

السلام عليكم

فكرت ملياً قبل أن أكتب موضوعي الأول في هذا المنتدى لأني أمل أن يكون فيه الفائدة الكبيرة لكم ولا أريد أن أكرر الموضوعات، وقد لاحظت من خلال تصفحي لعدة منتديات وجزئياً هذا المنتدى أن الأعضاء يقومون برفع الأمثلة وخصوصاً من المصادر الأجنبية وفي كثير من الأحيان تكون هذه الأمثلة قوية ونافعة بل ورائعة ولكنها لا تعطى حقها من المتابعة والنقاش وتضيع فائدتها وتختفي في غيابات المنتديات وتهجر دون أن يستفيد منها أحد.

سأقدم في هذه السلسلة مجموعة من الأمثلة من مصادر أجنبية مختلفة مبينا مصادرها (لحفظ الحقوق)، وسأقدمها بتصرف مع لمساتي الشخصية ومع ترجمة وشروحات ومتابعة حثيثة.

أملأ من المولى أن يوفقتي وإياكم لهذا العمل الذي لا ابتغي من خلاله إلا رضى الله سبحانه وتعالى.

تحياتي لكم

المثال الأول

كيف نحاكي قدرات الجر والإفلات drag-and-drop في أكسس .

المقدمة

في أكسس لا يوجد حدث معرف لعملية الجر والإفلات (DragDrop) كما هو الحال في فيجوال بيسك، في هذا المثال سيتم شرح كيفية عمل إجراء يعمل في أكسس لعملية الجر والإفلات من عنصر تحكم لأخر وتحديدًا من مربع قائمة لأخر وإذا تمت العملية مع استمرار الضغط على زر Ctrl فإن جميع البيانات الموجودة في القائمة الأولى سوف تنقل إلى القائمة الثانية.

الإجراءات

الإجراءات الأربعة الآتية سيتم استخدامها في هذا المثال وهي :

كود

```
Sub DragStart(SourceFrm As Form)
```

هذا الإجراء يستخدم النموذج المصدر sourceFrm الذي يحتوي على عنصر التحكم الذي سيتم جره (يعني جر البيانات التي فيه)

كود

```
Sub DragStop()
```

هذا الإجراء لإيقاف عملية الجر

كود

```
Sub DropDetect(DropFrm As Form, DropCtrl As Control, _  
    Button As Integer, Shift As Integer, _  
    x As Single, Y As Single)
```

حيث :

- **DropFrm** النموذج الذي ستقلت فيه البيانات
- **DropCtrl** العنصر الذي ستقلت فيه البيانات
- **Button** ، **X** ، **Y** المعلومات التي سنحصل عليها من حدث **MouseMove**

هذا الإجراء يمكن عنصر تحكم من أن يكون عنصر الإفلات (الذي سنعبئ البيانات فيه) ، يجب أن يطلب هذا الإجراء في حدث **MouseMove** للعنصر الذي يمكن أن يكون الهدف (الذي ستقلت عنده البيانات)

كود

```
Sub DragDrop(DragFrm As Form, DragCtrl As Control, DropFrm As Form, DropCtrl As Control, _  
    Button As Integer, Shift As Integer, x As Single, Y As Single)
```

حيث :

- **DragFrm** النموذج الذي سنجر منه
- **DragCtrl** العنصر الذي سنجر منه
- **DropFrm** النموذج الذي ستقلت فيه البيانات
- **DropCtrl** العنصر الذي ستقلت فيه البيانات
- **Botton** وضع أزرار الماوس عند عملية الإفلات
- وضع زر **Shift ALt Ctrl** عند إجراء الإفلات
- **X,Y** إحداثيات الماوس عند تنفيذ عملية الإفلات

هذا الإجراء سيتم طلبه للاستجابة على عملية الجر والإفلات من عنصر مفعل لأن يكون عنصر جر إلى عنصر مفعل لأن يكون عنصر إفلات. في هذا الإجراء يمكن أن نعدل في الكود كي يتناسب مع ما نحتاجه أو ما نريد عمله على سبيل المثال يمكن أن ننسخ محتويات العنصر المجرور في العنصر المستقبل (عنصر الإفلات)

ملاحظات حول تطبيق المثال

١. ملاحظة الوحدة النمطية **basdragdrop** والتي تحتوي على الإجراءات المشار إليها أعلاه

٢. لاحظ حقل **ysnSelected** الموجود في الجدول **tblDragListBox**

٣. ملاحظة جميع الأحداث المعرفة على عنصري التحكم مربع قائمة **list ١** و **list ٢** في النموذج **frmDragDropListBoxes**

وربما تتسألون لماذا كان هناك تعريف للنموذج المصدر الذي سنجر منه وللنموذج الهدف الذي ستقلت فيه البيانات، والجواب هو أننا في مثالنا الثاني سوف نقوم بعملية الجر من نموذج والإفلات في نموذج آخر

ما زلنا في نفس الموضوع وهو

المقدمة

كما ذكرت سابقا في أكسس لا يوجد حدث معرف لعملية الجر والإفلات (DragDrop) كما هو الحال في فيجوال بيسك

الآن سأقدم مثال آخر وهو جر عنصر تحكم من نموذج وإفلاته في نموذج آخر، طبعا الحديث هنا يدور عن نفس الإجراءات التي تم التعرف عليها وتعريفها سابقا مع طريقة معالجة جديدة للمثال الجديد

الإجراءات

كود

```
Sub DragDrop(DragFrm As Form, DragCtrl As Control, DropFrm As Form, DropCtrl As Control, _
    Button As Integer, Shift As Integer, x As Single, Y As Single)
    ' Which form was dropped on?
    ' It is a good idea to use the DragDrop procedure to
    ' determine which drag-and-drop operation occurred; then call
    ' appropriate code to handle the special cases.
    Select Case DropFrm.Name
        Case "frmDragDropListBoxes"
            ListBoxExample DragFrm, DragCtrl, DropFrm, DropCtrl, _
                Button, Shift, x, Y
        Case Else
            ' For all other cases, copy contents of Drag to Drop
            ' control.
            On Error Resume Next
            DropCtrl = DragCtrl
            If Err Then MsgBox Error$
    End Select
End Sub

Sub ListBoxExample(DragFrm As Form, DragCtrl As Control, DropFrm As Form, DropCtrl As Control, _
    Button As Integer, Shift As Integer, x As Single, Y As Single)

    Dim conn As ADODB.Connection
    Dim strSQL As String

    Set conn = CurrentProject.Connection

    ' create SQL statement to update Selected field of
    ' .. drag/dropped list box item.
    strSQL = "UPDATE tblDragListBox SET ysnSelected="

    ' drag from List1 toggle Selected=True, List2 toggles False.
    strSQL = IIf(DragCtrl.Name = "List1", strSQL & "True", strSQL & "False")

    ' if CTRL key not used, alter dragged value only.
    If (Shift And CTRL_MASK) = 0 Then
        strSQL = strSQL & " WHERE pkeyCustomerID='" & DragCtrl & "'"
    End If

    ' run update query to toggle Selected field of Customer record(s).
    conn.Execute strSQL
```

```
' requery the list box controls to show update lists.  
DragCtrl.Requery  
DropCtrl.Requery End Sub
```

لننتبه جيدا كيف يتم تكييف الإجراءات حسب المسألة :

أولا : نلاحظ أن إجراء الإفلات لا يتغير والذي يغير طبيعة (نتيجة) الإفلات هو استدعاء إجراء العمل وهو الذي يتغير من مثال إلى آخر

كود

```
Case DropFrm.Name  
    Case "frmDragDropListBoxes"  
        ListBoxExample DragFrm, DragCtrl, DropFrm, DropCtrl, _  
            Button, Shift, x, Y  
    Case Else  
        ' For all other cases, copy contents of Drag to Drop  
        ' control.  
        On Error Resume Next  
        DropCtrl = DragCtrl  
        If Err Then MsgBox Error$  
End Select
```

لاحظ أن هذا الكود يخدم المثالين فإذا كان النموذج **frmdragdroplistboxes** مفتوحا (وهو المثال الأول) فإنه يتم استدعاء الإجراء الخاص بذلك وهو في هذه الحالة **ListBoxExample** وعدا ذلك تتم فقط عملية الإفلات يعني في المثال الثاني وهو عملية الجر من نموذج والإفلات في آخر تتم العملية ببساطة من خلال الأمر **DropCtrl = DragCtrl** وهذه هي أبسط عمليات الإفلات أن نعبئ البيانات في حقل الإفلات دون أي إجراء إضافي.

ثانيا : نلاحظ أن الإجراء **ListBoxExample** هو خاص بالمثال الأول فقط وهو يقوم بضبط الأمور بعد الإفلات وذلك بانعاش مربع القائمة الأول ومربع القائمة الثاني بعد الجر والإفلات فإن أحدا القوائم ستزيد عنصرا والأخرى سنقص عنصرا وهذا هو ما يعالجه هذا الإجراء.

مرحلة التطبيق والتجربة

ما زلنا في موضوعنا الأول وهو

محاكاة قدرات الجر والإفلات **drag-and-drop** في أكسس

المقدمة

هذه من أهم المراحل بعدما نتعلم شيء معين يجب أن نطبقه ونستخدمه ولذلك قمت بعمل مثال يقوم بالتلوين باستخدام فكرة الجر والإفلات كما في الصورة



وفي هذا المثال البسيط قمت باستثمار ما تعلمناه سابقا وجدير بالملاحظة هنا ما يأتي

الإجراءات

الأحداث التي نستخدمها هي ثلاثة لكل لون حيث يعتبر هو العنصر المراد جره ومثال ذلك أحداث اللون الأحمر ١c وهي :

كود

```
Private Sub cl_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    DragStart Me
```

```
End Sub
```

```
Private Sub c1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    DropDetect Me, Me![c1], Button, Shift, X, Y
```

```
End Sub
```

```
Private Sub c1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    DragStop
```

```
End Sub
```

وحدث مربع النص المستقبل الذي سيتم عنده الإفلات وهو

كود

```
Private Sub dropto_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    DropDetect Me, Me![dropto], Button, Shift, X, Y  
End Sub
```

وطبعا في النهاية لا ننسى وضع الكود المناسب في الوحدة النمطية لإجراء عملية التلوين ويتم ذلك كالآتي :

كود

```
Sub Example3(DragFrm As Form, DragCtrl As Control, DropFrm As Form, DropCtrl As Control, _  
    Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    DropCtrl.ForeColor = DragCtrl.ForeColor  
    DropCtrl.Requery
```

```
End Sub
```

هذا كل شيء، كما ترون أن الأمر بسيط بعد أن تم فهمه

المثال الثاني

كيفية فتح مثيلات متعددة من نموذج MultipleInstance

المقدمة

هذا المثال يوضح كيفية استخدام **VBA : Visual Basic for Applications** لفتح المثيلات المتعددة لنموذج (نسخ عن النموذج) ويمكنك هذا من الاستفادة من هذه المثيلات باستطلاع عدة سجلات معا في آن واحد وبالتأكيد فإن الفائدة ستكون كبيرة في مجالات عدة.

التفاصيل

إن فتح مثيلات النموذج يتضمن بناء الجملة **Set** وذلك بواسطة عبارة تعيين الكلمة الأساسية (جديد) ومرجع للنموذج كما في المثال الآتي :

Set x = New Form_Form

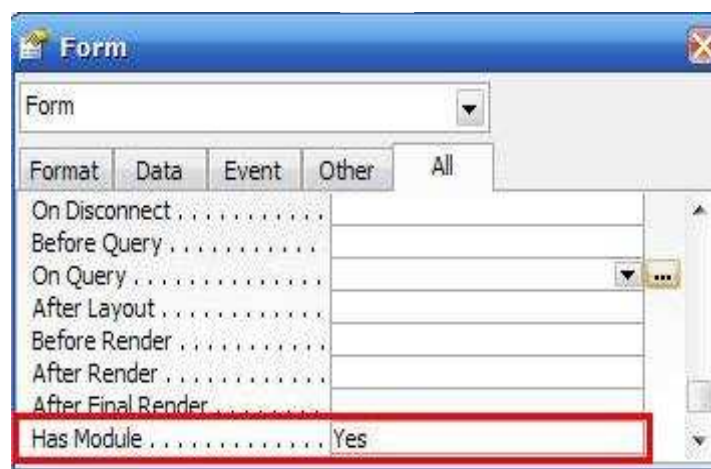
لأنك لا تستخدم الكلمة الأساسية **New** لإنشاء مثيل جديد من كائن خفيف (**lightweight Object**)، فيجب تعيين الخاصية **HasModule** الخاصة به إلى **True** يعني للنموذج الذي تنوي استخدام الكلمة الأساسية جديد **New** له.

عندما تقوم بفتح المثلثات المتعددة من نموذج فإن المثلث الأصلي (النموذج الأصلي) هو الكائن (النموذج) الوحيد الذي يتم تخزينه في قاعدة البيانات في النهاية عند إغلاقه. المثلثات الأخرى هي مؤقتة ويتم إزالتها من الذاكرة عندما تقوم بإغلاقها.

ملاحظة : عند إغلاق المثلث الأصلي (النموذج) سيتم إغلاق كافة المثلثات الأخرى منه كذلك.

ولفهم تطبيق هذا المثال وكيفية فتح المثلثات المتعددة من نموذج، يجب ملاحظة الخطوات الآتية :

١. تعيين خاصية **HasModule** له كائن إلى **true** كما ذكرنا سابقا كالآتي :



٢. حدث الضغط على زر أمر فتح مثيل جيد عن النموذج

كود

```
Private Sub cmdOpen_Click()  
Set frmNewForm = New Form_frmMultipleInstance  
    frmNewForm.SetFocus  
    intInstanceNum = intInstanceNum + 1  
    frmNewForm.Caption = "المثيل رقم - " & intInstanceNum  
End Sub
```

لاحظ هنا أننا أولاً نطلب مثيل جديد عن النموذج وما يمكننا من ذلك هو التعريف في الخطوة ١ و ثم ننقل التركيز إلى النموذج الجديد ونطلق عليه (المثيل رقم - ؟) حيث يستخدم المتغير **intInstanceNum** لمتابعة أرقام المثلثات التي تفتح.

٣. حدث فتح النموذج الأصلي

كود

```
Private Sub Form_Open(Cancel As Integer)

    DoCmd.MoveSize xPos, yPos

    xPos = xPos + 300
    yPos = yPos + 300

End Sub
```

يتم تعريف المتغير **xPos** و **yPos** و **intInstanceNum** في الوحدة النمطية بشكل **Public** لتعرف في كل مكان ويتم تحديد موقع ظهور النموذج ومثيلاته بإزاحة بالاتجاه الأفقي من اليسار إلى اليمين والعمودي من الأعلى للأسفل بمقدار ٣٠٠ نقطة (pixel) لنلا نفتح النماذج فوق بعضها.

٤. ما الذي يغلق المثلثات عندما يغلق الأصل علماً أنه لا يوجد في حدث الإغلاق أي كود يتعلق بذلك ؟

كود

```
Private Sub Form_Close()

    xPos = 300
    yPos = 300
    intInstanceNum = 0

End Sub
```

من المعروف أنه في البرمجة الكينونية عندما ننشأ كائناً (مثيلاً أو أبناً) من آخر فإنه عند إنهاء التعامل مع الكائن الأصلي سيتم إنهاء المثلثات كذلك، واعتقد غير جازماً أن هذا الأمر يتعلق بخاصية تعدد الأشكال أو ما يسمى في أدبيات أخرى تعدد الأوجه وبالانجليزية **polymorphism** ولا اعتقد جازماً أن هذه من خصائص التوارث **inheritance**.

المثال الثالث

كيفية استخدام عدداً ثابتاً من السطور في صفحات التقرير

المصدر : مثال من تصميم المشرف **Dream_Works**

المقدمة

في بعض التطبيقات قد نحتاج إلى عدد ثابت من السطور في كل صفحة، ونحن نعرف أن التحكم في عدد السطور في صفحات التقرير في أكسس يتم تلقائياً حسب حجم الصفحة. في هذا المثال سنتعلم كيف نتحكم في عدد السطور وكيف نغير هذا العدد من تقرير لآخر باستخدام خاصية **openargs** الخاصة بالتقارير عندما نطلبه من نموذج.

التفاصيل

١. للتحكم بعدد السطور يجب تعريف متغيراً يحدد فيه قيمة عدد السطور المطلوبة كالآتي :

٢. في حدث عند الفتح للتقرير نعرف قيمة هذا المتغير وهي عدد السطور المرغوب بها هكذا

كود

```
Private Sub Report_Open(Cancel As Integer)
    OnePageRecords = 15
End Sub
```

٣. في حدث عند التنسيق لقسم التفصيل في التقرير نقوم بزيادة هذا المتغير زيادة رتيبة مقدارها ١ هكذا

كود

```
Private Sub تفصيل_Format(Cancel As Integer, FormatCount As Integer)
    Static Count As Byte
    Count = Count + 1
    If Count = OnePageRecords Then
        Count = 0
        Me.Section(acDetail).ForceNewPage = 1
    End If
End Sub
```

لاحظ أن `Me.Section(acDetail).ForceNewPage = 1` هذا السطر هو الذي سيفرض صفحة جديدة عند وصولنا إلى عدد السطور المطلوب.

٤. ومهم جدا لكي تتم العملية بالشكل المطلوب تعريف ما يلي في حدث الصفحة `page` للتقرير

كود

```
Private Sub Report_Page()
    Me.Section(acDetail).ForceNewPage = 0
End Sub
```

وهذا سيمنع الإجراء الاعتيادي في أكسس من فرض صفحة جديدة كما هو معتاد بشكل طبيعي.

هذا كل شيء والنتيجة هي ما نتمناه وهو طباعة عدداً ثابتاً من السطور في صفحات التقرير.

ملاحظة مهمة: لاحظ التعريف `Static Count As Byte` لأنه مهم بهذه الطريقة (يعني `static`) وإذا عرفناه بطريقة أخرى لن يعمل لأن `Static` تعرف المتغير على مستوى الإجراء وتحفظ بقيمته طالما الكود شغال .

كما تلاحظون فإننا قمنا بتثبيت عدد السطور داخل حدث عند الفتح للتقرير وهذا غير مفضل خصوصاً عندما نطلبه من نموذج، حيث أننا يمكن أن نحتاج أن نغيره في كل مرة نطلبه وهذا يكون من خلال تعيين قيمته بواسطة خاصية `openargs` وذلك لو طلبه من النموذج كما يأتي:

Option Compare Database

```
Private Sub Command0_Click()  
    DoCmd.OpenReport "report1", acPreview, , , , Me.LinesPerPage  
End Sub
```

لاحظ عدد الفواصل

هذه هي خاصية openargs وهو القيمة التي ستمرر للتقرير

المثال الرابع

طريقة مثلى لإحضار المجموع الفرعي SubTotal من النموذج الفرعي إلى النموذج الرئيسي دون مشاكل

المقدمة

كثيرا ما نصمم نموذج رئيسي ونموذج فرعي ويكون في النموذج الرئيس المعلومات الرئيسة وفي الفرعي التفاصيل (الحركات)، وغالبا ما نحتاج إلى إحضار المجموع الفرعي من النموذج الفرعي إلى النموذج الرئيسي وقد نعالج هذه المسألة بطرق مختلفة ومتعددة وصعبة أحيانا وقد تقودنا إلى مشكلة معظمنا شاهدها وهي ? #name ، في هذا المثال سنتعلم طريقة مثلى لتحقيق المطلوب دون مشاكل.

التفاصيل

إذا حاولت جمع حقل في نموذج فرعي باستخدام عنصر تحكم محسوب على نموذج رئيسي قد تتلقى رسالة خطأ في عنصر تحكم النموذج الرئيسي.
التعبير الآتي في عنصر تحكم نموذج رئيسي على سبيل المثال، ينشئ رسالة إعلام بالخطأ "? #Name " الرسالة غير المرغوب بها:

كود

```
=Sum([MySubForm].Form![ExtendedPrice])
```

حيث MySubForm هو اسم النموذج الفرعي

ويمكن لتجنب الخطأ، جمع حقل النموذج الفرعي باستخدام عنصر تحكم محسوب في تذييل النموذج الفرعي، ويمكنك ثم الرجوع لعنصر التحكم المحسوب من النموذج الرئيسي لعرض القيمة المجموعة.

أنظر كيف يتم تعريف قيمة المجموع الفرعي في النموذج الفرعي

Form Header

Detail

القطعة	fkeyProductID	سعر الوحدة	curlUnitPrice
الكمية	intQuantity	الخصم	sngDiscount
		ExtendedPrice	المسعر

Form Footer

=Sum([ExtendedPrice]) المجموع الفرعي

Text Box: txtOrderSubtotal

txtOrderSubtotal

Format Data Event Other All

Name	txtOrderSubtotal
Control Source	=Sum([ExtendedPrice])
Format	Currency
Decimal Places	Auto
Input Mask	
Default Value	
IME Hold	No
IME Mode	No Control
IME Sentence Mode	Phrase Predict
Validation Rule	

وأنظر كيف نطلبها من النموذج الرئيسي

Text Box: txtSubtotal

Format Data Event Other All

Name: txtSubtotal

Control Source: =subSubTotal.Form!txtOrderSubtotal

Format: Currency

Decimal Places: Auto

Input Mask:

Default Value:

IME Hold: No

IME Mode: No Control

IME Sentence Mode: Phrase Predict

Validation Rule:

Validation Text:

Status Bar Text:

Enter Key Behavior: Default

Allow AutoCorrect: Yes

Visible: Yes

Display When: Always

Vertical: No

Form Footer

إغلاق هذا النموذج

المتفرع الفرعي: المجموع الفرعي

إخراج الشحن: الشحن بواسطة: قنصل

الطلبية: pkeyOrd: تاريخ: 5

تأكد أن المراجع للنموذج الفرعي هو من خلف خاصية الاسم للنموذج الفرعي حتى لا تحصل على الخطأ #NAME

وأنظر نتيجة هذا العمل الرائع

المثال الرابع : المجموع الفرعي من النموذج الفرعي

الفاتورة ل: **Alfred's Futterkiste**

Obere Str. 57

Berlin 12209

Germany

الشحن ل: **Alfred's Futterkiste**

Obere Str. 57

Berlin 12209

Germany

اسم البائع: **Buchanan, Steven**

تاريخ الشحن: 21-Mar-95

شحن بواسطة: ☒ سبيدي ☐ يونيتد ☐ فيدرال

مطلوبة قبل: 24-Apr-95

تاريخها: 13-Mar-95

الطلبية: 10952

القطعة	سعر الوحدة	الكمية	الخصم	السعر
Rösle Sauerkraut	\$45.60	2	0%	\$91.20
Grandma's Boysenberry Spread	\$25.00	16	5%	\$380.00

المجموع الفرعي: \$471.20

إجرة الشحن: \$40.42

المجموع: \$511.62

تأكد أن المرحع للنموذج الفرعي هو من خلال
خاصية الاسم للنموذج الفرعي حتى لا تحصل
على الخطأ #NAME

لاحظ أن هذه الطريقة المثلى سهلة ومضمونة النتيجة وبدون استخدام أكواد

المثال الخامس

الفروقات بين مربع القائمة List Box ومربع التحرير والسرد Combo Box

المقدمة

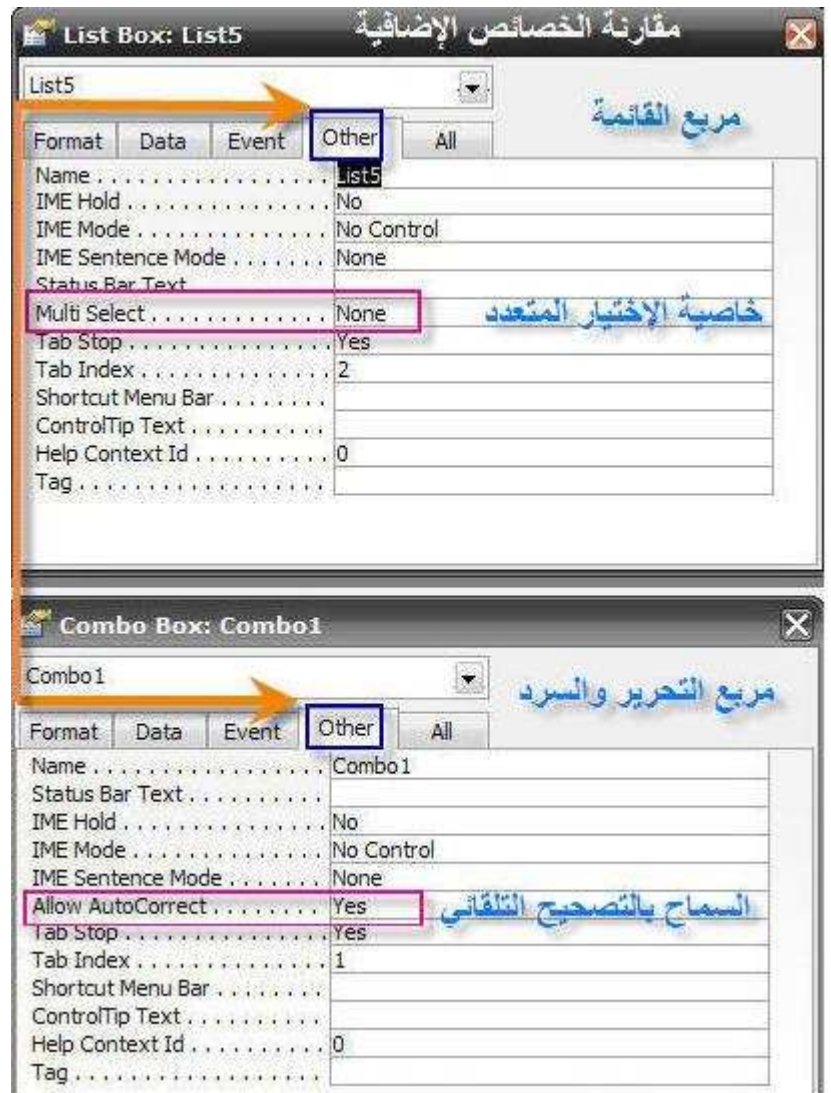
تسأل أحد الأعضاء في إحدى المشاركات (هنا) إن كان بالإمكان إخفاء سهم مربع التحرير والسرد ليظهر كأنه مربع القائمة. ويتضح ضمنا من سؤاله أنه يود أن يستخدم أحد الخصائص التي تتوفر في مربع التحرير والسرد وبنفس الوقت يستفيد من شكل مربع القائمة. إن هذا السؤال في محله بلا شك، رغم أن الكثيرين ممن لا يدققون في الأمور كثيرا قد يظنوا أن مربع القائمة لا يختلف عن مربع التحرير والسرد سوى في الشكل. في هذا المثال سيتم إلقاء الضوء على الفروقات بين List Box و Combo Box.

التفاصيل

لو دققنا في التسمية فقط نستطيع معرفة الفرق الأساس وهو أنه في مربع التحرير والسرد يسمح بالكتابة وصولا للمطلوب من القائمة بينما في مربع القائمة يتم انتقاء المطلوب بواسطة الماوس، ومما لا شك فيه أن هذا الفرق الأساس والجوهري سيقودنا إلى مجموعة تغيرات (فروقات) في الخصائص الخاصة بكل منهما، فما هي هذه الخصائص المختلفة؟

يوجد سبعة عشر فرقا بين مربع القائمة ومربع التحرير والسرد يمكن ملاحظتها بالتدقيق بالخصائص لكل منها كما يأتي :

١. في لسان التبويب other الخصائص الأخرى أو الإضافية



نلاحظ هنا أنه يمكن في مربع القائمة أن نحدد أكثر من اختيار بينما في مربع التحرير والسرد غير ممكن وهذه هي الخاصية الوحيدة الموجودة في مربع القائمة وليست موجودة في مربع التحرير والسرد. بمعنى آخر، فإن باقي الخصائص الستة عشر موجودة في مربع التحرير والسرد وغير موجودة في مربع القائمة.



لاحظ أربعة أحداث غير موجودة في مربع القائمة

List Box: List5 مقارنة فوارق البيانات

List5

مربع القائمة

Format Data Event Other All

Control Source

Row Source Type Table/Query

Row Source

Bound Column 1

Default Value

Validation Rule

Validation Text

Enabled Yes

Locked No

Smart Tags

Combo Box: Combo1

Combo1

مربع التحرير والسرد

Format Data Event Other All

Control Source

Input Mask

Row Source Type Table/Query

Row Source

Bound Column 1

Limit To List No

Auto Expand Yes

Default Value

Validation Rule

Validation Text

Enabled Yes

Locked No

Smart Tags

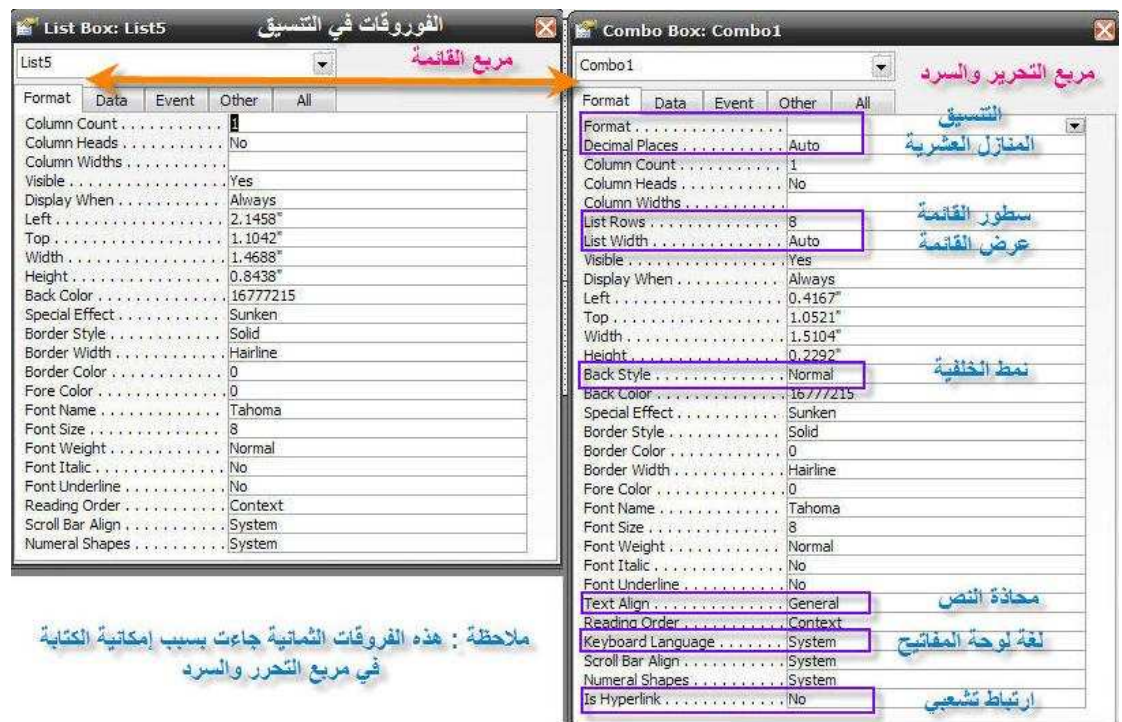
فناع الإدخال

الالتزام بالقائمة

التوسيع التلقائي

لاحظ ثلاث خصائص غير موجودة في مربع القائمة

٤. في لسان التبويب format التنسيق



ملاحظة : هذه الفروقات الثماتية جاءت بسبب إمكانية الكتابة في مربع التحرير والسرد

لاحظ ثمان تنسيقات غير موجودة في مربع القائمة

وهنا نأتي إلى تفسير مشاركة العضو بطلبه استخدام شكل مربع القائمة في مربع التحرير والسرد: مما لا شك فيه أنه يود الناحية الجمالية من مربع القائمة ويود استخدام أحد أو أكثر من الخصائص الستة عشر التي تم توضيحها أعلاه والخاصة بمربع التحرير والسرد.

ما زلنا في المثال الخامس

مرحلة التطبيق والتجربة

ما زال الحديث دائراً حول إخفاء السهم الذي يظهر في مربع التحرير والسرد وقد عرفنا لماذا قد نحتاج إلى ذلك. في الأمثلة الآتية سأعالج الأمر بطريقتين :

١. طريقة خداع اللون

إقتباس

من مربع الأدوات أضف مستطيل إلى النموذج واجعل حجمه يساوي تماماً المربع الخاص بالسهم لمربع التحرير والسرد وضعه فوق مربع السهم ليغطيه تماماً

واضبط خصائص المستطيل كما يلي:-

- لون الخلفية او (BackColor) غيره ليصبح مثل لون خلفية النموذج.

- نمط الخلفية او (BackStyle) اجعله عادي (Normal) وهذه هي القيمة الافتراضية لهذه الخاصية اتركها كما هي.

- نمط الحدود او (BorderStyle) اجعله شفاف (Transparent).

بهذا فان سهم مربع التحرير والسرد لن يظهر سوى عند التركيز عليه فقط وغير ذلك سيظهر كمربع نص. وهي طريقة رائعة سهلة التطبيق وناجحة ١٠٠ %

٢. طريقة إسناد قيمة مربع التحرير والسرد إلى مربع نص يوضع فوقه

وباستمرار الدراسة والبحث في هذا الموضوع الشيق استطعت تصميم طريقة أخرى لعلها بالألوان.

بحيث يعمل مربع نص ويجعل مصدره مربع التحرير والسرد ويوضع فوقه تماما ونغير خاصية (visible) لمربع التحرير والسرد إلى (لا) في وضع التصميم ليتم اخفائه عندما يفتح النموذج، وبواسطة حدث الدخول لهذا المربع وحدث الخروج من مربع التحرير والسرد يمكن تنفيذ المطلوب كما يأتي

كود

```
Private Sub Text_Enter()  
Me.Combo.Visible = True  
Me.Combo.SetFocus  
Me.Text.Visible = False  
End Sub  
Private Sub combo_Exit(Cancel As Integer)  
Me.Text.Visible = True  
End Sub
```

المثال السادس

كيفية التحقق من أن النموذج في وضع التعديل تلقائيا

المقدمة

كثيرا ما يتم تصميم النماذج لتلعب دور الإدخال ودور الإخراج معا، بمعنى أن النموذج يستعرض مجموعة السجلات وكذلك يستخدم لإضافة سجل أو تعديل سجل معروض وقد يؤدي هذا الأمر عن طريق الخطأ وخصوصا إذا كان التحكم في النموذج ضعيفا (أو مفقودا) فإن هذا قد يؤدي إلى ضياع بعض البيانات المهمة. في هذا المثال سيتم وصف أسلوبين يمكن استخدامهما للتحقق من أن النموذج في وضع التعديل تلقائيا ويمكن الاستفادة من هذا الأمر في زيادة التحكم في النموذج لعمل الإجراءات المناسب في هذا الوضع أو للتراجع عن التعديلات إن أحببنا.

الفوائد الجانبية للمثال

١. كيفية التراجع عن مجموعة التعديلات على سجل .
٢. كيفية التحكم في صفحة النموذج
٣. التعرف على حدث Dirty والذي يتفعل قبل أن يتم التعديل على حقل أو سجل

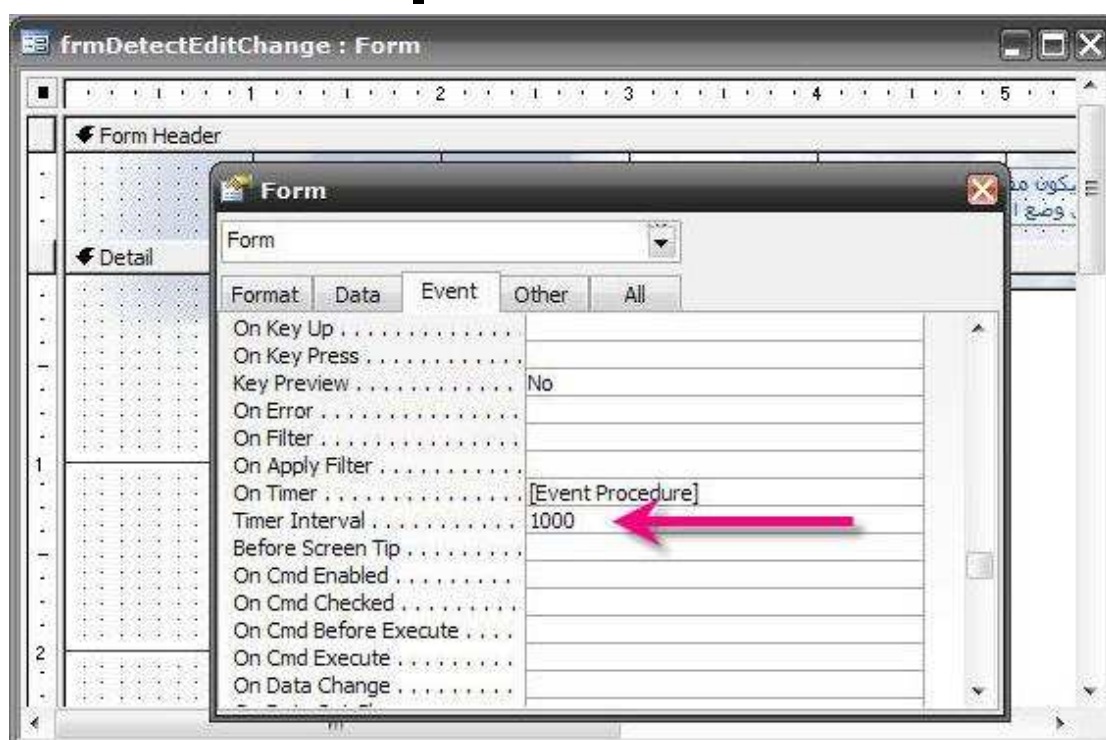
الأسلوبان المراد اتباعهما لتحقيق هدف المثال هما :

١. استخدام حدث عداد الوقت للنموذج
٢. استخدام خاصية dirty في تعبير

التفاصيل

الأسلوب الأول : استخدام حدث عداد الوقت للنموذج

١. يتم تحديد الزمن هكذا



٢. يتم تعريف إجراء عداد الوقت للنموذج هكذا

كود

```
Private Sub Form_Timer()  
    ' the Timer Interval has been set at 1000, that is, once per second.  
  
    Static bFlag As Boolean  
    If Me.Dirty Then  
        If Not bFlag Then  
            Me!cmdUndo.Enabled = True  
            bFlag = True  
        End If  
    Else  
        If bFlag Then  
            Me!txtFirstName.SetFocus  
            Me!cmdUndo.Enabled = False  
            bFlag = False  
        End If  
    End If  
End Sub
```

وما يقوم به هذا الإجراء هو التحقق بشكل دوري كل (ثانية) حسب الزمن المحدد في خطوة ١ إذا تم أي تعديل في السجل وتبديل حالة زر التراجع عن التعديلات بين التفعيل وعدمه وهكذا نكون قد حققنا ما نصبو له بحسب حالة هذا الزر.

٣. كما ذكرت سابقا فإن احد إجراءات التحكم في النموذج قد تكون التراجع عن هذه التعديلات وذلك يكون هكذا

كود

```
Sub cmdUndo_Click()
    ' same action as clicking Undo from the Edit menu
    DoCmd.RunCommand acCmdUndo
End Sub
```

والنتيجة تكون كأننا ضغطنا على زر التراجع عن التغييرات **ctrl+z** عدة مرات

الأسلوب الثاني : استخدام خاصية **dirty** في تعبير

١. تعريف الإجراء التالي في وحدة نمطية

كود

```
Function EditModeChange (F As Form) As Variant

    If F.Dirty Then
        F!btnUndo.Enabled = True
    Else
        F!btnUndo.Enabled = False
    End If

End Function
```

٢. تعريف حدث بعد التحديث للنموذج كما يأتي

كود

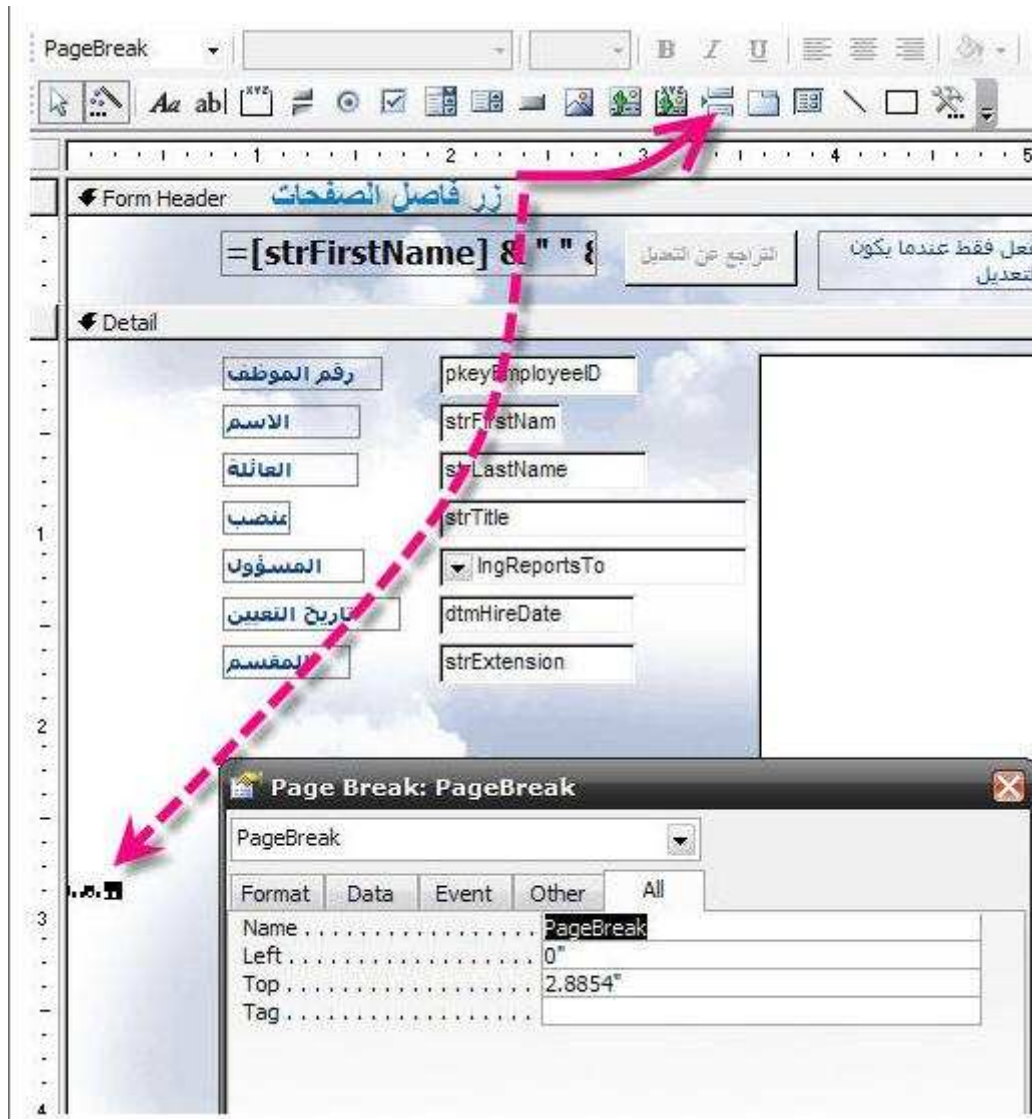
```
Sub Form_AfterUpdate ()
    Me!txtEditModeChange.Requery
End Sub
```

وهذا بدوره سيحقق المطلوب كما هو الحال باستخدام الأسلوب الأول

من الفوائد الجانبية لهذا المثال كيفية عمل صفحات في النموذج والتنقل بينها

وذلك يتم كما يلي

١. في وضع التصميم أضف **pagebreak** فاصل صفحات في موضع الفصل كم يأتي :



٢. ثم بناء على أزرار الأمر يتم التوجه للصفحة المطلوبة هكذا

كود

```
Private Sub cmdCompanyInfo_Click()
```

```
' move to first page of form.  
Me.GoToPage 1  
End Sub[/size]  
  
Private Sub cmdPersonalInfo_Click()  
' move to second page of form.  
Me.GoToPage 2  
End Sub
```

المثال السابع

أربعة أساليب لمنع ظهور رسائل التحذير التي ستظهر قبل تنفيذ استعلاماً إجرائياً

المقدمة

عندما نطلب تنفيذ استعلاماً إجرائياً لإضافة سجل جديد أو تعديل بيانات سجل أو لحذف سجل فإن رسالة تحذيرية ستظهر قبل تنفيذ الإجراء ستطالب المستخدم بتأكيد تنفيذ الإجراء. في بعض البرامج يتضايق المستخدم من هذه الرسائل. في هذا المثال سنتعرف على أربعة أساليب لمنع ظهور هذه الرسائل.

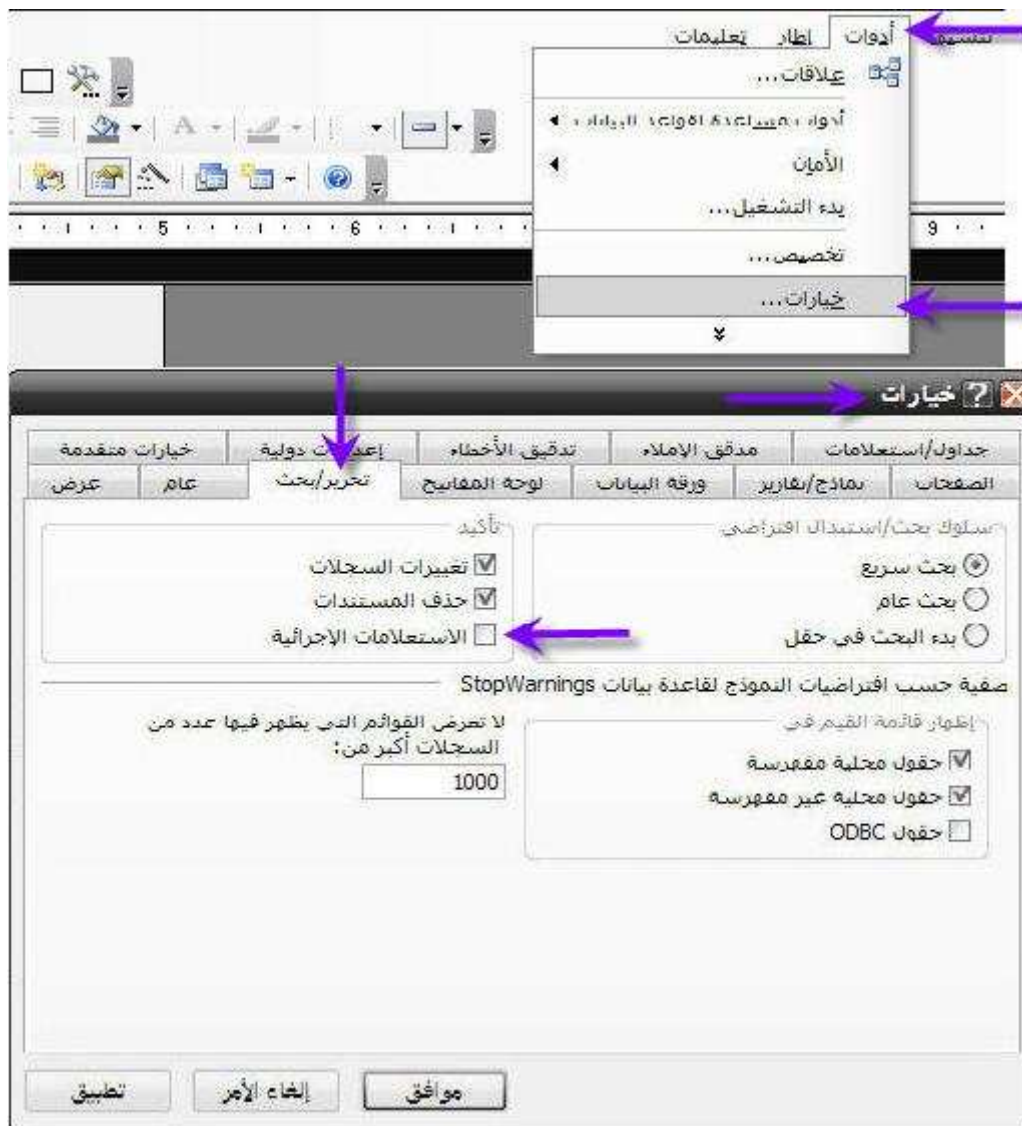
التفاصيل

الأساليب التي سنستخدمها لمنع الرسائل التحذيرية هي

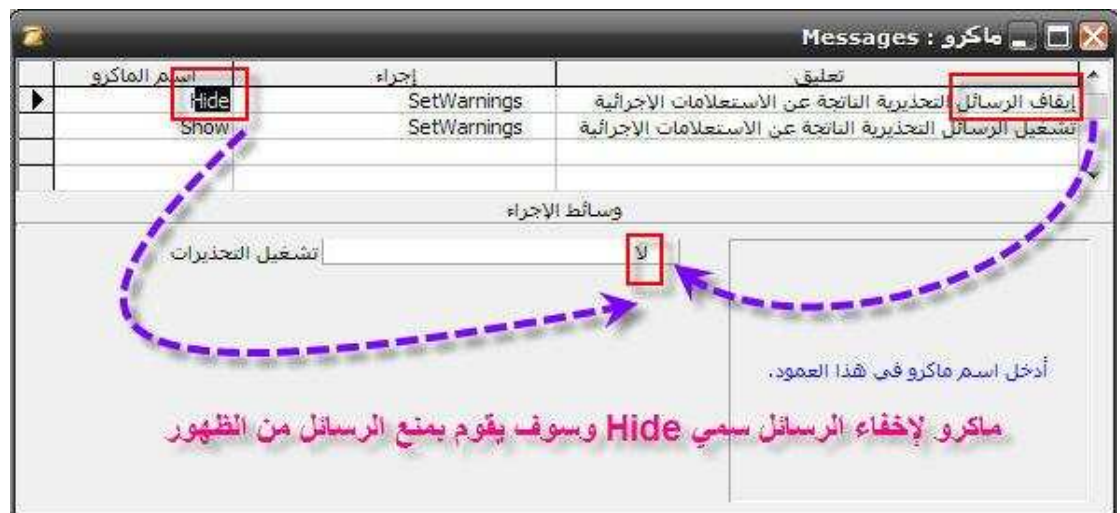
١. الطريقة اليدوية من الخيارات
٢. طريقة استخدام الماكرو
٣. باستخدام **CurrentDB.Execute**
٤. باستخدام **DoCmd.SetWarnings**

والآن لننتعرف على تفاصيل كل أسلوب من هذه الأساليب

الطريقة الأولى: يدويا من الخيارات وذلك يتم بحسب الشكل الآتي :



الطريقة الثانية : بواسطة الماكرو وذلك يتم بعمل ماكرو كما يأتي



الماكرو الأول Hide سيمنع الرسائل من الظهور
والماكرو الثاني Show سيعمل على إظهار الرسائل

ويطلب هكذا

كود

```
Private Sub أمر١_Click()
Dim SQL As String

SQL = "insert into students (name) " & _
      "values ('طالب')"
```

DoCmd.RunMacro "Messages.Hide"
DoCmd.RunSQL SQL
DoCmd.RunMacro "Messages.Show"

```
End Sub
```

الطريقة الثالثة : باستخدام CurrentDB.Execute لتنفيذ الإجراء بدلا من DoCmd.RunSQL هكذا

كود

```
Private Sub أمر٢_Click()
Dim SQL As String

SQL = "insert into students (name) " & _
      "values ('طالب')"
```

CurrentDb.Execute SQL

```
End Sub
```

الطريقة الرابعة : بواسطة الأمر DoCmd.SetWarnings هكذا

كود

```
Private Sub أمر٣_Click()
```



```
Dim SQL As String

SQL = "insert into students (name) " & _
      "values ('طالب')"
```

```
DoCmd.SetWarnings False
DoCmd.RunSQL SQL
DoCmd.SetWarnings True

End Sub
```

المثال الثامن

استخدام طرق البحث في البيانات بواسطة اقترانات البحث في مجال **functions aggregate**

المقدمة

لاحظت من خلال المشاركات أن بعض المشاركين يعانون من صعوبة البحث في الجداول باستخدام اقترانات تحديد المجال ويواجهون صعوبة في وضع شروط البحث. قمت بتصميم هذا المثال ليساعد في فهم الطريقة التي يتم بها البحث وطريقة تحديد الشروط التي لها علاقة بنوع البيانات المستخدمة.

التفاصيل

من اقترانات البحث في مجال معين مثل جدول **DMax** ، **DAvg** ، **DCount** ، **DSum** ، **DLast** ، **DFirst** ، **Dlookup** و **DMin** وغيرها

في المثال تم بناء جدول بيانات الطلاب : الرقم والاسم وتاريخ الميلاد و المصروف و إن كان للطلاب أخ وقد اختيرت هذه البيانات بالتحديد لاختلاف أنواع البيانات فيها فالرقم (رقم **Number**) والاسم (نص **Text**) وتاريخ الميلاد (تاريخ **Date**) والمصروف (رقم مزدوج **Double**) وله أخ (منطقي **Boolean**) ولكي نتعلم كيف نستخدم هذه العناصر في الشروط.

وقد قمت في هذا المثال بعمل عدة أسئلة وهي

- البحث عن طالب بواسطة الرقم ؟
- البحث عن طالب بواسطة الاسم ؟
- البحث عن طالب بواسطة تاريخ الميلاد ؟
- البحث عن طالب بواسطة الرقم و الاسم معا ؟
- البحث عن طالب بواسطة وقوع تاريخ ميلاده بين تاريخين محددين ؟
- البحث عن عدد الطلاب بواسطة سنة تاريخ الميلاد ؟
- البحث عن أول طالب بواسطة إبتداء اسمه بأحرف معينة؟
- البحث عن أصغر تاريخ ميلاد ضمن المجموعة ؟
- البحث عن مجموع المصاريف لكل الطلاب ؟
- البحث عن عدد الطلاب الذين لهم أخ ؟
- البحث عن مجموع مصروف الطلبة الذين ليس لهم أخ وولدوا في شهر بعد الذي سيتم تحديده ؟

- البحث عن معدل مصروف الطلبة الذين لم يولدوا في السنة المحددة ؟
- البحث عن آخر طالب في الجدول له أخ ؟
- عرض اسم وتاريخ ميلاد أول طالب يظهر في الجدول؟

ولنناقش هذه الطرق والأسئلة

١. البحث عن طالب بواسطة الرقم ؟

كود

```
Private Sub cmd1_Click()  
no = InputBox("الرجاء إدخال رقم الطالب", "إدخال")  
w = Nz(DLookup("[sno]", "[student]", "[sno] = " & Str(no)), Empty)  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
End Sub
```

استخدمت هنا الأقواس [] لحصر أسماء الحقول والجدول وهذا غير ضروري إذا كان الحقول من مقطع واحد ولكنه مهم جدا عند تكون الحقول أو اسم الجدول من مقطعين كأن يكون اسم الجدول مثلا جدول الطلاب أو يكون حقل التاريخ اسمه تاريخ الميلاد

الاقتران Nz يضم لنا عدم إرجاع القيمة الفارغة Null فإذا كان الجواب Null يرجع لنا كجواب للبحث القيمة الخالية Empty ومنها نعرف نتيجة البحث

لاحظ هنا كيف تم حصر الرقم بالاقتران str لتحويله إلى نص string

٢. البحث عن طالب بواسطة الاسم ؟

كود

```
Private Sub cmd2_Click()  
N = InputBox("الرجاء إدخال الاسم", "إدخال")  
w = Nz(DLookup("[sname]", "[student]", "[sname] = '" & N & "'"), Empty)
```

```

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "لا يوجد نتيجة"
End If

End Sub

```

لاحظ هنا مع الحقل من نوع نص كيف نستخدم علامات التنصيص المفردة ' للإحاطة بالمتغير النصي

٣. البحث عن طالب بواسطة تاريخ الميلاد ؟

كود

```

Private Sub cmd3_Click()
D = InputBox("الرجاء إدخال تاريخ الميلاد", "إدخال")

w = Nz(DLookup("[sdate]", "[student]", "[sdate] = #" & D & "#"), Empty)

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "نتيجة لا يوجد"
End If

End Sub

```

لاحظ هنا كيف نستخدم إشارة النمبر # للإحاطة بحقل من نوع تاريخ

٤. البحث عن طالب بواسطة الرقم و الاسم معا ؟

كود

```

Private Sub cmd4_Click()
no = InputBox("الرجاء إدخال رقم الطالب", "إدخال")
N = InputBox("إدخال الاسم", "إدخال الرجاء")

w = Nz(DLookup("[sno]", "[student]", "[sno] = " & Str(no) & " and [sname] = '" & N & "'"), _
Empty)

If Not IsEmpty(w) Then
    MsgBox "الطالب موجود"
Else
    MsgBox "نتيجة لا يوجد"
End If

End Sub

```

٥. البحث عن طالب بواسطة وقوع تاريخ ميلاده بين تاريخين محددتين ؟

كود

```

Private Sub cmd5_Click()
D1 = InputBox("الرجاء إدخال التاريخ الأول", "إدخال")
D2 = InputBox("إدخال التاريخ الثاني", "الرجاء")

w = Nz(DLookup("[sdate]", "[student]", "[sdate] between #" & D1 & "# and #" & D2 & "#"), Empty)

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "نتيجة لا يوجد"
End If

End Sub

```

عملية استخدام **between** ولا تنسى أن حدي البحث متضمنان في البحث

٦. البحث عن عدد الطلاب بواسطة سنة تاريخ الميلاد ؟

كود

```
Private Sub cmd6_Click()  
Y = InputBox("إدخال سنة تاريخ الميلاد", "إدخال الرجاء")  
  
w = DCount("*", "[student]", "year([sdate]) = " & Str(Y))  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

لاحظ هنا طريقة المقارنة مع السنة وهي رقم

٧. البحث عن أول طالب بواسطة إبتداء اسمه بأحرف معينة ؟

كود

```
Private Sub cmd7_Click()  
N = InputBox("إدخال بداية الاسم", "إدخال الرجاء")  
  
w = Nz(DFirst("[sname]", "[student]", "[sname] Like ' " & N & "'"), Empty)  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

لاحظ كيف استخدمنا **Like** وأين وضعنا *

٨. البحث عن أصغر تاريخ ميلاد ضمن المجموعة ؟

كود

```
Private Sub cmd8_Click()  
  
w = Nz(DMin("[sdate]", "[student]"), Empty)  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

في بعض الأحيان لا نحتاج إلى شروط في البحث

٩. البحث عن مجموع المصاريف لكل الطلاب ؟

كود

```
Private Sub cmd9_Click()  
w = Nz(DSum("[sexp]", "[student]", Empty)  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

١٠. البحث عدد الطلاب الذين لهم أخ ؟

كود

```
Private Sub cmd10_Click()  
  
w = Nz(DCount("[sexp]", "[student]", "[shasbrother] = True"), Empty)  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

لاحظ هنا كيف استخدمنا البحث عن حقل نعم/لا **boolean** منطقي واستخدام **true** ويمكن أيضا المقارنة العدد - ١

١١. البحث عن مجموع مصروف الطلبة الذين ليس لهم أخ وولدوا في شهر بعد الذي سيتم تحديده ؟

كود

```
Private Sub cmd11_Click()  
M = InputBox("إدخال الشهر", "إدخال الرجاء")  
w = Nz(DSum("[sexp]", "[student]", "[shasbrother] = false and month([sdate]) > " & Str(M)), _  
Empty)  
  
If Not IsEmpty(w) Then  
    MsgBox w  
Else  
    MsgBox "لا يوجد نتيجة"  
End If  
  
End Sub
```

لاحظ كيفية صياغة الشروط المركبة مع **and**

١٢. البحث عن معدل مصروف الطلبة الذين لم يولدوا في السنة المحددة ؟

كود

```
Private Sub cmd12_Click()
Y = InputBox("ادخال السنة","ادخال الرجاء")
w = Nz(DAvg("[sexp]", "[student]", "year([sdate]) <> " & Str(Y)), Empty)

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "لا يوجد نتيجة"
End If

End Sub
```

<> تعني لا يساوي

١٣. البحث عن آخر طالب في الجدول له أخ ؟

كود

```
Private Sub cmd13_Click()

w = Nz(DLast("[sname]", "[student]", "[shasbrother] = True"), Empty)

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "لا يوجد نتيجة"
End If
End Sub
```

١٤. عرض اسم وتاريخ ميلاد أول طالب يظهر في الجدول ؟

كود

```
Private Sub cmd14_Click()
w = Nz(DFirst("[sname] & ' ' & [sdate]", "[student]"), Empty)

If Not IsEmpty(w) Then
    MsgBox w
Else
    MsgBox "لا يوجد نتيجة"
End If
End Sub
```

لاحظ علامات التنصيص المفردة ' وذلك لترك فراغ بين الرقم والاسم

المثال العاشر

طريقة أمنة وسهلة لربط نمونجين معا وكان لهما نفس مجموعة السجلات وتحريك مؤشر السجلات في النمونجين

المقدمة

إذا قمنا بتصميم نموذج عرض مفرد (سجل تلو الآخر) وصممنا نموذج آخر عرض قائمة بيانات **datasheet** على نفس مجموعة السجلات (الجدول أو الاستعلام) وثم طلبنا القائمة من المفرد، فإننا نود بالطبع عند الرجوع إلى نموذج العرض المفرد أن يظهر السجل الذي كنا عنده عندما خرجنا من قائمة البيانات. هذا المثال يبين طريقة سهلة وآمنة لعمل ذلك بمعنى أننا لو قمنا بفتح كل من النموذجين لوحده لا يحدث مشكلة.

التفاصيل

لنفترض أن أسم النموذج المفرد : **Ambulance**

وأن النموذج قائمة البيانات هو : **AmbulanceDS**

وأننا سنطلب **AmbulanceDS** من **Ambulance** بواسطة زر وهو الزر الذي شكله كالقائمة في مثالنا يقع مباشرة فوق زر الخروج.

الفكرة مبنية على حدث (في الحال on current) للنموذج في عرض القائمة وهو الذي سيتم طلبه حيث نضع ما فيه ما يأتي

كود

```
Private Sub Form_Current()  
    Dim rs As Object  
    If IsLoaded("Ambulance") Then  
        Set rs = Forms!Ambulance.Recordset.Clone  
        rs.FindFirst "[ambId] = '" & Controls(0).Value & "'"   
        If Not rs.EOF Then Forms!Ambulance.Bookmark = rs.Bookmark  
    End If  
End Sub
```

لاحظ هنا أننا نفحص وضع النموذج الأب **Ambulance** أولاً قبل تحويل المؤشر بواسطة إقران فحص فتح النموذج المعروف وهو

كود

```
Function IsLoaded(ByVal strFormName As String) As Boolean  
    Const conObjStateClosed = 0  
    Const conDesignView = 0  
  
    If SysCmd(acSysCmdGetObjectState, acForm, strFormName) <> conObjStateClosed Then  
        If Forms(strFormName).CurrentView <> conDesignView Then  
            IsLoaded = True  
        End If  
    End If  
  
End Function
```

ثم بعد ذلك وبناء على النتيجة فإننا نقوم بتحويل المؤشر للنموذج الأب **Ambulance** للسجل الذي نكون عنده في الابن **AmbulanceDS**

لاحظ لو قمنا بفتح النموذج الابن بدون نموذج الأب فلا مشكلة.

اخوكم : محب العلم