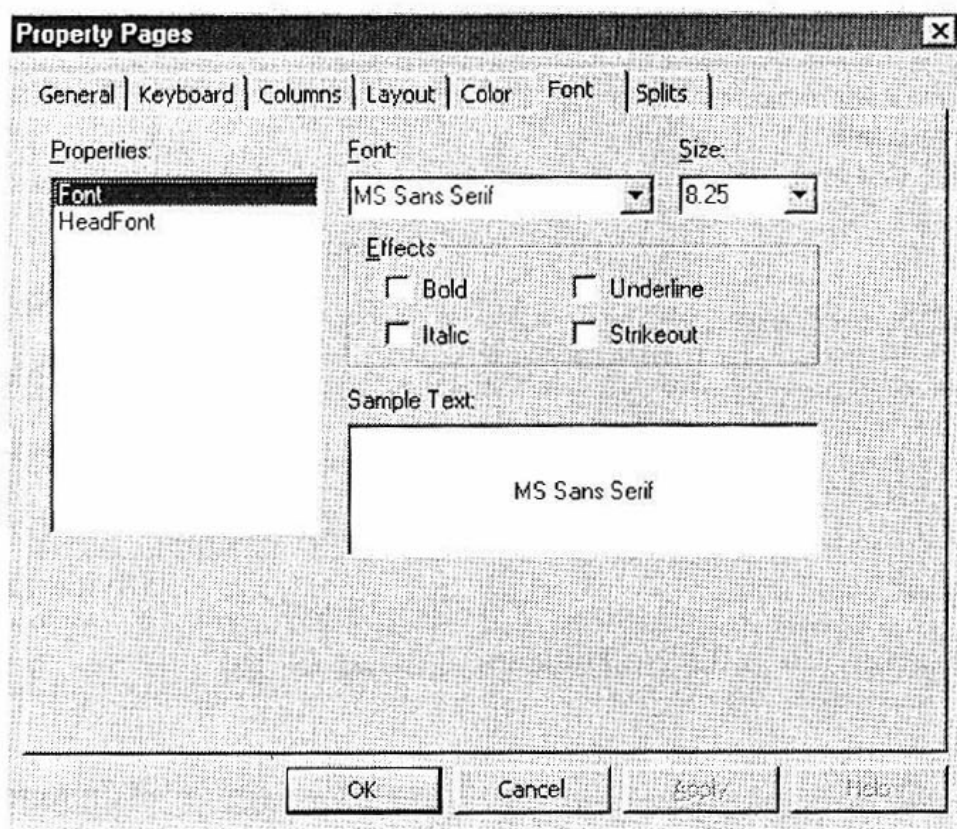
• Color: در این قسمت می‌توان رنگ قسمت‌های مختلف جدول را تعیین کرد.



شکل ۱۰-۱۰

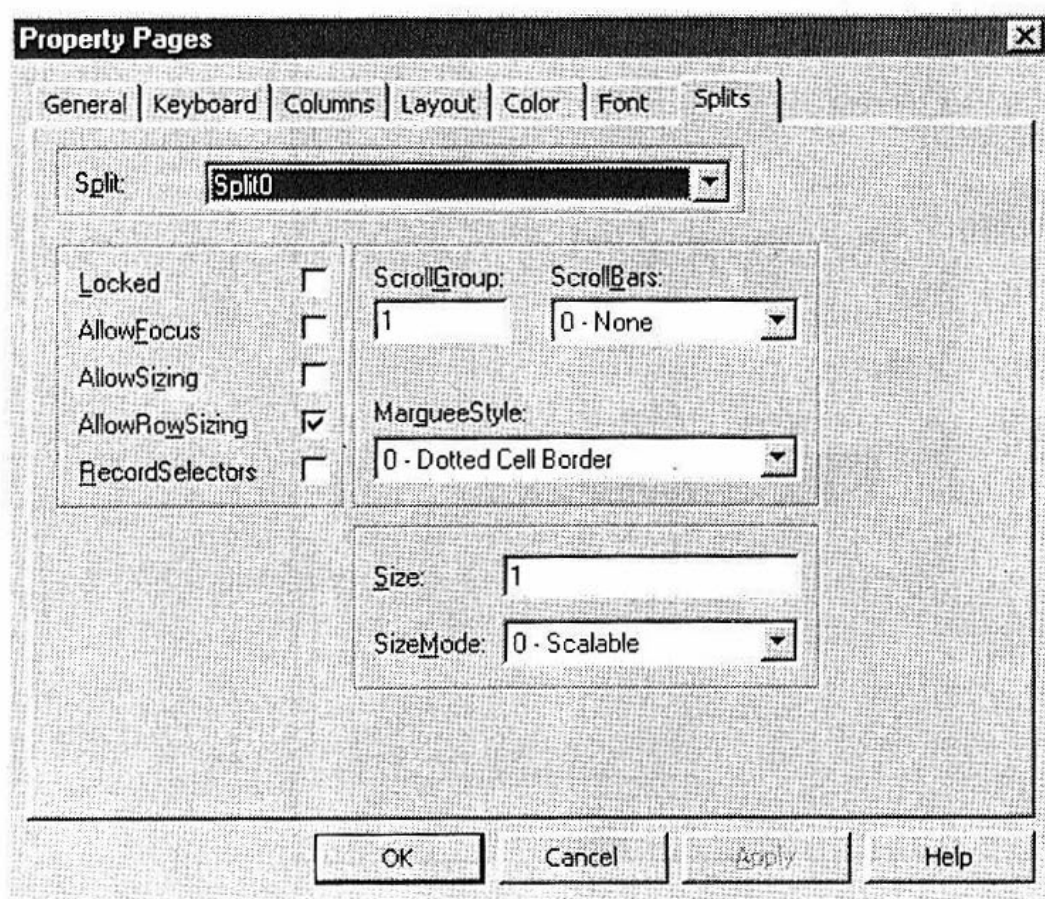
برگه Color

• **Font** : در این قسمت فونت متنها و داده‌های جدول تعیین می‌شود.



شکل ۱۱-۱۰.
برگه Font

• **Split** : در این قسمت یک سری مشخصات تکمیل کننده برای جدول DBGrid می‌باشد که در جدول ۴-۱۰ آنها را توضیح می‌دهیم.



شکل ۱۲-۱۰.
برگه Split

توضیح

نام قسمت

در صورتی که این خصوصیت را انتخاب کنیم تمام فیلدها قفل می‌شوند و نمی‌توان درون آنها تایپ نمود.	Locked
اجازه تغییر مکان از روی سلولهای جدول را تعیین می‌کند.	AllowFocus
اجازه تغییر اندازه سطرها و ستونها را تعیین می‌کند. در صورتی که این گزینه انتخاب شود در حالت اجرای برنامه می‌توان اندازه سطرها و ستونها را تغییر داد.	AllowSizing
میله‌های حرکت در طرفین جدول را فعال می‌کند.	ScrollBars
هنگامی که روی یک سلول از این جدول قرار می‌گیرد حالت‌های مختلف نشان دادن مکان توسط این خصوصیت تعیین می‌شود که شامل هفت گزینه است که گزینه ۶ به صورت پیش فرض قرار دارد. شکل ۱۲-۱۰ این حالتها را نشان می‌دهد.	MarqueeStyle



شکل ۱۲-۱۰.
حالت‌های مختلف Marquee Style

شکل ۱۳-۱۰ چهار ستون و ۱۲ سطر اول جدول Titles را از پایگاه داده Biblio.MDB نشان می‌دهد که در کنترل dbgTitle قرار دارد و تیترا جدول و ستونها به حالت Bold درآمده است. پس از اینکه در حالت اجرا مکان سطرها و ستونها را از طریق تکان دادن لبه‌های هر ستون و سطر، تنظیم کردیم در پنجره Immediate مقادیر این تنظیمات را به دست می‌آوریم تا در هنگامی که فرم باز می‌شود به وسیله کد، این تنظیمات را انجام دهیم. ما می‌توانیم این تنظیمات را در رویداد Activate در فرم قرار دهیم. برای این کار کافی است مقادیر را از پنجره Immediate کپی کنیم و در رویداد فرم قرار دهیم.

Form1

Titles Table of Biblio.mdb Database

Title	Year	ISBN	PubID
dBASE III : A Practical Guide	1985	0-0038307-6-4	469
The dBASE Programming Language	1986	0-0038326-7-8	469
dBASE III Plus	1987	0-0038337-8-X	469
Database Management : Developing Application S	1989	0-0131985-2-1	715
Wordstar 4.0-6.0 Quick Reference Guide	1990	0-0133656-1-4	460
Oracle Triggers and Stored Procedure Programmin	1996	0-0134436-3-1	715
Programming in Clipper	1988	0-0201145-8-3	9
Inside Macintosh	1994	0-0201406-7-3	9
Omni Online Database Directory	1983	0-0207992-0-9	156
Structured C for Engineering and Technology/Boo	1995	0-0230081-2-1	715
An Introduction to Assembly Language Programmir	1995	0-0230362-0-6	715
Applied Calculus With Linear Programming : For Bu	1995	0-0230650-8-7	119
Information Systems Literacy and Software Produc	1991	0-0230942-1-4	156

Biblio Titles

شکل ۱۳-۱۰.

```

Immediate
? dbgTitle.Columns(0).caption
Title
? dbgTitle.Columns(0).Width
3644.788
? dbgTitle.Columns(1).caption
Year Published
? dbgTitle.Columns(1).Width
585.0709
? dbgTitle.Columns(2).caption
ISBN
? dbgTitle.Columns(2).Width
1170.142

```

شکل ۱۴-۱۰.

تعیین مقادیر اندازه‌های سطرها
و ستونها در پنجره Immediate

باید توجه داشته باشید که پس از انتقال این کدها به فرم باید تغییراتی در آنها انجام دهیم. برای مثال بالا لیست زیر تایپ می‌کنیم :

```

Privata Sub Form_Activate()
    dbgTitle.Columus(0).Width = 3645
    dbgTitle.Columus(1).Width = 585
    dbgTitle.Columus(2).Width = 1170
    dbgTitle.Columus(3).Width = 840
    dbgTitle.Columus(4).Width = 1850
    dbgTitle.Columus(5).Width = 1500
    dbgTitle.Columus(6).Width = 3000
    dbgTitle.Columus(7).Width = 3000
End Sub

```

استفاده از رویداد Head Click

معمولاً جداول مورد استفاده برنامه‌های کاربردی دارای خاصیتی می‌باشند که می‌توان لیست داده‌های آن را براساس فیلد مورد نظر مرتب کرد. این کار با کلیک کردن روی تیتراستون فیلد میسر می‌شود.

اگر بخواهید این کار را در ویژوال بیسیک شبیه‌سازی کنید، باید کد مربوطه را در رویداد HeadClick وارد کنید. ساده‌ترین و سریعترین راه مرتب کردن یک لیست داده از طریق کدهای SQL می‌باشد. شما می‌توانید با تنظیم یک کد SQL برای لیست داده‌ای که DBGrid به آن مقید شده است، لیست داده را مرتب کنیم.

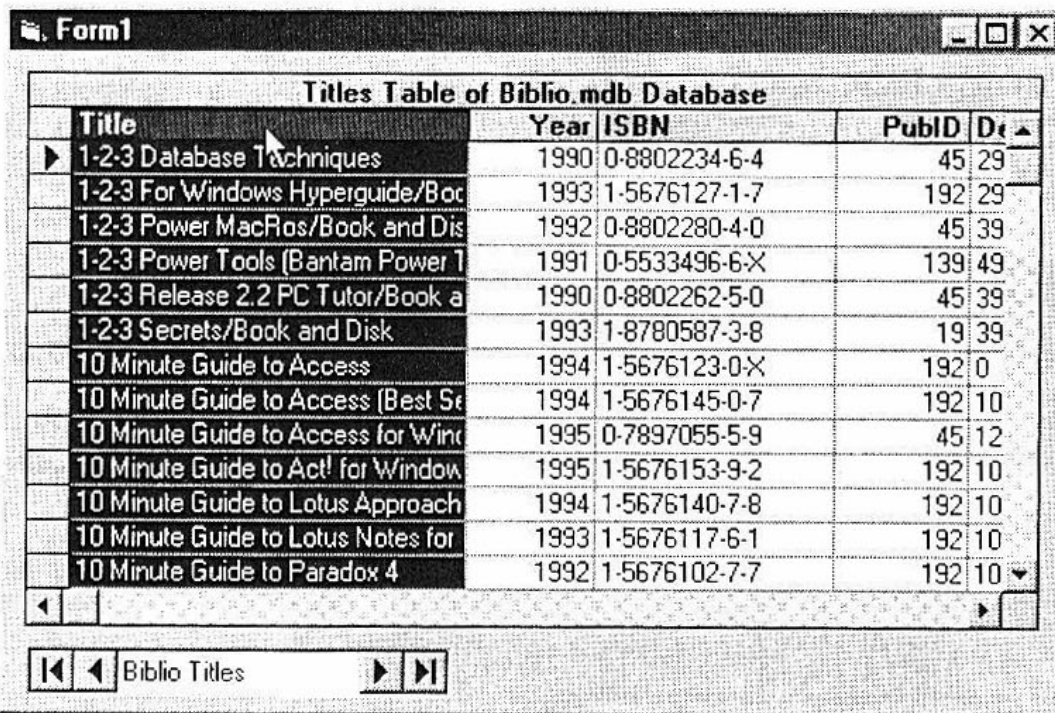
پس از آنکه لیست داده را مرتب کردیم باید لیست DBGrid را به هنگام کنیم تا تغییرات روی لیست داده به نمایش درآید. برای این کار از دستور Refresh استفاده می‌کنیم. کد زیر باعث می‌شود که هرگاه روی تیتراستون از DBGrid کلیک شود، لیست داده آن براساس آن فیلد مرتب گردد.

```
Private Sub dbgridTitle-HeadClick(ByVal ColIndex As Integer)
```

```
DtcTitles.RecordSource = "SELECT * FROM Titles ORDER BY [" & _  
dbgridTitle.Columns(ColIndex).DataField & "] ASC"
```

```
DtcTitles.Refresh
```

```
End Sub
```



Title	Year	ISBN	PubID	Date
1-2-3 Database Techniques	1990	0-8802234-6-4	45	29
1-2-3 For Windows Hyperguide/Book	1993	1-5676127-1-7	192	29
1-2-3 Power MacRos/Book and Disk	1992	0-8802280-4-0	45	39
1-2-3 Power Tools (Bantam Power T	1991	0-5533496-6-X	139	49
1-2-3 Release 2.2 PC Tutor/Book a	1990	0-8802262-5-0	45	39
1-2-3 Secrets/Book and Disk	1993	1-8780587-3-8	19	39
10 Minute Guide to Access	1994	1-5676123-0-X	192	0
10 Minute Guide to Access (Best Se	1994	1-5676145-0-7	192	10
10 Minute Guide to Access for Wind	1995	0-7897055-5-9	45	12
10 Minute Guide to Act! for Window	1995	1-5676153-9-2	192	10
10 Minute Guide to Lotus Approach	1994	1-5676140-7-8	192	10
10 Minute Guide to Lotus Notes for	1993	1-5676117-6-1	192	10
10 Minute Guide to Paradox 4	1992	1-5676102-7-7	192	10

شکل ۱۵-۱۰.

تنظیم رویداد Head Click
برای مرتب کردن هر ستون

تغییر نمایش ستون‌ها در حالت اجرا

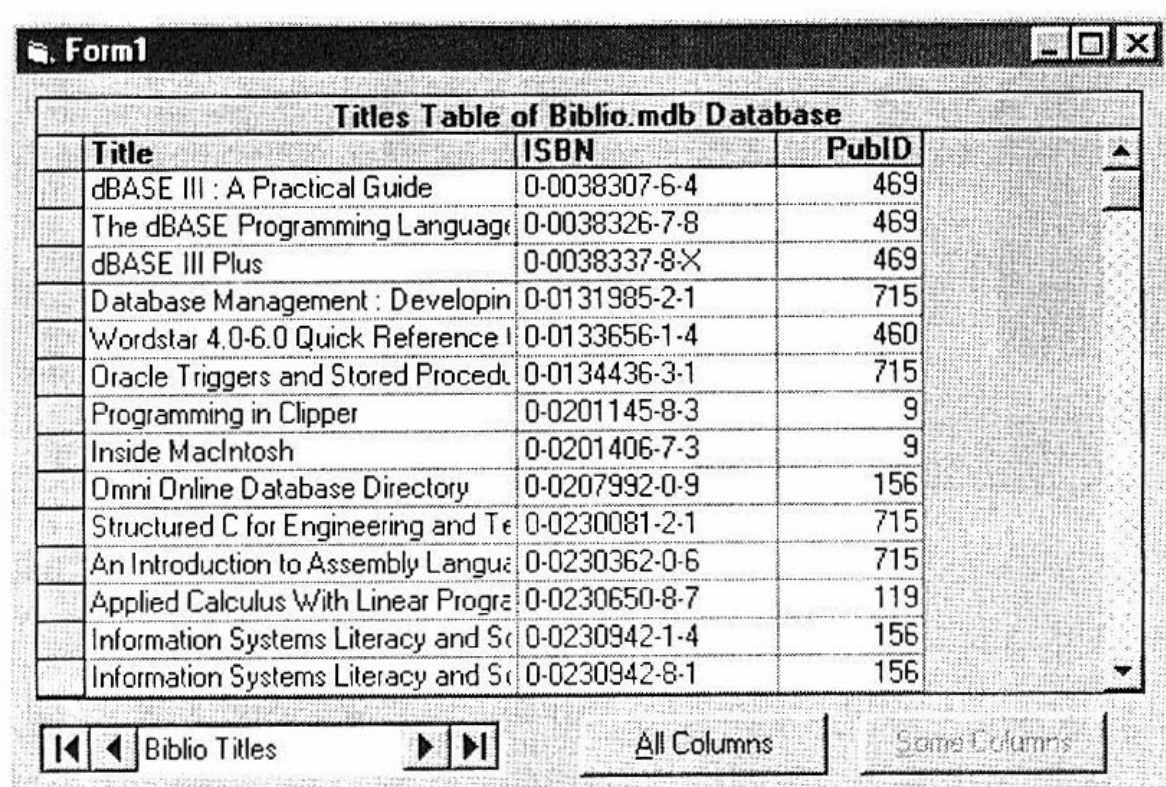
شما می‌توانید تعداد ستون‌هایی که در حالت اجرا نمایش داده می‌شوند را کنترل کنید. آسانترین راه برای این کار استفاده از خصوصیت Visible برای ستون‌ها می‌باشد، در صورتی که این خصوصیت False باشد ستون مربوطه در لیست DBGrid نمایش داده نمی‌شود. پنهان کردن ستون‌های اطلاعات در

بسیاری حالات مفید می باشد و می توان از آن برای محدود کردن محیط نمایش کاربرانی که نباید به اطلاعات بعضی از ستونها دسترسی داشته باشد، استفاده کرد.

کدهای زیر که در دو زیرروال نوشته شده است، طریقه استفاده از این امکانات را در کنترل DBGrid نشان می دهد.

```
Private Sub CmdAllCols_Click()
    'Display all columns
    Dim intCol As Integer
    cmdsomecols.Enabled = True
    For intCol = 1 To dbgTitle.Columns.Count - 1
        dbgTitle.Columns(intCol).Visible = True
    Next intCol
    CmdallCols.Enabled = False
End Sub
```

```
Private Sub cmdSomeCols_Click()
    'Hide unneeded columns
    cmdsomecols.Enabled = False
    dbgTitle.Columns(1).Visible = False
    dbgTitle.Columns(4).Visible = False
    dbgTitle.Columns(5).Visible = False
    dbgTitle.Columns(6).Visible = False
    dbgTitle.Columns(7).Visible = False
    CmdallCols.Enabled = True
End Sub
```



شکل ۱۶-۱۰

پنهان سازی ستونها توسط
 خصوصیت Visible

برای این عمل، راه کندتری نیز وجود دارد و آن استفاده از بازسازی لیست داده است که نیاز به Refresh کردن لیست داده و DBGrid دارد. سرعت این عمل به مراتب کمتر از استفاده از شیء Visible می‌باشد؛ بنابراین استفاده از این روش در برنامه‌ها پیشنهاد نمی‌شود.

استفاده از رویدادها برای تصدیق به روز رسانی داده‌ها

در کنترل DBGrid مجموعه‌ای از رویدادها وجود دارد که به وسیله آنها می‌توان عملیات به روز رسانی (Update) را روی داده کنترل کرد.

شما باید کدهای خود را برای این کار در رویدادهایی که با Befor... شروع می‌شوند، انجام دهید. در این رویدادها آرگومانی به نام Cancel وجود دارد که در صورتی که آن را به True تنظیم کنید تغییری روی داده‌ها اعمال نخواهد شد. جدول ۵-۱۰ مجموعه این رویدادها را نشان می‌دهد.

جدول ۵-۱۰: رویدادهای به روز رسانی

رویداد	زمان فعال شدن
BeforColUpdate	بعد از تغییر داده یک سلول و حرکت به سلول جدید و قبل از تغییر محتویات بافر DBGrid.
AfterColUpdate	بعد از تغییر محتویات بافر DBGrid برای به روز رسانی ستون.
BeforDelete	بعد از انتخاب یک سطر و زدن کلید Delete و قبل از اعمال حذف در لیست داده.
AfterDelete	بعد از حذف سطر از لیست داده.
BeforInsert	بعد از وارد کردن اولین کاراکتر در آخرین سطر لیست داده و قبل از ایجاد شدن یک سطر جدید.
AfterInsert	بعد از اضافه شدن سطر جدید به جدول.
BeforUpdate	بعد از تغییر مقادیر هر ستون و حرکت به رکورد بعدی و قبل از به روز رساندن لیست داده.
AfterUpdate	بعد از به روز رساندن یک سطر از لیست داده.

لیست ۱-۱۰ یک نمونه استفاده از رویدادهای جدول ۵-۱۰ و نمایش جعبه پیغام برای کنترل تصدیق داده‌ها را نشان می‌دهد.


```
Private Sub dbgTitle_AfterColUpdate(ByVal ColIndex As Integer)
    MsgBox "column " & ColIndex + 1 & " updated.", _
        vbInformation, "update Confirmation"
End Sub

Private Sub dbgTitle_AfterDelete()
    MsgBox "Row deleted.", vbInformation, "delete Confirmation"
End Sub

Private Sub dbgTitle_AfterInsert()
    MsgBox "New row added.", vbInformation, "Insert Confirmation"
End Sub

Private Sub dbgTitle_AfterUpdate()
    MsgBox "Row updated.", vbInformation, "Update Confirmation"
End Sub

Private Sub dbgTitle_BeforeColUpdate(ByVal ColIndex As Integer, _
    OldValue As Variant, Cancel As Integer)
    Dim strWarn As String
    strWarn = "update the column value from '" & _
        OldValue & "' to '" & _
        dbgTitle.Columns(ColIndex).Visible & "'?"

    If MsgBox(strWarn, vbYesNo + vbQuestion, "Comfirm Update") = _
        vbNo Then
        Cancel = True
    End If
End Sub

Private Sub dbgTitle_BeforeDelete(Cancel As Integer)
    Dim strWarn As String
    strWarn = strmsg & "delete this row?"
    If MsgBox(strWarn, vbYesNo + vbQuestion, "Confirm Delete") = _
        vbNo Then
        Cancel = True
    End If
End Sub

Private Sub dbgTitle_BeforeInsert(Cancel As Integer)
    Dim strWarn As String
    strWarn = "add this new row?"
```



```

If MsgBox(strWarn, vbYesNo + vbQuestion, "Confirm Insert") = _
    vbNo Then
    Cancel = True
End If
End Sub

```

```

Private Sub dbgTitle_BeforeUpdate(Cancel As Integer)
    Dim strWarn As String
    strWarn = "update this row?"
    If MsgBox(strWarn, vbYesNo + vbQuestion, "Confirm Update") = _
        vbNo Then
        Cancel = True
    End If
End Sub

```

انتخاب اقلام به وسیله DBCombo و DBList

کنترل‌های Dbcombo و در وسعت کمتری کنترل‌های DBlist ابزارهای مفیدی برای ایجاد یک لیست انتخابی روی مجموعه رکوردهای مجزای DBGrid می‌باشد. در دو قسمت بعد، توضیح می‌دهیم که :

- چگونه می‌توان از DBCombo در شرط SQL WHERE استفاده کرد.
- چگونه از چند DBCombo مرتبط برای انتخاب دقیقتر رکوردها استفاده کنیم.

استفاده از DBCombo برای محدود کردن لیست DBGrid

در اکثر موارد، یک لیست کنترل DBcombo به وسیله یک فیلد از یک جدول محصور می‌شود. برخلاف اکسس ۹۵ شما نمی‌توانید بیش از یک فیلد را برای کنترل DBCombo نمایش دهید و همچنین نمی‌توانید ترکیبی از مقادیر چند فیلد را در تمام DBlist تنظیم کنید؛ بنابراین شما باید از خصوصیت‌های مخصوص DBComb استفاده کنید تا به مقادیری که در لیست نشان داده نمی‌شود دسترسی پیدا کنید.

جدول ۶-۱۰ لیست این خصوصیات را که در DBCombo و DBList مورد استفاده قرار می‌گیرد نشان می‌دهد.

هدف	خصوصیت
در این خصوصیت نام ابزار کنترل داده‌ای که می‌خواهید از لیست داده آن در کنترل لیست استفاده کنید، قرار می‌گیرد.	RowSource
نام فیلدی از لیست داده که براساس آن کنترل لیست تنظیم می‌شود. در این قسمت قرار می‌گیرد.	ListField
نام فیلدی از لیست داده که قرار است در شرط SQL WHERE سطرهایی را به کنترل داده دیگر نسبت دهید در این قسمت مشخص می‌شود.	BoundColumn
مقدار ارزش فیلد Bound Column در این قسمت قرار می‌گیرد. نوع لیست را از جهت شکل ظاهری و نوع انتخاب، مشخص می‌کند.	BoundText Style

جدول Publisheres از پایگاه داده Biblio.MDB یک نمونه از جداولی است که از طریق فیلد کلیدی (PubID) به جداول دیگر متصل می‌شود. تعدادی فیلد توضیحی از قبیل Name و Company Name نیز در آن وجود دارد که مطابق اطلاعات رکورد است. در صورتی که این فیلدهای توضیحی را در DBCombo قرار دهید باید برای ارتباط از فیلد کلیدی استفاده کنید. برای این اعمال به ترتیب زیر عمل نمایید:

- ۱ - به فرم خود دو ابزار کنترل داده اضافه کنید. از یکی از آنها برای تعیین Row Source در DbCombo و از دیگری برای نمایش اطلاعات در DBGrid استفاده کنید.
- ۲ - مشخصه Database Name را در هر دو کنترل داده تنظیم کنید (در بیشتر مواقع تمام کنترل‌های داده از یک پایگاه داده استفاده می‌کنند).
- ۳ - مقدار مشخصه Record Source را در اولین کنترل داده به یک کد SQL در حالت کلی زیر تنظیم کنید :

```
SELECT BoundFieldName,ListFieldName FROM TableName ORDER BY ListFieldName
```

در اینجا فقط فیلدهای مورد نیازتان را از جدول انتخاب کنید و در ORDER BY برای مرتب کردن الفبایی موارد لیست استفاده نمایید.

- ۴ - در رویداد کلیک از DBCombo کدی را وارد کنید که عملیات انتساب کد SQL را به Record Source دومین کنترل داده را انجام دهد. این کد در حالت کلی به صورت زیر است :

```
SELECT * FROM Table2Name WHRER ForeignKeyName = ['] BoundFiledValue [']
```

که BoundFiledList در آن به خصوصیت Bound Text در DBCombo تبدیل می‌شود.

شکل ۱۷-۱۰ یک نمونه ساده استفاده از DBCombo را برای نمایش اطلاعات مرتبط نشان می‌دهد. در این حالت ListFieldName نام فیلد و BoundFieldName، فیلد Pub ID از جدول Publishers می‌باشد.

لیست ۲-۱۰ کد مربوط به این برنامه را نشان می‌دهد.

لیست ۲-۱۰:

```
Option Explicit
Dim fWidthSet As Boolean

Private Sub DBCpublishers_Click(Area As Integer)
    If Area = dbcAreaList Then
        ' area in list clicked
        Dim strSql As String
        DtcTitles.Caption = "Titles from " & DBCpublishers.Text
        strSql = "SELECT* FROM Titles WHERE PubID= " & DBCpublishers.BoundText
        DtcTitles.RecordSource = strSql
        DtcTitles.Refresh
        DtcTitles.UpdateControls
        If Not fWidthSet Then
            ' set the colume widths once
            Call SetColumnWidths
            fWidthSet = True
        End If
    End If
End Sub

Private Sub SetColumnWidths()
    ' Set width of column for readability
    dbgTitle.Columns(0).Width = 3500
    dbgTitle.Columns(1).Width = 570
    dbgTitle.Columns(2).Width = 1170
    dbgTitle.Columns(4).Width = 1850
    dbgTitle.Columns(5).Width = 1500
    dbgTitle.Columns(6).Width = 3000
    dbgTitle.Columns(7).Width = 3000
End Sub

Private Sub Form_Activate()
    ' Initalize the DBcombo control
    DBCpublishers.SetFocus
    SendKeys "{F4}{Down}{Enter}", True
    Call DBCpublishers_Click(2)
End Sub
```

شکل ۱۷-۱۰.

The screenshot shows a window titled 'Form1' with a table of book titles. A dropdown menu is open, listing publishers: M & T, MIT, MACMILLAN COMPUTER, HIGHTEXT PUBNS, SPRINGER VERLAG, O'REILLY & ASSOC (highlighted), ADDISON-WESLEY, and JOHN WILEY & SONS. The table below has columns for Year, ISBN, and PubID.

	Year	ISBN	PubID
	1992	0-9371751-1-0	8
ase	1992	0-9371751-2-9	8
	1991	0-9371752-6-9	8
erie:	1990	0-9371754-2-0	8
	1990	0-9371755-9-5	8
	1993	0-9371757-0-6	8
Essential System Administration	1991	0-9371757-4-9	8
The Z-Mail Handbook/3 Interfaces for E-Mail	1991	0-9371757-6-5	8
Power Programming With Rpc	1991	0-9371757-7-3	8
The X Resource : A Practical Journal of the X W	1991	0-9371757-9-X	8
Essential System Administration (A Nutshell Hanc	1991	0-9371758-0-3	8
X Window System Administrator's Guide (X Wind	1992	0-9371758-3-8	8

At the bottom, there are navigation buttons: 'Titles from O'REILLY & ASSOC' and 'Publishers'.

اتصال دو کنترل DBCombo برای انتخاب رکوردهای مخصوص

شما می‌توانید از دو DbCombo پشت سر هم، برای انتخاب یک رکورد مخصوص، جهت نمایش و اطلاع داده‌ها در جعبه‌های متن مقیدشده استفاده کنید.

فرمی که در شکل ۱۸-۱۰ دیده می‌شود توسط ویزارد طراحی فرم در ویژوال بیسیک ایجاد شده و دو کنترل DbCombo به آن اضافه شده است. کنترل DbCombo ای که در سمت چپ قرار گرفته است، ناشر را انتخاب می‌کند و این انتخاب باعث می‌شود که در DbCombo سمت راست لیست کتابهای چاپ‌شده توسط این ناشر نشان داده می‌شود و با انتخاب هریک اطلاعات فرم تنظیم می‌شود.

شکل ۱۸-۱۰.

The screenshot shows a window titled 'Titles' with a form. A dropdown menu is open, listing book titles: dBASE III : A Practical Guide (highlighted), The dBASE Programming Language, dBASE III Plus, Database Management : Developing Wordstar 4.0-6.0 Quick Reference G, Oracle Triggers and Stored Procedur, Programming in Clipper, and Inside MacIntosh.

Form fields:

- Title: dBASE III : A Pr
- Year Published: 1985
- ISBN: 0-0038307-6-4
- PubID: 469
- Description: 22.5
- Notes:
- Subject:
- Comments: 0

Buttons at the bottom: Add, Delete, Refresh, Update, Close.

لیست ۳-۱۰ رویداد Click را در دو کنترل DBCombo که در شکل ۱۸-۱۰ دیده می‌شود نشان می‌دهد. توجه داشته باشید که فیلد ISBN از نوع متنی است و در شرط SQL Where باید از علامت (") در طرفین آن استفاده کرد اما فیلد PubID از نوع عددی است و در شرط SQL Where نیازی به استفاده از (") در طرفین آن نیست.

لیست ۱-۱۰

```
private Sub DBCpublishers_Click(Area As Integer)
    If Area = dbcAreaList Then
        ' area in list clicked
        Dim strSql As String
        strSql = "SELECT Title, ISBN FROM Titles WHERE PubID = " &
        DBCpublishers.BoundsText
        dtcTitles.RecordSource = strSql
        dtcTitles.Refresh
        dtcTitles.UpdateControls
        If dtcTitles.Recordset.RecordCount > 0 Then
            ' display the first selection
            dbcTitles.SetFocus
            SendKeys "{F4}{Down}{Enter}", True
            Call dBcTitles_Click(2)
        End If
    End If
End Sub

Private Sub dBcTitles_Click(Area As Integer)
    If Area = dbcAreaList Then
        ' area in list clicked
        Dim strSql As String
        strSql = "SELECT * FROM Titles WHERE ISBN = " & dbcTitles.BoundsText
        dtcbook.RecordSource = strSql
        dtcbook.Refresh
        dtcbook.UpdateControls
    End If
End Sub
```

استفاده از DB List

این کنترل در تمام پارامترهای ارتباط با ابزار کنترل داده همانند Dbcombo می‌باشد و خصوصیات آن را داراست.

چاپ گزارش در ویژوال بیسیک

ویژوال بیسیک خود در زمینه چاپ گزارشات امکاناتی را ندارد و امکاناتی که به عنوان شیء چاپگر عرضه می شود مربوط به نسخه اصلی غیر ویژوال آن یعنی Quick Basic می باشد و این یک نقص برای یک محیط برنامه سازی تحت پایگاه داده بوده و لذا در ویرایشهای حرفه ای ویژوال بیسیک نسخه های ۴ به بعد برنامه ای با نام Crystal Report محصول شرکت Seagate Software به ویژوال بیسیک افزوده شد که قادر بود گزارشاتی در حد قسمت گزارشات اکسس ایجاد کند و در برنامه های کاربردی پایگاه داده مورد استفاده قرار گیرد. برنامه Crystal Report توسعه یافت و در نسخه های بعدی ویژوال بیسیک ، نسخه جدید آن عرضه شد.

در این فصل پس از بررسی چاپ گزارش به وسیله کدهای برنامه نویسی با Crystal Report 4.6 آشنا می شویم.

دسترسی به چاپگر از طریق کد

شیء چاپگر در ویژوال بیسیک این امکان را به شما می دهد تا با سیستم چاپگر ارتباط داشته باشید . مجموعه چاپگر شما را قادر می کند تا اطلاعات را به هر چاپگری که از طریق سیستم قابل دسترسی است ارسال کنید .
قالب استفاده از این اشیا عبارتند از:

Printer

Printers (Index)

index یک عدد است که از صفر تا 1 - Printers.Count می باشد. دستور Printers.Count تعداد پرینترها را مشخص می کند.

نکات

- شما می توانید با استفاده از متدهای گرافیکی نظیر Line , Paint Picture , Point , Pset , Print , Circle , Cls تصاویر و اشکال و متون را به شیء چاپگر ارسال کنید.
- در صورتی که بخواهید اطلاعات چاپ شده توسط چاپگر از چاپگر خارج شود از متد EndDoc استفاده نمایید .
- هنگامی که می خواهید عملیات چاپ را انجام دهید باید مکان کاغذ را چک کنید . برای مثال اگر از دستور Print Form برای چاپ یک تصویر گرافیکی استفاده می کنید ، ممکن است

نصف عکس در پایین کاغذ اول و بقیه آن در صفحه دوم چاپ شود ، بنابراین لازم است مکان کاغذ را توسط خصوصیت‌های مربوطه تنظیم کنید.

- شما می‌توانید از مجموعه چاپگرها ، یک چاپگر را به عنوان چاپگر پیش‌فرض تنظیم نمایید. برای مثال شاید بخواهید چاپگری را بیابید که دارای درایور مخصوصی باشد . در کد زیر تمام چاپگرهای قابل دسترسی جستجو می‌شود و در صورتی که کاغذ هر یک از این چاپگرها به صورت Portrait تنظیم شده بود به عنوان چاپگر پیش‌فرض تنظیم می‌گردد.

```
Dim x As Printer
For Each x In Printers
    If x.Orientation = vbDRORPortrait Then
        Set Printer = x
        Exit For
    End If
Next
```

همان طور که در کد بالا دیدید برای تنظیم یک چاپگر به صورت پیش‌فرض از دستور Set استفاده می‌کنیم .

- در صورتی که بخواهید از چاپگری در مجموعه چاپگرها از طریق index استفاده نمایید مشخصات این چاپگر فقط خواندنی می‌شود . برای اینکه بخواهید هم مشخصات را بخوانید و هم در صورت نیاز آنها را تغییر دهید ابتدا آن چاپگر را به چاپگر پیش‌فرض تنظیم کنید و سپس از مشخصات آن استفاده کنید.

متدهای مهم شیء چاپگر

جدول ۱-۱۱: لیست متدهای شیء چاپگر .

عملیات

نام متد

عملیات	نام متد
عملیات چاپ چاپگر به اتمام می‌رسد و کاغذ از چاپگر بیرون می‌آید و به صورت زیر به کار می‌رود:	EndDoc
Printer.EndDoc	
در صورتی که از این دستور بعد از NewPage استفاده کنید، کاغذ سفید رد نخواهد شد.	
عملیات فعلی چاپگر را متوقف می‌کند و به صورت زیر به کار می‌رود :	KillDoc
Printer. KillDoc	
ادامه صفحه جاری را سفید رد می‌کند و چاپگر را برای کاغذ جدید آماده می‌نماید .	New Page
Printer.NewPage	
محتویات یک فایل گرافیکی با پسوند های (BMP , Wmf , emf , ico , dib) را چاپ می‌کند و به صورت زیر استفاده می‌شود .	Paint Picture
Printer.PaintPicture Picture , X1 , Y1 , Width1 , Heigh1 , X2 , Y2 ,	

Width2 , heigh2

Pset یک نقطه با رنگ دلخواه را چاپ می کند .

Printer.Pset (x , y) , [color]

Printer.Pset (2,2),Qbcolor (Rnd * 15)

جدول ۱۱-۲ لیست پارامترهای این دستور را نشان می دهد .

جدول پارامترهای متد Paint Picture

جدول ۱۱-۲ : لیست پارامترهای متد Paint Picture.

پارامتر	توضیح
Picture	در این پارامتر منبع تصویر مشخص می شود . توجه داشته باشید که این منبع باید در مشخصه Picture در فرم یا Picture Box باشد.
(X1,Y1)	مکان شروع چاپ عکس را مشخص می کند.
Width1,Heigh1	دو پارامتر دلخواه می باشد که طول و عرض تصویر را در چاپ مشخص می کند . اگر طول یا عرض واقعی تصویر کمتر یا بیشتر از طول و عرض مشخص شده در این پارامتر باشد تصویر در چاپ کشیده و یا به هم فشرده می شود .
X2,Y2	مکان انتهای چاپ عکس را مشخص می کند.
Width2,Heigh2	دو پارامتر دلخواه می باشد که طول و عرض واقعی تصویر را مشخص می کند .

شیء چاپگر دارای سی و سه خصوصیت است که در جدول ۱۱-۳ مهمترین آنها را توضیح می دهیم .

جدول ۱۱-۳ : لیست خصوصیت های شیء چاپگر .

پارامتر	توضیح
Copies	تعداد نسخه های چاپ را مشخص می کند و به صورت زیر استفاده می شود : Printer.Copies [= Number]
	• در صورتی که از Dialog Box مربوط به چاپ استفاده کنید این خصوصیت مقداری را برمی گرداند که در جعبه متن تعداد نسخه های چاپ وارد شده است .
	• در حالت پیش فرض مقدار این خصوصیت ۱ می باشد .
Current X Current Y	مکان چاپ را در چاپگر تنظیم و یا تعیین می کند . طریقه استفاده از آن به صورت زیر است:

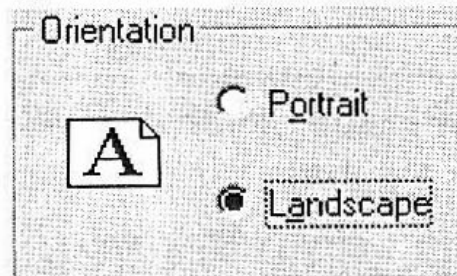
Printer . Current X [=X]
Printer . Current Y [=Y]

نام درایور چاپگر را بر می گرداند. DriverName
فونت چاپ را مشخص می کند. Font

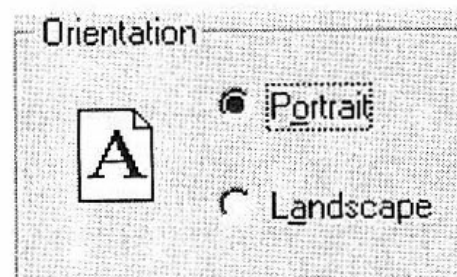
جدول ۳-۱۱: ادامه .

توضیح	پارامتر
نوع چاپ خط را مشخص می کند که به صورت زیر استفاده می شود . Printer . FontBold [= Boolean] Printer . FontItalic [= Boolean] Printer . FontStrikethry [= Boolean] Printer . FontUnderLine [= Boolean] Boolean یک مقدار در منطقی True یا False می باشد .	FontBold FontItalic FontStrikethry
نام فونت مورد نظر را برای چاپ تنظیم می کند . Printer . FontName [= Font]	FontName
اندازه فونت را برای چاپ تنظیم می کند . Printer . Font Size[= اندازه] Printer . Font Size = 8	FontSize
تعیین می کند که متن تحت چاپ ، تحت فونت تعریف شده در مشخصات قبل چاپ شود یا خیر . Printer . Font Transporent = {True False}	FontTransparen t

طول و عرض کاغذ را در چاپ تنظیم می کند . Height, Wiedth
تعیین می کند که چاپ به صورت Portrait یا Landscape باشد . Orientation



شکل ۱-۱۱
000114. BMP



شکل ۲-۱۱

و به صورت زیر استفاده می شود :

Printer . Orientation [= Value]

مقدار ۱ در Value برای Prortrait و مقدار ۲ برای Landscape می باشد.

Page

شماره صفحه جاری را بر می گرداند .

PaperSize

اندازه کاغذ را برای چاپگر جاری تعریف می کند و به صورت زیر استفاده می شود .

Printer . PaperSize [= Value]

مقادیر Value در جدول ۴-۱۱ نشان داده شده است .

Print

متن موردنظر را چاپ می کند .

جدول ۳-۱۱ : ادامه .

توضیح

پارامتر

Printer . Print Quality [= Value]

کیفیت چاپ را تعیین می کند.

PrintQuality

مقادیر Value به صورت زیر می باشد :

1. Draft
2. Low Resolution
3. Medium Resolution
4. High Resolution

Printer .

مقدار بزرگ‌نمایی را در چاپ تعیین می کند .

Zoom

Zoom [= Number]

مقدار Number برحسب درصد می باشد ، یعنی مقدار 100 اندازه واقعی را چاپ

می کند .

جدول ۴-۱۱ : لیست اندازه های کاغذ در خصوصیت PaperSize .

ثابت	ارزش	توضیح
VbPRPSLetter	1	Letter, 8 1 in.
vbPRPSLetterSmall	2	Letter Small, 8 11 in.
vbPRPSTabloid	3	Tabloid, 11 x 17 in.
vbPRPSLedger	4	Ledger, 17 x 11 in.
vbPRPSLegal	5	Legal, 8 1/2 x 14 in.
vbPRPSStatement	6	Statement, 5 1/2 x 8
vbPRPSExecutive	7	Executive, 7 1/2 x 10
vbPRPSA3	8	A3, 297 x 420 mm
vbPRPSA4	9	A4, 210 x 297 mm
vbPRPSA4Small	10	A4 Small, 210 x 297 mm
vbPRPSA5	11	A5, 148 x 210
vbPRPSB4	12	B4, 250 x 354 mm
vbPRPSB5	13	B5, 182 x 257 mm

vbPRPSFolio	14	Folio,8 13 in.
vbPRPSOuarto	15	Quarto, 215 x 275 mm
vbPRPS10x14	16	10 x 14 in.
vbPRPS11x17	17	11 x 17 in.
vbPRPSNote	18	Note,8 1/2 x 11 In.
vbPRPSEnv9	19	Envelope #9,3 7/8 x 8 7/8
vbPRPSEnv10	20	Envelope #10,4 1/4 x 9 1/2
vbPRPSEnv11	21	Envelope #11,4 1/2 x 10 3/8
vbPRPSEnv12	22	Envelope #12,4 1/2 x 11 In.
vbPRPSEnv14	23	Envelope #14, 5 x 11 1/2 in.
vbPRPSCSheet	24	C size sheet
vbPRPSDSheet	25	D size sheet
vbPRPSESHEET	26	E size sheet
vbPRPSEnvDL	27	Envelope DL, 110 x 220 mm
vbPRPSEnvC3	29	Envelope C3, 324 x 458 mm
vbPRPSEnvC4	30	Envelope C4, 229 x 324 mm
vbPRPSEnvC5	28	Envelope C5,162 x 229 mm
vbPRPSEnvC6	31	Envelope C6,114 x162 mm
vbPRPSEnvC65	32	Envelope C65,114 x 229mm
vbPRPSEnvB4	33	Envelope B4, 250 x 353 mm
vbPRPSEnvB5	34	Envelope B5,176 x 250 mm
vbPRPSEnvB6	35	Envelope B6, 176 x125 mm
vbPRPSEnvItaly	36	Envelope,110 x 230 mm
vbPRPSEnvMonarch	37	Envelope Monarch, 3 7/8 x 7 1/2 in.
vbPRPSEnvPersonal	38	Envelope,3 5/8 x 6 1/2 in.
vbPRPSFanfoldUS	39	U.S. Standard Fanfold,14 7/8 x 11 in.
vbPRPSFanfoldStdGerman	40	German Standard Fanfold, 8 1/2 x 12 in.

چاپ گزارش از لیست داده

کد زیر فیلدهای یک لیست داده را به صورت گزارش با استفاده از شیء چاپگر چاپ می کند. این کد در حالت کلی نوشته شده است و شما می توانید از آن در حالت های مختلف استفاده کنید. لیست ۱-۱۱: حالت کلی چاپ گزارش به وسیله کد .

```

Printer.Print ReportHeading      چاپ سربرگ گزارش
Do Until Recordset.EOF          تا زمانی که رکورد باقیمانده در Loop باقی می ماند
Printer.CurrentX = LeftMargin    تنظیم حاشیه چپ
Printer.CurrentY = HeadTopPosition  تنظیم فاصله چاپ از ابتدای کاغذ
Printer.Print PageHeading        چاپ سربرگ صفحه
Printer.CurrentY = DataTopPosition  تنظیم حاشیه بالا برای داده ها
LineCounter = 1
                                تا زمانی که رکوردهای محدود به هر صفحه باقیمانده در Loop می ماند
Do Until LineCounter >= MaxLines
Printer.CurrentX = LeftMargin

```

تک تک فیلدهای هر رکورد در این قسمت 'Print the fields of the recordset in sequence'

```

For FieldNumber = 1 To Recordset.field.Count
  Printer.CurrentX = Printer.CurrentX + PrintWidth(FieldNumber)
  Printer.Print Format(Recordset.field(FieldNumber).Value, Format(n))
If FieldNumber = n And Not_
  IsNull(Recordset.field(FieldNumber(n))) Then
  'Null values added to the sum give you a null sum
  SumFieldNumber(n) = SumFieldNumber(n) + _
    Recordset.field(Field_Number(n)).Value
  End If
Next FieldNumber
'move to next line انتقال به خط بعدی
Printer.CurrentY = Printer.CurrentY + LineHeight
Recordset.MoveNext
LineCounter = LineCounter + 1
Loop
'Print the page number as footer چاپ شماره صفحه در انتهای کاغذ
Printer.CurrentX = FooterLeftMargin
Printer.CurrentY = FooterTopMargin
Printer.Print PageCounter
'Issue a page eject آماده سازی کاغذ جدید
Printer.NewPage
PageCounter = PageCounter + 1
Loop
'Print one or more totals چاپ مجموعه ها
Printer.Print .Format(SumFieldNumber(n), "FormatSum" )
'Eject the last page بیرون آوردن آخرین کاغذ
Printer.EndDoc

```

استفاده از کنترل Crystal Report

مجموعه Crystal Computer Servere ناشر برنامه Crystal Report نسخه ۴، ۴،۵، ۴،۶ می باشد که نسخه ۴،۵ به بعد آن به عنوان یک برنامه مستقل برای تهیه گزارشات از پایگاههای داده شناخته شد و تنها مورد استفاده ویژوال بیسیک نبود. این برنامه با استفاده از قابلیت هایی که به آن اضافه شده بود می توانست از طریق یک DLL در ویژوال بیسیک کنترل شود و اکثر ویژگیهایش را در اختیار ویژوال بیسیک قرار دهد. این برنامه همچنین می توانست فایل های گزارش را کامپایل کرده و به صورت یک برنامه اجرایی تحت ویندوز درآورد.

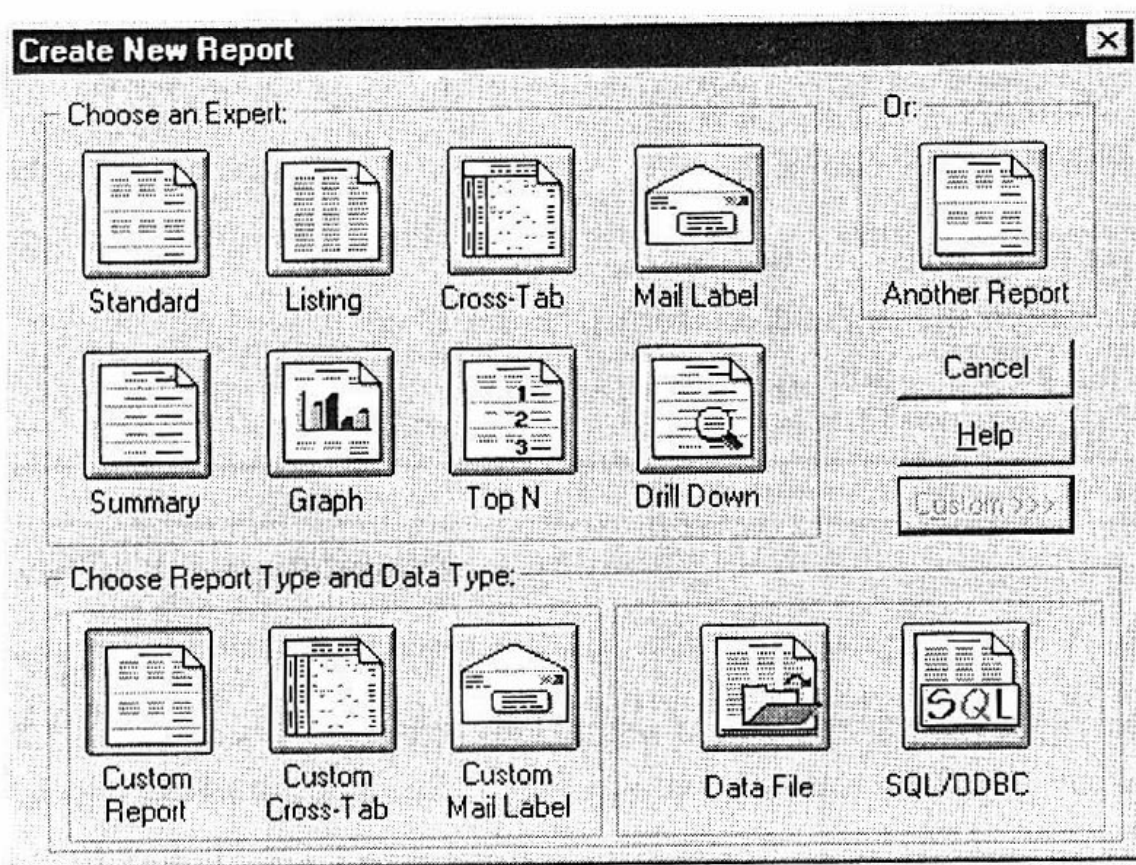
کنترل سفارشی گزارش تحت فایل Cryt132.OCX در ویژوال بیسیک فعال می‌شود و شما می‌توانید از طریق خصوصیت Report File Name نام گزارشی را که توسط Crystal report ایجاد کرده‌اید وارد کنید و در برنامه خود آن گزارش را فعال نمایید. بنابراین برای ایجاد یک گزارش در ویژوال بیسیک ابتدا باید فایل گزارش (.rpt) را به وسیله برنامه Crystal report بسازید و سپس کنترل سفارشی Cry132.OCX را به فرم برنامه خود اضافه کنید. در قسمتهای بعدی این فصل چگونگی ترکیب یک گزارش را در برنامه‌های ویژوال بیسیک توضیح خواهیم داد.

ایجاد یک فایل گزارش ساده توسط Crw32.EXE

برای ایجاد یک فایل گزارش به ترتیب زیر عمل کنید:

۱ - از شاخه ویژوال بیسیک، برنامه Crystal reports را اجرا کنید. در صورتی که این برنامه را برای اولین بار اجرا کرده باشید فرمی از مشخصات فردی اجرا می‌شود که اگر مایل بودید می‌توانید آن را پر کرده و عملیات ثبت را انجام دهید. اگر در اینجا فرم را Cancel کنید برنامه آماده کار می‌شود.

۲ - از منوی File گزینه New را انتخاب کنید تا پنجره ایجاد گزارش جدید نشان داده شود و سپس دکمه فرمان Custom را کلیک کنید.



شکل ۳-۱۱.

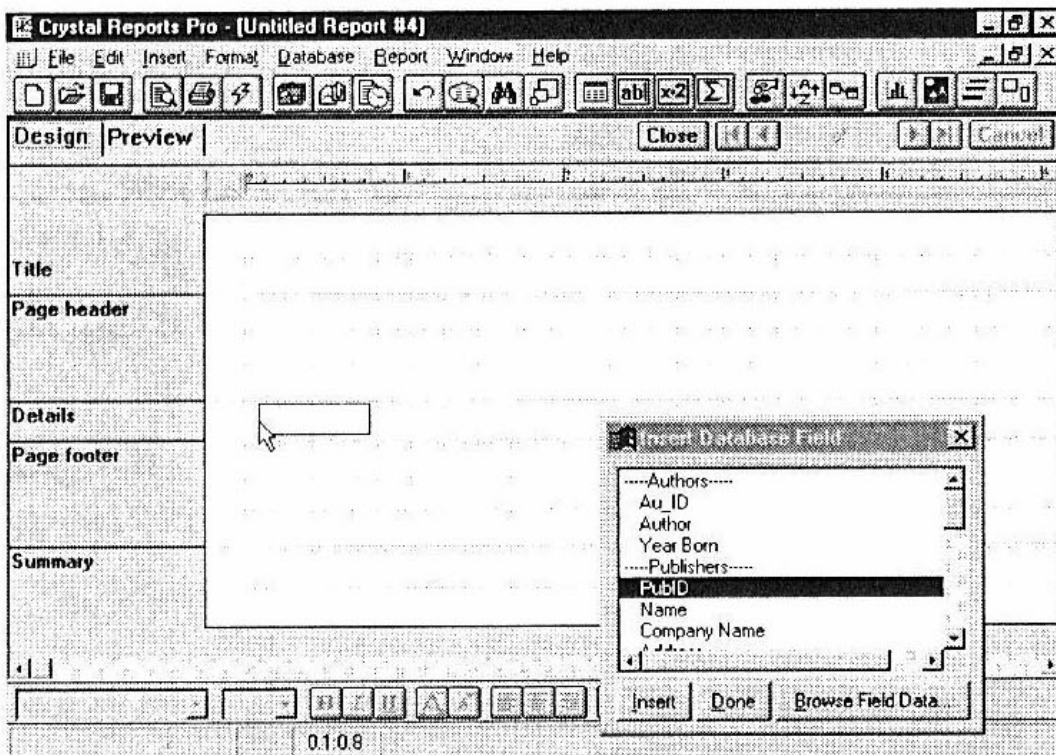
این پنجره از سه قسمت تشکیل شده است :

- **قسمت اول :** این قسمت Choose an Expert نام دارد و در واقع یک ویزارد برای ایجاد گزارش است.
- **قسمت دوم :** این قسمت Or نام دارد که در صورت انتخاب Another Report یک فایل با پسوند rpt یا rpl یا rpc خوانده و نمایش داده می شود .

• **قسمت سوم :** این قسمت که با انتخاب دکمه فرمان Custom فعال می شود مربوط به ایجاد یک گزارش جدید براساس یک پایگاه داده می باشد و تفاوت آن با قسمت اول در ویزارد نبودن آن است.

۳ - Custom Report را از قسمت سوم انتخاب و سپس روی Data File کلیک کنید تا پنجره انتخاب پایگاه داده فعال شود .

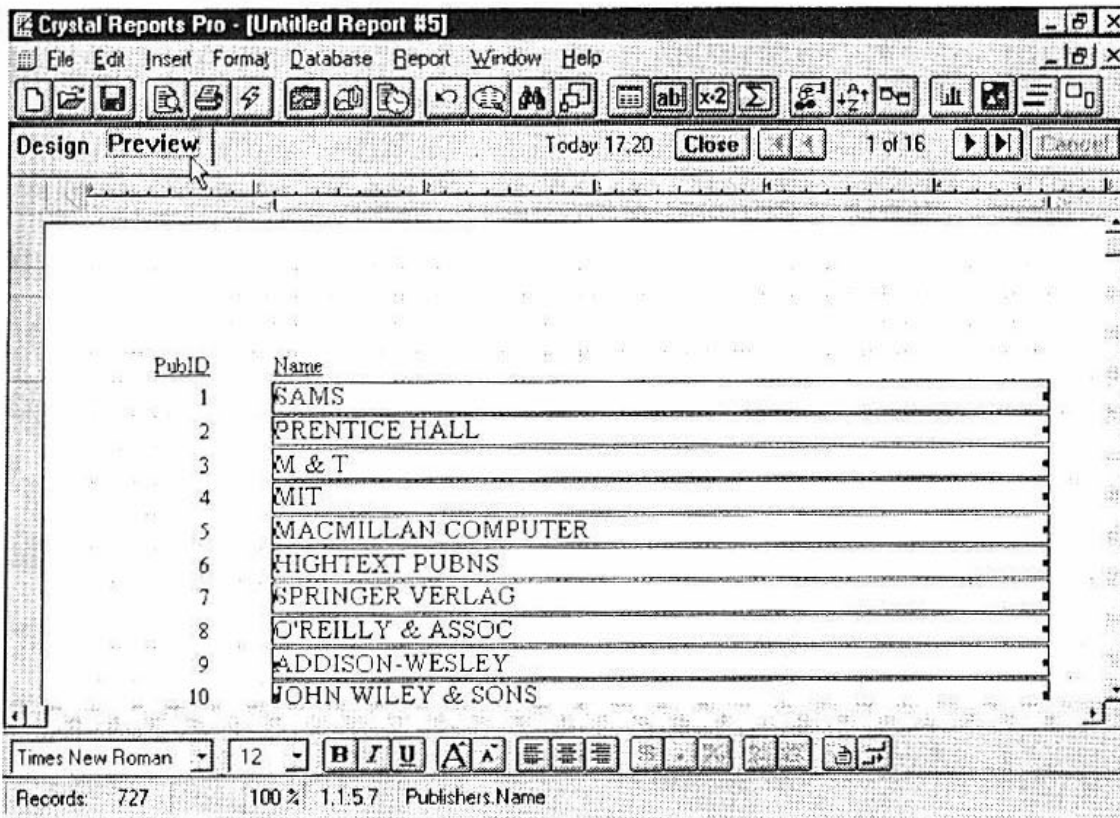
۴ - در اینجا می توانید گزارش خود را براساس پایگاه داده دلخواه ایجاد کنید . در صورتی که یک پایگاه داده را انتخاب نمایید تمامی جداول سؤالهای آن در گزارش قابل دسترسی خواهد بود. در اینجا برای مثال فایل Biblio.mdb را انتخاب کنید، حال گزارش شما آماده طراحی است . در پنجره ای که Insert Database Field نام دارد لیست تمام جداول و سؤالها به همراه فیلدهای آن قرار دارد و می توانید روی فیلد دلخواهی مانند Pub ID از جدول Publishers دوبار کلیک کنید و با جابجایی ماوس در قسمت طراحی گزارش (شکل ۴-۱۱) فیلد را در قسمت Details قرار دهید.



شکل ۴-۱۱.

- ۵ - مرحله ۴ را به تعداد فیلدهای دلخواه می توانید تکرار کنید و فیلدها را به ترتیب در جاهای مناسب قرار دهید . در اینجا فیلد Name را از جدول Publishers انتخاب کنید و آن را بعد از فیلد PubID قرار دهید. سپس پنجره فیلدها را ببندید .
- ۶ - حال Perview را انتخاب کنید تا گزارش شما به نمایش درآید .

شکل ۵-۱۱.

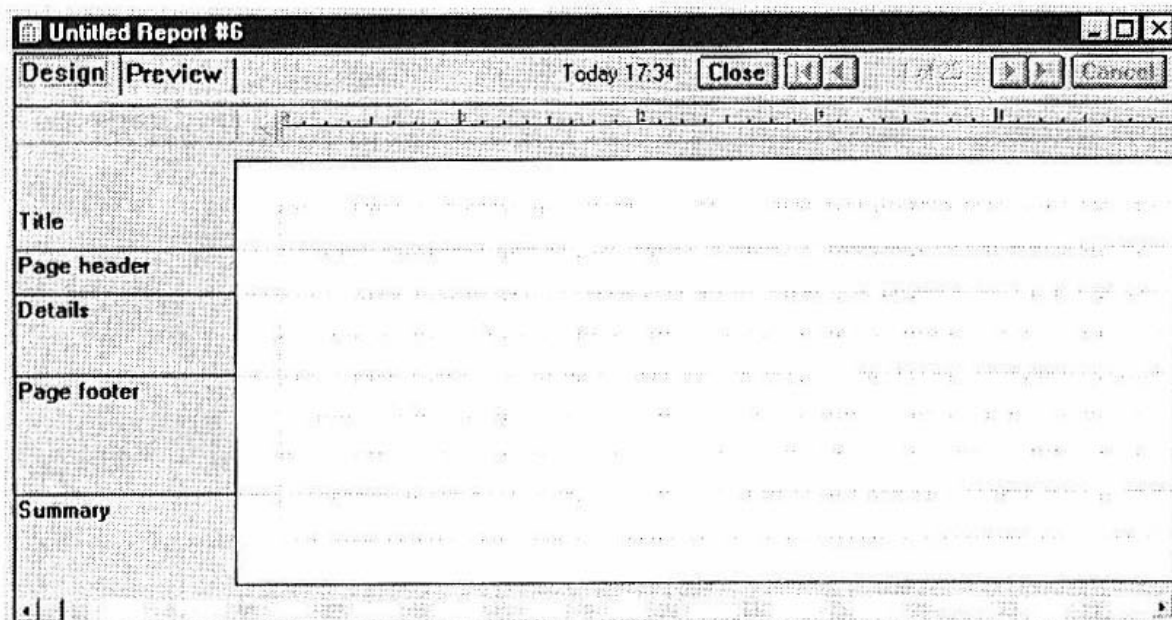


۷- از منوی File گزینه Save را انتخاب کنید و پس از وارد کردن نام گزارش در قسمت Filename آن را در شاخه مطلوب ذخیره کنید. همان طور که دیدید می‌توانید به راحتی یک گزارش ساده را از روی جدول و یا سؤال خود با فیلدهای دلخواه ایجاد کنید. در ادامه توضیحات بیشتری در مورد برنامه Crystal Report خواهیم داد.

بررسی ظاهری هر گزارش

همان طور که در شکل ۶-۱۱ می‌بینید هر گزارش از دو قسمت Design و Perview تشکیل شده است که قسمت اول مربوط به طراحی گزارش و قسمت دوم مربوط به پیش‌نمایش آن می‌باشد.

شکل ۶-۱۱.



• در قسمت طراحی، قسمت سفیدی در اختیاران قرار دارد که معرف کاغذ چاپ بوده و این کاغذ به قسمت‌های مختلفی تقسیم شده است.

• قسمت Title مربوط به تیتراژ شما می‌باشد. هر نوشته و فیلدی که در این قسمت قرار گیرد، فقط در صفحه اول گزارش چاپ می‌شود و در بقیه صفحات تکرار نمی‌شود. در صورتی که در این قسمت از فیلدی استفاده کنید در قسمت Title اولین رکورد این فیلد نمایش داده می‌شود.

• قسمت Page Header: این قسمت سربرگ هر صفحه می‌باشد و متن‌ها و فیلدهای آن در بالای هر صفحه تکرار می‌شود.

• قسمت Details: مهمترین قسمت بخش طراحی است و فیلدهایی که در این قسمت قرار بگیرند به ترتیب از ابتدای کاغذ، بعد از سربرگ به پایین لیست می‌شوند. در صورتی که در این قسمت از یک متن ثابت و یا تصویر استفاده کنید به ازای هر رکورد آن متن یا تصویر در هر خط تکرار می‌شود.

• قسمت Page Footer: این قسمت ته‌برگ هر صفحه می‌باشد و آنچه در این قسمت قرار بگیرد در انتهای هر صفحه از گزارش تکرار می‌شود.

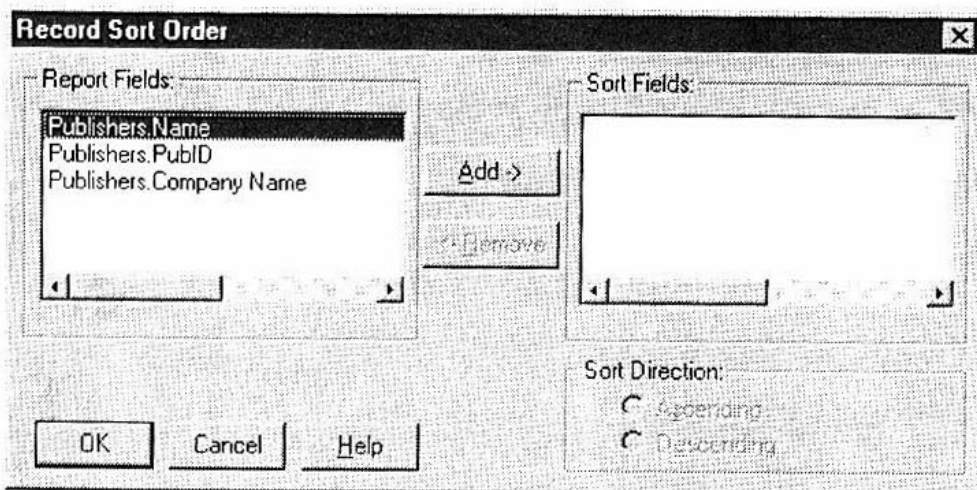
• قسمت Summary: آنچه در این قسمت قرار بگیرد در آخرین صفحه از گزارش به عنوان نتیجه و یا خلاصه اطلاعات چاپ می‌شود.

اضافه کردن یک پایگاه داده به گزارش

اگر خواسته باشید علاوه بر پایگاه داده‌ای که در ابتدای گزارش انتخاب کردید از اطلاعات پایگاه داده دیگری نیز استفاده کنید باید از منوی Database گزینه Add Database To Report را انتخاب کنید تا فیلدهای پایگاه داده جدید به لیست فیلدهای قبلی اضافه شود. برای اضافه کردن فیلد به گزارش می‌توانید از منوی Insert گزینه Database Field را انتخاب کنید و به همان ترتیب که در ابتدا گفته شد فیلدها را در جای مناسب قرار دهید.

مرتب کردن یک فیلد در گزارش

معمولاً امکان مرتب کردن (Sort) یک گزارش براساس یک فیلد بسیار ضروری است و در اکثر گزارشات به کار می‌رود. برای این کار کافی است از قسمت ابزارها آیکن ۵-۱۱ را انتخاب کنید تا پنجره زیر باز شود:



شکل ۷-۱۱.

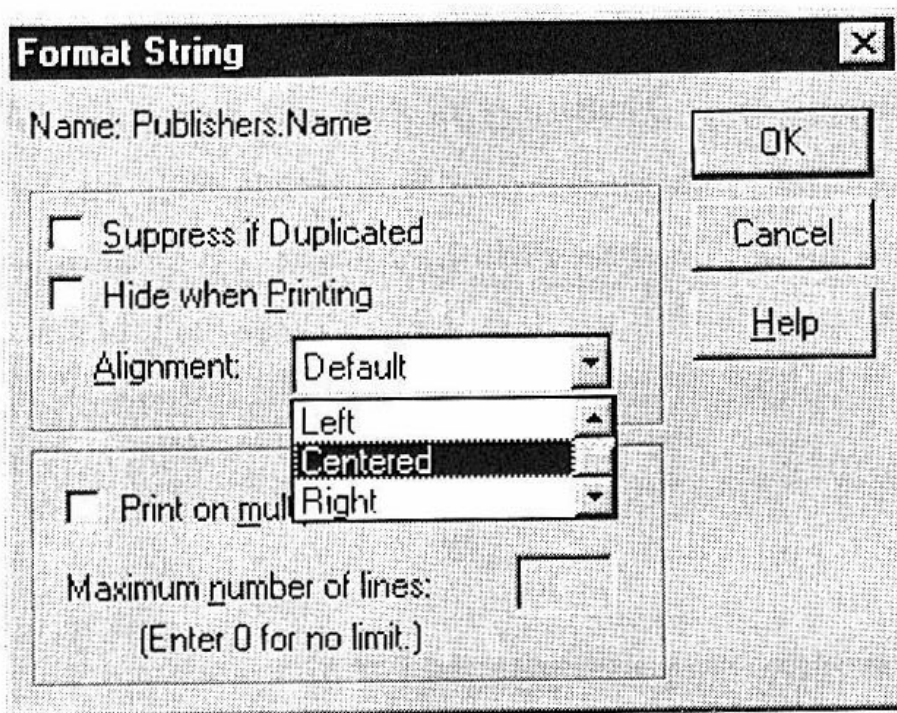
در قسمت Report Fields تمام فیلدهای استفاده شده در گزارش قرار دارد و با انتخاب هریک از آنها و کلیک کردن روی دکمه فرمان Add می‌توانید به ترتیب دلخواه آنها را مرتب کنید. همان طور که می‌دانید مرتب کردن یک فیلد به دو صورت می‌تواند انجام شود که یکی صعودی و دیگری نزولی است. پس از اینکه فیلد مناسب در قسمت Sort Field قرار گرفت قسمت Sort Direction در پایین پنجره فعال می‌شود که Ascending معرف صعودی و Descending معرف نزولی بودن ترتیب مرتب کردن می‌باشد. بعد از اینکه فیلدها را انتخاب کردید و نوع مرتب شدن آنها را هم مشخص ساختید، با زدن دکمه فرمان OK کار تمام است.

نکته: در صورتی که بخواهید عمل مرتب سازی را روی فیلدی انجام دهید که در گزارش آورده نشده است، باید ابتدا آن را در گزارش وارد کنید و عمل مرتب سازی را روی آن انجام دهید و به روشی که در قسمت بعدی توضیح می‌دهیم آن را در چاپ پنهان سازید.

نحوه تعیین قالب برای فیلد متن

در صورتی که در قسمت طراحی روی هریک از متن‌ها یا فیلدهای متنی دوبار کلیک کنید پنجره‌ای همانند آنچه در شکل ۸-۱۱ نشان داده شده است، ظاهر می‌شود که از طریق آن می‌توانید قالب نمایش فیلد را تعیین کنید.

شکل ۸-۱۱.

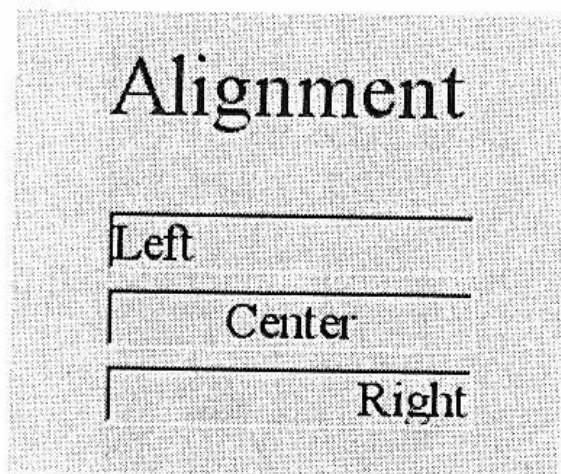


این پنجره دارای چند انتخاب می‌باشد که در ادامه آنها را توضیح می‌دهیم.

- **Suppress if Duplicated:** اگر این گزینه انتخاب شده باشد در صورتی که مقدار این فیلد در رکوردهای قبلی تکرار شده باشد از چاپ آن جلوگیری می‌شود.

• Hide When Printig: در صورتی که این گزینه انتخاب شده باشد، این فیلد در چاپ پنهان می گردد.

• Alignment: وضعیت قرار گرفتن داده‌های فیلد یا متن را تعیین می‌کند که از چهار قسمت پیش‌گزینه، چپ‌چین، راست‌چین و وسط‌چین تشکیل شده است. شکل زیر سه وضعیت مختلف را نشان می‌دهد:



شکل ۹-۱۱.

در صورت تنظیم این قسمت به Default، حالت پیش‌گزینه انتخاب می‌شود.

• Print On Multiple Lines: اگر این گزینه را فعال نمایید می‌توانید تعیین کنید در صورتی که اندازه فیلد شما در مقدار تعیین‌شده برای آن بیشتر شد تا چند خط به ردیف آن رکورد اضافه شود، در شکل زیر در ردیف 586 چون داده فیلد از اندازه آن در یک خط بیشتر شده است کلمات اضافه به خط‌های بعد منتقل شده است.

68	TAB BOOKS
478	TAB BOOKS SOFTWARE
529	TAB PROFESSIONAL & REFERENCE
668	TACTICAL TECHNOLOGIES
550	TAYLOR & FRANCIS
349	TEC SPEC
586	TECHNICAL COMMUNICATIONS ASSOCS

شکل ۱۰-۱۱.

در صورتی که این تنظیم را انجام ندهید و مقدار داده فیلد از اندازه آن بیشتر شد، کلمات اضافی در چاپ گزارش حذف می‌شود.

تعیین قالب برای فیلد عددی

در صورتی که داده شما از نوع عددی باشد، در صورتی که دوبار روی آن کلیک کنید پنجره زیر برای تعیین قالب آن باز خواهد شد.

Format Number

Name: Count of Publishers.Name

Use Windows Default Format

Suppress if Duplicated

Suppress if Zero

Hide when Printing

Alignment: Default

Currency Symbol: \$

One Symbol Per Page

Fixed Floating

Position: \$123

Decimals: 1

Rounding: None

Negatives:

Decimal Separator: .

Thousands Separator: .

Leading Zero

Sample: 555,555,555

OK

Cancel

Help

شکل ۱۱-۱۱.

در این پنجره علاوه بر آنچه در قالب متن داشتیم چند قسمت نیز اضافه شده است در ادامه این قسمتها را بررسی می کنیم .

- Use Windows Default Format : در صورت انتخاب این قسمت ، از قالب پیش گزینه ویندوز برای عدد استفاده می شود .

- Currency Symbol : در صورتی که داده عددی شما از نوع Currency (فیلدهای مبلغ و قیمت) باشد می توانید علامتی را که به آن اضافه می شود ، در این قسمت تعریف کنید و موقعیت آن را نسبت به آن تعیین نمایید . پیش گزینه این قسمت علامت \$ است و مکان قرارگیری آن قبل از عدد می باشد .

شما می توانید علامت \$ را به (ریال) تبدیل کنید و مکان آن را قبل از عدد قرار دهید . در این صورت مقدار عددی شما به صورت 555 (ریال) چاپ می گردد . قسمت های بعد مربوط به تعیین تعداد اعشار عدد و علامت اعشار می باشد .

تعیین اندازه کاغذ

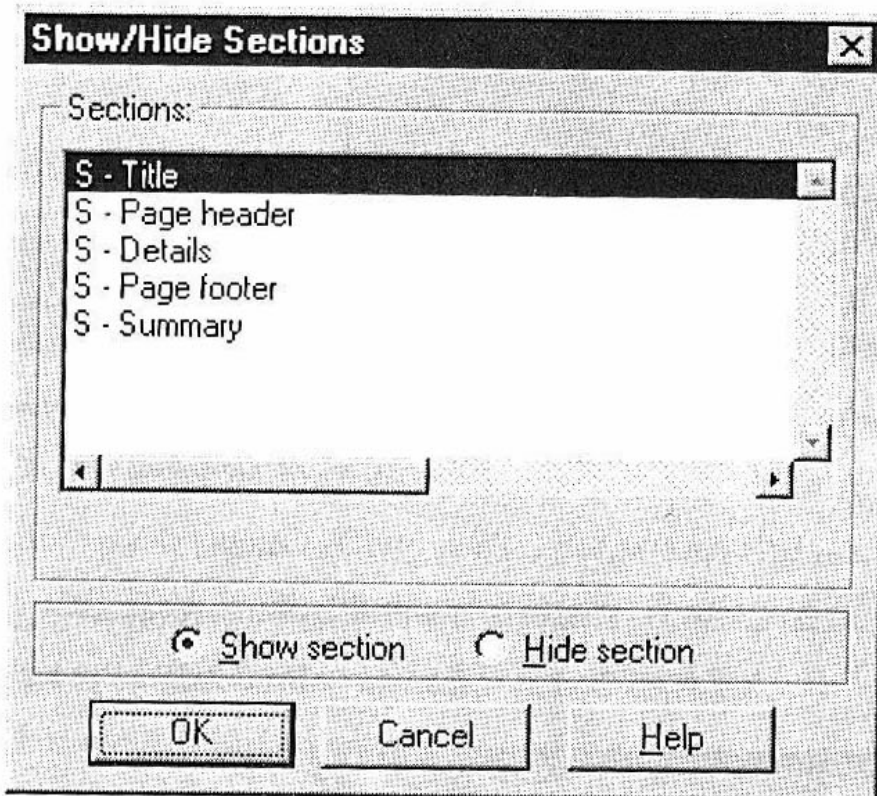
اگر بخواهید اندازه کاغذ خود را تغییر دهید و یا حالت چاپ آن را تعیین نمایید از منوی فایل گزینه Print Setup را انتخاب کنید. این پنجره همان پنجره استاندارد چاپگر شماست و در قسمت Orientation، Source و Size می‌توانید حالت‌های مختلف موردنظر خود را تنظیم کنید.

نکته

تنظیماتی که در این قسمت اعمال می‌کنید در فایل rpt ذخیره می‌شود و هنگام استفاده از آن در ویژوال بیسیک، این خواص حفظ خواهد شد.

وضعیت نمایش هر قسمت در چاپ

شما می‌توانید هر یک از بخش‌های گزارش خود نظیر Page Header، Page Footer، Title و غیره را در چاپ فعال یا غیر فعال کنید. برای این عمل می‌توانید از منوی Edit گزینه Show / Hide Section را انتخاب نمایید.



شکل ۱۲-۱۱.

با انتخاب هر بخش می‌توانید وضعیت نمایش آن را فعال یا غیر فعال کنید.

گروه‌بندی یک فیلد

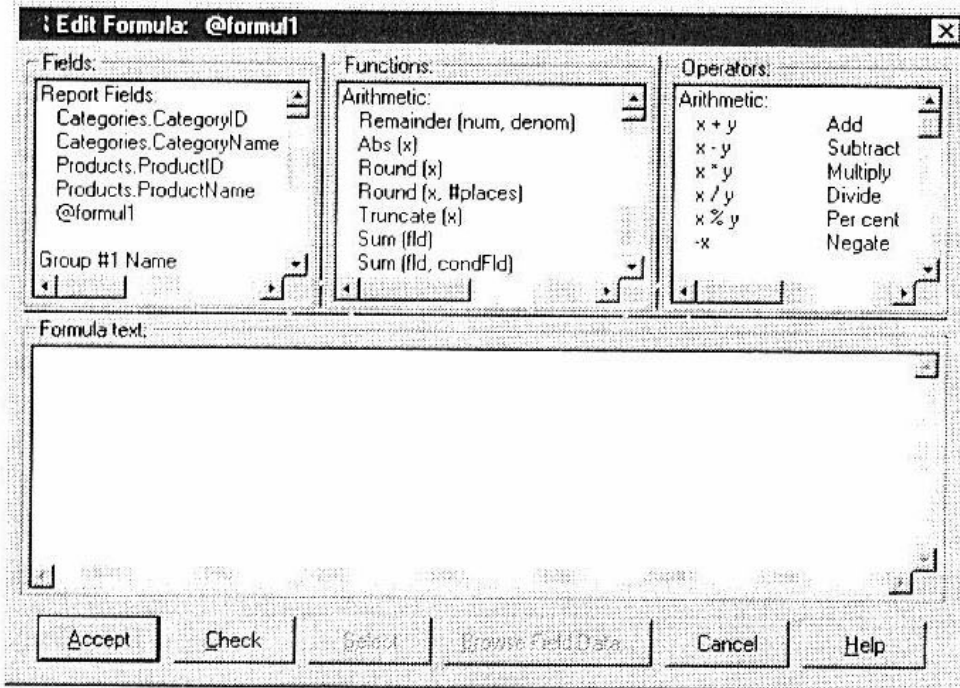
هنگامی که یک فیلد را گروه‌بندی می‌کنید، تمام داده‌هایی که از آن فیلد در رکوردهای مختلف قرار دارند و دارای مقدار مساوی هستند، در یک گروه قرار می‌گیرند و می‌توانید یکسری عملیات روی این گروه انجام دهید. مثلاً تعداد اعضای گروه را تعیین کنید، Min و Max عناصر را بیابید و یا جمع

۷ - قسمت Perview را برای پیش‌نمایش چاپ انتخاب کنید.

همان طور که در پیش‌نمایش چاپ مشاهده کردید، پس از قرار گرفتن هر فیلد در CategoryID اعضای گروه قرار گرفته‌اند.

فیلدهای فرمول

برای ایجاد این فیلدها باید از منوی Insert گزینه FormulaField را انتخاب کنیم و پس از وارد کردن نام فرمول پنجره شکل ۱۵-۱۱ باز می‌شود.



شکل ۱۵-۱۱.

شما می‌توانید از طریق این قسمت فیلدهای مختلف را با توابع مختلف و اوپراتورهای ریاضی ترکیب نمایید و نتیجه را به صورت یک فیلد در گزارش ظاهر کنید. مثلاً می‌خواهید در یک گزارش فروش، قیمت کالا در تعداد کالا ضرب گردد و به صورت فیلد مبلغ چاپ شود. برای این منظور از قسمت Fields، فیلد قیمت کالا را انتخاب و پس از وارد کردن علامت *، فیلد تعداد را وارد کنید و سپس دکمه فرمان Accept را انتخاب می‌کنیم و فیلد ایجاد شده را در قسمت مناسب گزارش قرار دهید.

در صورتی که نوع متغیرها در فیلد درست باشد با خطایی رو به رو نمی‌شویم، ولی اگر مثلاً یک فیلد متن را در فیلد عددی ضرب کنیم با خطا رو به رو می‌شویم. دکمه فرمان Check در پنجره فرمول، مقادیر فرمول و نوع استفاده از توابع و اوپراتورها را چک می‌کند و در صورت وجود خطا پیغام خطا می‌دهد.

کنترل گزارش در ویژوال بیسیک

کنترل گزارش ابزاری است که دسترسی شما را به فایل گزارش در ویژوال بیسیک امکان‌پذیر می‌کند و با آیکون 000131 نشان داده می‌شود. در صورتی که این ابزار در قسمت Toolbox و برنامه نباشند می‌توانید به ترتیب زیر عمل کنید.

۱ - از منوی Project زیر منوی Component را انتخاب کنید تا پنجره Componet باز گردد .

۲ - در قسمت Controls گزینه Crystal Report 4.6 را فعال نمایید و دکمه OK را انتخاب کنید .

از این طریق ویژوال بیسیک با فایل Crystal32.OCX ارتباط برقرار می کند و این ابزار را در ToolBox قرار می دهد .

امکانات اصلی کنترل گزارش در ویژوال بیسیک به سه قسمت تقسیم می شود :

• چاپ مستقیم یک گزارش روی چاپگر پیش گزینه : این عمل با تنظیم خصوصیت Destination به مقدار ۱ اجرا می شود .

• چاپ گزارش روی یک فایل : این عمل با تنظیم خصوصیت Dastination به مقدار 2 اجرا می شود. نوع فایل خروجی باید در خصوصیت Print File Type مشخص گردد که جدول ۵-۱۱، انواع این فایل های خروجی را نشان می دهد . پس از تنظیم نوع فایل باید در خصوصیت Print File Name نام فایل خروجی مشخص گردد تا گزارش به صورت فایل با نوع و نام و مسیر مشخص شده ذخیره گردد .

• چاپ گزارش در فرم به عنوان پنجره پیش نمایش چاپ : این عمل با تنظیم خصوصیت Diistination به 0 صورت می گیرد . شما می توانید قبل از چاپ صفحات مختلف گزارش را دیده و در صورت درست بودن آن را یا روی چاپگر یا روی فایل ارسال کنید.

جدول ۵-۱۱ : مقادیر خصوصیت Print File Type برای تعیین نوع فایل خروجی .

توضیح *	نوع فایل	Print File Type
این فایل متنی با طول ثابت می باشد که برای استفاده در سیستم های Client / Server و پایگاه های داده دستگاه های Main Fram مناسب است .	Record(ASC)	0
فیلدها به وسیله Tab از هم جدا می شوند و رکوردها به وسیله خط جدید از هم جدا می شوند ، یعنی هر خط معرف یک رکورد است	Tab-Separated (.TSV)	1
این گزینه همانند درایور چاپ روی فایل Generic / Text Only در ویندوز می باشد.	Text(.TXT)	2
این نوع مختلف Data Interchange Format است و برای تبدیل به صفحات گسترده که از DIF پشتیبانی می کنند ، به کار می رود .	DIF(.DIF)	3
فیلدها در این فایل در دو () قرار می گیرند و یک فیلد با کاما از فیلد دیگر جدا می شود .	Comma-Separated (.CSV)	4

توضیح	نوع فایل	Print File Type
فیلد در این فایل به وسیله کاراکتری دلخواه جدا می‌شود	Charaeter-Separate	5
فیلدها با کارکتر Tab و رکوردها با خط جدید از هم جدا شده‌اند. فیلدها دارای (") نمی‌باشند و قابل استفاده در Word برای ویندوز است.	Tab-Separatadtext (.TST)	6
این فایل قابل استفاده در خود برنامه Cryatal Report می‌باشد.	Crystal Report (.RPT)	7
این فایل برای استفاده در برنامه مایکروسافت اکسل نسخه ۲،۱ می‌باشد	Exeel 2.1 (.XLS)	8
این فایل برای استفاده در برنامه مایکروسافت اکسل نسخه ۳،۰ می‌باشد.	Excel 3.0 (.XLS)	9
این فایل برای استفاده در برنامه مایکروسافت اکسی نسخه ۴ می‌باشد.	Excel 4.0 (.XLS)	10
این فایلها برای برنامه Lotus 123 می‌باشد.	Lotus 123 WK1	11
	Lotus 123 WK3	12
	Lotus 123 WK5	13
این فایل برای برنامه‌هایی مانند Word است که از پسوند RTF پشتیبانی می‌کند.	Rich Text File (. RTF)	15
این فایل مخصوص برنامه مایکروسافت Word تحت ویندوز است.	Win Word (.DOC)	17
این فایل برای برنامه مایکروسافت اکسل ۵ می‌باشد.	Excel 50	19
این فایل به صورت HTML نسخه 30 ذخیره می‌گردد.	HTML 3.0	20
این فایل یک HTML برای برنامه Internet Explorer می‌باشد.	Internet Explorer	21
این فایل یک HTML برای برنامه Netscape Navigator می‌باشد.	Netscape	22

در ادامه تکنیکهای کدنویسی را برای تهیه گزارش بررسی می‌کنیم.

چاپ یک گزارش به وسیله کنترل سفارشی گزارش

چاپ یک نسخه‌ای یک گزارش روی چاپگر پیش گزینه نیازی به نمایش فرم به کاربر ندارد، اما اگر خواسته باشید کاربر عملیات چاپ را گسترده‌تر و با امکانات بیشتر انجام دهد، می‌توانید از یک فرم استفاده کنید. با اجرای کدی که در لیست ۲-۱۱ نشان داده شده است، فرم تبدیلی چاپ به کاربر نشان داده می‌شود و کاربر می‌تواند تعداد نسخه‌های چاپ و تنظیمات دیگر را انجام دهد و با زدن OK چاپگر شروع به چاپ می‌کند. در صورتی که دکمه Cancel کلیک شود عملیات چاپ قطع می‌شود و پیغامی نمایش داده می‌شود.

لیست ۲-۱۱:

```
Private Sub mnuReportsPrintProdByCat_Click()
    On Error GoTo PrintReportError
    With frmMIDReports
        .Show
        .Cls
        .Caption = "Printing Products by Category the window"
        ' Send Report to Printer
        .rptReports.Destination = 1
        .cdgPrint.CancelError = True
        .cdgPrint.Copies = 1
        .cdgPrint.FromPage = 1
        .cdgPrint.ToPage = 1
        .cdgPrint.Flags = cdIPDPrintSetup
        .cdgPrint.ShowPrinter
        DoEvents
        If .cdgPrint.PrintDefault Then
            .rptReports.copiesToPrinter = .cdgPrint.Copies
            .cdgPrint.Action = 1
        Else
            MsgBox prompt:="The printer you use must be" & _
                "selected as the default printer" , _
                buttons:=vbCritical
            GoTo BailOut
        End If
    End With
    DoEvents 'Let the Printer occur
    GoTo BailOut
PrintReportError:
    MsgBox prompt:=Err.Description, _
        buttons:=vbCritical, _
        Title:="Report Printing Error"
```

BailOut:

```
frmMIDGraph.Show  
Unload frmMIDReports  
DoEvents  
End Sub
```

چاپ یک گزارش در یک فایل

کد لیست ۱۱-۳ کادر تبادلی مشخص کردن فایل با فرمت دلخواه را به کاربر نشان می‌دهد (شکل ۱۱-۱۴) و پس از مشخص شدن فایل، گزارش را در فایل ایجاد می‌کند.
لیست ۱۱-۳:

```
Private Sub mnuReportsFileProdByCat_Click()  
    Dim strFilter As String  
    strFilter = "Record (*.asc) | *.asc | Tab-separated (*.tsv) | *.tsv |" & _  
                "Text (*.txt) | *.txt | Spreadsheet (*.dif) | *.dif |" & _  
                "Comma-separated (*.csv) | *.csv | (Reserved) | *.???" & _  
                "Tabbed Text (*.tst) | *.tst"  
  
    On Error GoTo PrintFileError  
    With frmMIDReports  
        .Show  
        .Caption = "Printing Products by Category Report to File"  
        end Report to Printer  
        .rptReports.Destination = 2  
        .cdgPrint.CancelError = True  
        .cdgPrint.DialogTitle = "Save Report to File"  
        .cdgPrint.Filter = strFilter  
        .cdgPrint.FilterIndex = 5  
        .cdgPrint.ShowSave  
        DoEvents  
        Select Case LCase(Right(.cdgPrint.filename, 3))  
            Case "asc"  
                .rptReports.PrinterFileType = 0  
            Case "tsv"  
                .rptReports.PrinterFileType = 1  
            Case "txt"  
                .rptReports.PrinterFileType = 2  
            Case "dif"  
                .rptReports.PrinterFileType = 3  
            Case "csv"  
                .rptReports.PrinterFileType = 4  
            Case "tst"  
                .rptReports.PrinterFileType = 6  
        End Select  
    End With  
End Sub
```

Case Else

.rptReports.PrinterFileType = 5

End Select

If Len(.cdgPrint.filename) > 5 And _

(cdgPrint.PrintFileType <> 5) Then

.rptReports.PrintFileName = .cdgPrint.filename

.rptReports.Action = 2

Else

MsgBox prompt:="Invalid file name" & _

.cdgPrint.filename & "' or type. ", _

button:=vbCritical, Title:="Print to File Error"

GoTo BailOut

End If

End With

DoEvents 'Let the Printer occur

GoTo BailOut

PrinterFileError:

MsgBox prompt:=Err.Description , _

buttons:=vbCritical , _

Title:="Report Printing Error "

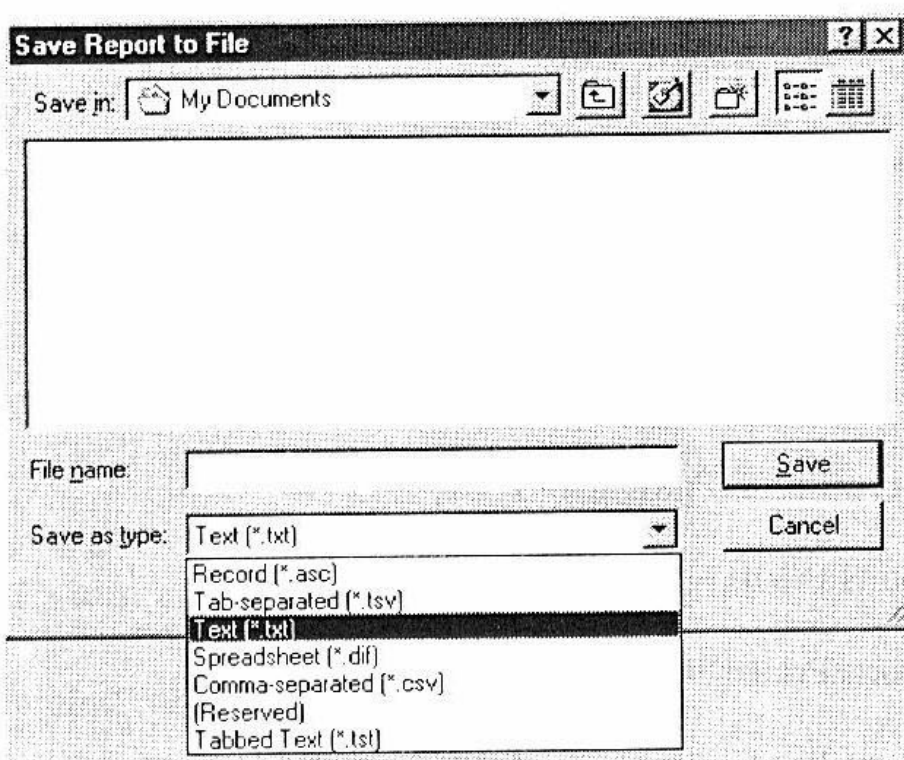
BailOut:

frmMIDGraph.Show

Unload frmMIDReports

DoEvents

End Sub



شکل ۱۱-۱۴




ایجاد پنجرهٔ پیش‌نمایش در فرم MDI Child

کد زیر یک پنجره به صورت MDI Child ایجاد می‌کند که در داخل پنجرهٔ برنامهٔ اصلی قرار می‌گیرد و پیش‌نمایش گزارش در آن دیده می‌شود. از طریق دکمه‌های فرمان پایین این پنجره می‌توان گزارش را کوچک و بزرگ نمود و صفحات مختلف آن را مرور کرد و سپس در صورت لزوم آن را در چاپگرها روی فایل چاپ کرد.

لیست ۴-۱۱:

```
Sub ShowPreviewWindow(strCaption As String)
    Dim sgnTPP As String
    Dim intWidthT As Integer
    Dim intWidthP As Integer
    Screen.MousePointer = 11
    UnloadReportsForm
    frmMDIReports.Show
    frmMDIReports.rptReports.Destination = 0
    frmMDIReports.rptReports.WindowParentHandle = frmMDIReports.hWnd
    frmMDIReports.rptReports.WindowTop = 0
    frmMDIReports.rptReports.WindowLeft = 0
    If frmMDIReports.WindowState = 0 Then
        frmMDIReports.Left = 0
        frmMDIReports.Top = 0
        frmMDIReports.Height = frmMDIToolbar.ScaleHeight
        frmMDIReports.Width = frmMDIToolbar.ScaleWidth
    End If
    intWidthT = frmMDIReports.ScaleWidth
    frmMDIReports.ScaleMode = 1
    sgnTPP = intWidthT / intWidthP
    frmMDIReports.rptReports.WindowWidth = _
        frmMDIReports.ScaleWidth /
sgnTPP
    frmMDIReports.rptReports.WindowHeight = _
        frmMDIReports.ScaleHeight /
sgnTPP
    frmMDIReports.Caption = "Preview " & strCaption & "Report "
    frmMDIReports.rptReports.WindowTitle = "Print Preview "
    DoEvents
    frmMDIReports.rptReports.Action = 1
    Screen.MousePointer = 0
End Sub
```

Category ID	Category Name	Product ID	Product Name	Quantity Per Unit
1	Beverages	1	Chai	10 boxes x 20 bags
		2	Chang	24 - 12 oz bottles
		24	Guaraná Fantástica	12 - 355 ml cans
		34	Sasquatch Ale	24 - 12 oz bottles
		35	Steeleye Stout	24 - 12 oz bottles
		38	Côte de Blaye	12 - 75 cl bottles
		39	Chartrreuse verte	750 cc per bottle
		43	Ippoh Coffee	16 - 500 g tins
		67	Laughing Lumberjack Lager	24 - 12 oz bottles
		70	Outback Lager	24 - 355 ml bottles
		75	Rhönbräu Klosterbier	24 - 0.5 l bottles
		76	Lakkalikööri	500 ml

به وسیله آیکون  می‌توان بزرگ‌نمایی را روی پیش‌نمایش انجام داد و جزئیات و کلیات صفحه پیش‌نمایش را دید. در صورتی که تعداد صفحات گزارش بیش از یک صفحه باشد به کمک دکمه‌های طرفین و اولین و آخرین می‌توان صفحات مختلف گزارش را مرور کرد. در صورتی که مایل به چاپ گزارش باشید می‌توانید با انتخاب آیکن  گزارش را چاپ کنید. برای ارسال گزارش به فایل با کلیک کردن روی  می‌توانید نوع فایل و نام آن را مشخص کنید و چاپ را روی آن انجام دهید. در صورتی که این پنجره را Cancel کنید هیچ عملیاتی روی گزارش صورت نخواهد گرفت.

Category ID	Category Name	Product ID	Product Name	Quantity Per Unit
1	Beverages	1	Chai	10 boxes x 20 bags
		2	Chang	24 - 12 oz bottles
		24	Guaraná Fantástica	12 - 355 ml cans
		34	Sasquatch Ale	24 - 12 oz bottles
		35	Steeleye Stout	24 - 12 oz bottles
		38	Côte de Blaye	12 - 75 cl bottles
		39	Chartrreuse verte	750 cc per bottle
		43	Ippoh Coffee	16 - 500 g tins
		67	Laughing Lumberjack Lager	24 - 12 oz bottles
		70	Outback Lager	24 - 355 ml bottles
		75	Rhönbräu Klosterbier	24 - 0.5 l bottles
		76	Lakkalikööri	500 ml

محافظت از پایگاه داده‌ها در ویژوال بیسیک

سیستم حفاظتی یک پایگاه داده اکسس به دو دسته تقسیم می‌شود. دسته ای از خصوصیات در خود پایگاه داده و دسته دیگر در فایلی به نام System.MDW و یا در نسخه های قبلی Jet تحت نام System.MDA طبقه بندی و نگهداری می‌شوند. فایل حفاظت از پایگاه داده را نمی‌توان در ویژوال بیسیک ایجاد کرد بلکه باید عملیات ایجاد این فایل را در اکسس انجام داد و سپس در ویژوال بیسیک از آن استفاده نمود. پس از ایجاد فایل حفاظت از پایگاه داده، به وسیله دو شیء User و Group و مجموعه های آنها در ویژوال بیسیک می‌توان کاربران و گروههای کاربران را ایجاد و محدودیتهای لازم را برای استفاده از اجزای پایگاه داده تعریف کرد. اما پیشنهاد ما برای این کار، استفاده از خود اکسس است زیرا توسط محیط مخصوص خود اکسس جهت این اعمال می‌توان به راحتی گروهها و کاربران را تعریف کرد و در صورتی که در سیستم، نیاز به ایجاد کاربر جدید بود می‌توان توسط شیء User، کاربر جدید را تحت یک گروه در ویژوال بیسیک ایجاد نمود.

با توجه به موارد بالا پس از اینکه سیستم حفاظتی خود را در اکسس ایجاد کردید برای تنظیم این سیستم حفاظت و استفاده از پایگاه داده حفاظت شده در ویژوال بیسیک بایستی مراحل زیر را انجام داد :

- مکان قرار گرفتن فایل حفاظت خود را در سیستم تعیین کنید.
 - خصوصیت SystemDB و یا IniPath را برای شیء DBEngine تعریف کنید.
 - در صورتی که بخواهید یک کاربر با کلمه عبور مخصوص در تمام برنامه تعریف شده باشد می‌توانید دو خصوصیت DefaultUser و DefaultPassword را برای DBEngine در ابتدای برنامه تعریف کنید.
 - در صورتی که بخواهید به طور موضعی از اجزای پایگاه داده استفاده نمایید می‌توانید به وسیله متد CreateWorkspace یک محیط کاری جدید برای یک کاربر و کلمه عبور آن ایجاد کنید.
- مشخص کردن فایل حفاظت برای DBEngine :

در صورتی که از ویژوال بیسیک ۴ تحت محیط ۳۲ بیتی و نسخه های بعد از آن استفاده می‌کنید، شیء DBEngine دارای خصوصیتی به نام SystemDB می‌باشد که به وسیله آن می‌توان فایل حفاظت را برای DBEngine تعریف نمود. دستور زیر این عمل را نشان می‌دهد.

```
DBEngine.SystemDB = "C:\Windows\System\System.MDW"
```

و در صورتی که از ویژوال بیسیک در محیط ۱۶ بیتی استفاده می کنید باید برای تعریف فایل حفاظت به ترتیب زیر عمل کنید :

۱. یک فایل با نام دلخواه و پسوند INI در سیستم ایجاد و اطلاعات زیر را در آن وارد کنید :

[Options]

SystemDB=C:\Windows\System\System.MDA

۲. سپس در برنامه توسط خصوصیت IniPath مسیر این فایل را مشخص نمایید :

DBEngine.IniPath="C:\Windows\IniFile.INI"

مکان تعریف SystemDB و یا IniPath در برنامه معمولاً در رویداد Load در اولین فرم می باشد زیرا با یک بار تعریف در ابتدای برنامه، این خصوصیات تا پایان برنامه باقی خواهد ماند.

تعریف یک کاربر مشخص در تمام طول کار در برنامه

در صورتی که بخواهید یک کاربر مشخص را با یک کلمه عبور مخصوص در تمام طول برنامه به صورت پیش فرض برای DBEngine تعریف کنید می توانید از دو خصوصیت DefaultUser و DefaultPassword استفاده نمایید. شما می توانید مقدار این دو خصوصیت را در ابتدای برنامه به وسیله پنجره ای که اصطلاحاً Logon نام دارد از کاربر گرفته و سپس آنها را تنظیم نمایید. دستورات زیر نمونه استفاده از این خصوصیات را نشان می دهد.

DBEngine.DefaultUser = "UserName"

DBEngine.DefaultPassword = "Password"

توجه داشته باشید که این دو خصوصیت باید پس از مشخص شدن فایل حفاظت در برنامه مورد استفاده قرار گیرد.

استفاده از CreateWorkspace

یکی دیگر از روشهای دسترسی به پایگاه داده حفاظت شده، استفاده از محیط کاری است. به این ترتیب که به وسیله متد CreatWorkspace یک شیء محیط کاری جدید برای یک کاربر با کلمه عبور مخصوص ایجاد می کنیم و تحت آن محیط کاری عملیات دلخواه را روی جداول و سؤالهای یک پایگاه داده انجام می دهیم.

شیء ایجاد شده توسط CreateWorkspace می تواند هم برای پایگاههای داده Jet و هم برای

ارتباطات ODBC مورد استفاده قرار گیرد. دستور کلی استفاده از این متد به صورت زیر است.

Set WorkspaceVar = CreateWorkspace(Name,User,Password,Type)

جدول ۱-۱۲ هر یک از قسمتهای این دستور را توضیح می دهد.

WorkspaceVar	یک متغیر از نوع Workspace می باشد که به وسیله دستور Dim ابتدا تعریف می شود.
Name	یک نام از نوع String می باشد که به شیء Workspace اختصاص داده می شود.
User	نام کاربر را در محیط کاری مشخص می نماید.
Password	کلمه عبور کاربر را در محیط کاری مشخص می کند.
Type	یک ثابت می باشد که نوع محیط کاری را تعیین می کند. جدول ۲-۱۲ حالت های مختلف این پارامتر را نشان می دهد.

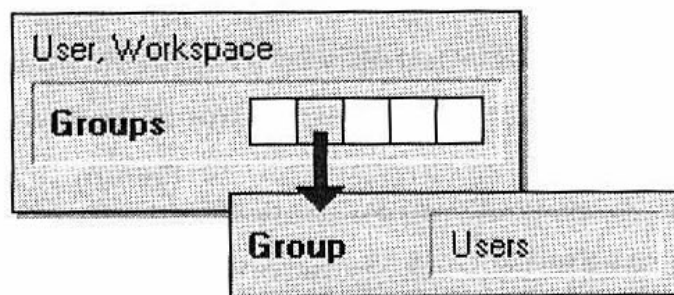
جدول ۲-۱۲:

DbUseJet	محیط کاری ایجاد شده برای استفاده در Jet می باشد.
DbUseODBC	محیط کاری ایجاد شده برای استفاده در ODBC می باشد.

در صورتی که بخواهید این شیء ایجاد شده را از مجموعه Workspace حذف کنید باید از متد Close استفاده نمایید.

شیء Group و مجموعه Groups

یک شیء Group نشان دهنده یک دسته از مجوزهای کاربران است که دسترسی هر کاربر را به پایگاه داده کنترل می کند و این عملیات تحت یک محیط کاری صورت می گیرد. مجموعه Groups شامل تمام شیء های Group است که تحت یک محیط کاری تعریف شده باشد.



شکل ۱-۱۲.

شما ابتدا باید اشیای گروه را ایجاد کنید و سپس با استفاده از نام آنها مجوزهای هر گروه را مشخص نمایید.

نکته: در صورتی که فایل حفاظت System.MD? را فعال نکرده باشید امکان تخصیص مجوز را نخواهید داشت.

شما می توانید یک شیء Group را به مجموعه Groups در شیء User اضافه کنید تا تمام مجوزهای آن کاربر به گروه منتقل شود و نیز می توان یک شیء User را به مجموعه Users در یک شیء Group اضافه کرد تا آن کاربر عضو گروه مورد نظر در آید.

با استفاده از خصوصیات شیء Group می توانید:

- به وسیله خصوصیت Name نام یک Group را تعیین کنید.

- با تنظیم خصوصیات Name و PID (Personal Identity) برای یک گروه جدید می توان یک گروه شخصی ایجاد نمود. PID کدی است که باعث می شود دو گروه همنام از یکدیگر تمایز داشته باشند و با شبیه سازی نام یک گروه در یک سیستم حفاظت دیگر دسترسی به اجزای پایگاه داده در خطر نیفتد.

Jet DBEngine دارای سه گروه از پیش تعریف شده است که Admins، Users و Guests نام دارد. به وسیله دستور زیر می توان به گروههای ایجاد شده از طریق نام آنها دسترسی داشت:

Groups(

برای ایجاد یک گروه جدید باید از متد CreateGroup تحت شیء User و یا Workspace استفاده نمود. شیء Group و مجموعه Groups مجموعاً دارای ~ خصوصیت می باشد که در جدول ۱۲-۳ نشان داده شده است.

جدول ۱۲-۳: خصوصیت شیء Group

توضیح	خصوصیت
این خصوصیت تعداد اعضای مجموعه Groups را مشخص می کند.	Count
نام یک گروه را تنظیم و یا تعیین می کند.	Name
توسط این خصوصیت می توان PID یک گروه را تنظیم نمود.	PID
نکته: توجه داشته باشید که توسط این خصوصیت نمی توان PID یک گروه را تعیین کرد.	

متدهای شیء Group و مجموعه Groups در جدول ۱۲-۴ نشان داده شده است.

جدول ۱۲-۴: متدهای شیء Group

توضیح	متد
یک شیء Group ایجاد شده را به مجموعه Groups اضافه می کند.	Append
یک شیء User جدید را تحت شیء Group و یا شیء Workspace ایجاد می نماید.	CreateUser

مثال زیر یک نمونه استفاده از این متد را نشان می دهد:

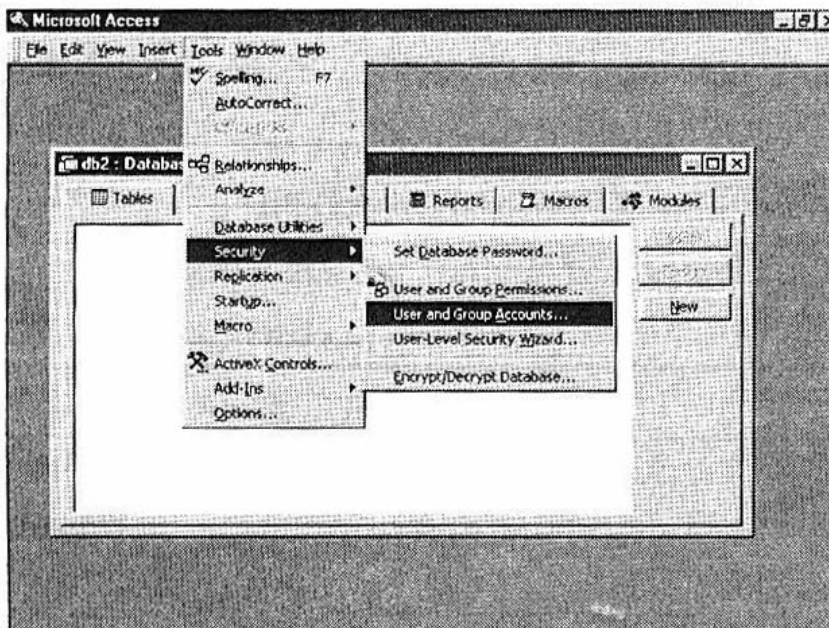
`Dim MyUser As User`

```
Dim ManagersPID As String
ManagersPID = "1122aabb"
Set MyUser = Workspaces(0).CreateUser("Senior Partner" ,
ManagersPID)
MyUser.Password = "NewPassword"
Workspaces(0).Users.Append MyUser
MyUser.Groups.Append MyUser.CreateGroup("Users")
```

یک شیء Group را از مجموعه Groups حذف می کند. Delete
اشیاء مجموعه Groups را به روز می رسانی. Refresh

شیء User و مجموعه Users

شیء User مجوز یک کاربر را نشان می دهد که برای دسترسی محدود به اجزای پایگاه داده تحت یک محیط کاری محدود شده، ایجاد شده است.



شکل ۲-۱۲

مجموعه Users شامل اشیای User ذخیره شده در یک گروه کاری می باشد. شما یک شیء User را ایجاد می کنید تا به وسیله نام آن بتوانید به پایگاه حفاظت شده دسترسی داشته باشید. همچنین با خصوصیات که برای یک شیء User تعریف شده است می توانید یک شیء محیط کاری ایجاد کنید و همان دسترسیها را به اجزای پایگاه داده داشته باشید. (ایجاد چنین محیط کاری ای در ابتدای فصل بررسی شد).

شما می توانید یک شیء User را به مجموعه Users تحت شیء group اضافه کنید تا مجوزهای گروه به User داده شود. همچنین با اضافه کردن یک شیء Group به مجموعه Groups تحت یک شیء User مجوزهای یک کاربر را به آن گروه منتقل کنید.

توسط خصوصیات شیء User می توانید :

- به وسیله خصوصیت Name نام کاربر را تنظیم و یا تعیین نمایید.
 - به وسیله تنظیم خصوصیات Name, PID و Password برای یک شیء User جدید، می توان یک کاربر شخصی ایجاد نمود. در صورتی که خصوصیت Password را تنظیم نکنید این خصوصیت به یک متغیر تهی () تبدیل می شود.
- Jet DBEngine دارای دو کاربر از پیش تعریف شده می باشد که Admin و Guest نام دارد. کاربر Admin عضو دو گروه Admins و Users می باشد و کاربر Guest فقط عضو گروه Guest می باشد. برای ایجاد یک شیء جدید User می توان از متد CreateUser استفاده نمود.
- شیء User و مجموعه Users مجموعاً دارای چهار خصوصیت می باشند که در جدول ۴-۱۲ نشان داده شده اند.

جدول ۴-۱۲ : خصوصیات شیء User

خصوصیت	توضیح
Count	این خصوصیت تعداد اعضای مجموعه Users را مشخص می کند.
Name	نام یک شیء User را تنظیم و یا تعیین می نماید.
Password	کلمه عبور شیء User را تنظیم می کند.
PID	PID (Personal Identity) یک شیء User را تنظیم می نماید.

متدهای شیء User و مجموعه Users در جدول ۵-۱۲ نشان داده شده اند.

جدول ۵-۱۲ : متدهای شیء User و مجموعه Users

خصوصیت	توضیح
Append	یک شیء User ایجاد شده را به مجموعه Users اضافه می کند.
CreateGroup	یک شیء Group ایجاد می نماید.
Delete	مثال زیر یک نمونه استفاده از این متد را نشان می دهد: Dim MyGroup As Group, ManagersPID As String ManagersPID = "2222abc" Set MyGroup = Workspaces(0).CreateGroup("Managers", ManagersPID) DBEngine.Workspaces(0).Groups.Append MyGroup
NewPassword	یک شیء User را از مجموعه Users حذف می کند. کلمه عبور یک کاربر را تغییر می دهد. این متد دارای دو پارامتر بوده و دستور کلی استفاده از آن به صورت زیر است. UserObject.NewOassword oldPassword,NewPassword
Refresh	اشیای مجموعه Users را به روز می رساند.

نکته بسیار مهم

همان طور که در شیء User دیدید، کاربری با نام Admin تحت گروه Admins و Users در هر محیط سیستم حفاظت Jet از پیش تعریف شده می باشد و نمی توان این کاربر و سایر کاربرهای از پیش تعریف شده را از مجموعه کاربران حذف نمود.

چون PID تمام این کاربران در تمام سیستمهای حفاظتی یکسان است و قابل حذف هم نمی باشد و چون این کاربر عضو گروه Admins می باشد و به تمام اجزای پایگاه داده دسترسی دارد. به طور مثال اگر یک پایگاه داده حفاظت شده با یک کاربر Admin با یک رمز مخصوص وجود داشته باشد در صورتی که این فایل به محیط دیگری منتقل شود که کاربر Admin در آن دارای یک رمز دیگری باشد و یا دارای رمز نباشد، دسترسی به تمام اجزا ممکن خواهد بود. بنابراین برای رفع این مشکل باید به ترتیب زیر در محیط اکسس عمل نمود.

۱. ابتدا کاربری با نام دلخواه برای گروه Admins ایجاد کنید.

۲. یک پایگاه داده جدید ایجاد نموده و از طریق Import Database تمام اجزای پایگاه داده را به آن منتقل کنید (این کار برای تغییر صاحب پایگاه داده از کاربر Admin به کاربر مورد نظران می باشد).

۳. حال این کاربر را عضو گروه Admins در آورید.

۴. کاربر Admin را از گروه Admins خارج سازید.

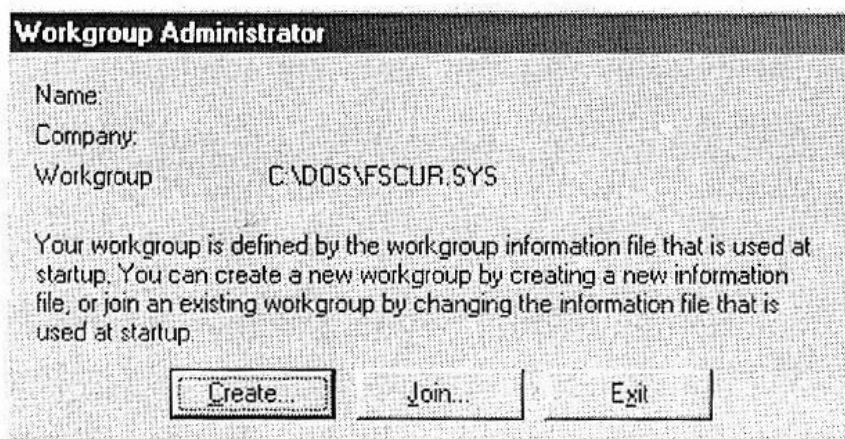
توجه داشته باشید که تا عضوی از مجموعه Users عضو گروه Admins نشده باشد نمی توان کاربر Admin را از گروه Admins خارج نمود.

۵. حال تمام مجوزهای کاربر Admin را گرفته و به کاربر ایجاد شده تمام مجوزها را اختصاص دهید. پس از انجام این عملیات پایگاههای حفاظت شده شما غیر قابل دسترسی برای سایر کاربران خواهد شد مگر اینکه مجوزهای لازم را به آن کاربران اختصاص دهید.

در ادامه این فصل، نحوه ایجاد فایل حفاظت، تعریف کاربران و گروههای کاربران و نیز تخصیص محدودیتها به گروهها و کاربران را از طریق Access 97 بررسی می کنیم.

ایجاد فایل محافظت

۱- از شاخه برنامه اکسس Workgroup Administrator را اجرا کنید.



شکل ۳-۱۲

۲- برای ایجاد یک فایل حفاظت جدید روی دکمه فرمان Create ، کلیک کنید تا پنجره Owner Information Workgroup نشان داده شود (شکل ۴-۱۲) .

Workgroup Owner Information

The new workgroup information file is identified by the name, organization, and case-sensitive workgroup ID you specify.

Use the name and organization information below, or enter a different name or organization. If you want to ensure that your workgroup is unique, also enter a unique workgroup ID of up to 20 numbers or letters.

Name: MyName

Organization: MyOrganization

Workgroup ID: MyWorkgroupID

OK Cancel

شکل ۴-۱۲

در این فرم سه فیلد قرار دارد که برای شناسایی یک فایل حفاظتی باید پر شود. دو فیلد Name و Organization ، فیلدهایی هستند که برای شناخته شدن فایل به کار می‌روند و می‌توان تعداد زیادی فایل حفاظت با Organization Name یکسان ایجاد کرد، اما آنچه که باعث تفکیک در فایل حفاظت می‌شود، فیلد Workgroup ID می‌باشد. این فیلد فقط در زمان ایجاد یک فایل حفاظت مقداره‌ی می‌شود و بعد از آن نمی‌توان آن را تعیین و یا تنظیم نمود . اهمیت این فیلد از آن جهت است که علاوه بر عامل تفکیک پذیری، باعث ایجاد امنیت برای مشابه سازی از فایل حفاظت نیز می‌باشد. این فیلد حداکثر دارای ۲۰ کاراکتر می‌باشد که شامل اعداد و حروف می‌شود.

برای مثال مقادیر نشان داده شده در شکل ۴-۱۲ در این فیلدها وارد شده است.

۳- دکمه OK را فشار دهید تا پنجره Workgroup Information File فعال شود(شکل ۵-۱۲).

Workgroup Information File

Use the path and name below, or enter a different path and name for the new workgroup information file.

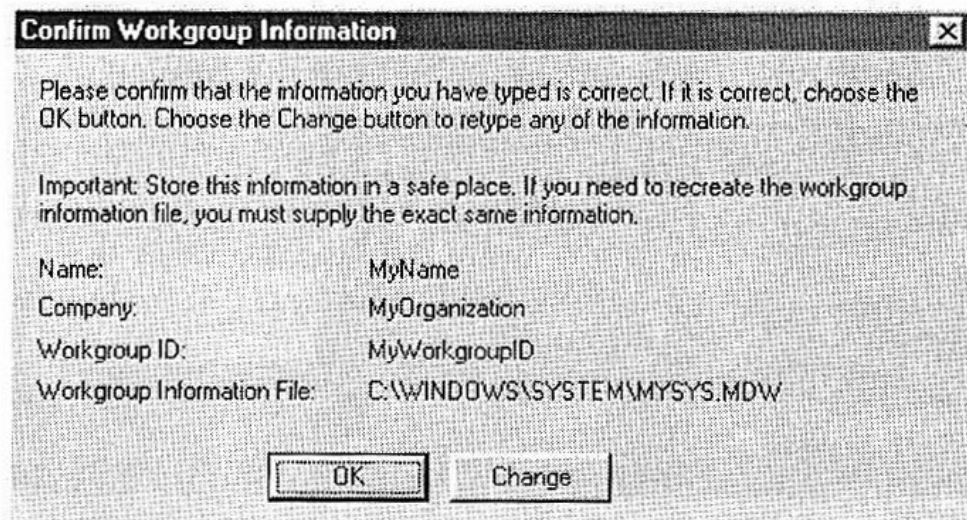
Database: C:\WINDOWS\SYSTEM\MYSYS.MDW Browse...

OK Cancel

شکل ۵-۱۲

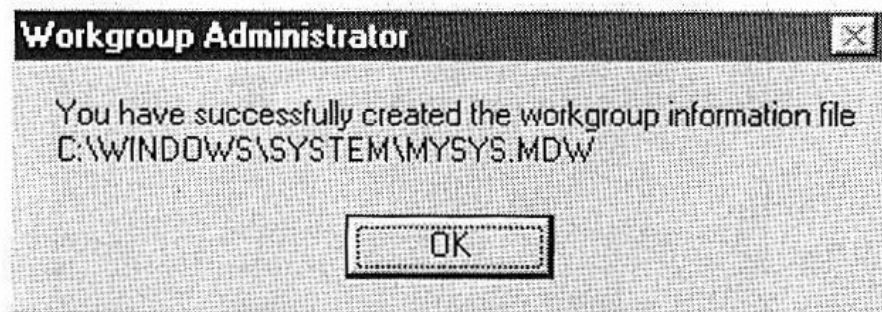
پس انجام مراحل قبل، حال باید مسیر و نام فایل حفاظت خود را در این پنجره وارد کنید و دکمه OK را فشار دهید.

۴- در این مرحله اطلاعات فایل حفاظت شما در پنجره Confirm Workgroup Information نشان داده می‌شود که می‌توانید توسط دکمه فرمان Change آنها را تغییر دهید، در غیر این صورت دکمه فرمان OK را فشار دهید.



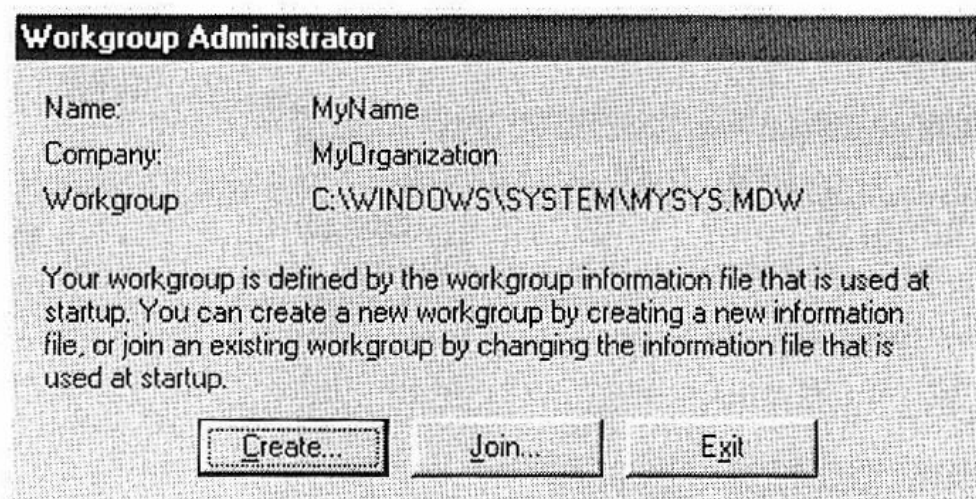
شکل ۶-۱۲

۵- در این مرحله فایل حفاظت شما به طور کامل ایجاد می‌گردد و در پنجره‌ای همانند شکل زیر، این امر نشان داده می‌شود.



شکل ۷-۱۲

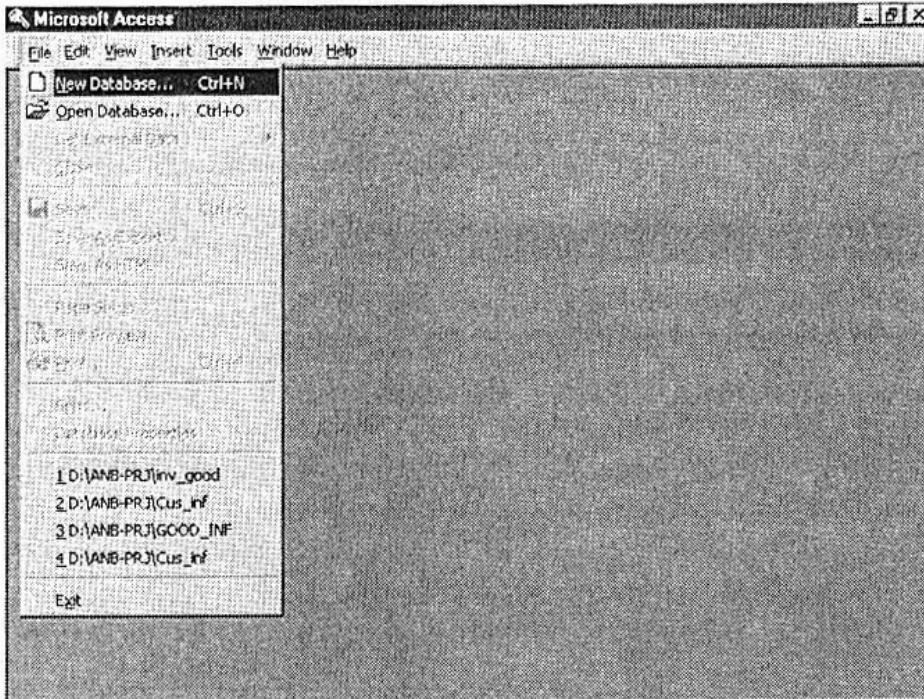
۶- حال می‌توانید OK را فشار دهید تا پنجره اولیه باز گردد و با کلیک کردن دکمه فرمان Exit عملیات تمام می‌شود.



شکل ۸-۱۲

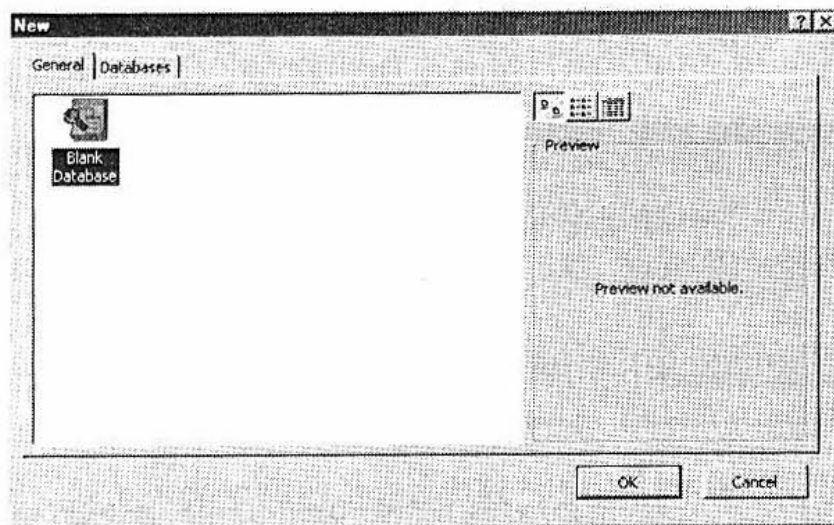
تعریف کاربران و گروههای کاربران

۱- ابتدا وارد برنامه اکسس شوید و منوی File New Database را اجرا کنید.



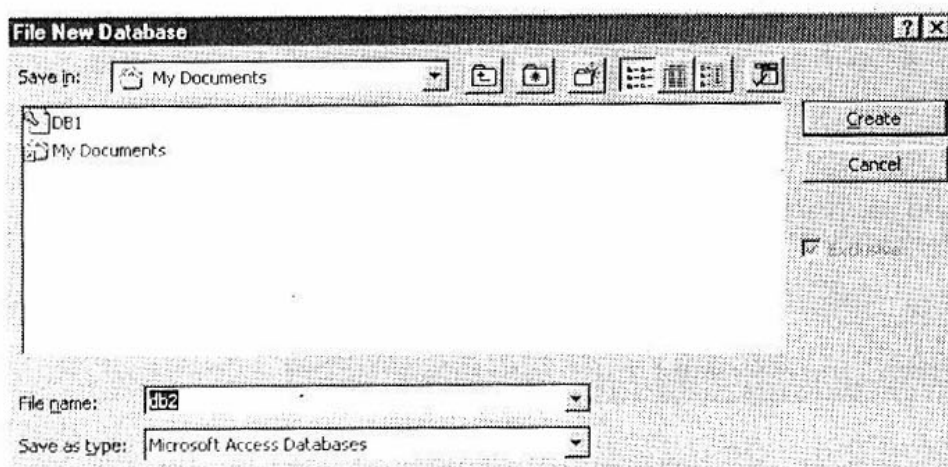
شکل ۹-۱۲

۲- در پنجره New روی آیکن Blank Database را دو بار کلیک کنید.



شکل ۱۰-۱۲

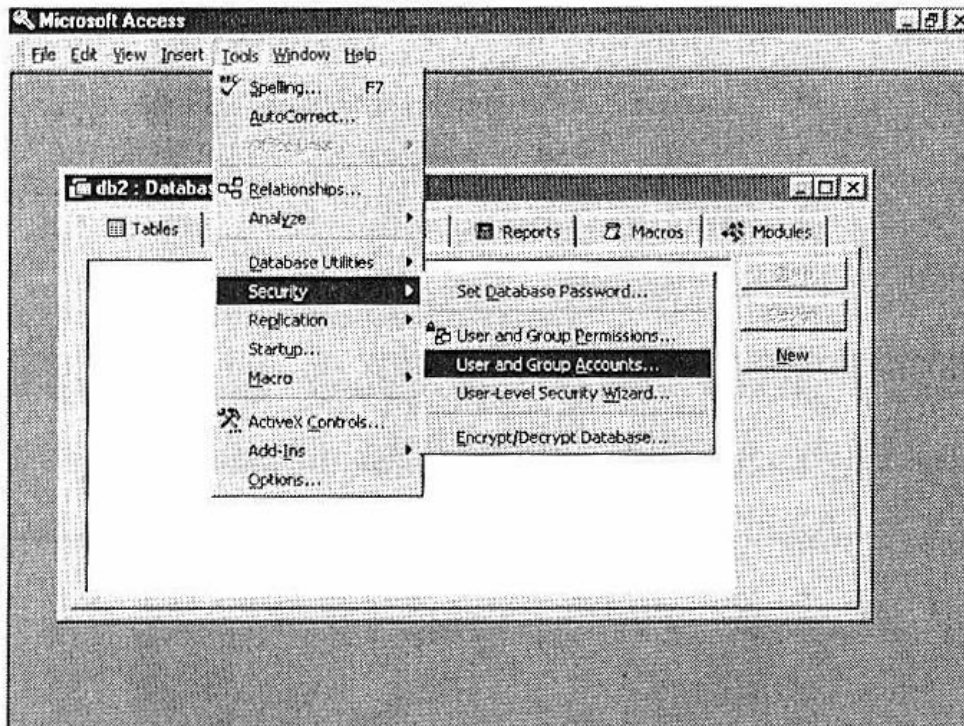
۳- در پنجره File | New Database نام و مسیر پایگاه داده خود را مشخص کنید. برای مثال در قسمت File Name ، DB2 را وارد نموده و OK کنید.



شکل ۱۱-۱۲

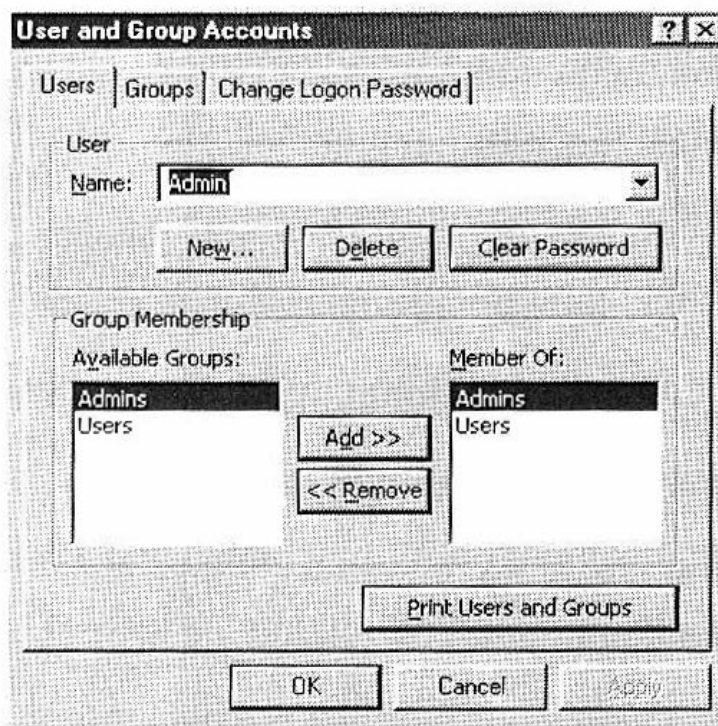
۴- حال منوی Security | User and Group Account را اجرا کنید.

نکته: در صورتی این منو فعال می‌باشد که کاربر جاری سیستم Admin و یا عضو گروه Admins باشد.



شکل ۱۲-۱۲

۵- در این مرحله پنجره User and Group Account فعال می‌شود.



شکل ۱۲-۱۳

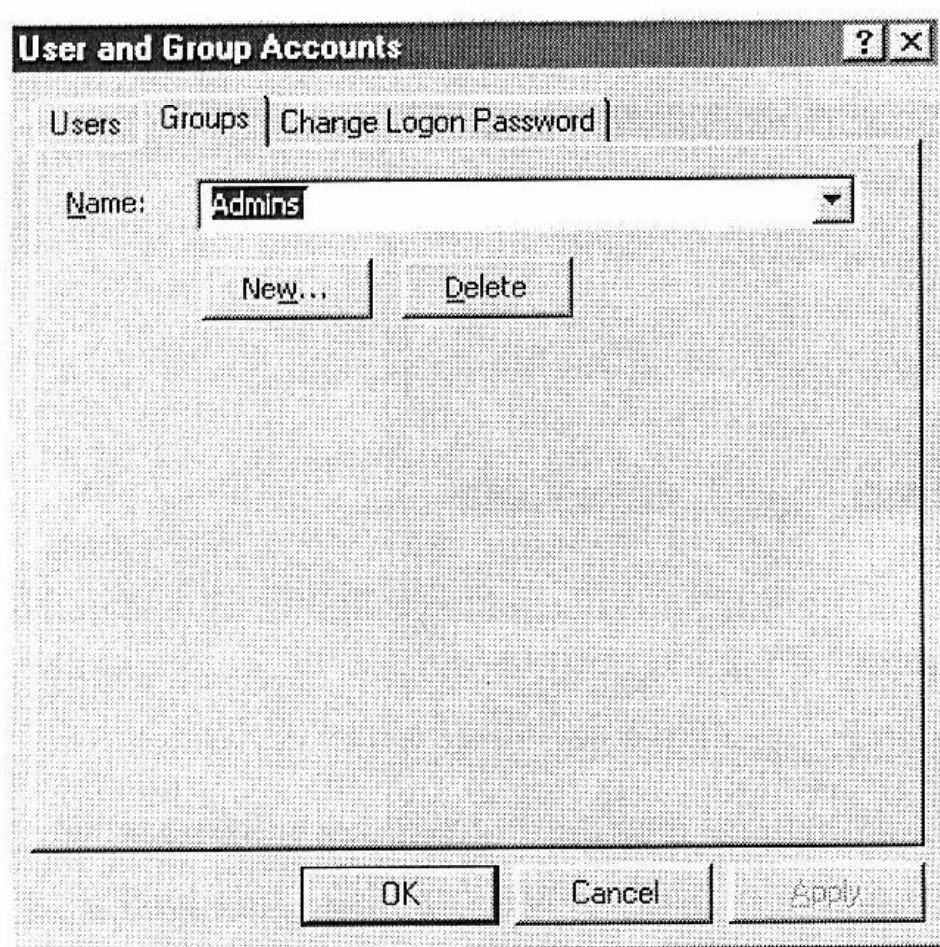
این پنجره از سه بخش اصلی تشکیل شده که به ترتیب User، Groups و Change Logo Pass نام دارد.

• Users: در این قسمت کاربران سیستم و عضویت هر یک در گروهها نشان داده می‌شود. همچنین از طریق این قسمت می‌توان یک کاربر جدید ایجاد نمود و نیز یک کاربر را از لیست کاربران حذف کرد

و یا کلمه عبور آنها را پاک نمود . همچنین می توان هر کاربر را عضو گروهی در آورد و یا عضویت کاربر را از گروه حذف نمود.

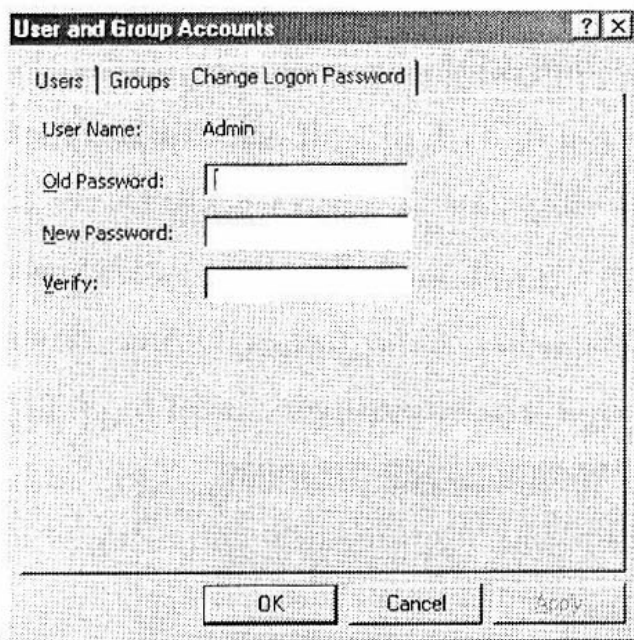
دو بخش اصلی این قسمت، User و Group Membership می باشد که در بخش User کاربران انتخاب می شوند و در بخش Group Membership عضویت کاربر در گروهها نشان داده می شود . برای ایجاد یک کاربر جدید باید دکمه فرمان New را در بخش User کلیک کنید و مشخصات کاربر جدید را در آن وارد نمایید. سپس عضویت کاربر را در گروهها توسط بخش بعدی یعنی User و Group Membership انجام دهید.

• Groups در این قسمت گروههای کاربران ایجاد و یا حذف می گردند یک گروه کاربری نامی لیست به یک عده از کاربران اطلاق می شود که همگی دارای محدودیتهای مشخصی هستند. این گروهها بسیار حائز اهمیت می باشند زیرا در زمان طراحی یک سیستم کافی، لیست گروههای کاربران خود را مشخص کنید و به هر گروه محدودیتهای لازم را بدهید و سپس کاربران مورد نظر خود را در برنامه ای ایجاد کنید و عضو هر گروه قرار دهید. با این کار دیگر لازم نیست به ازای تعریف هر کاربر جدید، تمام محدودیتهای آن کاربر ایجاد کرد.



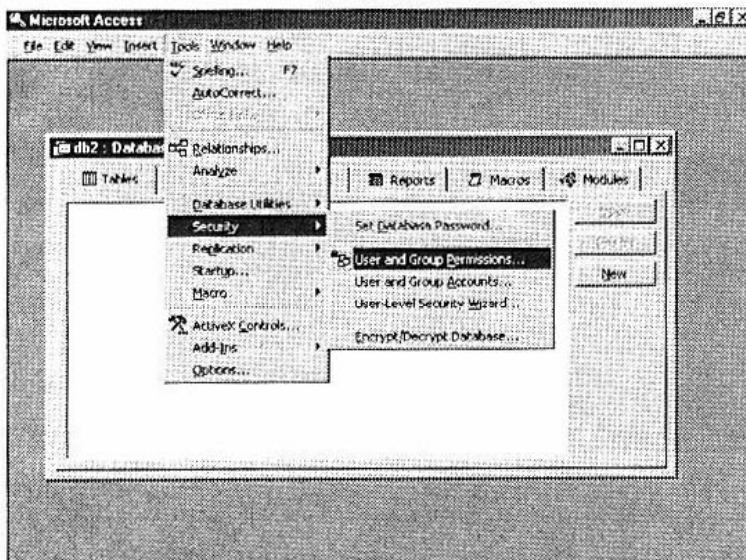
شکل ۱۴-۱۲

از طریق این بخش می توان کلمه عبور هر کاربر را تغییر داد . برای تغییر کلمه عبور یک کاربر باید ابتدا کلمه قبلی کاربر را در قسمت OldPassword و Verify وارد نمود .



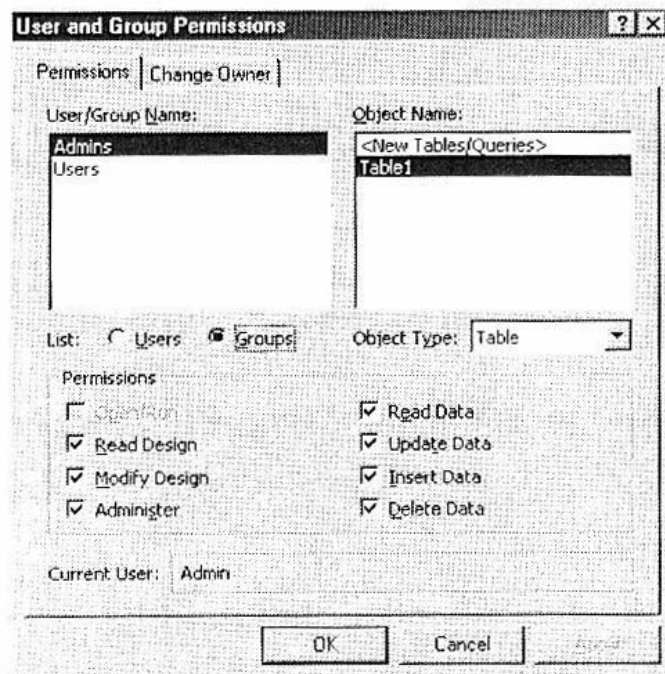
شکل ۱۵-۱۲

تخصیص محدودیتها به گروهها و کاربران
 ۱- منوی Tools|Security|User and Group Permissions را اجرا کنید.



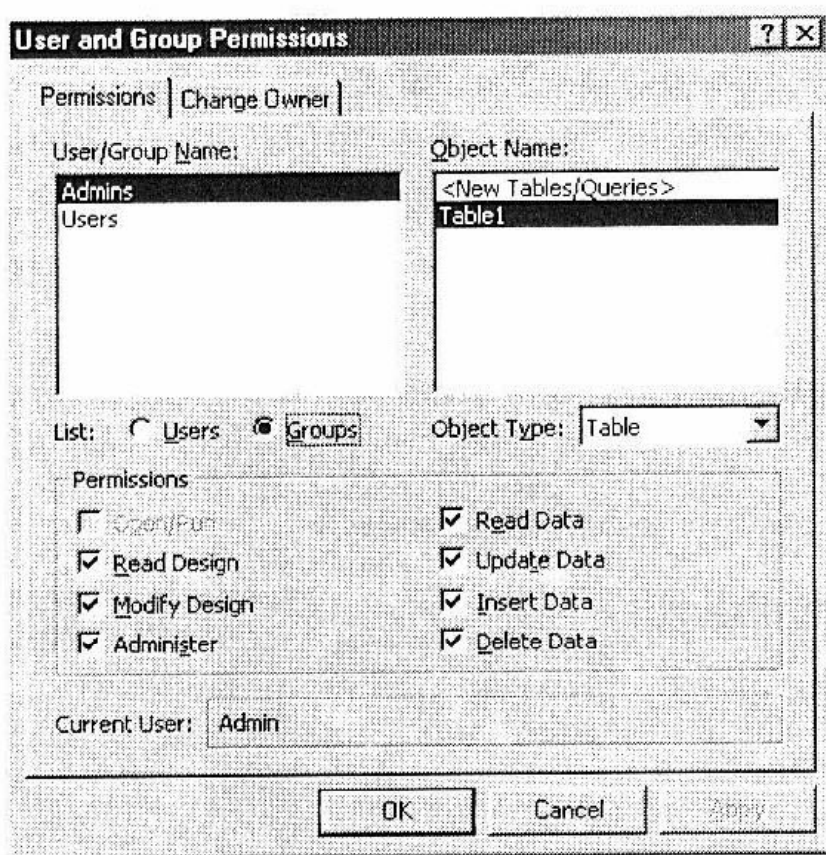
شکل ۱۶-۱۲

۲- پنجره User and Group Permissions فعال می گردد (شکل ۱۷-۱۲).



شکل ۱۷-۱۲

این پنجره از دو قسمت اصلی تشکیل شده است که Permissions و Change Owner نام دارند. در قسمت Permissions محدودیتهای کاربران و گروهها روی هر یک از اعضای پایگاه داده تعریف می شود و در قسمت Change Owner صاحب هر عنصر پایگاه داده اکسس را می توان تغییر داد. ۳- در قسمت Permission می توانید هر گروه و یا کاربری را از لیست User | Group انتخاب کنید و سپس در قسمت Object Name روی هر یک از عناصر پایگاه داده نظیر جداول کلیک نموده و محدودیتهای مورد نظرتان را از لیست Permissions برای آن عنصر تعیین کنید.



شکل ۱۸-۱۲
P5.BMP

عناصری که در این کتاب مورد نظر ماست جداول و سؤالها می باشد. این دو عنصر را می توانید از قسمت Object Type انتخاب نمایید پس از انتخاب Object Type لیست تمام عناصر این دسته در قسمت Object Name نشان داده می شود.

محدودیتهایی که می توان روی جداول و سؤالها اعمال کرد به ترتیب زیر می باشد.

Read Design: با انتخاب این محدودیت، کاربر قادر خواهد بود ساختار یک جدول و یا سؤال را ببیند اما امکان تغییر آنها را نخواهد داشت این محدودیت در صورت انتخاب هر یک از محدودیتهای دیگر خود به خود انتخاب خواهد شد.

Modify Design: توسط این محدودیت کاربر قادر خواهد بود ساختار یک جدول و یا سؤال را تغییر دهد.

Read Data: این محدودیت امکان خواندن اطلاعات را به کاربر می دهد اما اجازه تغییر را به کاربر نمی دهد.

Update Date : این محدودیت امکان تغییر اطلاعات را به کاربر می‌دهد. با انتخاب این محدودیت خود به خود محدودیت Object Type به کاربر داده می‌شود زیرا برای تغییر اطلاعات باید ابتدا آنها خوانده شود.

Insert Data : این محدودیت اجازه اضافه کردن به رکوردهای یک جدول و یا سؤال را به کاربر می‌دهد .

Delete Date : این محدودیت اجازه اضافه کردن به رکوردهای یک جدول و یا سؤال را به کاربر می‌دهد.

Adminster : این محدودیت اجازه تغییر محدودیتهای جدول و یا سؤال را به کاربر می‌دهد.

پس از اعمال این محدودیتها با زدن دکمه فرمان OK همه تغییرات در سیستم حفاظت ثبت می‌شود و اکنون فایل حفاظت قابل استفاده در سیستم پایگاه داده شما خواهد بود.