قام بتجميع الدروس : مشرف ناصر

صاحب الدروس: هيثم القلاف

قمت بتجميع الدروس للفائدة فقط

الكتاب هدية الى:

جامعة أهلا عرب

يوجد مع الكتاب أمثله

PDF created with pdfFactory trial version www.pdffactory.com

كيفية إجراء اتصال مع قواعد البيانات

أولا قد يتساءل البعض عن مكتبة ADO ما هي وما فائدتها وكيف يمكن استغلالها من خلال لغة الفيجوال بيسك ٦,٠ ؟؟ .

ADO -هي اختصار ل Microsoft ActiveX Data Object

-مكتبة ADO تعتبر الأسلوب الجديد (قبل الأداة الجديدة ADO.NET) للتعامل مع قاعد البيانات بعيدا عن البرنامج الأم الذي أنشئت منها قاعدة البيانات ... فمثلا يمكننا الاتصال وقراءة محتويات قاعدة بيانات اكسس Access بدون الحاجة لوجود Access فهذه المكتبة توفر خدمة الاتصال والإجراءات المختلفة للتعامل معها .

-إما كيفية استغلالها فهذا ما سوف نتعرف علية في السطور القليلة القادمة .

أولا وقبل إن نبدأ بأي خطوة علينا توفير مكتبة ADO للمشروع . ولتوفيرها نتبع الخطوات التالية :

وبعد فتح مشروع جديد :

من خلال القائمة نختار Project ثم ... References بعدها سوف يظهر فورم ((Form معنون بي References – Project1. ويحتوي هذا الفورم على مكتبات عديدة وما يهمنا منها هي مكتبة ADO وسوف نجدها بسم /

Microsoft ActiveX Data Objects 2.X Library .

وال X يعبر عن الإصدار المثبت على جهازك . وهناك إصدارات عديدة من هذه المكتبة وكلها تقريبا بالشكل التالي /

1- Microsoft ActiveX Data Objects 2.0 Library
2- Microsoft ActiveX Data Objects 2.1 Library
3- Microsoft ActiveX Data Objects 2.5 Library
4- Microsoft ActiveX Data Objects 2.6 Library
5- Microsoft ActiveX Data Objects 2.7 Library

بعد إن تختار أحدها سنكون بهذه الخطوة قد أمنا مكتبة ADO لمشروعنا . تمام . نكمل.

وألان وبعد تأمين مكتبة ال ADOفيمكننا الشروع في العمل ، وستكون خطة العمل كالتالي /

المستوى الأول

-1كيفية إجراء اتصال مع قواعد البيانات .

-2كيفية فتح الجداول .

PDF created with pdfFactory trial version www.pdffactory.com

3-تحرير طرق السجلات .
4-إدخال البيانات ، تحريها ، حذفها .
5-عملية البحث(Search)
a.بحث بحالة الأحرف
b.بحث شامل
6-التصفية(Filter)
7-الفرز ((Sort
c.تصاعدي
d.تنازلي

بالنسبة للمستويات الأخرى سوف نذكرها في وقتها- بإذن الله - وأيضا يتوقف طرحها على المستفيدين من هذا الموضع . ومدى رغبتهم في مواصلة الأمثلة والنماذج الخاصة بقواعد البيانات .

*كيفية إجراء اتصال مع قواعد البيانات .

لإجراء اتصال مع قاعد البيانات أكسس تكون بالتالي /

نضع المتغير في قسم الجنرال: General

```
Dim db As New ADODB.Connection
```

ثم نكمل الكود بالذي يليه ونضعه في قسم التحميل/Form_Load

```
db.Provider = "Microsoft.JET.OLEDB.4.0;"
db.Open App.Path & "\db1.mdb"

MsgBox "" تم الاتصال بقاعدة البيانات بنجاح
```

السطر الأول

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

والاستفادة منه هو تحديد المزود الذي سنتصل من خلاله بقاعدة البيانات إن كانت اكسس أو SQLاو Dbase أو غيرها .
فالمزود Microsoft.JET.OLEDB.4.0 هو ا لخاص بقواعد بيانات اكسس .
السطر الثاني

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

والغرض منه تحديد مسار قاعدة البيانات db1.mdb ... هو اسم قاعدة البيانات

بالخطوات السابقة انتهينا من عملية الاتصال مع قواعد بيانات اكسس

لإجراء اتصال مع قاعد البيانات SQL تكون بالتالي /

نتبع الخطوات السابقة بالتفصيل ولكن التغير سوف يحدث أين .... طبعا سيكون في المزود والذي يتعاطى مع قاعدة بيانات ، SQL إضافة إلى اسم قاعدة البيانات ، كما سوف ندمج السطرين السابقين في سطر واحد فقط ليصبح الكود هكذا /

```
db.Open "Provider=SQLOLEDB.1;Integrated
Security=SSPI;Persist Security Info=False;Initial
Catalog=Northwind"
```

تم بفضل من الله .

PDF created with pdfFactory trial version www.pdffactory.com

يوجد مثال لدرس مرفق مع الكتاب

كيفية فتح الجداول

سابقا قد تعرفنا على هوية الأكواد التي نستطيع من خلالها فتح اتصال مع قواعد البيانات والآن ننتقل إلى الخطوة التالية وهي كيفية فتح الجداول الموجودة في قاعدة البيانات .
ولكي نفتح جدول في قاعدة البيانات يجب أولا تأمين اتصال مع قاعدة البيانات الموجود بها الجدول المراد فتحة . يعني نستخدم الكود السابق لتفح قاعدة البيانات مع زيادة طفيفة وهي كالتالي

```
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
```

أنظر للمتغير rs حيث أسندنا إليه خصائص Recordset وهو الذي يُعنى بكل العمليات التي تجرئ على الجداول من فتح وحذف وتعديل و ....الخ .
وأيضا لاحظ السطر

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

حيث الجدول هنا هو Table1 و db تحدد الجدول الموجود في قاعدة البيانات .
وبنفس الخطوات نستطيع فتح جدول في قواعد بيانات SQL

يوجد مثال لدرس مرفق مع الكتاب

PDF created with pdfFactory trial version www.pdffactory.com

طريقة عرض السجلات

استكمالا لما سبق .
تعرفنا آنفا كيفية أجراء اتصال مع قواعد البيانات ومن ثم تعرضنا لطريقة فتح الجداول واليوم بمشيئة الله سنتعلم كيفية عرض السجلات ، والإبحار ما بينها . وكما نعرف إن كل خطوة تبنى عليها خطوات ، بمعنى لا يمكن مثلا فتح جدول بدون تأمين اتصال مع قاعدة البيانات وكذلك أيضا لا يمكن عرض السجلات وقراءة بياناتها بدون فتح الجدول ، فكل عملية مطلوبة وحتمية ولا يمكن تجاهلها ، لأنها بمثابة سلسلة متكاملة .

وهنا سوف نحتاج إلى نشاء بعض الأدوات وتعريفها بالشكل المناسب كالصناديق النصوص Text Boxوال label وبعض المفاتيح . Command على فكرة قبل هذا ارجوا الإطلاع على قاعدة البيانات والتي هي هدفنا . قاعدة البيانات موجودة في الملف المرفق .

أسم قاعدة البيانات Db1.mdb وتحتوي على جدول واحد ويحمل اسم Table1 ويوجد في الجدول ثلاثة سجلات :
السجل الأول : ID وهو يحوي بيانات رقمية (المفتاح الأساسي)
السجل الثاني :CategoryName
السجل الثالث :Description

من الملاحظ إن أسم الحقول وضعت باللغة الإنجليزية وفي الحقيقة لا يهم وضعنها بحروف إنجليزية أو عربية وهنا اخترتها إنجليزية لكي تكون مميزة وملاحظة .

تنبيه / في حالة استخدامك للحروف العربية أو الأسماء الإنجليزية المتفرقة مثل Category Nameفدائما ضع اسم الحقل ما بين الأقواس وليست الأقواس المعتادة إنما استخدم هذه الأقواس [ ] وضع الاسم وسطها مثل. [Category Name]

على كل نشرع في تصميم الفورم وكتابة الأكواد المناسبة .

أولا ومثل ما نعرف نختار مشروع جديد New Project ثم . Standard بعد ظهور الفور ندرج ثلاثة Label فيه . بالنسبة لترتيب ال Label فلا يهم رتبها بالشكل الذي تراه مناسب . وعنونها بالشكل التالي ، طبعا سنستخدم خاصية Caption لعنونة ال. Label

الأول :ID
الثاني :Category Name
الثالث :Description

وبعد ذلك أيضا نضيف ثلاثة صناديق نصوص Text Box ونضعها أمام كل واحد مرتبة حسب وضع ال Label فمثلا ال Label رقم واحد يكون أمامه ال Textbox رقم واحد . وهكذا بالنسبة للبقية . بعد ذلك نضيف أربعة Command إلى الفورم وسوف نستخدمها في عملية الإبحار ما بين السجلات .

PDF created with pdfFactory trial version www.pdffactory.com

عناوين ال Command تكون بالشكل التالي ومتسلسلة الأرقام .

Commandرقم ١ يحمل اسم First Record :
Commandرقم٢ يحمل اسم Previous Record :
Commandرقم٣ يحمل اسم Next Record :
Commandرقم٤ يحمل اسم Last Record :

ولا تنسى إن تحفظ المشروع ولنسميه مثلا . ViewRecord

والآن وصلنا إلى دور كتابة الكود . ومثل ما عرفنا سابقا إن لإجراء إي اتصال مع إي قاعدة بيانات يجب تهيئة مكتبة ADO وقد شرحنا سابقا طريقة تهيئتها وبشكل مبسط .

وهنا سنكمل الكود الذي تم فيه الاتصال مع الجدول / تذكير بالكود .

```
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
```

أولا نقوم بإنشاء Sub وسنسميه ViewRecord

```
'
Sub ViewRecord()
'

End Sub
```

والهدف منه هو عرض السجل الحالي الذي يتم استدعائه من قبل الضغط على ال Commandالذي تم إنشائه سابقا . وستتضح الرؤية بشكل افضل بعد إتمام الكود . نتابع.

بالمناسبة توجد عدة طريق لعرض السجلات وقصد بعرض السجلات هو إظهار البيانات في صناديق النصوص TextBox وليس الانتقال ما بينها . وسأستعرض طريقتين الأولى المفضلة لي والثاني لا أحبذ استخدامها ولكن للمعرفة .

سأدرج الكود أولا ثم سأضع التعليقات حول طريقة اشتغال الكود .

```
Sub ViewRecord()
'
If Not rs.RecordCount = 0 Then
'
   Text1.Text = "
   Text2.Text = ""
   Text3.Text = ""
'
  If Not IsNull(rs![ID]) Then Text1 = rs![ID]
  If Not IsNull(rs![CategoryName]) Then Text2 = rs![CategoryName]
  If Not IsNull(rs![Description]) Then Text3 = rs![Description]
'
End If
End Sub
```

الجملة الشرطية والتي هي: If Not rs.RecordCount = 0 Then
قبل إن اشرح ما فائدتها دعني أوضح فائدة الخاصية RecordCount وهذه الخاصية في

PDF created with pdfFactory trial version www.pdffactory.com

الحقيقة ترجع عدد السجلات الموجودة في الجدول . فمثلا يوجد عندك ١٠ سجلات معرفة في الجدول فإن هذه الخاصية ترجع لنا العدد ١٠ ، فبها نستطيع معرفة ما يحتويه الجدول من عدد السجلات وأيضا نستطيع من خلالها التأكد من إن الجدول فارغ ولا يحتوي على بيانات . أتوقع بدأنا نفهم سبب استخدامها هنا .
وسبب استخدامها في الجملة الشرطية هو التحقق من عدم خلوّ الجدول من البيانات فإذا كان الجدول فارغ فلا يتم إرجاع قيم الحقول ، وإلا سوف نقع في دائرة الأخطاء البرمجية .

أما السطور الثلاثة التي تلي الجملة الشرطية فهو تُعنى بإفراغ محتويات ال Textbox لكي يتم تجهيزها لاحتضان بيانات جديدة.

أما هذا الكود :

```
'
  If Not IsNull(rs![ID]) Then Text1 = rs![ID]
  If Not IsNull(rs![CategoryName]) Then Text2 = rs![CategoryName]
  If Not IsNull(rs![Description]) Then Text3 = rs![Description]
'
```

إذ جئنا لنترجم الكود السابق لنص مقروء فالناتج يكون :

إذا كان الحقل الذي اسمهم ID فارغ ولا يحتوي على بيانات حينئذ تجاهل إسناد القيمة الفارغة إلى صندوق النصوص . وهذه هي الترجمة الحقيقة للكود .

IsNull(rs![ID])تتحقق من إن حقل ID لا يحوي قيمة فارغة.
Text1 = rs![ID]هنا نرجع قيمة الحقل إلى ال Textbox
وهكذا بالنسبة للبقية .

والآن وبعد الانتهاء من كتابة الكود السابق يكون بمقدورنا استعراض البيانات . ولكن قبل تجربة هذه المرحلة يجب وضع كود في قسم التحميل Form_Load وهذا الكود يقوم باستدعاء ال Sub الذي هو ViewRecord فلذلك نكتب الكود التالي /
Call ViewRecordويكون مكانه اسفل الكود الخاص بفتح الجدول . والآن تستطيع تشغيل البرنامج لترى ما تم إنجاز .

تلك كانت الطريقة الأولى والتي أفضلها أما الثانية فيه قريبة من الطريقة الأولى ولكن عوضا عن نستخدم اسم الحقل سوف نستخدم م رقم الحقل أنظر الكود /

```
Sub ViewRecord()
'
If Not rs.RecordCount = 0 Then
'
   Text1.Text = ""
   Text2.Text = ""
   Text3.Text = ""
'
   If Not IsNull(rs.Fields(0).Value) Then Text1 = rs.Fields(0).Value
   If Not IsNull(rs.Fields(1).Value) Then Text2 = rs.Fields(1).Value
   If Not IsNull(rs.Fields(2).Value) Then Text3 = rs.Fields(2).Value
'
End If
End Sub
```

وأيضا توجد طريقة أخرى سأذكرها من باب العمل بالشيء . وفي هذه الطريقة نضع الكود في قسم التحميل Form_Load

انظر الكود/

```
'
Private Sub Form_Load()
```

PDF created with pdfFactory trial version www.pdffactory.com

```
  db.Provider = "Microsoft.JET.OLEDB.4.0;"
  db.Open App.Path & "\db1.mdb"
  rs.Open "[Table1]", db, adOpenStatic, adLockReadOnly
'
  Set Text1.DataSource = rs
  Set Text2.DataSource = rs
  Set Text3.DataSource = rs
      Text1.DataField = "ID"
      Text2.DataField = "CategoryName"
      Text3.DataField = "Description"
End Sub
```

وهكذا باب البرمجة والإبداع موجود ولكل شخص الطريقة التي يفضل استخدامها بغض النظر عن طول الكود أو قصره .

والآن وبعد أن تعرفنا على بعض الطرق المختلفة لعرض البيانات سوف نقوم بكتابة كود لتصفح السجلات أو البيانات والانتقال ما بينها.

نبدأ بكتابة كود الانتقال إلى السجل الأول Record First وسنضع الكود في Command1 الكود

```
 If Not rs.BOF Then rs.MoveFirst
 If rs.BOF And rs.RecordCount > 0 Then
    Beep
    Exit Sub
End If
Call ViewRecord
```

السطر الأول :

If Not rs.BOF Then rs.MoveFirst

ومعنى الكود السابق لفظيا هو إذا كان موقع السجل ليس بالأول فانتقل إلى السجل الأول .
والخاصية BOF هي المسؤولة عن إرجاع القيمة التي تشير إلى موقع السجل المعروض ، و MoveFirst هي المسؤولة عن تحقيق الانتقال و في حالة عدم تحديد الجملة الشرطية سنقع في خطئ برمجي فادح.

إما باقي الكود :

```
If rs.BOF And rs.RecordCount > 0 Then
    Beep
    Exit Sub
End If
```

وفي الحالة الأخرى وإذا كانت نقطة موقع العرض موجودة على السجل الأول وكانت RecordCount اكبر من الصفر فإنها تخرج من طور التنفيذ ولا يتم إتمام الكود . فبإتمام الكود سنحصل على خطأ . بالنسبة للأكواد المتبقية فيها تقريبا بنفس الفكرة ولكن سيختلف المضمون فعوضا عن استخدام MoveFirst سنستخدم مثلا MovePrevious مع إضافات بسيطة .
أما آخر سطر من الكود هو Call ViewRecord فالغاية منه هو إظهار ما تم تنفيذه في الكود الذي سبقه .

ولآن سأعرض باقي الأكواد ، واترك لكم فهم الكود /
كود التراجع أو الانتقال إلى السابق . MovePrevious ونضعه في Command2

PDF created with pdfFactory trial version www.pdffactory.com

```
 If Not rs.BOF Then rs.MovePrevious
 If rs.BOF And rs.RecordCount > 0 Then
    Beep
    rs.MoveFirst
    Exit Sub
 End If
Call ViewRecord
End Sub
```

لعرض السجل التالي MoveNext نستخدم الكود الذي يلي ونضعه في:Command3

```
Private Sub Command3_Click()
'
 If Not rs.EOF Then rs.MoveNext
 If rs.EOF And rs.RecordCount > 0 Then
    Beep
    rs.MoveLast
 End If
'
Call ViewRecord
End Sub
```

ولعرض السجل الأخير MoveLast والكود يسكون موضعه في: Command4

```
Private Sub Command4_Click()
'
If Not rs.EOF Then rs.MoveLast
    If rs.EOF And rs.RecordCount > 0 Then
       Beep
    Exit Sub
 End If
'
Call ViewRecord
End Sub
```

فبهذا سنكون قد انتهينا من إعداد مفاتيح الانتقال ما بين السجلات وعرض البيانات في صناديق النصوص. TextBox

يوجد مثال لدرس مرفق مع الكتاب

PDF created with pdfFactory trial version www.pdffactory.com

إدخال البيانات ، تحريها ، حذفها

مررنا سابقا بعدة مراحل كانت الأولى إجراء اتصال مع قواعد البيانات ومن ثم فتح الجدول وكان أخرها هو عرض السجلات وتصفحها . والآن سنأتي على ذكر العمليات أ و 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 أمنا 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
Add .سيحمل اسم1- Command5
Edit .سيحمل اسم2- Command6

PDF created with pdfFactory trial version www.pdffactory.com

Delete .سيحمل اسم 3- Command7
Save .سيحمل اسم 4- Command8
Cancel .سيحمل اسم 5- Command9

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 ستنجم 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 SUBمفهومة لدى الجميع .

PDF created with pdfFactory trial version www.pdffactory.com

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 "Connot 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 "Connot 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 Boxرقم 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 شخصيا 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
```

PDF created with pdfFactory trial version www.pdffactory.com

لها تقوم 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 كتالي :

```
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 Boxيحوي 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 Commandsوهنا يجب أقفالها فعندما ننتهي من إي إجراء يجب قفل ما تم فتحة أو إتاحته .

```
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
```

يوجد مثال لدرس مرفق مع الكتاب

PDF created with pdfFactory trial version www.pdffactory.com

تحرير أو تعديل البيانات

تحرير البيانات هي العملية التي يتم فيها استبدال البيانات السابقة ببيانات جديدة والفرق الأساسي ما بين عملية إضافة سجل جديد وتعديله تكمن في خاصية AddNew. حيث هذه الخاصية نستخدمها عند إضافة سجل جديد إما في العملية الأخرى وهي التحرير فلا نستخدمها فالسطور القادمة بإذن الله تبين ما سبق .

نبدأ بإضافة المتغير الذي سوف يأتي ذكره والهدف منه هو تعريف البرنامج إن المستخدم الآن يقوم بعملية إضافة أو تحرير وسيفيدنا هذا المتغير في عدم تكرار كتابة الأكواد حيث بدونه سوف نضطر إلى كتابة كود أخر وتخصيص مفتاح ليعمل على حفظ البيانات المعدلة .

المتغير :

```
Dim EditORAdd As Boolean
```

ونضع المتغير في. General

ثم نضيف المتغير إلى مفتاح إضافة سجل جديد Addوجعل قيمته . True ليكون الكود كاملا بالشكل التالي :

```
Private Sub Command5_Click()
'
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
'
 Call EnabledUnEnabled(False)
      Text2.SetFocus
'
      EditORAdd = True
End Sub
```

والآن كلما نقرنا على مفتاح Add ستكون قيمة المتغير EditORAdd دائما . True ثم نضيف الكود التالي إلى مفتاح . Edit

الكود :

```
Private Sub Command6_Click()
'
 Call EnabledUnEnabled(False)
      Text2.SetFocus
      EditORAdd = False
End Sub
```

وهنا نضع قيمة المتغير دائما تساوي ، False إذا مما سبق يتبين لنا إن المتغير إذا كانت قيمته تساوي False فمعنا ذلك إن المستخدم يريد تحرير البيانات إما إذا كانت قيمة المتغير تساوي Trueفذلك يقودنا إلى إن المستخدم بصدد إدخال بيانات جديدة .

والآن سنضيف جملة شرطية إلى مفتاح الحفظ . Save وهي ستحل محل السطر التالي من الكود الموجود في مفتاح الحفظ. Save

الكود المستبدل :

```
.AddNew
```

الكود الجديد :

PDF created with pdfFactory trial version www.pdffactory.com

هل يمكنك ترجمة الكود السابق .؟؟؟! هو ببساطة إذا كانت قيمة المتغير EditORAdd تساوي True فسيتم تنفيذ الخاصية AddNew إما إذا كانت قيمة المتغير EditORAdd لا تساوي True فإنه يتجاهل تنفيذها .

وأيضا سنضيف نفس الجملة الشرطية في آخر الكود من المفتاح . Save انظر الكود بعد آخر تعديل .

```
Private Sub Command8_Click()
'
If Trim(Text2.Text) = "" Then
   MsgBox "Connot contain a Null Value .", vbCritical, "Message"
   Text2.SetFocus
   Exit Sub
End If
'
With rs
     If EditORAdd = True Then .AddNew
'
     ![CategoryName] = Trim(Text2.Text)
     If Not Trim(Text3.Text) = "" Then ![Description] = Trim(Text3.Text)
'
     .Update
End With
'
If EditORAdd = True Then
     rs.Requery
     rs.MoveLast
End If
'
Call EnabledUnEnabled(True)
Call ViewRecord
End Sub
```

وبهذا قد انتهينا من عملية تعديل البيانات فمثل ما جاء آنفا إن الفرق الأساسي بين العمليتين هي خاصية AddNew فحين تنفيذها سيتم إضافة سجل جديد أما حين تجاهلها فسيتم تعديل البيانات . وأيضا قد استخدمنا المتغير لكي يتجاهل خاصية الإنعاش Requey وكذلك خاصية الانتقال إلى السجل الأخير MoveLast

يوجد مثال لدرس مرفق مع الكتاب

تم وبفضل الله تجميع الدروس

للمراسلة:

Plantsman9009@hotmail.com
Moh1394rl@hotmail.com

PDF created with pdfFactory trial version www.pdffactory.com