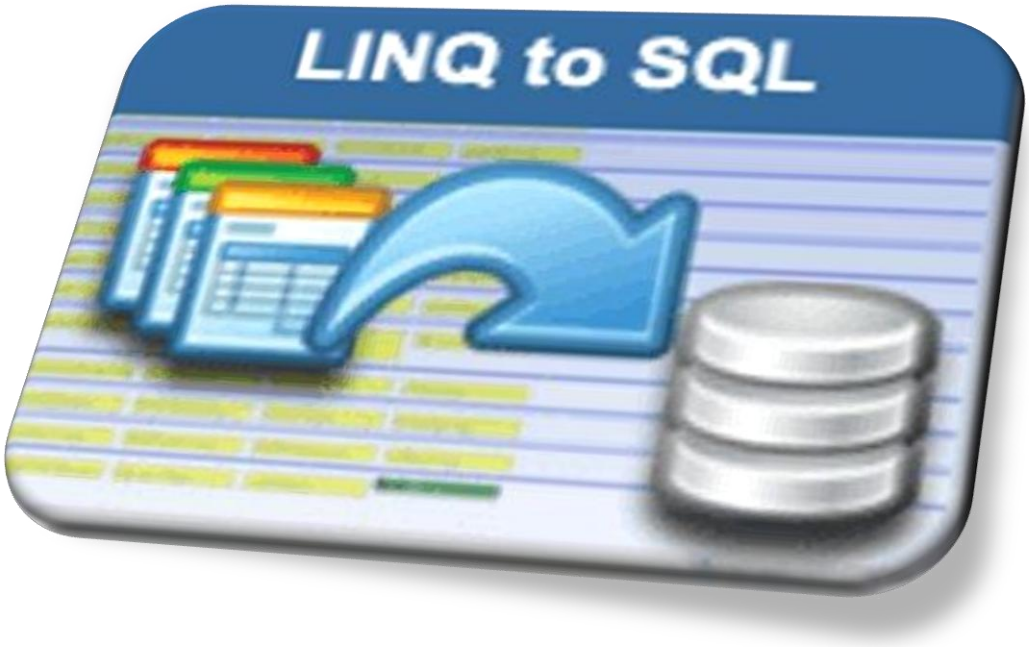


LINQToSQL Using VB.NET

اعداد :

سجاد محمد باقر (Sajad)

مدرس حاسوب

مدير منتدى فيجوال بيسك لكل العرب vb4arb.com

العراق – كركوك

بسم الله الرحمن الرحيم

وصلى الله على محمد وآله الطاهرين

السلام عليكم ورحمة الله وبركاته

مقدمة:

سبق وتطرقت في الكتيب السابق الذي تجدونه على هذا الرابط:LINQToSQL Using C#.NET عن هذه التقنية التي كانت بلغة C#.NET لكن في هذا الكتيب سنتعرف عن هذه التقنية في التعامل مع قواعد البيانات بلغة VB.NET مع وقواعد بيانات SQL Server 2008.

تعريف:

LINQ: مختصر لجملة Language Integrated Query أي ما معناه لغة الاستعلام المتكامل, وهي تقنية او نموذج ومنهج برمجي مقدم من Microsoft لتضيف قدرات الاستعلام الاساسية المستخدمة مع قواعد البيانات الى لغات البرمجة المستندة على الـ.NET, حيث تقدم هذه التقنية تعابير واضحة وذكية لمعالجة البيانات, وتكمن القدرة الحقيقية لهذه التقنية في قابليتها على تطبيق نفس الاستعلام على قاعدة بيانات SQL, على Dataset, مصفوفة من البيانات في الذاكرة, XML, الكائنات والعديد من الانواع الاخرى من البيانات.

الجزء المتكامل معناه أن التقنية جزء من بناء الجملة في لغة البرمجة, اما الجزء الاخر (الاستعلام) يوضح قدرة وقوة هذه التقنية من حيث معالجتها العديد من الانواع من البيانات ,ويمكن وصف هذه التقنية أيضا بأنها تعابير او جمل برمجية يستخدم للاستعلام عن البيانات.

فوائدها:

1. **سهولة التعامل مع البيانات.**
2. **تحويل الجداول الى كائنات والاعمدة الى خصائص وانشاء العلاقات تلقائيا اثناء التحويل.**
3. **الاستعلام عن العديد من الانواع من البيانات, حيث لا يقتصرعلى قواعد البيانات العلائقية.**
4. **استخدام تعابير لامبدا.**
5. **استخدام LINQ مع قواعد البيانات هي أكثر أمانا من تقنية ADO.NET.**
6. **أكوادها قصيرة.**

عيوبها:

1- أن تغيير طريقة الوصول الى البيانات يحتاج منك اعادة الترجمة.
2- من الصعب فهم الاستعلام من خلال التعابير المعقدة.
3- عملية الربط (Joins) تتسم بالبطئ.
4- عدم وجود مخطط واضح للطبقات (Tiers).
5- ارسال الاستعلام بأكمله الى الى قاعدة البيانات وهذا يؤدي الى أخذ الكثير من حركة المرور في الشبكة.

انواع تقنية LINQ:

1- DLINQ: استخدام تقنية LINQ مع قواعد بيانات SQL. (وهذا ما سنشرحه في هذا الكتيب ان شاءالله).
2- XLINQ: استخدام تقنية LINQ مع ملفات XML.
3- LINQ: استخدام تقنية LINQ مع الكائنات (Objects).

مقارنة بين استعلام LINQ واستعلام SQL

اولا: صيغة الاستعلام الخاص بتقنية LINQ:

```
IEnumerable(Of T)or Anonymous (Dim) = from <<element>> in <<Source
collection>> where <<Condition As Boolean Output>>
             select <<expression>>
```

مثال:

```
Dim items = From name In names Where name = "أحمد" Select name
```

ان ناتج الاستعلام يكون من نوع (generic collection) بشكل عام ومن الممكن أن يكون الناتج قيم مجهولة حيث يتم التعرف على نوع الارجاع من قبل المترجم ففي هذه الحالة لايجب استخدام الفئة IEnumerable(Of T) كناتج للاستعلام ,أما نوع الارجاع (Of T) يحدده العبارة Select بمعنى هل الارجاع من نوع عدد صحيح ,سلسلة نصية ,فئة والخ. حسب شرط معين ويمكن اهمال الشرط في حال عدم الحاجة اليه ,اما <Source Collection> فهي مصدر البيانات المراد الاستعلام منها.

ثانيا: صيغة الاستعلام الخاص لـSQL المشابه لاستعلام LINQ أعلاه فهي:

```
Select <Columns> from <table> where <Conditions>
```

لاحظ التشابه الكبير بين صيغتي الاستعلامين الفرق فقط في التقديم والتأخير بين الكلمات المحجوزة مثل كلمات (from ,select ..... etc).

نفس المثال اعلاه لكن في SQL

```
Select name from table where name = 'محمد'
```

سنأخذ مثالا بسيطا لنبين كيفية استعمال تقنية LINQ:

لو كانت لدينا مصفوفة من الاسماء

VB.NET

```
Dim names As String() = New String() {
"سجاد", "أحمد", "محمد", "أمير", "سهى", "عمار", _
                "جمال"}
```

وطلب منا البحث عن اسم معين فيكون الحل كالتالي:

VB.NET

```
            For i As Integer = 0 To names.Length - 1
                If names(i) = "محمد" Then
                    MessageBox.Show("Found")
                End If
            Next
```

هذا طبعا بدون استخدام LINQ ,أما لو بحثنا عن الاسم باستخدام LINQ فسيكون البحث بهذه الطريقة:

الحل:

VB.NET

```
Dim namecollection As IEnumerable(Of String)
= From name In names Where name = "محمد" Select name

            For Each name As String In namecollection
                MessageBox.Show(name)
            Next
```

في المثال اعلاه تم استخدام الفئة IEnumerable(Of) كناتج للاستعلام وهذا طبيعي لان مصدر البيانات عبارة عن سلسلة نصية فليس هنالك من داع لاستخدام متغير من نوع Dim لكن ماذا لو كانت مصدر البيانات عبارة عن قيم غير معروفة النوع؟ في هذه الحالة يسمى ناتج الاستعلام بالناتج المجهول لانه لايمكن التكهن بناتج الاستعلام وهنا تأتي دور الكلمة المحجوزة Dim.

Dim: عبارة عن نوع من الانواع الضمنية حيث يقوم بتمثيل أي نوع من البيانات ويحدد نوع البيانات كما قلنا قبل المترجم.

لنأخذ مثالا يبين ذلك:

لدينا الفئة التالية باسم Books:

VB.NET

```
Class Books
        Public Property BookID() As Integer
            Get
                Return m_BookID
            End Get
            Set
                m_BookID = Value
            End Set
        End Property
        Private m_BookID As Integer
        Public Property BookName() As String
            Get
                Return m_BookName
            End Get
            Set
                m_BookName = Value
            End Set
        End Property
        Private m_BookName As String
    End Class
```

المطلوب أن يكون ناتج الاستعلام البحث عن كتاب معين حسب التسلسل ومطلوب ايضا تغيير اسم الخاصية BookName الى اسم آخر مثلا BName.

ملاحظة: تتيح لك تقنية LINQ تغيير اسماء الخصائص اثناء الاستعلام باستخدام الكلمة المحجوزة New With باسماء اخرى غير معروفة النوع ,بمعنى آخر يمكن أختيار حقول دون أخرى.

الحل:

VB.NET

```
Dim books As Books() = New Books() {
            New Books() With { _
    .BookID = 1, _
    .BookName = "C#" _
    },
    New Books() With { _
    .BookID = 2, _
    .BookName = "VB" _
    }
  }

        Dim book = From b In books Where b.BookID = 1
        Select New With {.BName = b.BookName}

        For Each item In book
            MessageBox.Show(item.BName)
        Next
```

لاحظ هذا السطر:

```
Select New With {.BName = b.BookName}
```

تم تغيير اسم الكتاب الى اسم أخر غير معروف النوع وذلك باستخدام الكلمتين المحجوزتين New With ففي هذه الحالة يجب تعريف متغير من نوع Dim كناتج للاستعلام وايضا داخل الـForEach ويتم عرض الاسم عن طريق الاسم الجديد .BName

المزيد من استعلامات LINQ:

VB.NET

```
' Display all books order by book name
        Dim book = From b In Books Order By b.BookName Select b
        ' Display all books order by book id but this time descending
        Dim book = From b In Books Order By b.BookID Descending Select
```

```
b
        ' Using Lambda Expression and Function OrderBy
        Dim book =
Books.OrderBy(Function(c) c.BookName).Select(Function(c) c)
        ' Retrieving books according to BookID and Renaming the Source
Name of properties
        Dim book = From b In Books Where b.BookID = 1 Select New
With { _
    .ID = b.BookID, _
    .BName = b.BookName _
   }
        '
        Dim numbers As Integer() = New Integer() {2, 4, 1, 3, 6, 7, _
         5, 8, 9}
        ' Display odd numbers
        Dim odds = From num In numbers Where num Mod 2 <> 0 Select num
        ' Retrieving Names witch start with 'a' letter
        Dim name = From n In names Where n.Name.StartsWith("a") Select
n
        ' Displaying Single Book using Lambda Expression depending on
book id
        Dim b As Books = books.Single(Function(c) (c.BookID = 1))
        ' Displaying book name where book id equal to 2 (output as
IEnumerable(Of string) or Dim)
        Dim book = Books.Where(Function(bid) bid.BookID
= 2).Select(Function(bn) bn.BookName)
        ' Returning number books in collection
        Dim bcount As Integer = Books.Count()
        ' you can retrieving data from two collection as showing below
        Dim book As ba = From b In books From a In aouthers Select New
With { _
         b.BookName, _
         a.aname _
        }
        ' You can join multiple collection using join keyword as
showing below
            Dim bajoin = From b In books Join a In aouthers On b.aid =
a.aid Select New With { _
                a.aid, _
                b.BookName, _
                a.aname _
            }
```

بناء مشروع نتعرف من خلاله على كيفية التعامل مع قواعد البيانات SQL باستخدام تقنية LINQ:

سنقوم ببناء مشروع يتعامل مع تقنية LINQ مع قواعد بيانات SQL يؤدي العمليات الاساسية التي تحتاجها كل برنامج قاعدة بيانات من (اضافة ,حذف ,تعديل ,بحث ,طباعة تقرير سنستخدم تقارير مايكروسوفت (Report.rdlc)) واستعراض السجل السابق واللاحق) لكي يكون مرجعا لكل من يريد تعلم هذه التقنية واستخدامها مع قواعد بيانات.SQL

بسم الله نبدأ

اولا : قم بإنشاء قاعدة بيانات باسم std عن طريق برنامج ال SQL Management Studio ومن ثم أنشئ جدول باسم StdInfo
وأضف اليه الحقول (ID, FName, DoB, Adress) واضبط خصائص الحقول كما في الشكل الآتي:

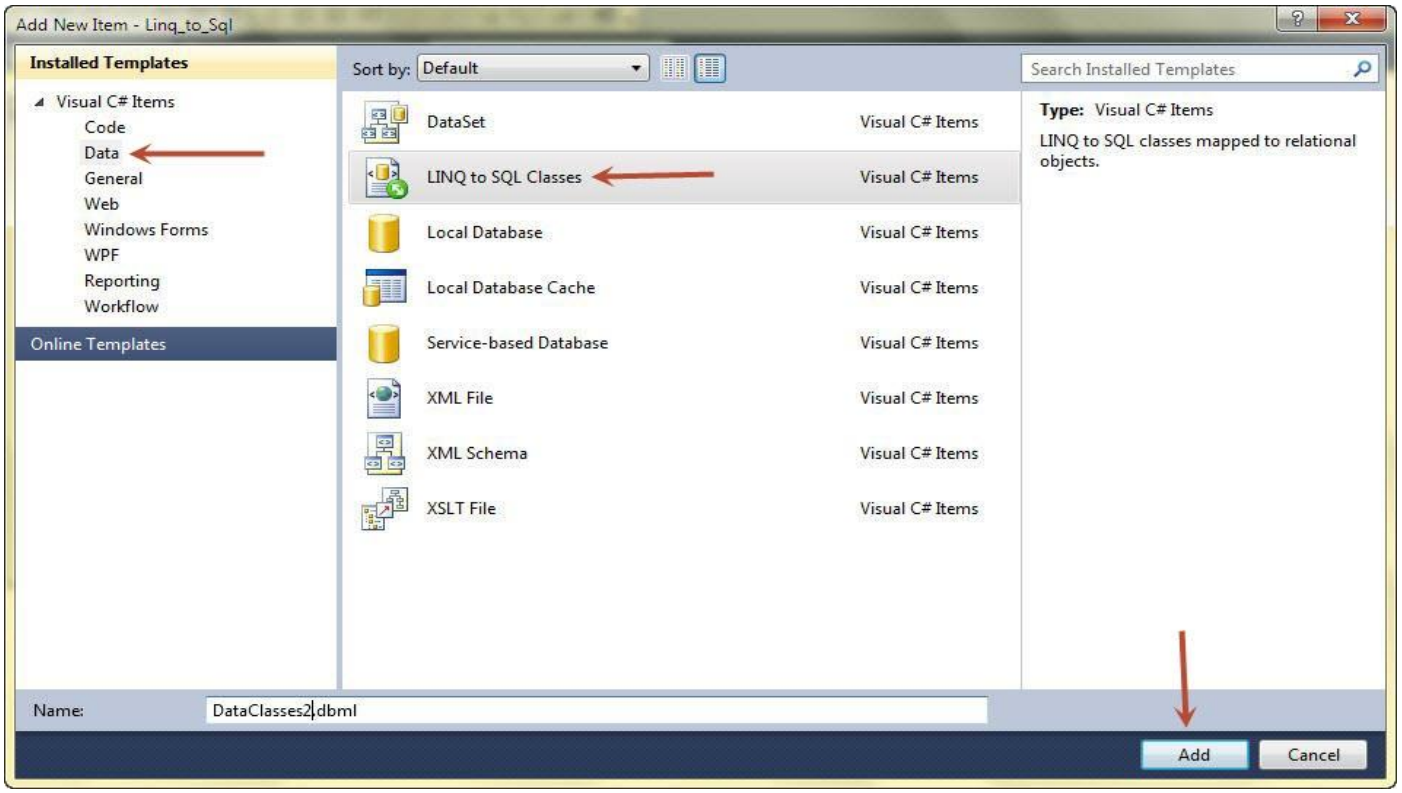
مع جعل حقل ال IDمفتاح اساسي و ترقيم تلقائي من خاصية Identity column كما هو موضح في الصورة اعلاه.

ثانيا : افتح مشروع جديد باسم Lint_to_SQL وصمم الواجهة بالشكل الآتي:

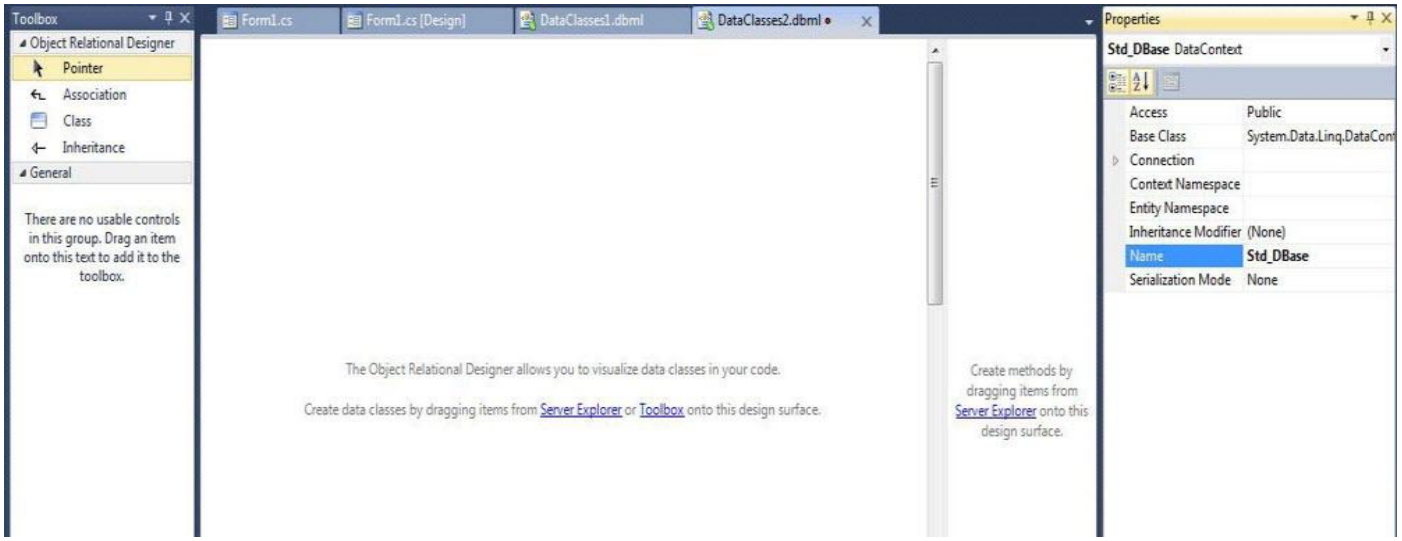
قم بضبط خصائص الادوات كما في الجدول الآتي:

الاداة	الاسم	الاستخدام	خصائص اضافية
TextBox	Id_txt	تسلسل الطالب	RightToLeft=Yes Enabled=False
TextBox	name_txt	اسم الطالب	
TextBox	address_txt	عنوان الطالب	
MaskedTextBox	dob_txt	تاريخ الولادة	RightToLeft=No TextAlign=Right
TextBox	srch_txt	حقل البحث بالاسم	RightToLeft=Yes AutoCompleteMode= SuggestAppend AutoCompleteSpurce= CustomSource
button	new_btn	تفعيل اضافة طالب جديد	
button	cancel_btn	الغاء تفعيل اضافة طالب جديد	
button	load_btn	تحميل البيانات	
button	insert_btn	اضافة طالب الى قاعدة البيانات	Enabled=False
button	delete_btn	حذف طالب من قاعدة البيانات	
button	update_btn	تعديل المعلومات	
button	dosrch_btn	البحث عن طالب معين	
button	next_btn	السجل اللاحق	
button	prev_btn	السجل السابق	
button	first_btn	السجل الاول	
button	last_btn	السجل الاخير	
button	show_btn	عرض البيانات في الDgridview	
button	nextpage_btn	عرض الصفحة التالية	
button	prevpage_btn	عرض الصفحة السابقة	
button	current_btn	طباعة السجل الحالي	
button	selected_btn	طباعة السجلات المختارة من الDgridview	
button	all_btn	طباعة كل السجلات الموجودة في قاعدة البيانات	
Datagridview	DGV	لعرض البيانات	

ثالثا : من Project ثم Add New Item تظهر لك الشكل الآتي:

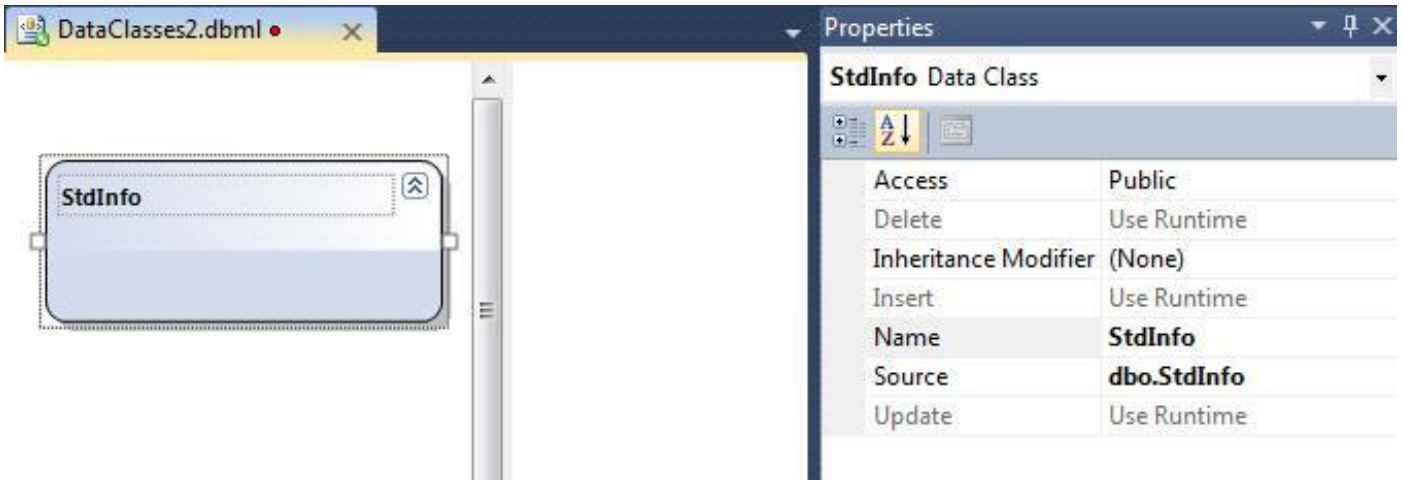


افعل كما في الصورة أي تختار (LINQ to SQL Classes) هذا الـ Class وسيلة لربط قاعدة بيانات SQL بالبرنامج بتقنية الLINQ ومن ثم تضغط على Add فتظهر الشكل الآتي:



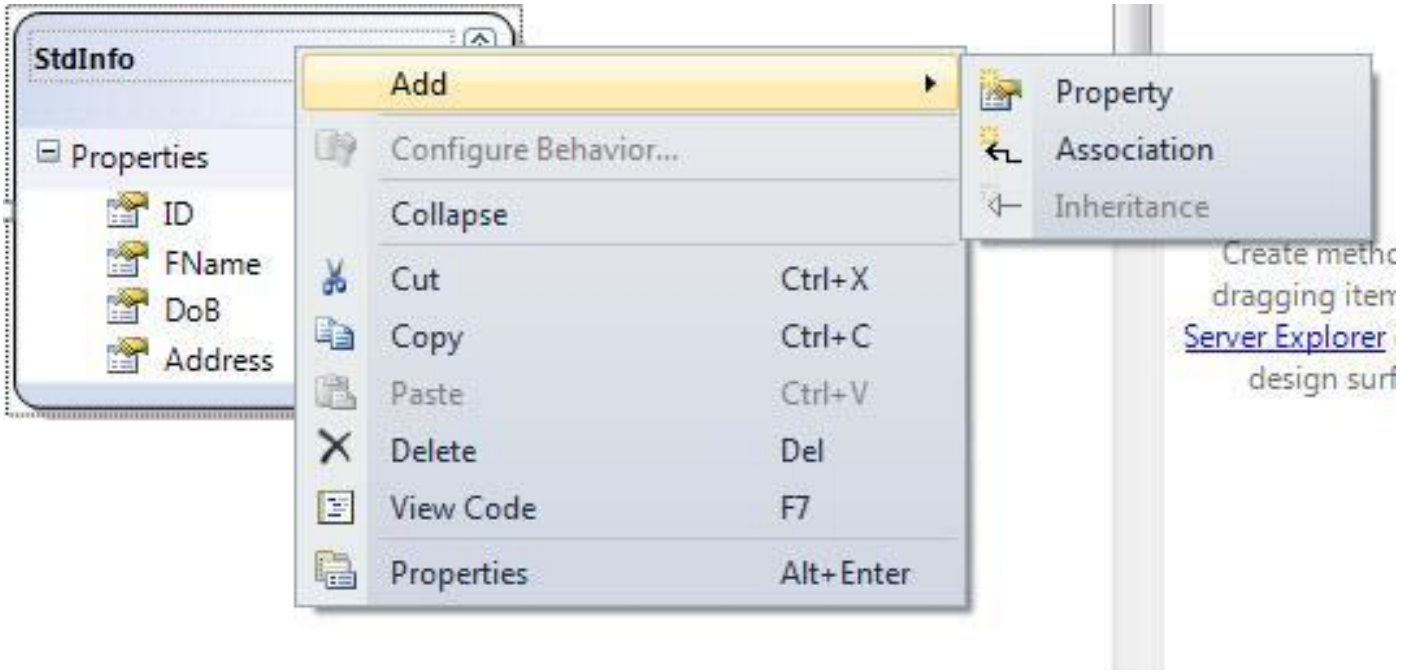
من الخصائص غير الاسم الى Std_DBase

وبعد ذلك من الـ ToolBox التي على اليسار اسحب أداة Class لاحظ الشكل الآتي:

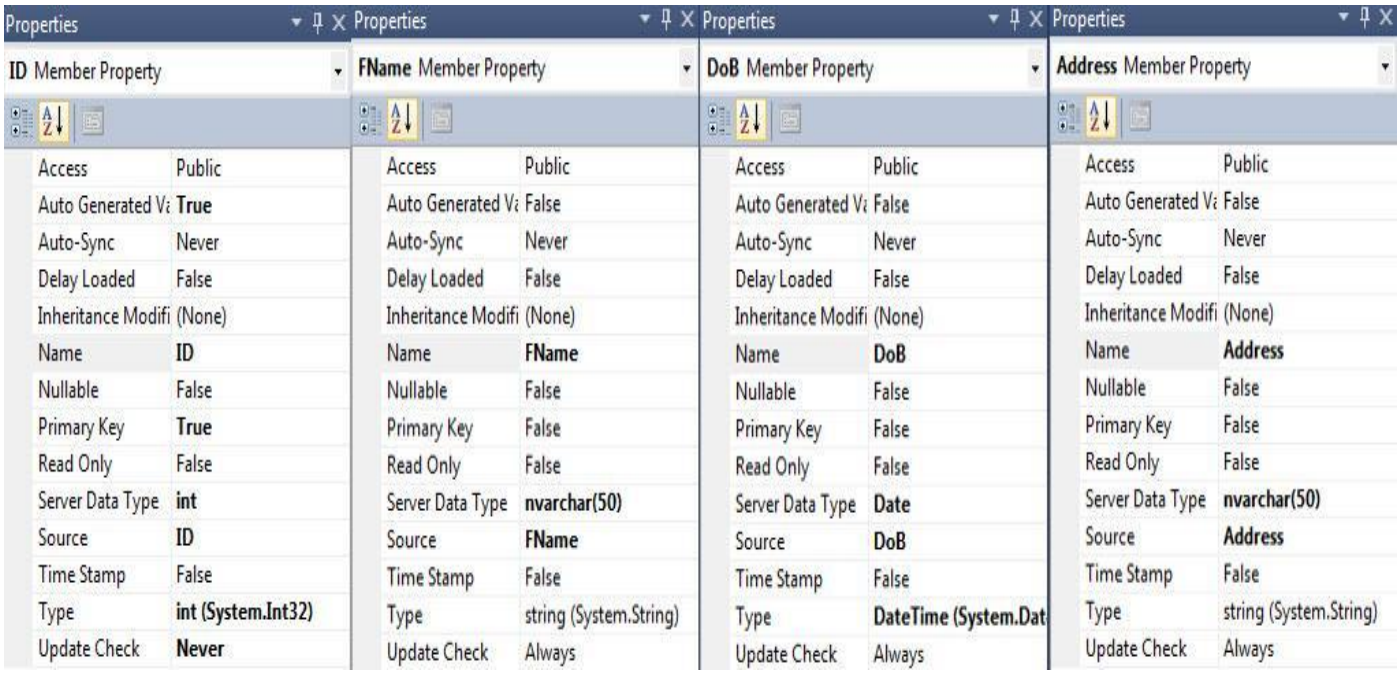


غير اسم الـ (Class) والتي يعادل جدول في قاعدة البيانات الى StdInfo او أي اسم تختاره ,وفي حقل الـ Source اكتب اسم الجدول في قاعدة البيانات وهذا مهم جدا ,لاحظ الشكل اعلاه.

اضف اربعة خصائص (Properties) (حقول) الى الـ Class بهذا الشكل:



وقم بضبط خصائص الحقول كما مبين في الشكل أدناه:



Properties — ID Member Property	
Access	Public
Auto Generated Va	True
Auto-Sync	Never
Delay Loaded	False
Inheritance Modifi	(None)
Name	ID
Nullable	False
Primary Key	True
Read Only	False
Server Data Type	int
Source	ID
Time Stamp	False
Type	int (System.Int32)
Update Check	Never

Properties — FName Member Property	
Access	Public
Auto Generated Va	False
Auto-Sync	Never
Delay Loaded	False
Inheritance Modifi	(None)
Name	FName
Nullable	False
Primary Key	False
Read Only	False
Server Data Type	nvarchar(50)
Source	FName
Time Stamp	False
Type	string (System.String)
Update Check	Always

Properties — DoB Member Property	
Access	Public
Auto Generated Va	False
Auto-Sync	Never
Delay Loaded	False
Inheritance Modifi	(None)
Name	DoB
Nullable	False
Primary Key	False
Read Only	False
Server Data Type	Date
Source	DoB
Time Stamp	False
Type	DateTime (System.Dat
Update Check	Always

Properties — Address Member Property	
Access	Public
Auto Generated Va	False
Auto-Sync	Never
Delay Loaded	False
Inheritance Modifi	(None)
Name	Address
Nullable	False
Primary Key	False
Read Only	False
Server Data Type	nvarchar(50)
Source	Address
Time Stamp	False
Type	string (System.String)
Update Check	Always

ملاحظة : اجعل اسماء الحقول تتطابق مع اسماء الحقول في قاعدة البيانات لسهولة التعامل ,لكن تستطيع تغييرها كما تريدها انت.

اهم الملاحظات:

1- جعل خاصية ال Auto Generated Value لحقل الـ ID يساوي True لان الحقل الذي يقابله في قاعدة البيانات ترقيم تلقائي.

2- جعل خاصية الـ Primary Key للـ ID يساوي True .

3- أهم شيء هو تحديد الـ Source لكل الحقول ومعناه اسم الحقل الذي يقابله في قاعدة البيانات.

4- تحديد الـ Server Data Type كما في الجدول في قاعدة البيانات (int, varchar, double , ...)

5- تحديد الـ Type للحقول التي أنشأتها في الـ Class بما يطابق نوع الحقول في الجدول في قاعدة البيانات.

رابعا : اضف نموذج الى المشروع لعرض التقرير وسمه Report Form ومن ثم اضف اليه اداة Report Viewer بعد ذلك من Project ثم Add New Item اضف تقرير فارغ الى المشروع (لمعرفة المزيد عن تصميم التقارير راجع هذا الرابط: (كيفية تمرير البيانات الى Microsoft Report عن طريق البراميترات)

اضيف اربع باراميترات الى التقرير بأسماء, (ID, Fname, Dob, Address) ايضا يمكنك مراجعة الرابط اعلاه لمعرفة المزيد. (وأنصح بشدة بذلك).

صمم التقرير بهذا الشكل:

حيث (معلومات الطلبة والعناوين والخطوط الافقية في ال (Page header) والوقت والترقيم في ال Page footer لكي تتكرر في كل الصفحات في حال كون التقرير يتكون اكثر من صفحة.

واضف الباراميترات التي انشأتها الى التقرير كما في الشكل اعلاه.

ملاحظة : يمكن اضافة الـ Page header and footer من شريط الـTool bar Report

ملاحظة : الـ Page Number والـ Execution Time تستطيع اضافتها من نافذة ال Report Data من الـ Built-in-Fields.

الآن نأتي الى كتابة الاكواد ,وستكون هذه المرة بلغة VB.NET.

اولا: اضف مجالات الاسماء التالية:

VB.NET

```
Imports System.Collections
Imports System.Data.SqlClient
Imports Microsoft.Reporting.WinForms
Imports System.IO
```

ثانيا : المتغيرات العامة:

VB.NET

```
' نص الاتصال بقاعدة البيانات
    Private constr As String = "نص الاتصال"
    'linqانشاءinstance من قاعدة البيانات التي تتعامل مع تقنية ال
    Private dbo As DataBase
    ' انشاء قائمة من نوعStudents تحميلها من الجدول في قاعدة البيانات  لتضم بيانات الطلاب اثناء
    Private student As List(Of Students)

    Private i As Integer_
, pos As Integer = 0_' موقع السجل (للتنقل بين السجلات)
, start As Integer = 0_'  بداية الصفحة من السجل الاول
, end1 As Integer = 5'  نهاية الصفحة (هنا حددنا 5 سجلات فقط للصفحة اثاء عرض السجلات في
ال DataGrudView)

    Private rep_source As ReportDataSource
    ' لتحديد مصدر البيانات للتقرير
    Private rep_param As List(Of ReportParameter)
    '  مصفوفة من البارامترات التي ستمرر الى التقرير
```

ثالثا : اضف Class الى المشروع باسم Students لغرض تصفح المعلومات:

VB.NET

```
Public Class Students
    Private m_id As Integer
    Private m_fname As String, m_address As String
    Private m_Dob As DateTime' (Date of Birth) تاريخ الولادة

    Public Property ID() As Integer
        Get
            Return m_id
        End Get
        Set(value As Integer)
            m_id = value
        End Set
    End Property
    Public Property Fname() As String
        Get
```

```
            Return m_fname
        End Get
        Set(value As String)
            m_fname = value
        End Set
    End Property
    Public Property DoB() As DateTime
        Get
            Return m_Dob
        End Get
        Set(value As DateTime)
            m_Dob = value
        End Set
    End Property
    Public Property Address() As String
        Get
            Return m_address
        End Get
        Set(value As String)
            m_address = value
        End Set
    End Property
End Class
'Class End
```

تفعيل اضافة سجل جديد:

VB.NET

```
Private Sub new_btn_Click(sender As Object, e As EventArgs)
        Enabled_btns(True)
        id_txt.Text = InlineAssignHelper(name_txt.Text,
InlineAssignHelper(dob_txt.Text,
InlineAssignHelper(addr_txt.Text, "")))
    End Sub
```

دالة التفعيل:

VB.NET

```
Private Sub Enabled_btns(b As Boolean)
        name_txt.Enabled = InlineAssignHelper(dob_txt.Enabled,
InlineAssignHelper(insert_btn.Enabled,
InlineAssignHelper(addr_txt.Enabled, b)))
    End Sub
```

الغاء التفعيل:

VB.NET

```
        Private Sub cancel_btn_Click(sender As Object, e As EventArgs)
           Enabled_btns(False)
        End Sub
```

دالة ملئ الحقول:

يتم استخدام كائن الـStudents لغرض تصفح البيانات.

VB.NET

```
Private Sub Fill(pos As Integer)
        Enabled_btns(True)
        insert_btn.Enabled = False
        id_txt.Text = student(pos).ID.ToString()
        name_txt.Text = student(pos).Fname
        dob_txt.Text = student(pos).DoB.ToString()
        addr_txt.Text = student(pos).Address
    End Sub
```

ملاحظة: الكائن StdInfo هو نفس الكائن الذي أنشأتها داخل كائن الـLINQ (DBML).

تحديد نص الاتصال لقاعدة البيانات في حدث الـ Load للـForm

VB.NET

```
Private Sub Form1_Load(sender As Object, e As EventArgs)
        dbo = New DataBase(constr) ' Linq تحديد نص الاتصال لقاعدة البيانات التي تتعامل
مع تقنية
    End Sub
```

شرح الكود:

في هذا الحدث يتم اسناد نص الاتصال بقاعدة البيانات الى Std_Dbase لانشاء الاتصال بقاعدة البيانات.

1- اضافة البيانات:

VB.NET

```
Private Sub insert_btn_Click(sender As Object, e As EventArgs)
        'Insert Data
        Dim std1 As New StdInfo()

        std1.FName = name_txt.Text
        std1.DoB = DateTime.Parse(dob_txt.Text)
        std1.Address = addr_txt.Text

        dbo.StdInfos.InsertOnSubmit(std1)
        dbo.SubmitChanges()

        Enabled_btns(False)
        pos = dbo.StdInfos.Count() - 1
        load_btn_Click(Nothing, Nothing)
    End Sub
```

شرح الكود :

اولا يتم تعريف متغير من الكائن StdInfo ومن ثم يتم اسناد القيم الى خصائص هذا الكائن وأخيرا يمرر الكائن الى دالة InsertOnSubmit ليتم اضافة القيم الى الجدول في حال تم تأكيد ذلك عن طريق هذه الدالة SubmitChanges لتأكيد التغييرات على الجدول. (جرب ان تحذف الدالة SubmitChanges ثم لاحظ ماذا يحصل).

2- حذف البيانات:

VB.NET

```
Private Sub delete_btn_Click(sender As Object, e As EventArgs)
        'Delete Data
        If dbo.StdInfos.Count() <> 0 Then

            Dim del As StdInfo =
dbo.StdInfos.SingleOrDefault(Function(c) (c.ID
= Integer.Parse(id_txt.Text)))
```

```
            dbo.StdInfos.DeleteOnSubmit(del)
            dbo.SubmitChanges()
            pos = dbo.StdInfos.Count() - 1
            If pos < 0 Then
                pos = 0
            End If
            load_btn_Click(Nothing, Nothing)
        End If
    End Sub
```

شرح الكود:

في البداية يتم التحقق من عدم خلو الجدول من سجلات ,ومن ثم نستخدم تقنية LINQ لاختيار السجل المساوي لحقل الـID ليتم حذفه.
لاحظ الفرق بين الاستعلام العادي وهذا الاستعلام الخاص بتقنية LINQ المستخدم لحذف السجلات:

```
Delete From table Where Id = @Id
```

الاستعلام أعلاه يتم استخدامه لحذف سجل حسب Id معين من جدول في قاعدة بيانات ما ,وتستخدم ايضا تقنية ADO.NET.

الان لاحظ الاستعلام الخاص بتقنية LINQ لحذف سجل معين

```
 Dim del As StdInfo = (From s In dbo.StdInfos Where s.ID
= Integer.Parse(id_txt.Text) Select s).Single()
```

في نهاية الاستعلام تم استخدام الدالة Single ,هذه الدالة ترجع سجل واحد فقط من نوع الفئة StdInfo وسيتم حدوث استثناء في حال وجود أكثر من سجل يحمل نفس الـ ID.

ويمكن استخدام هذا الكود بدلا من الكود أعلاه وذلك باستخدام تعابير لامبدا (سنستخدم ذلك في تعديل البيانات).

```
Dim del As StdInfo = dbo.StdInfos.SingleOrDefault(Function(c) (c.ID
= Integer.Parse(id_txt.Text)))
```

3- تعديل البيانات:

VB.NET

```
    Private Sub update_btn_Click(sender As Object, e As EventArgs)
        'Update Data
        Dim id As Integer = Integer.Parse(id_txt.Text)

        Dim s As StdInfo = dbo.StdInfos.Single(Function(c) (c.ID = id))
        s.FName = name_txt.Text
        s.DoB = DateTime.Parse(dob_txt.Text)
        s.Address = addr_txt.Text

        dbo.SubmitChanges()
        load_btn_Click(Nothing, Nothing)
    End Sub
```

شرح الكود:

التعديل يتم ايضا اعتمادا على الـID الذي هو PrimaryKey ,كما بينا سيتم استخدام تعابير لامبدا مع تقنية LINQ في التعديل حسب الـID ,بعد اسناد الحقول الى الخصائص يتم استدعاء دالة تأكيد التغييرات على الجدول.

جلب البيانات من قاعدة البيانات:

VB.NET

```
    Private Sub load_btn_Click(sender As Object, e As EventArgs)
        'Load
        If dbo.StdInfos.Any() Then

            Dim ss As IEnumerable = From s In dbo.StdInfos Select s

            student = New List(Of Students)()

            For Each std As StdInfo In ss
                student.Add(New Students() With { _
                 .ID = std.ID, _
                 .Fname = std.FName, _
                 .DoB = std.DoB, _
                 .Address = std.Address _
```

```
                })

                'for autocomplete during searching
                srch_txt.AutoCompleteCustomSource.Add(std.FName)
            Next

            Fill(pos)
        Else
            MessageBox.Show("There is no record in table")
        End If
```

يتم جلب كافة البيانات عن طريق الاستعلام التالي:

```
Dim ss = From s In dbo.StdInfos select s
```

ملاحظة: من الممكن كتابة نوع المتغير في صيغة الاستعلام كما في الاستعلام اعلاه ,حيث تم تعريف المتغير من نوع StdInfo.

وبعد ذلك يتم المرور على كل السجلات واضافتها الى القائمة التي هي من نوع الكائن Students في هذا الكود:

```
For Each std As StdInfo In ss
                student.Add(New Students() With { _
                 .ID = std.ID, _
                 .Fname = std.FName, _
                 .DoB = std.DoB, _
                 .Address = std.Address _
                })
                'for autocomplete during searching
                srch_txt.AutoCompleteCustomSource.Add(std.FName)
            Next
```

في آخر سطر يتم اضافة الاسم الى مصدر بيانات التكملة التلقائية للكلمات الخاص بحقل البحث.

عرض اول سجل:

VB.NET

```
    Private Sub first_btn_Click(sender As Object, e As EventArgs)
```

```
        pos = 0
        Fill(pos)
    End Sub
```

عرض آخر سجل:

VB.NET

```
Private Sub last_btn_Click(sender As Object, e As EventArgs)
        pos = stdent.count - 1
        Fill(pos)
    End Sub
```

عرض السجل التالي:

VB.NET

```
Private Sub next_btn_Click(sender As Object, e As EventArgs)
        If pos < student.Count - 1 Then
            pos += 1
            Fill(pos)
        End If
    End Sub
```

عرض السجل السابق:

VB.NET

```
Private Sub prev_btn_Click(sender As Object, e As EventArgs)
        If pos > 0 Then
            pos -= 1
            Fill(pos)
        End If
    End Sub
```

البحث:

VB.NET

```
Private Sub dosrch_btn_Click(sender As Object, e As EventArgs)

        'Search Data
        load_btn_Click(Nothing, Nothing)
```

```
        Dim stud As StdInfo = (From s In dbo.StdInfos Where s.FName =
srch_txt.Text).SingleOrDefault()

        id_txt.Text = stud.ID.ToString()
        name_txt.Text = stud.FName
        addr_txt.Text = stud.Address
        dob_txt.Text = stud.DoB.ToShortDateString()
    End Sub
```

شرح الكود:

تتم عملية البحث عن طريق الاسم عبر هذا الكود:

```
Dim stud As StdInfo = (
From s In dbo.GetTable(Of StdInfo)() Where s.FName =
srch_txt.Text).SingleOrDefault()
```

لاحظ عملية المقارنة ,تكون ناتج المقارنة إما True أو False لهذا السبب يجب كتابة رمز المساواة مرتين بعكس استعلام SQL حيث يتم كتابة الرمز مرة واحدة فقط ,وأثناء الكتابة في حقل البحث ستظهر قائمة منسدلة بالاسماء المطابقة حسب خاصية التكملة التلقائية للكلمات ,وبما أننا بصدد البحث عن شخص واحد فقط سنستخدم الدالة SingleOrDefault() في نهاية الاستعلام وهذه الدالة تقوم بجلب بيانات سجل واحد فقط من الجدول.

عرض السجلات في الDataGridView:

VB.NET

```
  Private Sub show_btn_Click(sender As Object, e As EventArgs)
        DGV.DataSource = ""

        If dbo.StdInfos.Count() > 0 Then
            DGV.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells
            If dbo.StdInfos.Count() <= 5 Then
                [end] = dbo.StdInfos.Count()
            End If
            DGV.DataSource = student.GetRange(start, [end])
            DGV.Columns(0).HeaderText = "التسلسل"
            DGV.Columns(1).HeaderText = "الاسم الكامل"
```

```
            DGV.Columns(2).HeaderText = "تاريخ الولادة"
            DGV.Columns(3).HeaderText = "العنوان"
        End If
    End Sub
```

شرح الكود:

ملاحظة مهمة: تم الاعتماد على الـPaging في عرض السجلات في الـDataGridView, حيث يتم عرض 5 سجلات فقط من الجدول باستخدام دالة الـGetRange للـList الذي هو من نوع الكائن Students , ويمكن تصفح الصفحات في حال وجود اكثر من 5 سجلات في الجدول.

الدالة GetRange تأخذ مدخلين الاول بداية النسخ والثاني كمية النسخ.

عرض الصفحة التالية في الـDataGridView (5 سجلات في كل صفحة):

VB.NET

```
Private Sub nextpage_btn_Click(sender As Object, e As EventArgs)
        If start < student.Count - 5 Then
            start += end1
            If start + 5 > student.Count Then
                end1 = student.Count - start
            End If
            DGV.DataSource = student.GetRange(start, end1)
        End If
    End Sub
```

شرح الكود:

في البداية احب أن أبين المقصود بالـPaging: التصفح معناه جلب جزء من البيانات من الجدول والتعامل معها والفائدة منها هو تخفيف الحمل على الذاكرة ,تخيل لو أن الجدول يحتوي على ألاف السجلات فبالتأكيد أن قراءة هذه السجلات يستغرق وقتا طويلا لذا يلجأ الكثير الى مبدأ التصفح أي قراءة جزء من البيانات وهكذا ,هنالك طريقة أخرى لقراءة جزء من البيانات وهي عن طريق الاستعلام باستخدام الكلمة المحجوزة Top(number) كالتالي:

```
Select Top(10) * from table
```

عرض الصفحة السابقة في الـDataGridView:

VB.NET

```
    Private Sub prevpage_btn_Click(sender As Object, e As EventArgs)
        If start > 0 Then
            If end1 < 5 Then
                end1 = 5
            End If
            start -= end1
            DGV.DataSource = student.GetRange(start, end1)
        End If
    End Sub
```

الطباعة والتقرير:

1- طباعة السجل الحالي:

VB.NET

```
    Private Sub current_btn_Click(sender As Object, e As EventArgs)
        Dim repf As New ReportForm()

        rep_source = New ReportDataSource("student", student)
        repf.reportViewer1.LocalReport.DataSources.Add(rep_source)

        rep_param = New List(Of ReportParameter)()
        rep_param.Add(New ReportParameter("ID", id_txt.Text))
        rep_param.Add(New ReportParameter("Fname", name_txt.Text))
        rep_param.Add(New ReportParameter("Dob",
DateTime.Parse(dob_txt.Text).ToShortDateString()))
        rep_param.Add(New ReportParameter("Address", addr_txt.Text))

        repf.reportViewer1.LocalReport.SetParameters(rep_param)
        repf.reportViewer1.RefreshReport()

        repf.ShowDialog()
    End Sub
```

شرح الكود:

في البداية يجب تحديد مصدر البيانات للتقرير وبعد ذلك تمرير البيانات الى التقرير ,وبما أننا نعتمد على الباراميترات يجب قبل تمرير الباراميترات ان نسند البيانات التي ستظره في التقرير الى هذه الباراميترات وذلك عن طريق اسماء البارميترات المصممة في التقرير ويمكن ان تسند البيانات إما عن طريق اسم الباراميتر أو الـIndex للباراميتر ,ويتم ذلك عن طريق الكائن ReportParameter وهذا الكائن يأخذ مدخلين الاول إما اسم الباراميتر او الـIndex للباراميتر حسب الترتيب والثاني البيان (مفرد بيانات) التي ستسند الى الباراميتر ,وبما أننا بصدد طباعة بيانات السجل الحالي سنسند القيم الحالية الظاهرة في أدوات الـTextBox الى الباراميترات كما هو مبين في الكود أعلاه.
وأخيرا نمرر قائمة الباراميترات الى التقرير عن طريق الدالة SetParameters.

2- طباعة السجلات المحددة من الـ DataGridViewعن طريق الفأرة:

VB.NET

```
Private Sub selected_btn_Click(sender As Object, e As EventArgs)
        Dim repf As New ReportForm()

        Dim rc As DataGridViewSelectedRowCollection = DGV.SelectedRows

        rep_source = New ReportDataSource("student", student)
        repf.reportViewer1.LocalReport.DataSources.Add(rep_source)

        Dim i As Integer = 0
        Dim s1 As String = "", s2 As String = "", s3 As String = "",
s4 As String = ""

        For i = 0 To rc.Count - 1
            s1 += rc(i).Cells("ID").Value.ToString() &
Environment.NewLine & Environment.NewLine
            s2 += rc(i).Cells("FName").Value.ToString() &
Environment.NewLine & Environment.NewLine
            s3 +=
DateTime.Parse(rc(i).Cells("DoB").Value.ToString()).ToShortDateString()
& Environment.NewLine & Environment.NewLine
            s4 += rc(i).Cells("Address").Value.ToString() &
Environment.NewLine & Environment.NewLine
        Next
```

```
        rep_param = New List(Of ReportParameter)()
        rep_param.Add(New ReportParameter("ID", s1))
        rep_param.Add(New ReportParameter("Fname", s2))
        rep_param.Add(New ReportParameter("Dob", s3))
        rep_param.Add(New ReportParameter("Address", s4))

        repf.reportViewer1.LocalReport.SetParameters(rep_param)
        repf.reportViewer1.RefreshReport()

        repf.ShowDialog()
    End Sub
```

شرح الكود:

في هذا الجزء سيقوم المستخدم بطباعة السجلات التي سيحددها من الـDataGridView عن طريق الماوس ,كأن يحدد سجل او أكثر.

في البداية وقبل تحديد مصدر البيانات للتقرير يجب احتواء الصفوف المحددة باستخدام هذا الكود:

```
Dim rc As DataGridViewSelectedRowCollection = DGV.SelectedRows
```

وظيفة الكود أعلاه هو وضع الصفوف المحددة في Collection من نوع DataGridViewSelectedRowCollection ,وبعد ذلك يتم تحديد مصدر البيانات ومن ثم تعريف 4 متغيرات من نوع String كل متغير خاص بحقل واحد من الـDataGridView وأخيرا يتم تمرير هذه المتغيرات التي تضم الحقول المحددة الى التقرير.

3- طباعة كل السجلات:

VB.NET

```
    Private Sub all_btn_Click(sender As Object, e As EventArgs)
        Dim repf As New ReportForm()

        rep_source = New ReportDataSource("student", student)
        repf.reportViewer1.LocalReport.DataSources.Add(rep_source)

        Dim i As Integer = 0
        Dim s1 As String = "", s2 As String = "", s3 As String = "",
s4 As String = ""
```

```
        For i = 0 To student.Count - 1
            s1 += student(i).ID.ToString() & Environment.NewLine &
Environment.NewLine
            s2 += student(i).Fname.ToString() & Environment.NewLine &
Environment.NewLine
            s3 +=
DateTime.Parse(student(i).DoB.ToString()).ToShortDateString() &
Environment.NewLine & Environment.NewLine
            s4 += student(i).Address.ToString() & Environment.NewLine &
Environment.NewLine
        Next

        rep_param = New List(Of ReportParameter)()
        rep_param.Add(New ReportParameter("ID", s1))
        rep_param.Add(New ReportParameter("Fname", s2))
        rep_param.Add(New ReportParameter("Dob", s3))
        rep_param.Add(New ReportParameter("Address", s4))

        repf.reportViewer1.PageCountMode = PageCountMode.Actual
        repf.reportViewer1.LocalReport.SetParameters(rep_param)
        repf.reportViewer1.RefreshReport()
        'Showing the report form
        repf.ShowDialog()
    End Sub
```

شرح الكود:

أخيرا طباعة كل ما موجود من سجل في الجدول ,بينا فيما سبق أنه اثناء تحميل البيانات يتم وضعها في قائمة من نوع Students اذن كل البيانات توجد في هذه القائمة لذا يتم المرور على كافة السجلات ويتم وضعها كما ذكر سابقا في المتغيرات النصية ومن ثم يتم تمريرها الى التقرير.

وأخيرا لا تنسوني و والدي من صالح دعواتكم وإن اخطأت فهو مني وإن اصبت فمن الله

والله ولي التوفيق

البريد الألكتروني: sajad_88m@yahoo.com .