

* طريقة عرض السجلات .

استكمالا لما سبق .
تعرفنا آنفا كيفية إجراء 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 إنشاء بعض الأدوات وتعريفها بالشكل المناسب كالصناديق النصوص Text Box وال label وبعض المفاتيح Command . على فكرة قبل هذا أرجوا 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 وضعها بحروف إنجليزية أو عربية وهنا اخترتها إنجليزية لكي تكون مميزة وملاحظة .

تنبيه / في حالة استخدامك للحروف العربية أو الأسماء الإنجليزية المتفرقة مثل Category Name فدائما 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 . وعنوانها 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 .

عناوين ال Command تكون بالشكل التالي ومتسلسلة الأرقام .

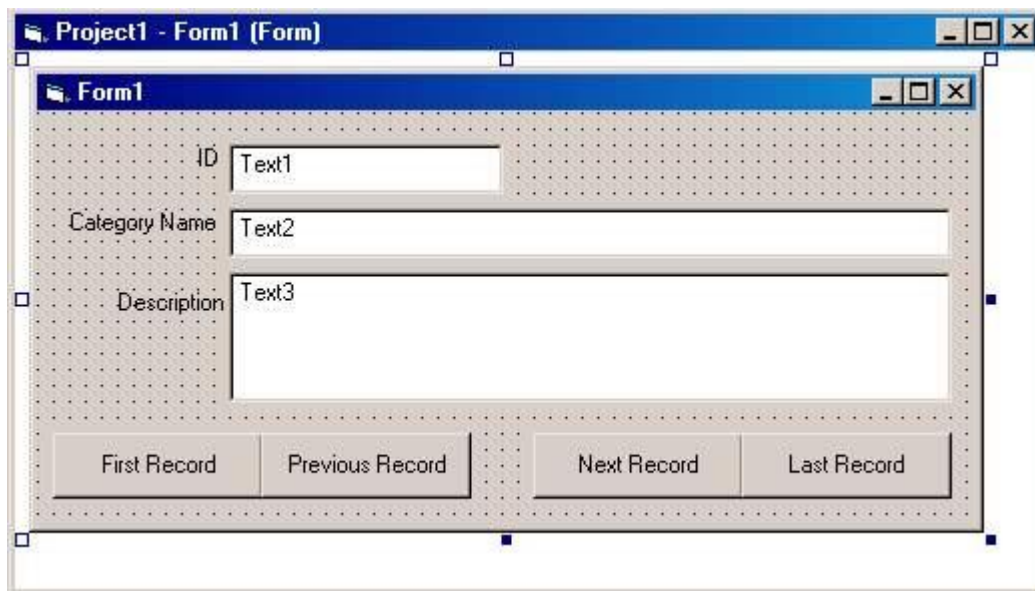
Command رقم ١ يحمل اسم : First Record

Command رقم ٢ يحمل اسم : Previous Record

Command رقم ٣ يحمل اسم : Next Record

Command رقم ٤ يحمل اسم : Last Record

ولا تنسى إن تحفظ المشروع ولنسميه مثلا ViewRecord . الصورة التالية تبين الخطوات التي تم إتباعها.



والآن وصلنا إلى دور كتابة الكود . ومثل ما عرفنا سابقا إن لإجراء أي اتصال مع أي 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 Command الذي 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 الحقيقة ترجع عدد السجلات الموجودة في الجدول ، فمثلا يوجد عندنا ١٠ سجلات مخزنة 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 خلو 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]
```

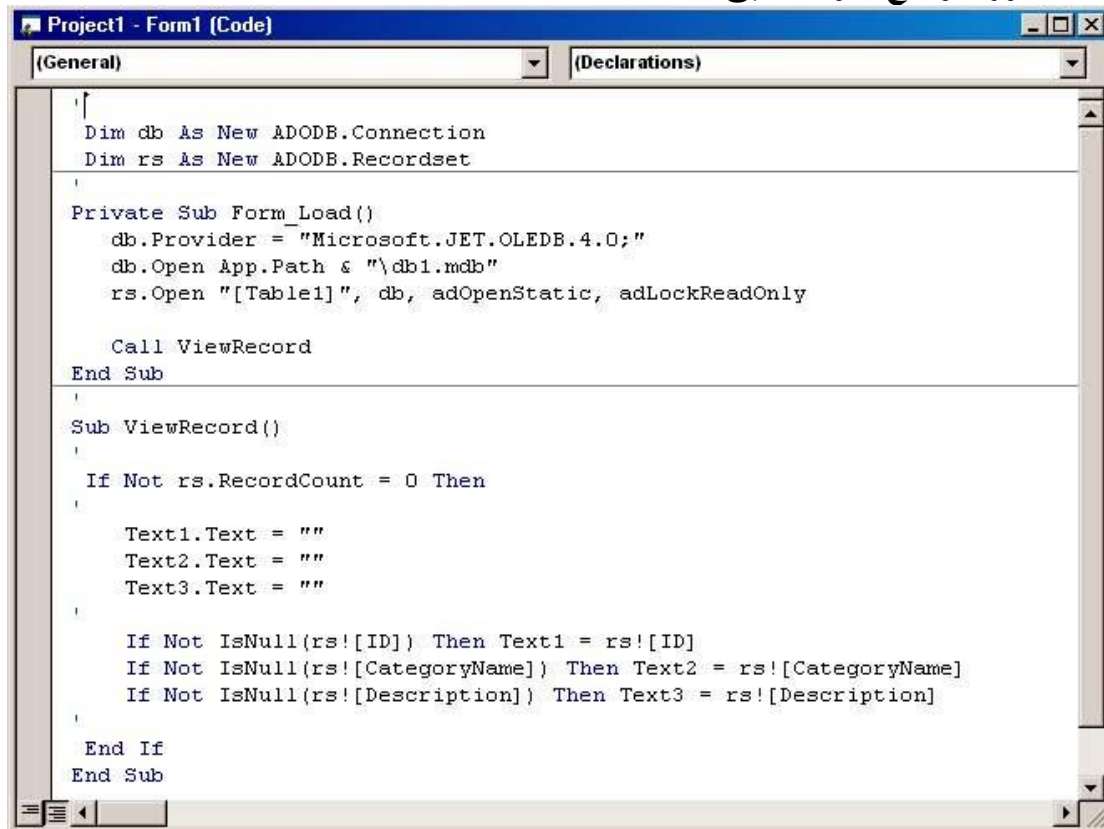
إذ جئنا لترجم الكود السابق لنص مقروء فالنتائج يكون :

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

وهذه صورة توضح الكود السابق .



```
Project1 - Form1 [Code]
(General) (Declarations)

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

    Call ViewRecord
End Sub

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
```

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** فذلك نكتب الكود التالي /
Call ViewRecord ويكون مكانه اسفل الكود الخاص بفتح الجدول . والآن تستطيع تشغيل البرنامج لترى ما تم إنجازه .

تلك كانت الطريقة الأولى والتي أفضّلها أما الثانية فهي قريبة من الطريقة الأولى ولكن عوضاً عن استخدام اسم الحقل سوف نستخدم 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
```

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

```
Private Sub Form_Load()
```

```
    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 **First Record** 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 خطأ 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 . فإتمام 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 .

ولأن سأعرض باقي الأكواد ، واترك لكم فهم الكود /

كود التراجع أو الانتقال إلى السابق **MovePrevious** . ونضعه في **Command2** :

```
Private Sub Command2_Click()
    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 .

هذا وعسى إن تكون الكلمات السابقة فيها نوع من الفائدة والتحصيل البرمجي الخاص بقواعد البيانات . وإلى الدرس القادم لكم مني أخلص تحية والسلام عليكم .

هيثم القلاف

منتدى الإبداع الإسلامي

Haa10@hotmail.com

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