

# راهنمای استفاده از DAO در ویژوال بیسیک (VB)

دانلود شده از سایت [www.pcseven.com](http://www.pcseven.com)

تألیف : امیرمسعود ایرانی (AMIB)

[AMIBCT@Gmail.com](mailto:AMIBCT@Gmail.com)

۸۴/۹/۱۵

## مقدمه

برنامه‌نویسی بانک‌های اطلاعاتی و ذخیره و بازیابی داده‌ها در دنیای امروز از اهمیت زیادی برخوردار است و ارتباط سریع و آسان با حجم انبوهی از داده‌ها خواسته‌ی هر کاربر و برنامه‌نویس است. یکی از بهترین رابط‌ها برای انجام این کار، DAO است. در ادامه قصد آشنایی بیشتر با این رابط و طرز کار آن را در VB داریم.

## ۱- DAO چیست؟

DAO کوتاه‌شده‌ی عبارت «Data Access Objects» به معنی «اشیای دسترسی به داده» است. ذخیره و بازیابی داده‌ها در فایل‌های اطلاعاتی مستلزم آشنایی برنامه‌نویس با فرمت و قالب‌بندی این گونه فایل‌هاست. برای حل این مشکل و آسان نمودن دسترسی به اطلاعات، رابط‌های بسیاری ایجاد گشته‌اند که میان برنامه و بانک اطلاعاتی نقش میانجی را بازی می‌کنند. در این گونه رابط‌ها تعداد محدودی دستور برای ارتباط با برنامه وجود دارد و در طرف دیگر انواع مختلف بانک‌های اطلاعاتی می‌تواند مورد استفاده قرار گیرد.



## ۲- شروع کار

گام اول در استفاده از DAO انتخاب آن در لیست References یا «ارجاعات» پروژه است. برای انجام این کار به منوی «Project» رفته و سپس بر روی عنوان «References...» کلیک کنید. لیستی از اشیای قابل استفاده را مشاهده خواهید کرد. شیء «Microsoft DAO 3.x Object Library» را پیدا کنید، CheckBox مجاور آن را تیک دار کنید و با تأیید تغییرات به محیط VB بازگردید.

هم‌اکنون استفاده از اشیای موجود در این کتابخانه امکان‌پذیر است. در نسخه‌ی ۳٫۶ از این کتابخانه ۱۷ نوع شیء قابل استفاده است. اشیایی از قبیل Database، Recordset، Field و غیره.

## ۳- آشنایی با «فضای کاری» یا «Workspace»

ایجاد هر یک از ۱۷ شیء ذکر شده مستلزم وجود یک فضای کاری است. فضای کاری را مانند مکانی برای کار کردن کارمندان تصور کنید. برای ایجاد یک فضای کاری شیئی از نوع «Workspace» تعریف می‌کنیم و با کمک دستور «DAO.CreateWorkspace» یا به اختصار «CreateWorkspace» یک فضای کاری خالی ایجاد می‌نماییم. فراموش نکنید که پس از پایان کار، باید فضای کاری ایجاد شده را با دستور «Close» ببندیم تا حافظه‌ی اختصاص یافته به آن رها گردد.

```
Dim wrkJet As Workspace
```

```
Set wrkJet = CreateWorkspace("", "admin", "", dbUseJet)
```

باز کردن بانک اطلاعاتی '

```
wrkJet.Close
```

#### ۴- باز کردن بانک اطلاعاتی

پس از آنکه فضای کاری با موفقیت ایجاد شد، می‌توانیم بانک اطلاعاتی خود را در این بخش از حافظه بارگذاری کنیم. برای این کار نیاز به شیئی از نوع «Database» داریم. همان‌گونه که پیش‌تر ذکر شد، بستن این شیء نیز پس از پایان کار باعث آزادسازی فضای اختصاص داده شده به آن می‌گردد.

```
Dim dbs As Database
```

```
Set dbs = wrkJet.OpenDatabase("C:\MyDir\AMIB.mdb", False, False, "MS Access;")
```

دستورات مربوط به ذخیره و بازیابی داده‌ها '

```
dbs.Close
```

#### ۵- ذخیره و بازیابی داده‌ها

با توجه به تنوع اشیای موجود در DAO، برای نوشتن و خواندن اطلاعات راه‌های گوناگونی وجود دارند. ما به ساده‌ترین و کوتاه‌ترین راه می‌پردازیم.

برای دسترسی به داده‌های موجود در جداول (Tables) موجود در بانک اطلاعاتی ابتدا باید آن را باز کنیم. برای این کار از شیء «Recordset» استفاده می‌نماییم. همان‌گونه که از نام این شیء آشکار است، مجموعه‌ای از رکوردها محتویات آن را تشکیل می‌دهند. ولی با توجه به تنظیمات پیش‌فرض، DAO تلاش می‌کند در صورت امکان Recordset را معادل جدول به حساب آورد. بنابراین در صورتی که ما Recordsetی از رکوردهای یک جدول را ایجاد کنیم، در حقیقت با خود جدول سروکار خواهیم داشت.

ایجاد Recordset مستلزم آشنایی ابتدایی با زبان SQL است. در شکل زیر می‌توانید با دستور «SELECT» و پارامترهای ساده‌ی آن آشنا شوید.

شرط انتخاب رکوردها WHERE نام جدول FROM نام فیلدها SELECT

در حالت خاص و برای انتخاب تمامی فیلدهای یک جدول، می‌توان از علامت ستاره (\*) به عنوان نام فیلد استفاده نمود. در بخش «شرط انتخاب رکورد» می‌توان از دستوراتی مشابه زبان VB استفاده نمود. برای مثال به شرط زیر توجه کنید.

```
WHERE (Age = 18 AND Gender = "Male")
```

هم‌اکنون آماده هستیم تا جدول مورد نظر را باز کنیم و به ثبت و بازخوانی داده‌های آن پردازیم.

```
Dim rstPhone As Recordset
```

```
Set rstPhone = dbs.OpenRecordset ( "SELECT * FROM Phone" )
```

دستورات مربوط به اضافه، ویرایش، خواندن و حذف رکوردها '

```
rstPhone.Close
```

هنگامی که یک **Recordset** ایجاد شد، می‌توان با کمک دستورات و صفات زیرمجموعه‌ی آن رکوردهای موجود را مدیریت نمود. مهمترین و پرکاربردترین دستورات عبارتند از **AddNew**، **Edit**، **Delete**، **Update**، **FindFirst**، **FindNext**، و دستورات جابجایی میان رکوردها. همچنین مهمترین صفات عبارتند از **Fields**، **NoMatch**، **RecordCount**، و **EOF**. برای روشن‌تر شدن موضوع به قطعه برنامه‌ی کوتاه زیر توجه کنید.

```
Do While Not rstPhone.EOF
```

```
    Debug.Print rstPhone.Fields ("Name"); " = "; rstPhone.Fields("PhoneNo")
```

```
    rstPhone.MoveNext
```

```
Loop
```

در هر **Recordset** یک رکورد فعال وجود دارد که تمام عملیات خواندن، نوشتن و حذف بر روی آن صورت می‌پذیرد. برای حرکت میان رکوردها از دستورات انتقالی که با پیشوند «**Move**» آغاز می‌شوند، استفاده می‌کنیم. همان‌طور که در شکل مشاهده کردید، حلقه‌ای با شرط خاتمه‌ی «پایان نیافتن رکوردها» ما را در مدیریت یاری می‌کند و همین‌طور دستور «**MoveNext**» رکورد جاری را پس از چاپ محتویات آن، یک خانه به جلو منتقل می‌کند.

برای جستجوی رکوردها از دو دستور **FindFirst** و **FindNext** و همین‌طور صفت **NoMatch** بهره می‌گیریم.

```
rstStudents.FindFirst "Age > 25"
```

جستجوی اولین دانشجو با سن بیش از ۲۵ سال '

```
Do While Not rstStudents.NoMatch
```

شرط ورود به حلقه : پیدا شدن رکوردی با شرایط ذکر شده '

```
    Debug.Print rstStudents.Fields ("Name")
```

چاپ نام دانشجوی پیدا شده '

```
    rstStudents.FindNext
```

جستجوی مجدد از محل فعلی به بعد '

```
Loop
```

## ۶- اضافه کردن و ویرایش رکوردها

برای انجام این کارها ابتدا باید **DAO** را آگاه کنید که قصد انجام این اعمال را دارید. به همین منظور از دستورات «**AddNew**» و «**Edit**» استفاده می‌کنیم. دستور «**AddNew**» رکوردی خالی را در حافظه ایجاد می‌کند و آن را به رکورد فعال تبدیل می‌نماید. همچنین دستور «**Edit**» رکورد جاری را به حافظه منتقل می‌کند و آن را برای ویرایش آماده می‌کند. پس از اجرای این دو دستور رکورد جاری قابل ویرایش خواهد بود. هر تلاشی برای نوشتن بر روی رکوردها پیش از اجرای این دستورات با خطایی از سوی **DAO** مواجه خواهد شد.

<code>rstStudent.Edit</code>	آگاه‌سازی رابط از قصد انجام ویرایش رکورد جاری '
<code>rstStudent.Fields ("Name") = "AMIB"</code>	ایجاد تغییرات لازم...
<code>rstStudent.Update</code>	ذخیره‌ی تغییرات به طور دائم بر روی بانک اطلاعاتی '

<b>With</b> <code>rstPhone</code>	انتخاب شیء فعال برای صدا زدن بدون ذکر نام '
<code>.AddNew</code>	اضافه کردن رکورد جدید و فعال کردن آن برای ویرایش '
<code>.Fields ("Name") = "AMIB"</code>	ذخیره‌ی مقادیر جدید در فیلدها '
<code>.Fields ("PhoneNo") = "12345678"</code>	
<code>.Update</code>	به روز رسانی بانک اطلاعاتی با داده‌های جدید '
<b>End With</b>	

## ۷- چند نکته

+ صفت **RecordCount** در ابتدای ایجاد **Recordset** فقط در صورتی تعداد کامل رکوردها را نمایش خواهد داد که تمام رکوردها پیمایش شده باشند. به همین دلیل در صورتی که قصد پیدا کردن تعداد رکوردها را دارید، باید ابتدا از دستور **MoveLast** استفاده نمایید تا تمام رکوردها پیمایش شوند و رکورد پایانی پیدا شود.

+ در صورت استفاده از تنظیمات پیش‌فرض در هنگام باز کردن بانک اطلاعاتی، حالت دسترسی به صورت «اشتراکی» خواهد بود. به این معنی که هر کاربر دیگری نیز قادر خواهد بود همزمان با شما رکوردها را اضافه، ویرایش و حذف نماید. برای جلوگیری از این اتفاق، کافی است در هنگام صدا زدن دستور **OpenDatabase** مقدار «True» را به عنوان دومین پارامتر (**Options**) به تابع ارسال نمایید.

+ برای وارد کردن رمز بانک اطلاعاتی هنگام باز کردن آن، صفت **PWD** در پارامتر چهارم یا **Connect** را تعیین کنید.

```
Set dbs = wrkJet.OpenDatabase("AMIB.mdb", False, False, "MS Access;PWD=MyPassword")
```

+ رابط **DAO** توانایی باز کردن بانک‌های گوناگونی غیر از **Access** را نیز داراست. ولی با توجه به فرمت و قالب‌بندی‌های متفاوت ممکن است تمام اعمالی که بر روی بانک‌های **Access** قابل انجام هستند، در دیگر انواع **Database** در دسترس نباشند.

تألیف : امیرمسعود ایرانی (AMIB)

AMIBCT@Gmail.com

۸۴/۹/۱۵