

## تعديل وتحديث البيانات

السلام عليكم

كمدخل لمفاهيم تعديل وتحديث البيانات أحب أن اذكر الجميع بأننا في Linq أصبحنا نتعامل مع كائنات وبالتالي مفهوم تحديث البيانات هو عبارة عن تعديل لخصائص هذه الكائنات والمجموعات أو حذف أو اضافة كائنات ثم يقوم مزود Linq to SQL بتحويل تعديلاتنا إلى جمل SQL يتم ارسالها لقاعدة البيانات .

### أولا إضافة البيانات :

#### ١- إضافة البيانات لجدول واحد :

نقوم أولا بانشاء نسخة من كائن الجدول ( مثلا Customer أو Category ) ثم نسند القيم للحقول المطلوبة ونستخدم الكائن DataContext لإلحاق البيانات بقاعدة البيانات والميزة الجميلة أن LINQ يقوم لوحده بتحديث بقية الحقول التي لم ندخلها لقيمها الجديدة. مثلا الكود التالي يضيف بيانات لجدول الفئات Categories في NorthWind ثم يعرض قيمة ID للفئة الجديد .

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim MyDB As New NorthWind.NorthWind("Northwind.sdf")
    MyDB.Log = Console.Out
    Dim myCat As New NorthWind.Category
    With myCat
        .CategoryName = "Arabicaa"
        .Description = "Arabs"
    End With
    MyDB.Categories.InsertOnSubmit(myCat)

    MyDB.SubmitChanges()
    MsgBox("new ID : " & myCat.CategoryID)

End Sub
```

لا تنسى أن تلقي نظرة على نافذة Output .

#### ٢-إضافة البيانات لجدول مترابطة Master/Details:

غالبا ما نحتاج ادخال المستخدم لإضافة البيانات لعدة جداول مرتبطة مثلا عند ادخال العميل لفاتورة ما نحتاج لإدخال البيانات في جدول الفواتير ثم نضيف تفاصيل كل فاتورة في جدول التفاصيل ... وفي قاعدة Northwind لنفرض أننا نريد إدخال منتجات Products جديدة تتبع لفئة جديدة ..ما سنقوم به هو انشاء كائن جديد من نوع Category وسنجد في خصائصه تلقائيا مجموعة Products فنضيف لها ما نحتاجه من منتجات و نرسل التغييرات لقاعدة البيانات .وهذا مثال الكود :

```
Dim myCat As New NorthWind.Category
With myCat
    .CategoryName = "Arabicooo"
    .Description = "Arabians"
    .Products.Add(New NorthWind.Product With _
        {.ProductName = "AAAAAA", .Discontinued = False})
    .Products.Add(New NorthWind.Product With _
        {.ProductName = "BBBBBB", .Discontinued = False})
    .Products.Add(New NorthWind.Product With _
        {.ProductName = "CCCCCC", .Discontinued = False})
End With

MyDB.Categories.InsertOnSubmit(myCat)

MyDB.SubmitChanges()
MsgBox("CategoryID ID : " & myCat.CategoryID & " My Products : " & _
    myCat.Products.Count)
```

ولاحظ أننا لم نضع CategoryID في كودنا لأن Linq to SQL سيتكفل بإحضارها . وإذا القيت نظرة على نافذة Output ستجد أن كودنا السابق تمت ترجمته لخمس جمل SQL الأولى لإضافة فئة جديدة Category والثاني للحصول على CategoryID للفئة الجديدة و الثلاث البقية لإضافة المنتجات Product الجديدة .

### ثانيا تعديل البيانات :

إذا فهمت الفقرة السابقة فالتالي أسهل باختصار طريقة التعديل . ( أحضر الكائن المطلوب .. عدله ... أرسله ) مثلا لنعدل أحد المنتج السابق الذي أضفناه المسمى CCCCCC ليصبح اسمه XXXXXX هذا الكود

```
Dim myProduct = From pro In MyDB.Products _
                  Where pro.ProductName = "CCCCCC" Select pro
If myProduct.Count > 0 Then myProduct.First.ProductName = "XXXXXX"
MyDB.SubmitChanges()
```

### ثالثا حذف البيانات :

ايضا بنفس الطريقة السابقة : أحضر الكائن ثم احذفه ثم أخبر القاعدة (قاعدة البيانات طبعا ...) وهذا كود لحذف المنتج XXXXXX :

```
Dim myProduct = From pro In MyDB.Products _
                  Where pro.ProductName = "XXXXXX" Select pro
If myProduct.Count > 0 Then _
    myDB.Products.DeleteOnSubmit(myProduct.First)
MyDB.SubmitChanges()
```

ولحذف كل ما أضفناه في الأكواد السابقة لقاعدة بيانات Northwind حيث سنحذف Category التي تبدأ ب Arab ومنتجاتها بهذا الكود :

```
Dim myCat = From cat In MyDB.Categories _
             Where cat.CategoryName Like "Arab*" Select cat

If myCat.Count > 0 Then
    For Each cat In myCat
        MyDB.Products.DeleteAllOnSubmit(cat.Products)
    Next
End If

MyDB.Categories.DeleteAllOnSubmit(myCat)
MyDB.SubmitChanges()
```

## رابعا :نطاق العمليات Transactions :

اعتدنا دائما لإنشاء برنامج يعتمد عليه أن نستخدم عن تغيير البيانات كائنات Transaction خاصة عند حدوث خطأ ما أثناء ادخال بيانات مترابطة فيجب الغاء العملية برمتها ..فليس من المعقول أن نسجل الفاتورة مثلا ناقصة بعض الاغراض .

وبالنسبة ل Linq To SQL فهي تدعم Transaction داخليا فعند أي استدعاء ل SubmitChanges سيقوم مزود Linq تلقائيا بإنشاء Transaction خاص بهذا الاستدعاء حيث ولدى حدوث أي خطأ فإن عملية تحديث البيانات سيتم الغائها تلقائيا .. جرب هذا الكود أن Category المقصودة لازالت موجودة . ( حيث تعمدت أن أولد خطأ عن طريق حذف كائن من Category مرتبط بكائنات من Product وهذا لا تسمح به قاعدة البيانات حيث يجب حذف الطرف الآخر من العلاقة ).

```
Dim myCat = From cat In MyDB.Categories Where cat.CategoryID = 2
MyDB.Categories.DeleteAllOnSubmit(myCat)
Try
    MyDB.SubmitChanges()
Catch ex As Exception
    MsgBox("Error !! all changes canceled")
End Try
```

ويمكن تعديل الكود السابق بحيث نتتبع الخطأ و نعالجه في جملة Catch ثم نستدعي SubmitChanges مرة أخرى.

وهناك طريقة أخرى عن طريق خاصية Transaction للكائن DataContext حيث بإمكاننا استخدام نطاق Transaction من كائنات ADO.net وستكون كل العمليات التي ستتم ضمن هذا النطاق وإذا عملنا RollBack ستلغى كل التغييرات .

بهذا ينتهي مقال اليوم .. امل أن يكون مفيدا للجميع .. إلى لقاء مع الموضوع القادم الذي سيكون عن ميزة رائعة يمكن الاستفادة منها في Linq وهي Extensions .. وايضا سنلقي نظرة أعمق على الكائن DataContext .

والسلام ...