LINQ (Language Integrated Query)

可以整合不同類型(SQL/XML/DataSet/C#object)的查詢語法

LINQ to Object

LINQ to DataSets

LINQ to SQL

LINQ to XML

--------------------------------------------

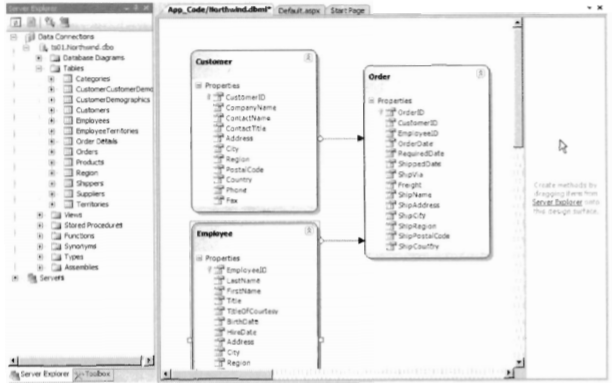
P 105

LINQ to SQL

Object Relational Designer 圖形化界面

簡稱︰O/R Designer

在資料庫和程式之間傳送和接收資料

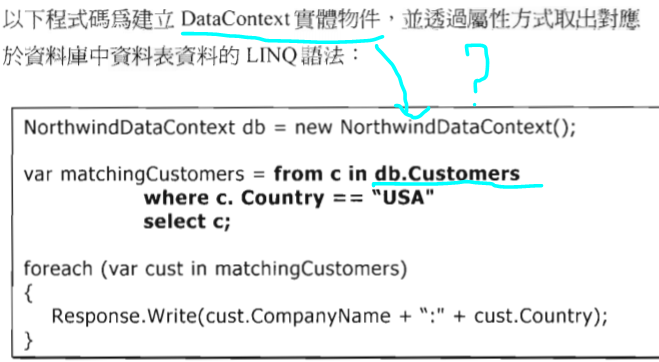


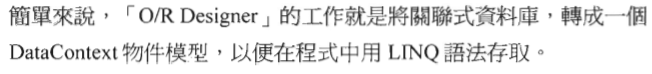
資料表 <對應為> Entity類別

欄位 <> Entity 屬性

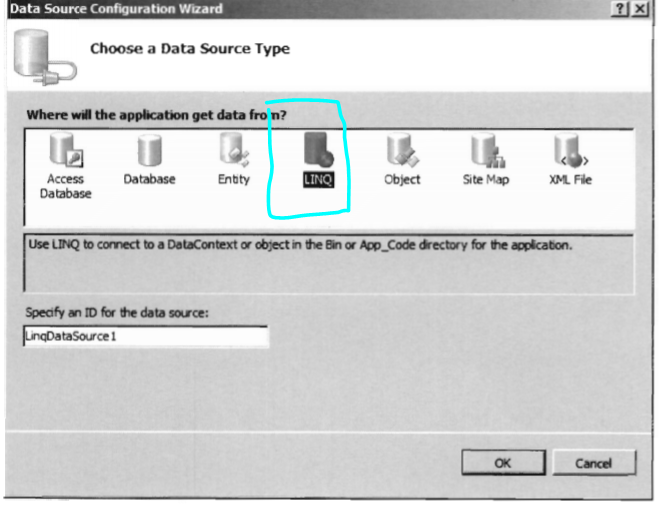
每筆資料 <> Entity實例

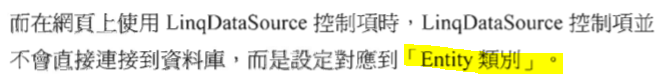
**DataContext 類別**



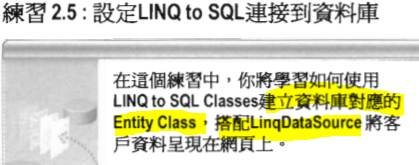
 ??????

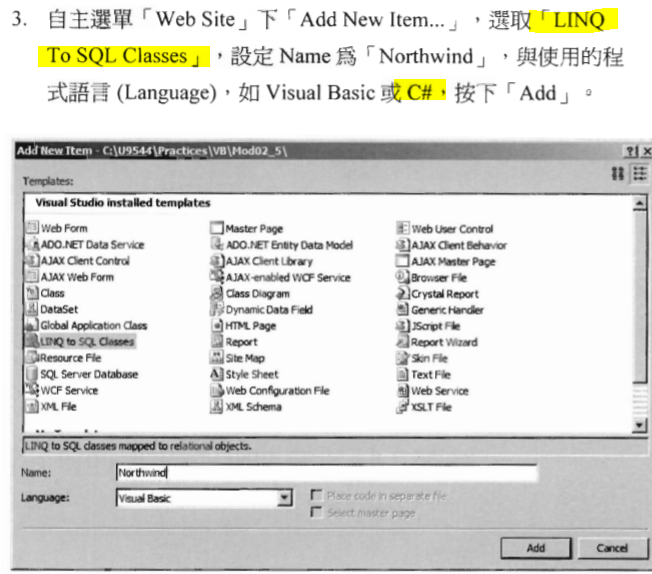
LinqDataSource P 108



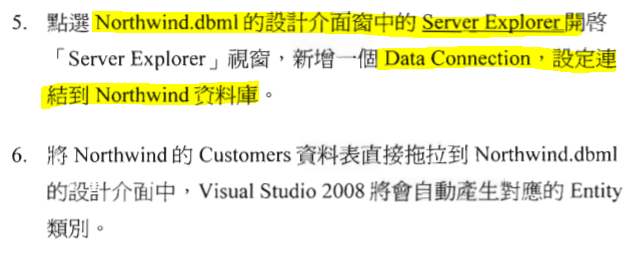


實作︰ P 110

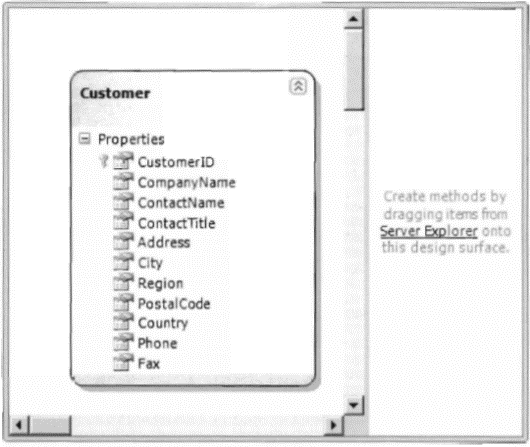




是否放入 App\_Code 資料夾 -- > Yes

???? 一臉矇逼

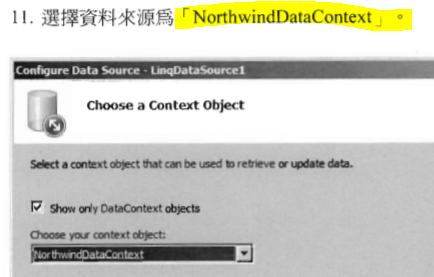
這些是在哪裡呀？

傳說中的效果 ( 沒看過 )

Build一下

建立 Web Form

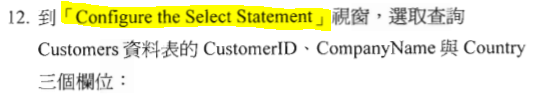
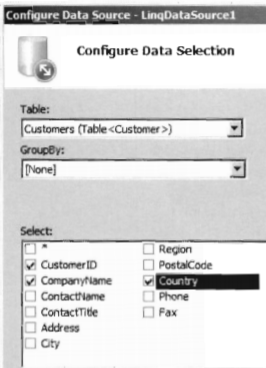
拉一個ListView > Choose Data Source > New data source > LinqDataSource



NorthwindDataContext

是在上方「傳說中的效果」中

建好的？







跑出來的效果 ( 還沒有實作過 )

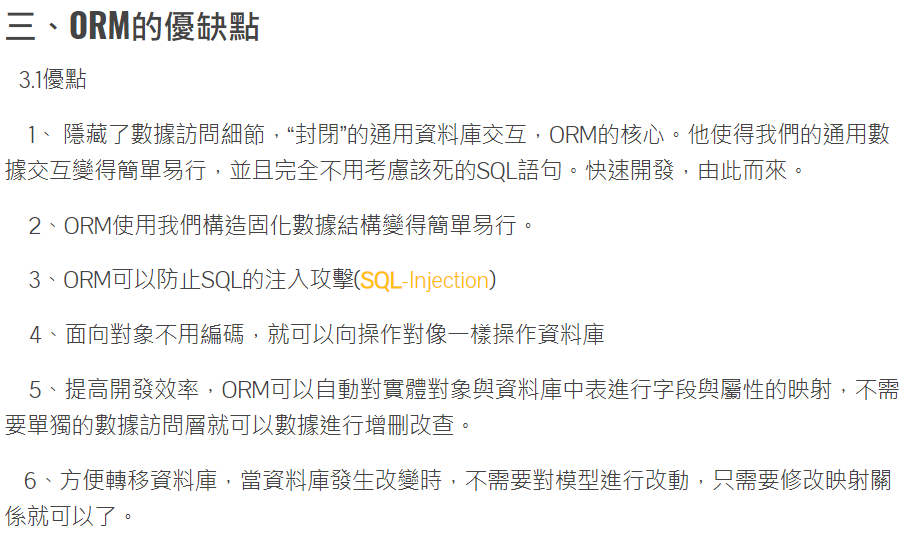
教材二的第二章完結，但我一﹑二章還是很多都沒有實作過…

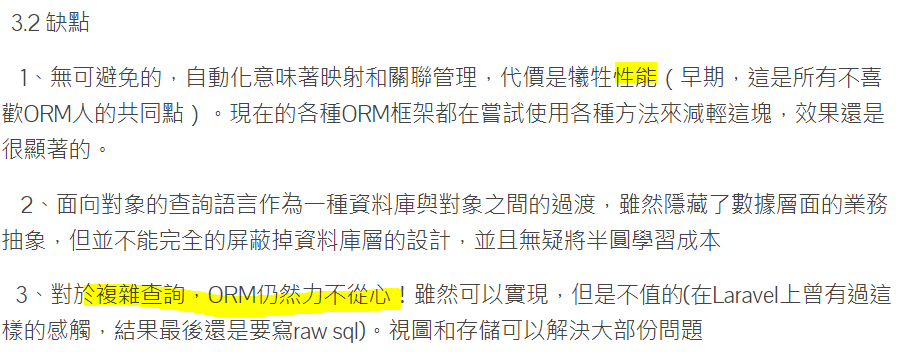
課外補充這一個主題︰

課外補充︰

O/R Designer (OR Mapping)的優缺︰

<http://blog.twbryce.com/what-is-orm/>

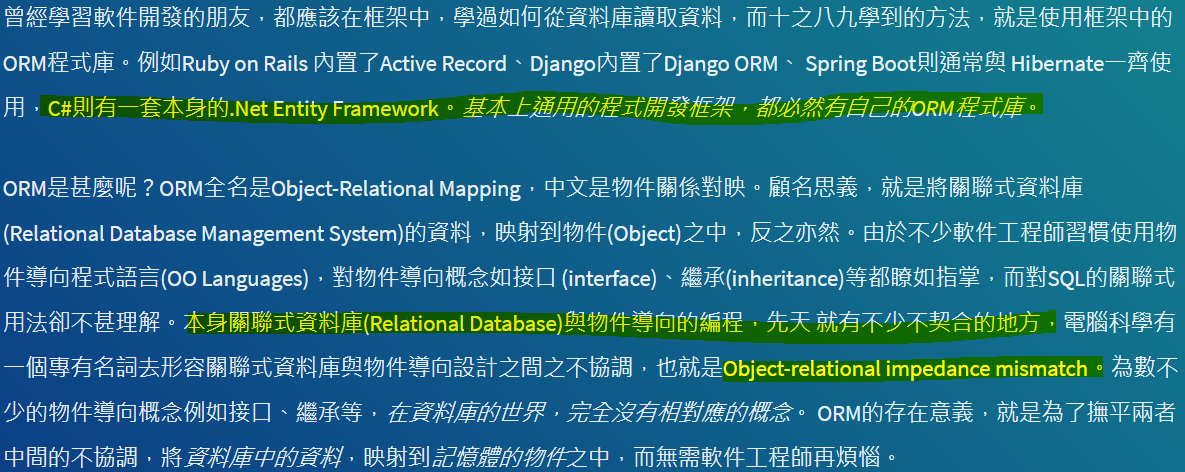


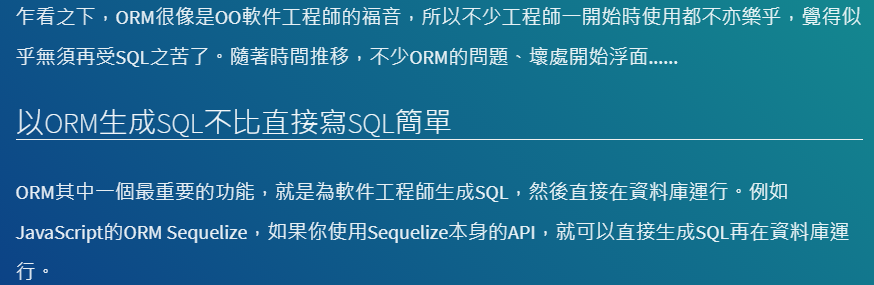


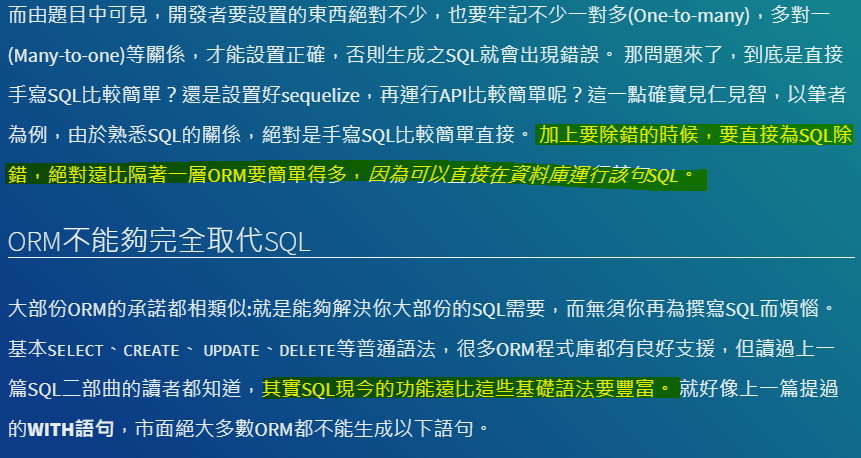
<https://tecky.io/en/blog/SQL%E4%B8%89%E9%83%A8%E6%9B%B2:%E4%BD%A0%E4%B8%8D%E9%9C%80%E8%A6%81ORM/>

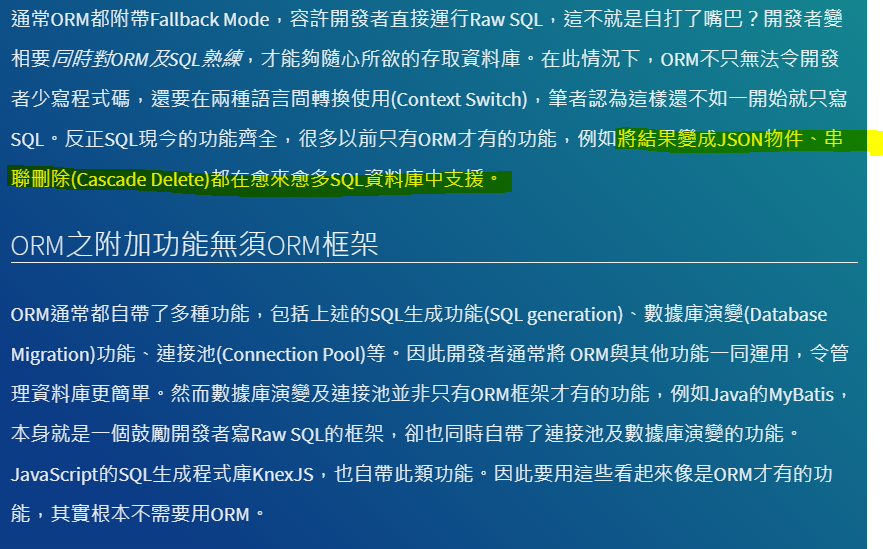
SQL三部曲:你不需要ORM

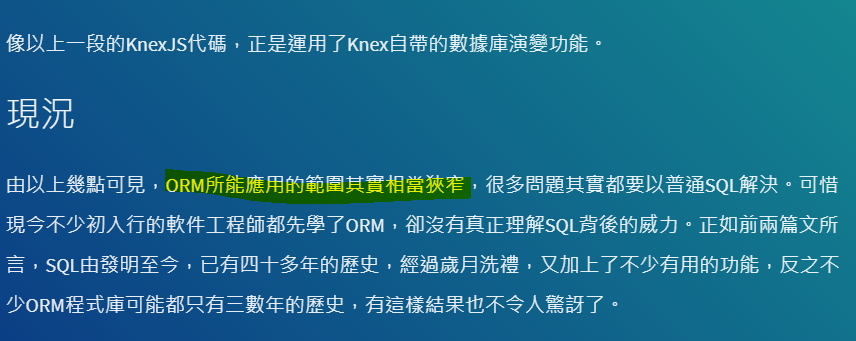
Gordon Lau 2019-10-22







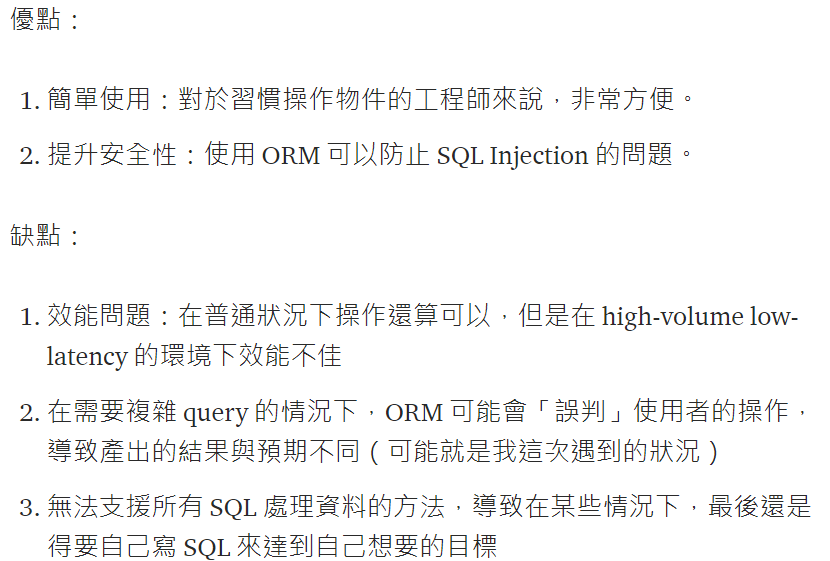




總的來說，此文不推O/R Designer

<https://medium.com/tds-note/orm-v-s-sql-91e003089a61>

也不推︰



<http://www.ruanyifeng.com/blog/2019/02/orm-tutorial.html>

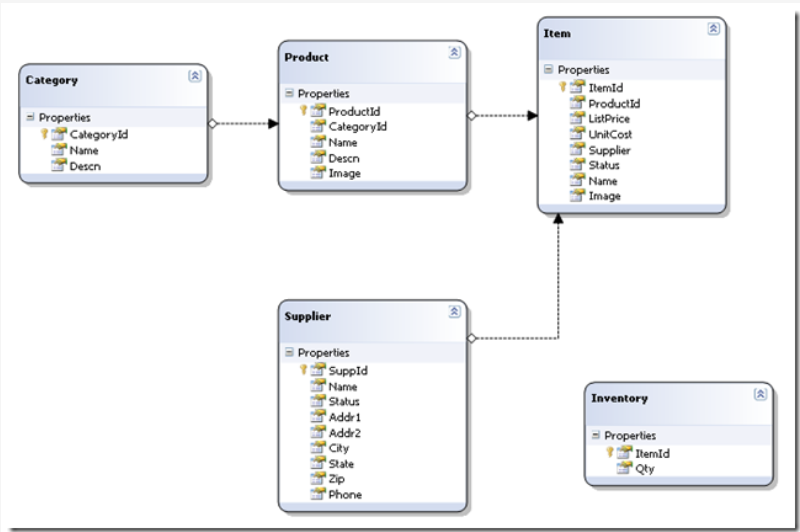
也不推…

總的來說，還是學好 SQL 才是正道…

但以下是一個網路上的應用例子︰

<https://dotblogs.com.tw/alonstar/2009/09/12/10583>





由上圖我們可以知道，Category和Product是二個相關聯的table。接下來新增一個頁面，放入LinqDataSource及GridView…，資料來源設定為Product

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataKeyNames="ProductId" DataSourceID="LinqDataSource1">

<Columns>

<asp:BoundField DataField="ProductId" HeaderText="ProductId" ReadOnly="True" SortExpression="ProductId" />

<asp:BoundField DataField="CategoryId" HeaderText="CategoryId" SortExpression="CategoryId" />

<asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />

<asp:BoundField DataField="Descn" HeaderText="Descn" SortExpression="Descn" />

<asp:BoundField DataField="Image" HeaderText="Image" SortExpression="Image" />

</Columns>

</asp:GridView>

<asp:LinqDataSource ID="LinqDataSource1" runat="server" ContextTypeName="DataClassesDataContext" TableName="Product">

</asp:LinqDataSource>

以上是我們基本一般單一資料表的寫法，如果今天我想同時顯示Category的名稱，那要怎麼做呢？

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataKeyNames="ProductId" DataSourceID="LinqDataSource1">

<Columns>

<asp:BoundField DataField="ProductId" HeaderText="ProductId" ReadOnly="True" SortExpression="ProductId" />

<asp:TemplateField HeaderText="CategoryName">

<ItemTemplate>

<%#Eval("Category.Name") %>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="CategoryId" HeaderText="CategoryId" SortExpression="CategoryId" />

<asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />

</Columns>

</asp:GridView>

<asp:LinqDataSource ID="LinqDataSource1" runat="server" ContextTypeName="DataClassesDataContext" TableName="Product">

</asp:LinqDataSource>

只要短短一行的<%#Eval(“Category.Name”)%>他就可以幫你帶出來囉！再加上他本身是物件，所以甚至可以回傳後再處理，就可以達到我想要的效果。例如：（記得存取權限問題，方法要宣告為protected。）

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataKeyNames="ProductId" DataSourceID="LinqDataSource1">

<Columns>

<asp:BoundField DataField="ProductId" HeaderText="ProductId" ReadOnly="True" SortExpression="ProductId" />

<asp:TemplateField HeaderText="CategoryName">

<ItemTemplate>

<%#addLabel(Eval("Category")) %>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />

</Columns>

</asp:GridView>

<asp:LinqDataSource ID="LinqDataSource1" runat="server" ContextTypeName="DataClassesDataContext" TableName="Product">

</asp:LinqDataSource>

-------------------------------------------------

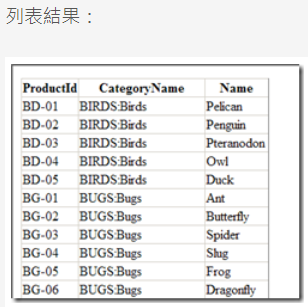
protected string addLabel(object o)

{ Category c = (Category)o;

return String.Format("{0}:{1}", c.CategoryId, c.Name);

}

-------------------------------------------------



此外，教材沒有細講，但是我想知道︰

LINQ to DataSet

LINQ to C# Object

於是我開新一份Word

放了在 小補充> C#語法那一邊。