

* إدخال البيانات ، تحريرها ، حذفها .

مررنا سابقا بعدة مراحل كانت الأولى إجراء اتصال مع قواعد البيانات ومن ثم فتح الجدول وكان آخرها هو عرض السجلات وتصفحها . والآن سنأتي على ذكر العمليات أو الإجراءات والتي تعتبر جزء مهم من أجزاء التعامل مع قواعد البيانات والتي تتبلور في عمليات الإضافة وتحرير وحذف للبيانات .

وسنبدأ بعملية الإضافة :

قد رأينا في قسم استعراض السجلات إن هناك عدة طرق للوصول إلى هدف معين في البرمجة ولكل مبرمج طريقه أو أسلوبه المفضل. والبرمجة في الحقيقة هي طريق للإبداع.

طبعة للوصول إلى هذه النقطة نقطة إضافة البيانات يجب توفير :

- اتصال مع قواعد البيانات.
- فتح الجدول بالطريقة المعتادة ولكن مع إعطاء صلاحية الإضافة أو صلاحيات المعالجة .

والآن لتأمين اتصال يتيح لنا معالجة البيانات يجب إجراء تغيير طفيف في كود الاتصال مع الجدول .

أنظر إلى الكود السابق :

,

Dim db As New ADODB.Connection

Dim rs As New ADODB.Recordset

,

Private Sub Form_Load()

db.Provider = "Microsoft.JET.OLEDB.4.0;"

db.Open App.Path & "\db1.mdb"

rs.Open "[Table1]", db, adOpenStatic, adLockReadOnly

End Sub

الكود المعدل :

,

Dim db As New ADODB.Connection

Dim rs As New ADODB.Recordset

,

Private Sub Form_Load()

db.Provider = "Microsoft.JET.OLEDB.4.0;"

db.Open App.Path & "\db1.mdb"

rs.Open "[Table1]", db, adOpenStatic, adLockPessimistic

End Sub

هل لاحظت الفرق ما بين الكود الذي استخدمناه في السابق والكود الجديد ، نعم يوجد فرق في السطر :

السطر من الكود السابق :

rs.Open "[Table1]", db, adOpenStatic, adLockReadOnly

السطر من الكود الجديد :

rs.Open "[Table1]", db, adOpenStatic, adLockPessimistic

في الكود السابق قد أمانا اتصال مع الجدول في وضعية القراءة فقط ، ولكن عندما نريد إجراء عمليات المعالجة يجب تأمين اتصال في وضعية الصلاحيات (**Pessimistic**) ، لهذا استبدلنا خاصية **adLockReadOnly** بالخاصية **adLockPessimistic** .

نكمل . بعد إن تعرفنا على المسار الجديد والتغيير المحدث في فتح الجداول ، سنقوم بإضافة خمسة Commands .

- ١ - Command5 سيحمل اسم Add .
- ٢ - Command6 سيحمل اسم Edit .
- ٣ - Command7 سيحمل اسم Delete .
- ٤ - Command8 سيحمل اسم Save .
- ٥ - Command9 سيحمل اسم Cancel .

الشكل العام للبرنامج بعد إضافة الCommands /

بعد ذلك سنقوم بإضافة Sub والغرض منه عدم إتاحة للخاصية الأخرى فمثلا عندما تضغط على مفتاح Add فمن المفروض إن المفاتيح الأخرى تكون غير متاحة أو ممكنة حيث لا يمكن إجراء عمليتين مع بعضهما البعض في نفس الوقت . على العموم سوف تتضح فكرة ال Sub عند إكماله .

أسم ال : SUB فليكون **EnabledUnEnabled** .

```
Sub EnabledUnEnabled(n As Boolean)
```

```
End Sub
```

والكود الذي سيحويه ال SUB هو :

```
Sub EnabledUnEnabled(n As Boolean)
```

```
Text1.Locked = True
```

```
Text2.Locked = n
```

```
Text3.Locked = n
```

```
Command1.Enabled = n
```

```
Command2.Enabled = n
```

```
Command3.Enabled = n
```

```
Command4.Enabled = n
```

```
Command5.Enabled = n
```

```
Command6.Enabled = n
```

```
Command7.Enabled = n
```

```
Command8.Enabled = Not n
```

```
Command9.Enabled = Not n
```

```
If rs.RecordCount = 0 Then
```

```
Command6.Enabled = False
```

```
Command7.Enabled = False
```

```
End If
```

```
End Sub
```

قد تلاحظ الحرف **n** الذي أضفناه إلى SUB من **Boolean** و ال **Boolean** هو متغير يرجع قيمة **True** أو **False** . وسوف نستفيد منه في عملية الاستدعاء **Calling** .

فمثلا عندما نستدعي ال SUB السابق بهذا الشكل :

```
Call EnabledUnEnabled(True)
```

فسيكون الناتج هو إتاحة جميع المفاتيح ما عدى مفتاح **Save** وال **Cancel** أما البقية ستكون متاحة . والعكس صحيح .

أما السطور الثلاثة الأولى فمهمتها قفل ال Textbox ولا يمكن الكتابة فيه إلا بعد الضغط على مفتاح إضافة سجل جديد Add وأما ما بعدها فهي مفهومة الغرض :

أما الجزء الأخير من الكود والذي هو :

```
If rs.RecordCount = 0 Then
```

```
,
```

```
Command6.Enabled = False
```

```
Command7.Enabled = False
```

```
End If
```

فهذا يتكفل بعملية إغلاق مفتاح Edit و Delete وهذا الإغلاق مشروط وهو إذا كان الجدول فارغ ولا يحتوي على بيانات إما إذا كان الجدول يحتوي على البيانات فلا يتم تنفيذ الكود السابق . طبعا بدون هذه الخاصية ستنتج أخطاء برمجية إذا ما تم النقر على المفتاحين السابقين وفي حالة عدم وجود إي بيانات حيث كيف يمكن تعديل أو تحرير بيانات في جدول معين والجدول في الأصل لا يحتوي على بيانات . وبهذا قد انتهينا من ال SUB . وأتمنى إن تكون فكرة ال SUB مفهومة لدى الجميع .

وألان سننتقل إلى مفتاح الإضافة Add . وكل الذي سيحويه هذا المفتاح هو التالي :

```
,
```

```
Private Sub Command5_Click()
```

```
,
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
,
```

```
Call EnabledUnEnabled(False)
```

```
Text2.SetFocus
```

```
End Sub
```

```
,
```

أتوقع الكود السابق لا يحتاج إلى شرح .

وألان سوف ننتقل إلى مفتاح الحفظ Save . وقبل إن اشرع في كتابة الكود أحببت انوه على إن عملية الحفظ هي العملية التي سوف يتم تخزين البيانات المدخلة فيها . فيجب إن نحرص على وجود بيانات ، إذا يجب وضع كود يقوم بتحقق من وجود البيانات قبل عملية الحفظ بخطوات .

وهذا الكود الذي سنضعه في مفتاح الحفظ Save . وبعدها سنتناول الكود خطوة خطوه .

```
,
```

```
Private Sub Command8_Click()
```

```
,
```

```
If Trim(Text2.Text) = "" Then
```

```
MsgBox "Cnnot contain a Null Value .", vbCritical, "Message"
```

```
Text2.SetFocus
```

```

Exit Sub
End If
'
With rs
. AddNew
'

    ![CategoryName] = Trim(Text2.Text)
    If Not Trim(Text3.Text) = "" Then ![Description] =
Trim(Text3.Text)
'

. Update
End With
'

rs.Requery
rs.MoveLast
Call EnabledUnEnabled(True)
Call ViewRecord

```

```

End Sub
'

```

أولا :

```

If Trim(Text2.Text) = "" Then
MsgBox "Cnnot contain a Null Value .", vbCritical, "Message"
Text2.SetFocus
Exit Sub
End If

```

ونستفيد منه في عملية التحقق المذكورة سابقا وفي حالة لم يتم العثور على نص في Text Box رقم ٢ فإنه سيقوم بأخبارك إنه لا يمكن إسناد قيمة فارغة . وسيعيدك إلى ال Text Box الذي من المفترض تعبئته بالبيانات . ومن ثم يخرج من ال SUB ولا يتم تنفيذ ما بعده وهذا النوع من الأكواد أنا شخصا أطلق عليه اسم (صائد الأخطاء) .

أما الكود الذي يليه :

```

With rs
. AddNew
'

    ![CategoryName] = Trim(Text2.Text)
    If Not Trim(Text3.Text) = "" Then ![Description] =
Trim(Text3.Text)
'

. Update
End With

```

فهنأ سنقوم بإضافة السجل الجديد أو البيانات الجديدة والأمر AddNew هو المسؤول عن الإضافة . وعلامة التعجب ! تفيد إن ما بعدها هو حقل وطبعا بعد تحديد الحقل سوف نسند له البيانات التي تم إدخالها . وبالنسبة للحقول الضرورية أو المطلوبة فنكتفي بوضعها بهذا الشكل /

![CategoryName] = Trim(Text2.Text)

أما الحقول غير ضرورية والمتروكة كخيار للمستخدم يقوم بتعبئتها أو يتركها فارغة . فهذه الحقول يجب وضع لها كود التحقق ويكون كآالي :

If Not Trim(Text3.Text) = "" Then ![Description] =Trim(Text3.Text)

والكود السابق يقوم بفحص ال Text Box وإذا كان فارغ فهو يتجاهل الحقل أما إذا كان ال Text Box يحوي على بيانات فإنه سيسند القيمة إلى الحقل الصحيح .

أما الخاصية **Update** فهي المسؤولة عن حفظ القيم في السجل وبدونها يسكون ما سبق عقيما .

وسنأخذ السطور المتبقية كل على حده :

rs.Requery

ال **Requery** هي بمثابة Refresh في DAO حيث تقوم بتحديث للبيانات .

rs.MoveLast

وال **MoveLast** الانتقال إلى السجل الأخير .. ولماذا نستخدمها هنا .. ببساطة لأنه السجل الجديد سيكون آخر سجل تم إضافته

Call EnabledUnEnabled(True)

ومثل ما عرفنا سابقا إن **EnabledUnEnabled** يقوم بعملية قفل أو فتح ال Text Box وال **Commands** وهنا يجب أقفالها فعندما ننتهي من إي إجراء يجب قفل ما تم فتحة أو إتاحتة .

Call ViewRecord

وال **ViewRecord** يقوم بعرض السجل الجديد . أنتهي .

بالنسبة لمفتاح إلغاء Cancel قم فقط بإضافة الكود التالي والكود مفهوم ولا يحتاج إلى توضيح :

Private Sub Command9_Click()

Call EnabledUnEnabled(True)

Call ViewRecord

End Sub

انتهى .

أخواني أأتمنى إن تعذرني على تقصيري وحقيقة كتبت الشروحات السابقة وأنا في عجلة من أمري وحتى إنني لا أستطيع من مراجعته ما كتبت والتأكد من خلوة من الأخطاء . على العموم عسى إن تستفيدوا منه ولكم مني أحلى تحية .

الدرس القادم سأحدث عن عملية تحرير أو تعديل البيانات .

هيثم القلاف

Haa10@hotmail.com

<http://www.al-ebda3.info/ib>