

السلام عليكم ورحمة الله وبركاته

في الموضوع السابق انشأنا برنامجا يرتبط بقاعدة بيانات NorthWind.sdf أنشأنا فئات عمل Linq باستخدام SQLMetal

ولمن يريد الاستفادة من الدرس يفضل انشاء البرنامج حسب الدرس السابق لأنني سأمزج بين المعلومات النظرية والتطبيق العملي مباشرة في كافة المراحل .

الان سأنتقل للموضوع المهم وهو التعامل مع LINQ بعد انتهينا من بناء الفئات باستخدام SQLMetal .
ان الفئات التي تشكلت لدينا هي صلتنا بقاعدة البيانات فبعد الان اصبح كل سجل بقاعدة البيانات ممثل بكائن وكل مجموعة سجلات ممثلة ب List - أو بنية تشبهها بالاحرى - مثلا الجدول Customers الان نصل اليه عن طريق الكائن Customers والذي يحتوي على كائنات من نوع Customer التي انشأها لنا SQLMetal وستجد لكل فئة من نوع Customer خصائص هي عبارة عن أعمدة الجدول Customer وعند الشرح العملي ستوضح الصورة أكثر .

الان في البرنامج ضع اداة DataGridView على كامل الفورم حيث سنستخدمها لعرض نواتج تجاربنا مع استعلامات Linq .

الخطوة الأولى لاستخدام LINQ TO SQL هو انشاء الكائن DataContext والذي مهمته تأمين الاتصال بقاعدة البيانات وطبعاً نحن لن ننشئ هذا الكائن مباشرة انما هناك الفئة Northwind التي انشأها لنا SQLMetal والتي هي اصلاً مشتقة من DataContext وبالتالي سننشئ كائن من هذه الفئة كمتغير محلي Field في الفورم . تحت كلمة Public Class

```
Public Class Form1
```

```
Dim MyDB As New NorthWind.NorthWind"(Northwind.sdf)"
```

طبعاً تلاحظ اننا كتبنا NorthWind.NorthWind وذلك لأننا عند انشاء الفئات sqlmetal طلبنا منه أن ينشئ نطاق للأسماء Namespace باسم NorthWind فلا بد للوصول لفئاتنا استخدام هذا النطاق قبلها .

الان اصبح كل البيانات في متناول يدك وسنبدأ بتجربة اكواد مختلفة لتتعرف على استعلامات LINQ أكثر ونقول وداعاً لاستعلامات SQL وكود ADO.Net الممل والمكرر .

استعلامات LINQ

بشكل عام استعلامات LINQ اما تبدأ ب From وتعيد مجموعة سجلات أو ب Aggregate وتعيد عندها قيمة واحدة وساقوم بترجمة كل استعلام نستخدمه لاستعلام SQL المقابل له ليسهل فهمه .

اولاً: استعلامات From :

صيغتها العامة المبسطة

```
From element In collection [Where condition] Select elements[vars]
```

وفي مثالنا تصبح Collection تصبح MyDB التي تمثل قاعدة البيانات كاملة وطبعاً ما بين اقواس مربعة هو اختياري .

الان سنضع جملة from بسيطة في الحدث Form_Load لتعرض البيانات في DataGridView .

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    MyDB.Log = Console.Out

    Dim mySource = From cust In MyDB.Customers Select cust

    DataGridView1.DataSource = mySource

End Sub
```

نفذ البرنامج وسترى الجدول Customers كاملا لديك . ولقد اضفت العبارة MyDB.Log = Console.Out حيث ستكتب لك في نافذة Output ما يتم ارساله لقاعدة البيانات وسترى انها استعلامات SQL المعروفة فهذه هي وظيفة LINQ to SQL ترجمة ما نكتبه لقاعدة البيانات ولاتنسى ان تلقي نظرة على نافذة Output بعد كل استعلام . أعود للاستعلام السابق بإمكانك اعتبار cust سجل ،، فترجمته لعبارة مفهومة هو "كل سجل نفترض ان اسمه cust في مجموعة السجلات MyDB.Customers اعطني cust " يعني كل السجلات ومقابلته ب SQL :

```
SELECT * FROM Customers
```

اما ناتج الاستعلام mySource فهو نوع مجهول Anonymous Type ولاتشغل بالك كثيرا به فالتطبيقات البسيطة ل LINQ لاتحتاج لمعرفة قوية بهذه الانواع . يمكنك اعتبار الناتج مجموعة تحتوي كائنات من نوع Customer .

وناتج الاستعلام يمكنك الاستفادة منه بعدة طرق مثلا في المثال السابق استخدمناه ك Data Source . أو يمكنك التنقل خلاله من خلال حلقة For..Each مثال :

```
For Each mycust In mySource

    Console.WriteLine(mycust.ContactName)

Next
```

أو اخذ اول سجل منه فقط واستعراض بياناته مثلا ...

```
Dim FirstCust = mySource.First

Console.WriteLine(FirstCust.CompanyName, FirstCust.ContactName,
FirstCust.City, FirstCust.Address)
```

وكل ما ينطبق على List(of T) ينطبق عليه .. وفوق ذلك يمكن الاستعلام عنه في جملة From جديدة .. مثلا

```
Dim LondonCust = From x In mySource Where x.City = "London" Select x
```

وبإمكانك ان تضع ناتج الاستعلام LondonCust كمصدر DataSource للداتا جريد .

لاحظ اثناء كتابتك للاستعلامات السابقة أن Visual BASIC يقترح عليك العناصر و الحقول ويصحح لك الأخطاء وهذا سبب قولي أن استخدام استعلامات LINQ اسهل واصح من استخدام استعلامات SQL فلامجال مع Linq لخطأ لاتعرف ما هيته ترسله لك قاعدة البيانات وتحتار في امره .ويكون سببه بسيط كاختيارك لنوع بيانات خاطئ مثلا String بدلا من Integer فهنا المحرر سيخبرك بالخطا مباشرة.

نعود لبعض التطبيقات وساسرد هنا استعلام LINQ وقم انت باستبداله في حدث Form_Load وشاهد النتيجة وبعده ساذكر استعلام SQL المقابل ولسهولة فهم الاستعلامات ساكتب كل بداية كلمة محجوزة في سطر باستخدام _ .

```
Dim mySource = From cust In MyDB.Customers _
                Select cust.ContactName, cust.City

SQL : SELECT ContactName,City FROM Customers

Dim mySource = From cust In MyDB.Customers _
                Select cust.ContactName, cust.City

SQL : SELECT ContactName,City FROM Customers

Dim mySource = From cust In MyDB.Customers _
                Where cust.City = "London"
                Select cust.ContactName, cust.ContactTitle, cust.Address
SQL : SELECT ContactName, ContactTitle, Address FROM Customers WHERE
City='London'

Dim mySource = From ord In MyDB.Orders _
                Where ord.Customer.City = "London" _
                Select ord.OrderDate,ord.OrderID,ord.Customer.ContactTitle

SQL : SELECT Orders.[Order Date],Orders.[Order ID],
Customers.[Contact Title] FROM Orders
INNER JOIN Customers ON Customers.[Customer ID] = Orders.[Customer ID]
WHERE Customers.City = 'London'
```

لاحظ الاستعلام الاخير أن العلاقات ممثلة بشكل مجموعة كائنات فكل كائن Order يحتوي على كائن Customer يمثل العلاقة بين Customer ID في الجدولين وهذه طريقة ذكية وسهلة لتمثيل العلاقات خاصة العلاقات المتشابهة بين عدة جداول حيث كنت سابقا اكتب اكثر من جملة Inner Join في الاستعلام . وطبعاً يمكنك استخدام الكلمات المحجوزة التالية مع الاستعلامات Like, Order By, Distinct واقترح عليك ان تجرب ما يقترحه عليك المحرر وتحاول بناء جمل استعلامات LINQ اكثر تعقيدا لتعتاد على استخدامها . الى لقاء قريب .حيث ساتطرق ل Aggregate و استعلامات اكثر تعقيدا .