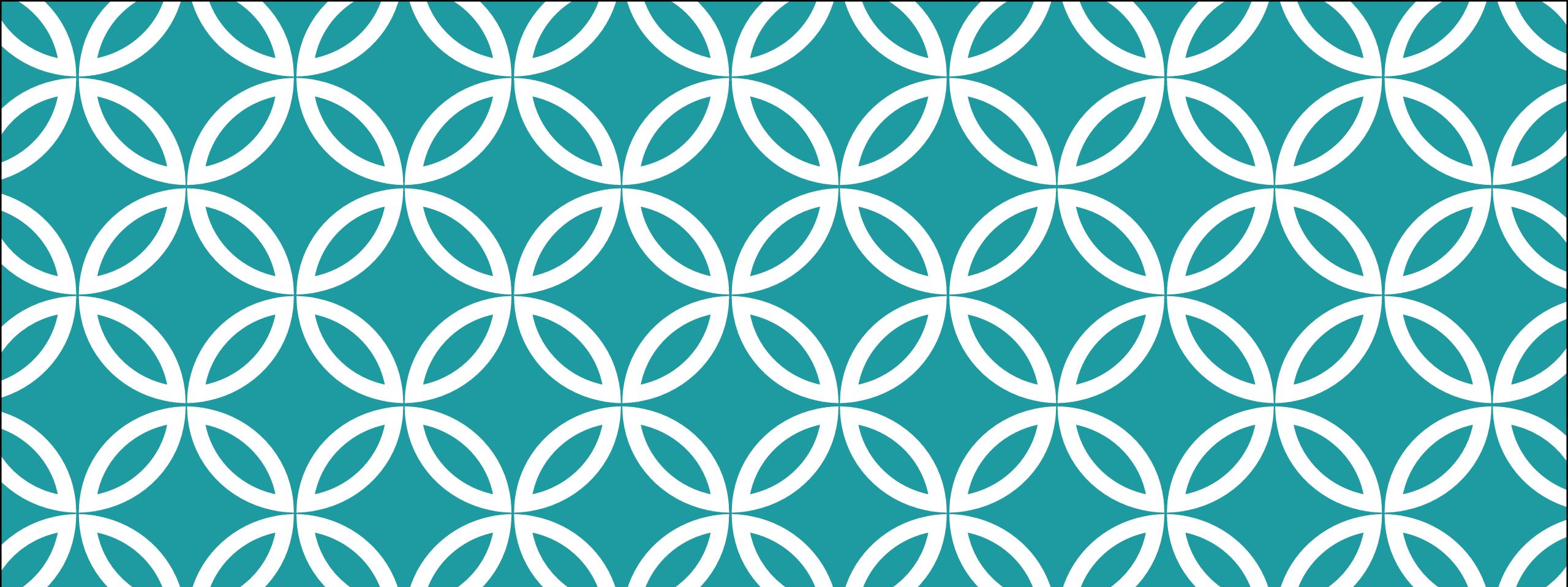


ASP.NET Web Form





SQL SERVER



SQL SERVER

資料庫系統下載及安裝

- 下載SQL Server 2019 Express：<https://www.microsoft.com/zh-tw/sql-server/sql-server-downloads>



Express

SQL Server 2019 Express 是免費的 SQL Server 版本，非常適合用來開發和生產桌面、Web 與小型伺服器應用程式。

- 下載完成後請選擇安裝免費的基本款



SQL SERVER

資料庫系統下載及安裝

- 安裝完SQL Server後，接著下載SSMS(SQL Server Management Studio)：

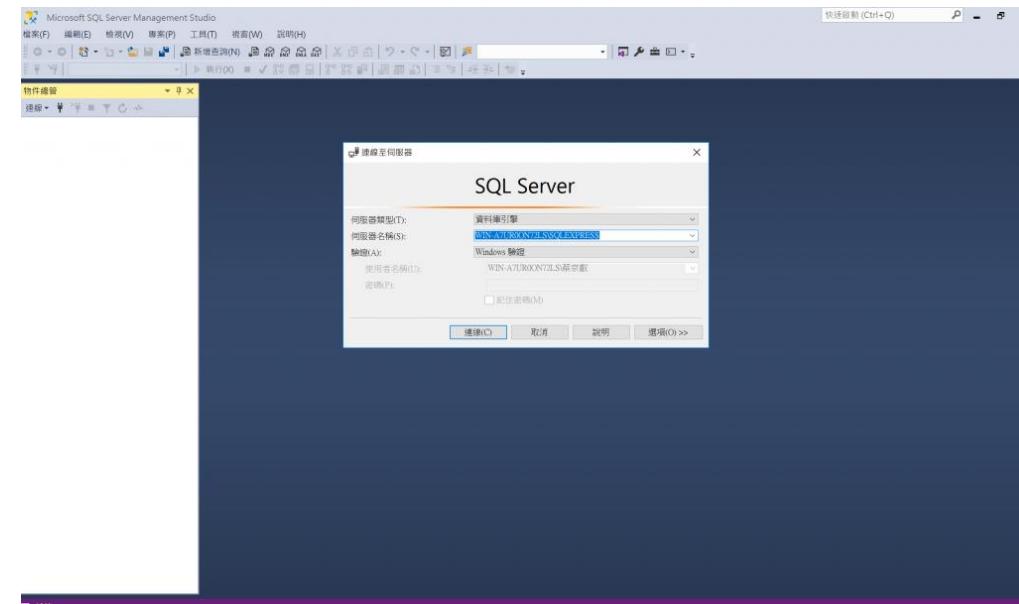
- <https://go.microsoft.com/fwlink/?LinkId=2099720>



SQL SERVER

資料庫系統下載及安裝

- 接著打開SQL Server資料庫系統：
- 開啟後會看到需要連線登入的畫面，就先連自己的伺服器及選擇Windows驗證登入即可(也可使用SQL Server驗證，不過這需要創個帳號及密碼)

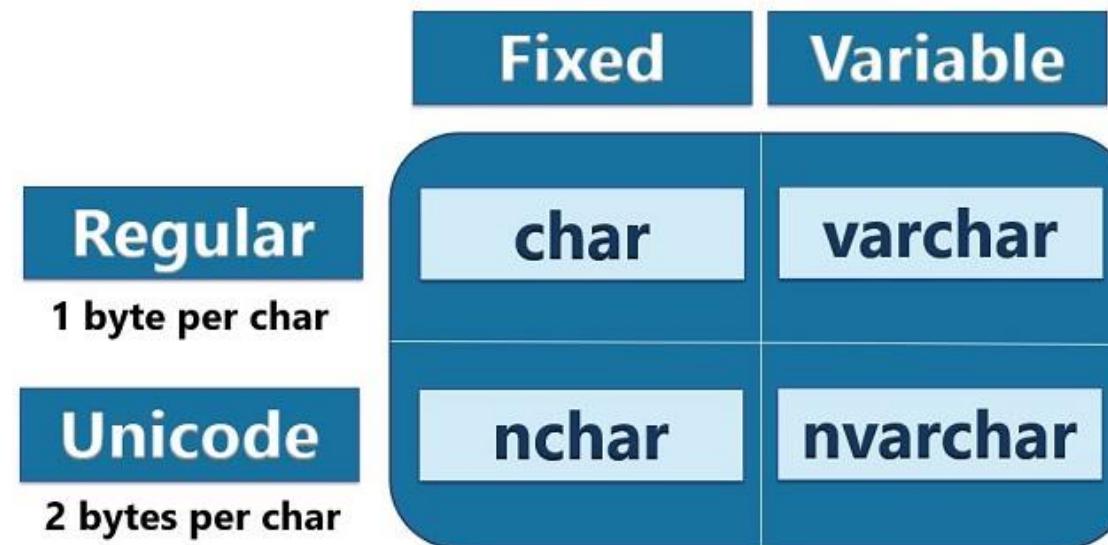


- 成功登入後會看到物件總管內有系統內建的資料庫，代表成功登入資料庫系統了

SQL SERVER

SQL Server 資料型態

- 建立資料表時需要先選擇欄位的資料型態，而當要輸入文字存入資料時，在SQL Server大致上有**char varchar nchar nvarchar** text ntext 這6種可以選擇。
- 仔細觀察以上型態從字面上的差別只在有無var以及有無n。



SQL SERVER

SQL Server 資料型態

- 定長與變長
 - 定長(沒有var)
 - 文字長度固定，當輸入的數據長度沒有達到指定的長度時將自動以英文空格在其後面填充，讓長度達到相對應的長度。
 - 變長(有var)
 - 實際存儲空間是變長的，當輸入的數據長度沒有達到指定的長度時不會在後面填充空格。(不過text所存儲的也是可變長的)。
- 舉個例子來說，現在資料表有兩個欄位：

rocket1 char(10)、rocket2 varchar(10)

- 存入相同的資料「HiIU」，
 - rocket1 會自動在「HiIU」後面補空白，存滿10 Bytes。
 - rocket2 會根據資料長度變更儲存空間，「HiIU」共四個英文字符，所以則存了4 Bytes。
- 使用varchar相對節省空間。

SQL SERVER

SQL Server 資料型態

- Unicode或非Unicode
 - 在資料庫中，每個英文字符的儲存空間只需要一個 1 Byte，但若是非英文字及符號，如中文、泰文、日文、韓文等非英文字符，則需要 2 Byte來儲存。
 - 如果英文與中文同時存在，由於占用空間數不同，容易造成混亂，導致讀取出來的字符串是亂碼。所以就要使用Unicode字符集(N)。
 - Unicode(有n)：
 - 所有的文字都用2 Byte來儲存，即使是英文字也是使用2 Byte來儲存，就可以解決中英文字符集不兼容的問題。
 - 非Unicode(沒有n)：
 - 文字是英文字符就是1 Byte;非英文字符就是使用2 Byte來儲存。
- 舉例來說，一樣有兩個欄位：

rocket1 varchar(50)、rocket2 nvarchar(50)

- 存入相同的資料「我愛TsungHua」
 - rocket1: 我愛 = 22= 4 Byte ; TsungHua =81= 8 Byte ，所以總共 $4+8=12$ Byte的儲存空間。
 - rocket2: 「我愛TsungHua」共10個字，所以儲存空間為 $10*2=20$ Byte。
- 雖然rocket2較佔儲存空間，但若儲存的資料不限於英文，建議都以n開頭的欄位型態為主，避免亂碼。

SQL SERVER

SQL Server 資料型態

- 有var的：
 - 在存不固定長度的資料時，較省空間。
- 有n的：
 - 支援Unicode，不過儲存的資料會大兩倍(全部都英文字的話啦)。
 - 然後建議開資料表欄位時，要以儲存的資料為考量，
- 建議：
 - char 存資料有固定長度，並且都為英文數字。
 - nchar 存資料有固定長度，但不確定是否都是英文數字。
 - varchar 存資料沒有固定長度，並且都為英文數字。
 - nvarchar 存資料沒有固定長度，且不確定是否皆為英文數字。

SQL SERVER

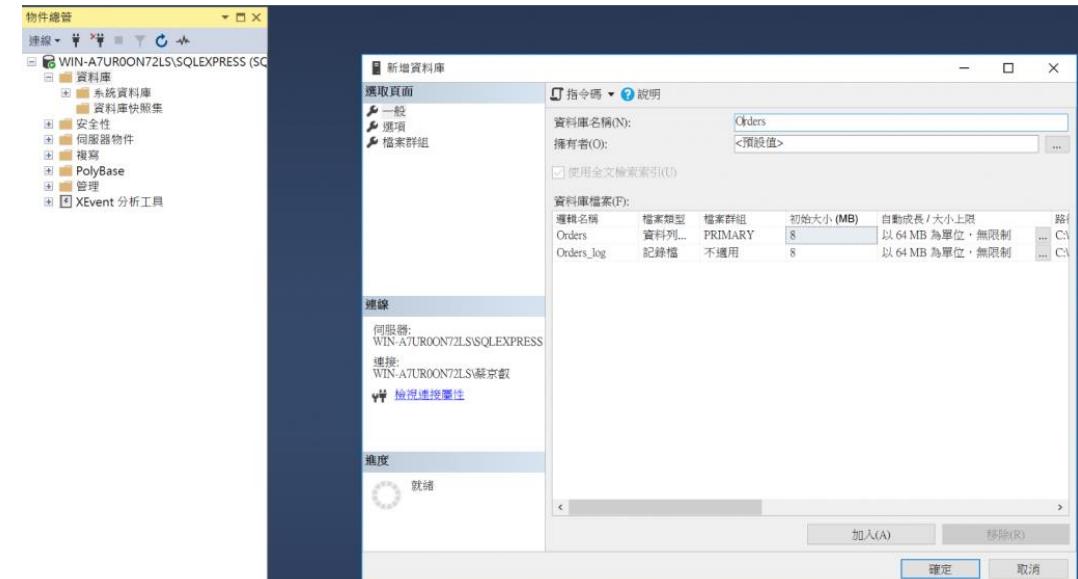
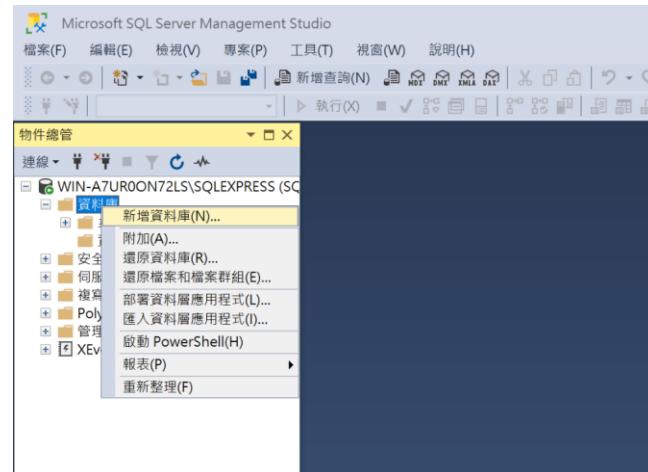
SQL Server 資料型態

- **char(n) 定長：**
 - n 必須是一個介於 1 和 8000 之間的數值，最多8000個英文字，4000個非英文字，儲存大小為 n 個位元組。
- **nchar(n) 定長：**
 - n 的值必須介於 1 與 4000 之間。儲存大小為 n 位元組的兩倍。
- **varchar(n) 變長：**
 - n 必須是一個介於 1 和 8000 之間的數值，可存儲4000個字，無論英文還是非英文，儲存大小為輸入資料的位元組的實際長度。
- **nvarchar(n) 變長：**
 - n 的值必須介於 1 與 4000 之間，位元組的儲存大小是所輸入字元個數的兩倍。
- **text(n) 變長：**
 - 非Unicode數據，最大長度為 $2^{31}-1(2,147,483,647)$ 個字符。
- **ntext(n) 變長：**
 - 最大字串長度為 $2^{30} - 1 (1,073,741,823)$ 位元組的可變長度 Unicode 資料。儲存體大小 (以位元組為單位) 是輸入字串長度的兩倍。

SQL SERVER

建立資料表

- 上圖這張出貨單，把它分成Customers,Products,Orders,OrderDetail四塊資料表。可以把資料表想像成是像Excel一樣的表單，裡面會有欄跟列來做排序。
 - 建立資料表前呢，首先要先建立一個訂單的資料庫。請對系統資料庫按右鍵選新增資料庫。



- 建立一個名叫Orders的資料庫並按下確定。

SQL SERVER

建立資料表

- 建立資料庫後，接下來就要創立資料表了，
 - 請對**Orders**資料庫內的資料表資料夾按右鍵新增資料表，會看到可填入資料行的畫面：

資料行名稱	資料類型	允許 Null
		<input type="checkbox"/>

- 資料行就是要填入資料的項目名稱，以出貨單為例就是：產品名稱、客戶姓名、數量等...
 - 不過一般而已，第一項資料行通常都會設叫id做關聯圖時的主索引鍵，主索引鍵(Key)可以讓資料庫快速的找到相關的資料。而id的資料型態就先設定為int，接著看到正下方有一堆屬性，找到最下面有一個識別規格，預設為"否"：

資料行名稱	資料類型	允許 Null
id	int	<input checked="" type="checkbox"/> <input type="checkbox"/>

資料行屬性	
	有非 SQL Server 訂閱者
決定性	否
定序	是
為疏鬆	<資料庫預設值>
為資料行集	否
> 計算資料行規格	
描述	否
發佈的 DTS	否
發佈的合併	否
資料類型摘要	int
> 識別規格	
識別規格	否

SQL SERVER

建立資料表

- 接著對識別規格點兩下左鍵會往下展延，然後對(為識別)一樣點兩下左鍵將否改為"是"，這個意思就是說，每次在這個資料表建立資料時，id這個項目的內容每次都會自動+1(就不用打了)。

為疏鬆	否
為資料行集	否
➤ 計算資料行規格	
描述	
發佈的 DTS	否
發佈的合併	否
資料類型扼要	int
▼ 識別規格	
(為識別)	是
識別值種子	1
識別值增量	1
(為識別)	

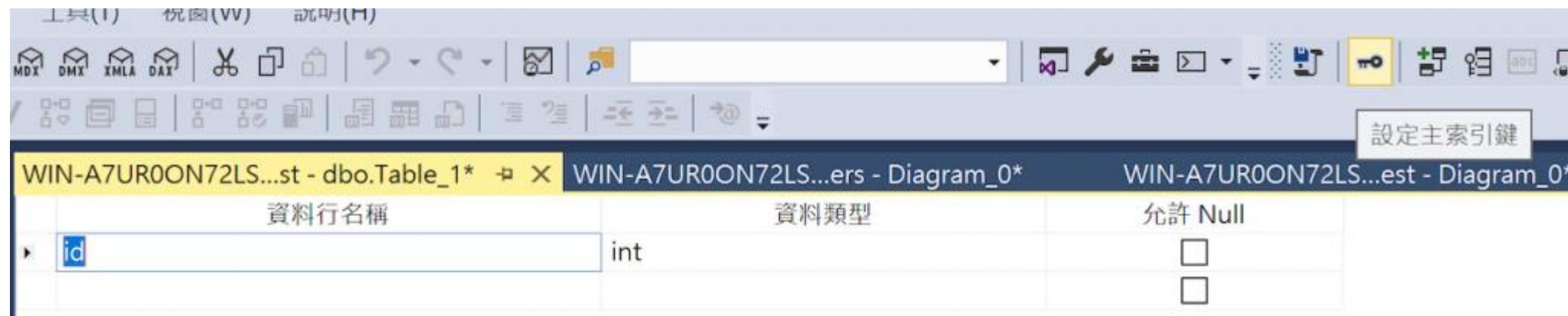
X X X公司				
出貨單		出貨日期		
客戶姓名		客戶電話		
運送地址				
產品名稱	數量	單價	金額	備註
客戶簽名	總計			

SQL SERVER

建立資料表

- 最後要幫id加上主索引鍵，在上方有個鑰匙圖案，按下後就可以設定主索引鍵。

X X X公司				
出貨單			出貨日期	
			出貨單號	
客戶姓名			客戶電話	
運送地址				
產品名稱	數量	單價	金額	備註
客戶簽名			總計	



- 設定後會發現，id列前方的小圖示會變成一個鑰匙，就代表已經是主索引鍵了！(預設不可為NULL)



SQL SERVER

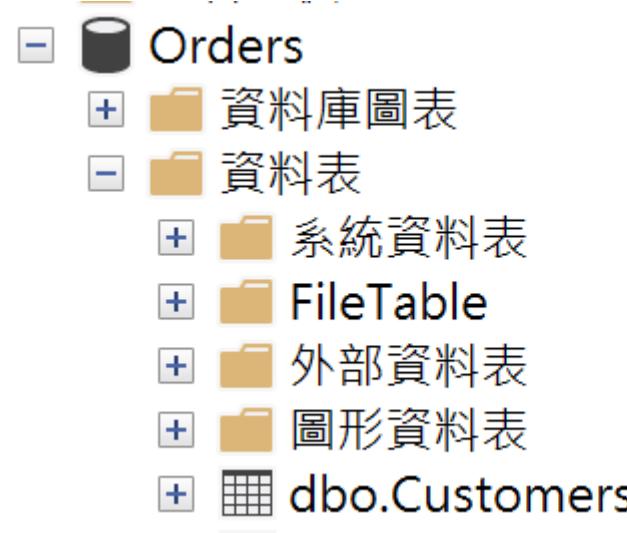
建立資料表

- 接下來依序填入其他項目，不過其他項目就不用設定成主索引鍵了(通常一個資料表都只會有一個主索引)
 - 比較要一提的是initDate，通常在資料表內都會加上這個資料行的項目是代表建立資料的時間，而資料型態使用datetime，並在下方一般的預設值或繁結輸入`getdate()`方法，會自動抓取創立時間。

SQL SERVER

建立資料表

- 接著按儲存(ctrl+s)命名為 Customers(這是用來存放客戶資料的資料表)
- 接著對資料表右鍵重新整理後，Customers的資料表就出現了。



- 現在建立完資料表了，接下來繼續完成其他資料表並輸入資料與索引鍵連線。

X X X公司				
出貨單		出貨日期		
客戶姓名		客戶電話		
產品名稱	數量	單價	金額	備註
客戶簽名		總計		

SQL SERVER

存入資料內容

- 將假設的資料輸入進資料表內，因為這是客戶的資料表，所以要輸入客戶的資料。首先對**Customers**資料表按下右鍵並選擇"編輯前200個資料列"：



- 會看到一開始因為還沒有輸入資料，所以一筆資料都沒有。

	id	name	tel	addr...	initD...
▶*	NULL	NULL	NULL	NULL	NULL

SQL SERVER

存入資料內容

- 接著請在name那欄填入客戶名稱：apple；tel輸入電話：123456789；address輸入地址：USA(如果點了無法輸入，請先按一下Enter鍵，就可以輸入了)
- 接著輸入第二筆客戶資料 name:asus；tel：987654321；address：Taiwan，然後輸入第三筆客戶資料 name:sony；tel：52012345；address：Japan，並按下儲存後，接著對資料表重新執行編輯前200個資料列

	id	name	tel	addr...	initDate
▶	1	apple	123456789	USA	2019-09-02 10:40:22.147
	2	asus	987654321	Taiwan	2019-09-02 10:40:42.553
	3	sony	52012345	Japan	2019-09-02 10:41:22.153
*	NULL	NULL	NULL	NULL	NULL

- 會發現id以及initDate，系統已經自動幫忙依序輸入1,2,3還有輸入建立的時間了。
- 這是因為id有將為識別開啟為是以及initDate設定getdate()的關係。

SQL SERVER

存入資料內容

- 接下來繼續創立另外三個資料表及存入資料進去。
 - orders** 資料表：訂單資料，這邊的customerID是要用來接customers資料表的索引值的。

資料行名稱	資料類型	允許 Null
id	int	<input type="checkbox"/>
customerID	int	<input checked="" type="checkbox"/>
total	int	<input checked="" type="checkbox"/>
initDate	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

- Products** 資料表：創一個用來存放產品資料的資料表，並且將兩筆資料存進去。

資料行名稱	資料類型	允許 Null
id	int	<input type="checkbox"/>
name	nvarchar(50)	<input checked="" type="checkbox"/>
price	int	<input checked="" type="checkbox"/>
initDate	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

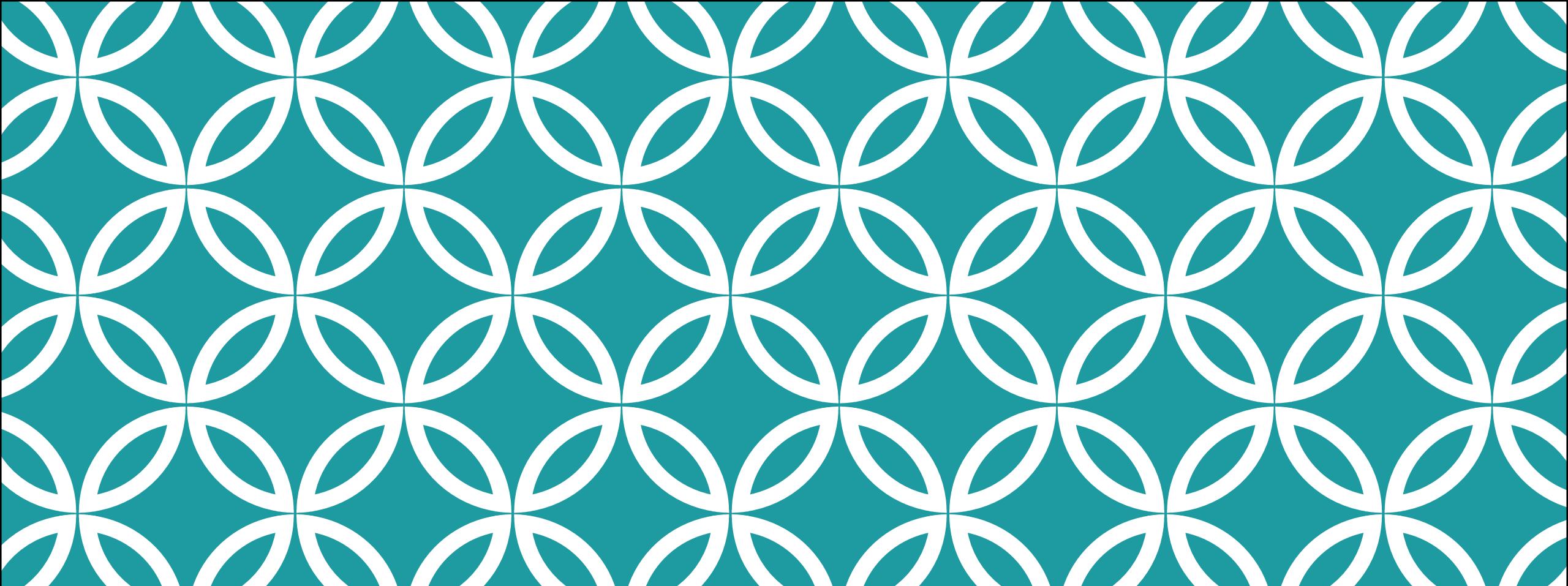
	id	name	price	initDate
1	1	糖果	100	2019-09-02 14:15:08.723
2	2	餅感	200	2019-09-02 14:15:21.187
▶*	NULL	NULL	NULL	NULL

SQL SERVER

存入資料內容

- 接下來繼續創立另外三個資料表及存入資料進去。
 - Orderdetial**資料表：客戶所購買的內容資料都在這個資料表裡，orderID及productID是用來接索引值的

	資料行名稱	資料類型	允許 Null
•	id	int	<input checked="" type="checkbox"/>
	orderID	int	<input checked="" type="checkbox"/>
	productID	int	<input checked="" type="checkbox"/>
	amount	int	<input checked="" type="checkbox"/>
	price	int	<input checked="" type="checkbox"/>
	total	int	<input checked="" type="checkbox"/>
	memo	nvarchar(50)	<input checked="" type="checkbox"/>
	initDate	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>



建立資料庫圖表



建立資料庫圖表

接下來要建立資料庫圖表，首先對著資料庫圖表按下"新增資料庫圖表"：



接著選取所有資料表，並按加入。

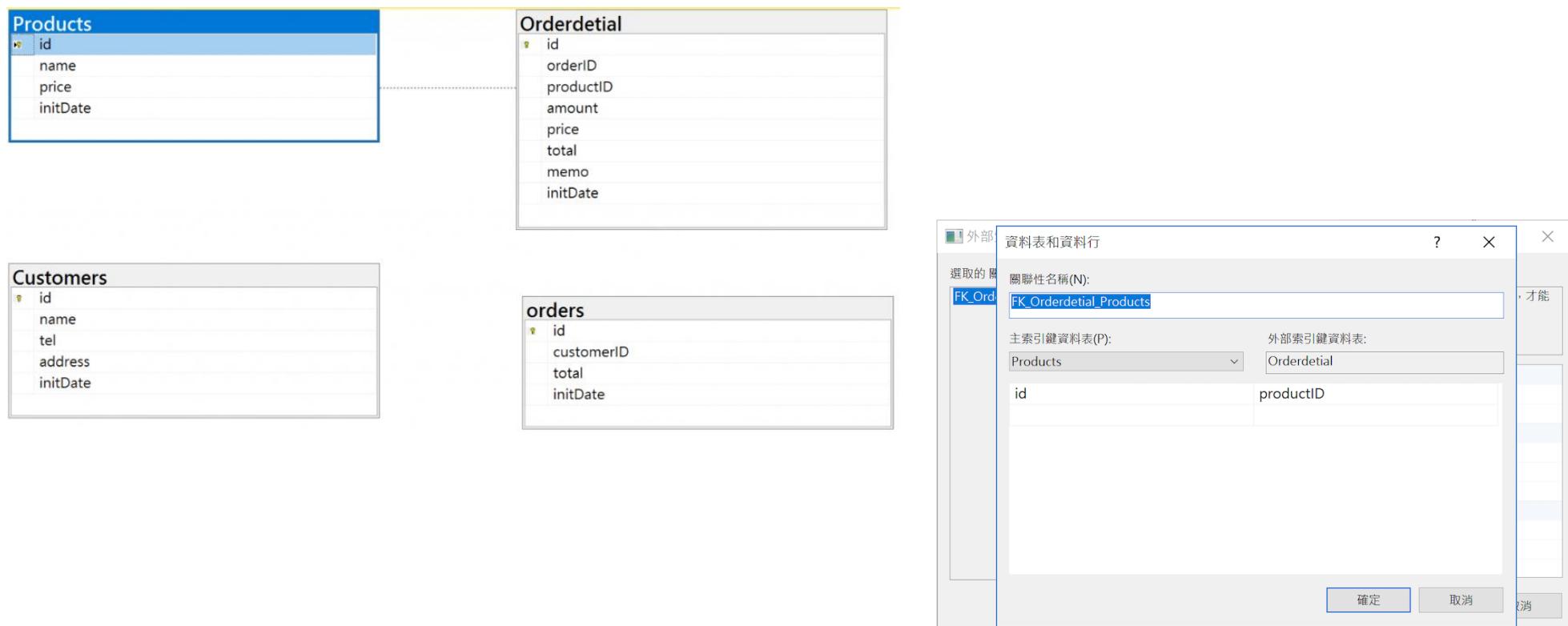
The screenshot shows the 'Add Table' dialog with the 'Tables' tab selected, displaying 'Customers', 'Orderdetial', 'orders', and 'Products'. Below the dialog are five database tables:

- Products**: id, name, price, initDate
- Orderdetial**: id, orderID, productID, amount, price, total, memo, initDate
- Customers**: id, name, tel, address, initDate
- orders**: id, customerId, total, initDate
- (Partially visible): id, name, price, initDate

建立資料庫圖表

接下來開始拉主索引鍵，建立資料表與資料表互相的關聯。

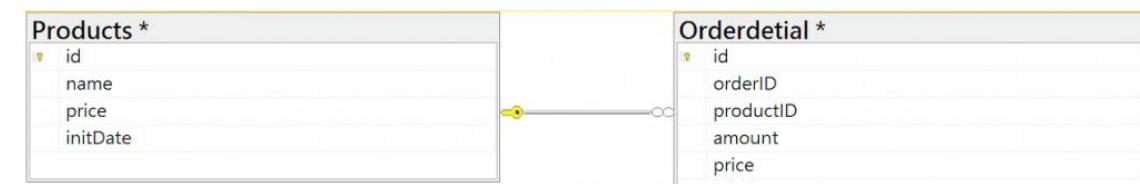
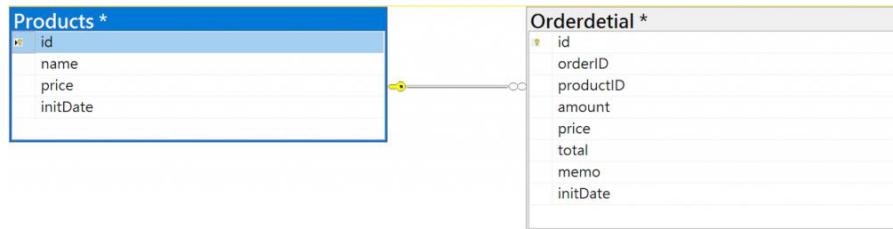
- 首現要將Products與Orderdetial建立關聯，請按著Products的id主索引鍵不放並拉向Orderdetial的productID然後放開：



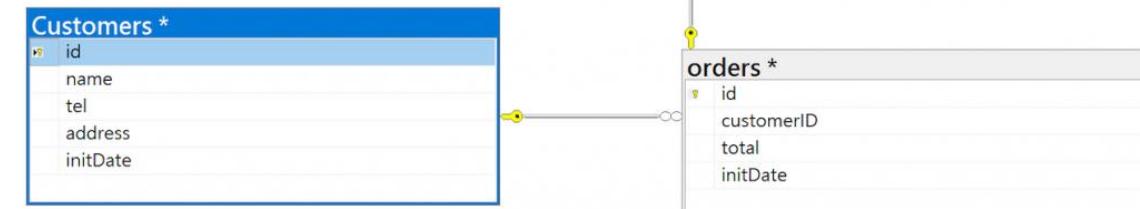
建立資料庫圖表

接下來開始拉主索引鍵，建立資料表與資料表互相的關聯。

- 按確定後，就會出現一條線聯繫著兩個資料表，而鑰匙的圖案那邊代表主索引鍵的那一端，而另一端就是外部索引鍵。



- 將剩下的關聯線也連一連：



建立資料庫圖表

接下來開始拉主索引鍵，建立資料表與資料表互相的關聯。

- 假如在建立資料表時忘記建立主索引鍵，在資料庫圖表這裡也可建立，在資料表按右鍵，選擇設定主索引鍵就可建立了。



建立資料庫圖表

接下來開始拉主索引鍵，建立資料表與資料表互相的關聯。

- 接下來將存orders及orderdetail的資料：
 - 先在orders資料表內存入資料，customerID那欄只需要輸入1即可，就可以帶表是Customers資料表的id=1內的資料，這就是有連接主索引鍵的地方！Total可以先空白沒關係，等orderdetail存入資料後再填寫。

	id	custo...	total	initD...
▶	1	1	NULL	2019-...
*	NULL	NULL	NULL	NULL

- 而有了orders的第一筆id後就可以填寫orderdetial的資料表了，這邊的orderID兩筆都可填入1，就代表是屬於orders id=1內的資料。

	id	orderID	productID	amount	price	total	memo	initD...
▶	1	1	1	2	100	200	NULL	2019-...
	2	1	2	3	200	600	NULL	2019-...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- 建立後，就可在orders的total內填入資料了，也就是orderdetial內兩筆資料的total的總和(200+600)。就是最後這筆訂單應付的總金額(800)了。

	id	customerID	total	initD...
...	1	1	800	2019-...
*	NULL	NULL	NULL	NULL

驗證使用者帳戶



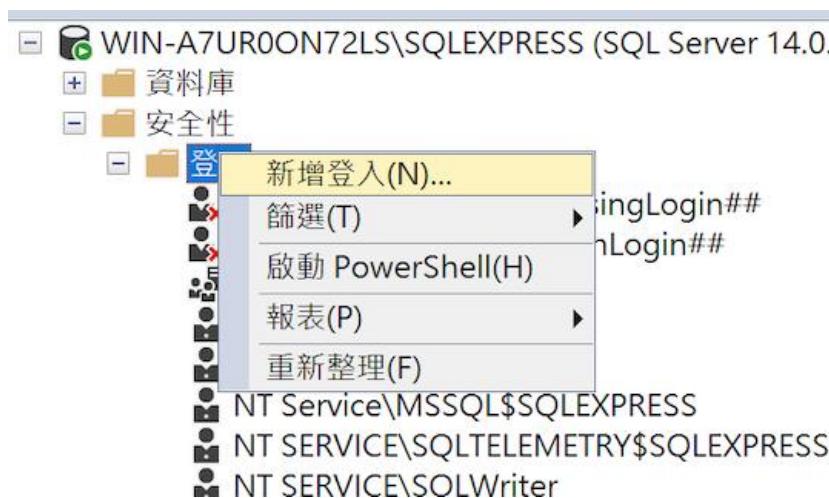
驗證使用者帳戶

SQL Server 驗證登入

- 在SQL Server中，除了Windows 驗證登入外，其中還有一種登入方式，那就是使用SQL Server驗證登入。
 - 使用SQL Server驗證來登入前需要建立帳號及密碼，之後與Visual Studio做連接只要將帳密寫在連線字串內即可。

建立帳號

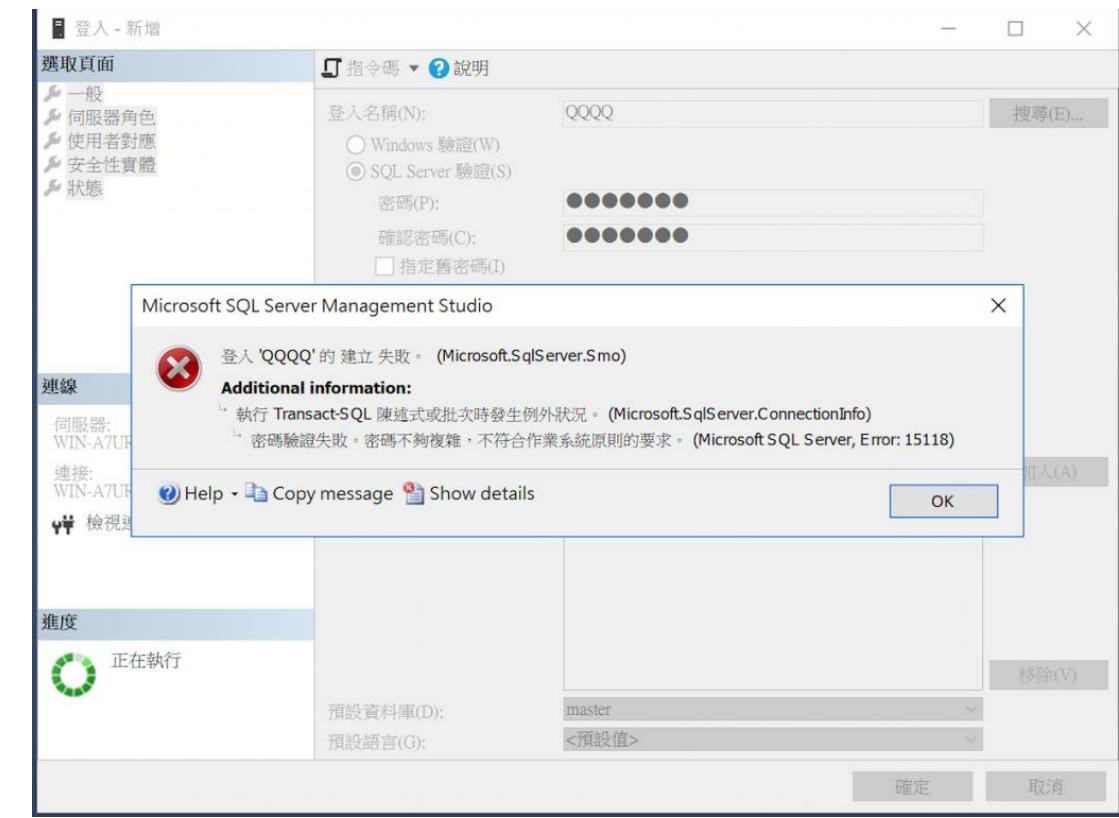
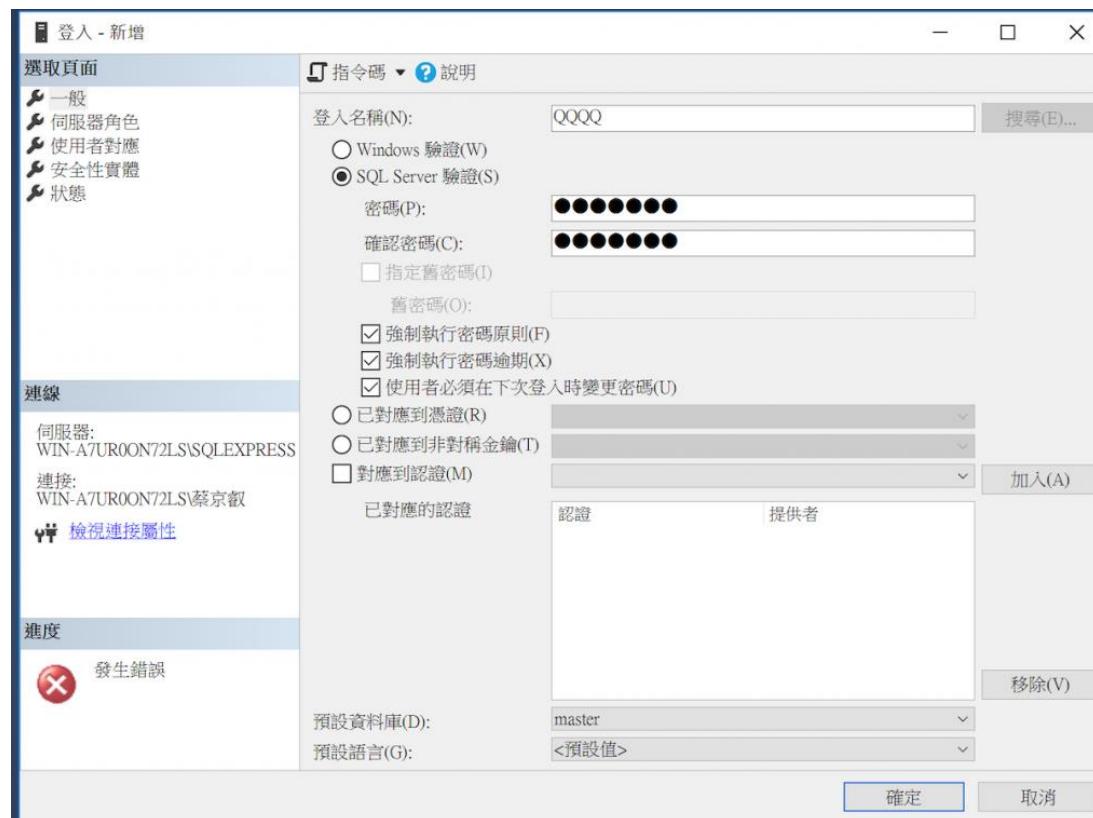
- 首先使用Windows驗證登入後，安全性->登入按右鍵選擇新增登入



驗證使用者帳戶

建立帳號

- 輸入登入名稱(帳號)及SQL Server驗證的密碼，並按確定，若顯示密碼驗證失敗，代表密碼設的不夠嚴謹



驗證使用者帳戶

建立帳號

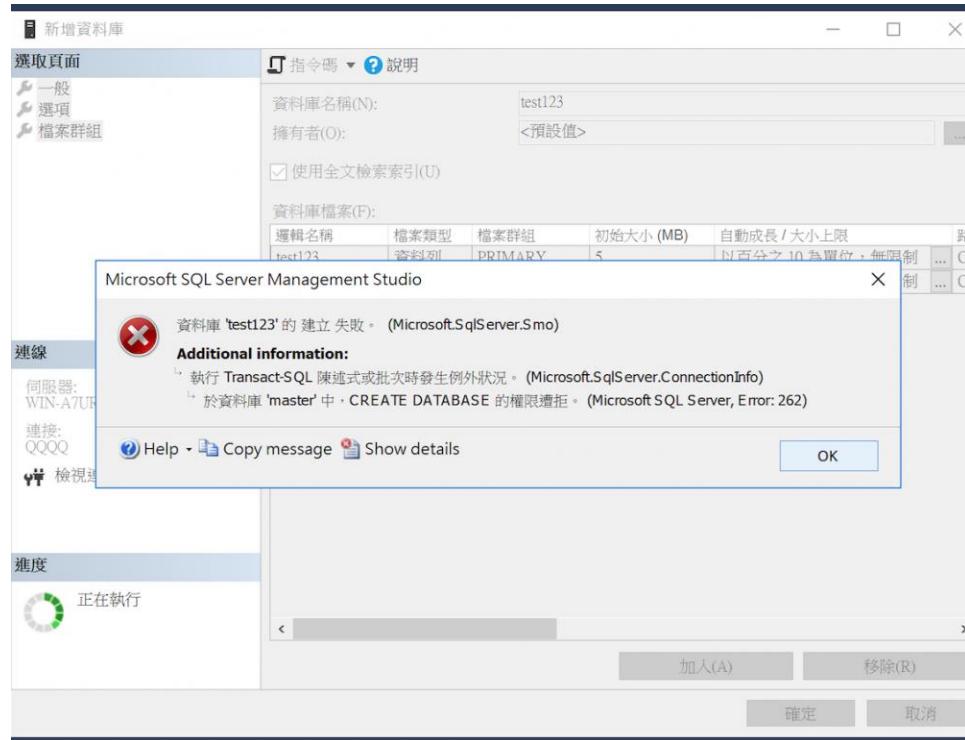
- 原因是強制執行密碼原則有被啟用，強制執行密碼原則必須符合以下規範：
 - 密碼不包含使用者的帳戶名稱。
 - 密碼長度至少為八個字元。
 - 密碼包含下列四種類別的其中三種：
 - 拉丁文大寫字母 (A 到 Z)。
 - 拉丁文小寫字母 (a 到 z)。
 - 以 10 為基底的數字 (0 到 9)。
 - 非英數字元，例如：驚嘆號 (!)、錢幣符號 (\$)、數字符號 (#) 或百分比符號 (%)。
- 密碼長度最多可達 128 個字元。盡可能使用長且複雜的密碼。所以如果不想那麼嚴謹，可以把強制執行密碼原則關掉，按確定就可以了。
- 重新登入選擇SQL Server驗證登入，打上帳密，便可看到安全性的登入有出現，代表創立帳號成功了。



驗證使用者帳戶

建立與開啟資料庫

- 不過當要開啟原本創建好的資料庫或要創建新的資料庫時，會出現建立失敗，原因是還沒有給這帳號設定權限。

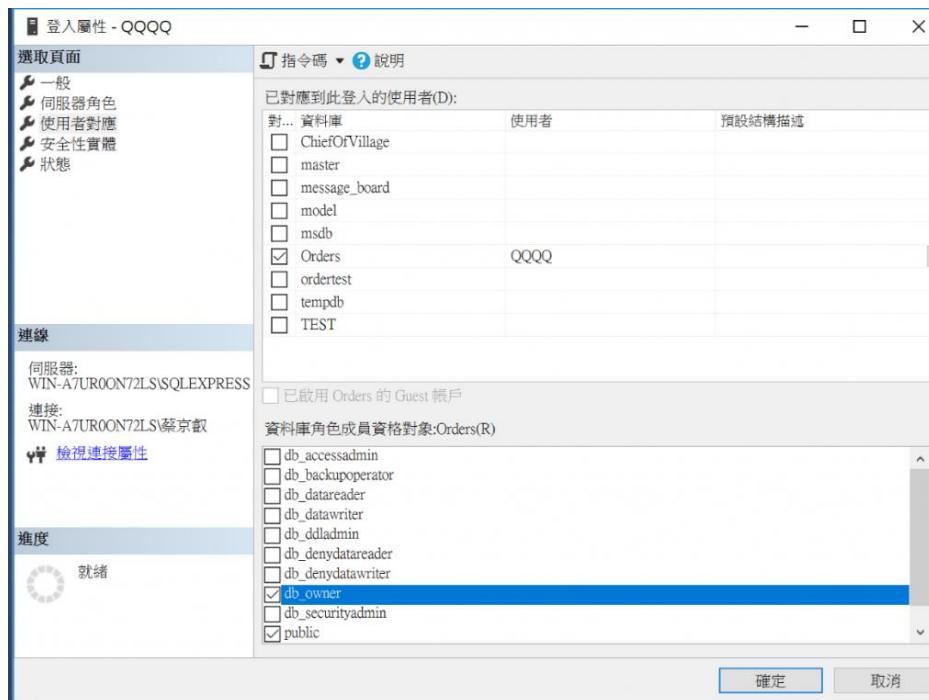


- 這時候請使用Windows驗證登入，登入後對使用者右鍵屬性，並選擇使用者對應。

驗證使用者帳戶

建立與開啟資料庫

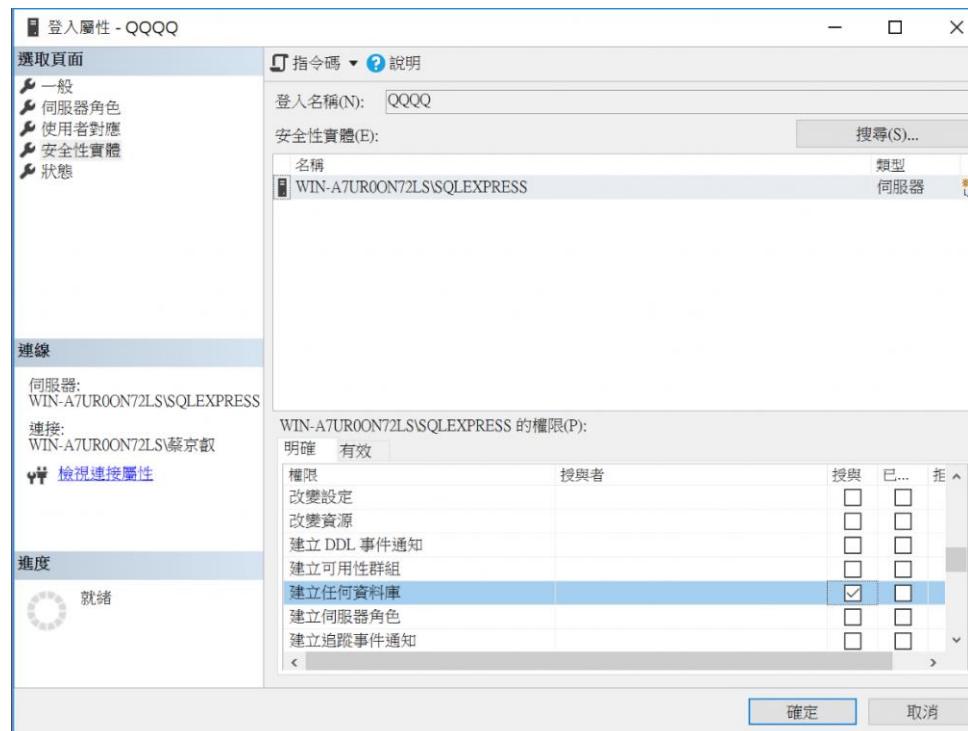
- 選擇要賦予權限的資料庫(orders) , 並將下方 db_owner打勾。
- db_owner 固定資料庫角色的成員可以在資料庫上執行所有的組態和維護活動 , 也可以在 SQL Server中卸除資料庫。 (在 SQL Database 和 SQL 資料倉儲中 , 某些維護活動需要伺服器層級的權限 , 而且無法由 db_owners執行。)



驗證使用者帳戶

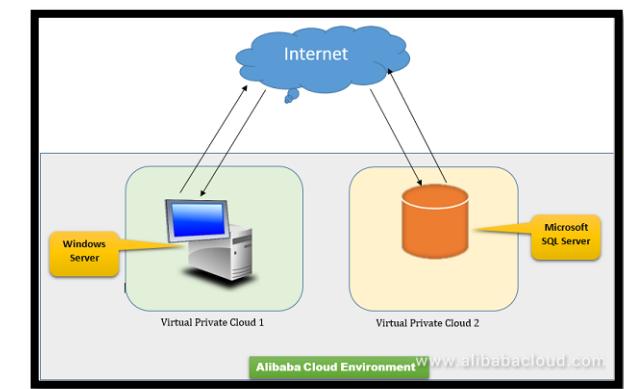
建立與開啟資料庫

- 接著選安全性實體，並允許建立資料庫，按確定後就完成了。



- 重新登入後就可以建立資料庫及開啟orders資料庫了。

ASP.NET & SQL SERVER



ASP.NET & SQL SERVER

在ASP.NET要與資料庫做連接，需要把連接字串寫進**Web.config**檔案裡，SQL Server 支援兩種驗證模式：
Windows 驗證模式和**SQL Server驗證模式**。

SQL Server驗證

- 必須建立儲存在 SQL Server中的SQL Server登入。需要登入使用者名稱及密碼，在連接字串中也需要提供。

```
<connectionStrings>
<add name="DbConnectionString" connectionString="Data Source=資料庫來源;Initial
Catalog=資料庫名稱;User ID=帳號;Password=密碼" providerName="System.Data.SqlClient" />
</connectionStrings>
```

Windows 驗證

- 使用 Windows 驗證的好處是因為使用者已登入 Windows，所以不必再另行登入 SQL Server，且不需要使用者名稱或密碼。
會使用一系列的加密訊息在 SQL Server 中驗證使用者。
- 與SQL Server驗證差在**integrated security=true;**這段。

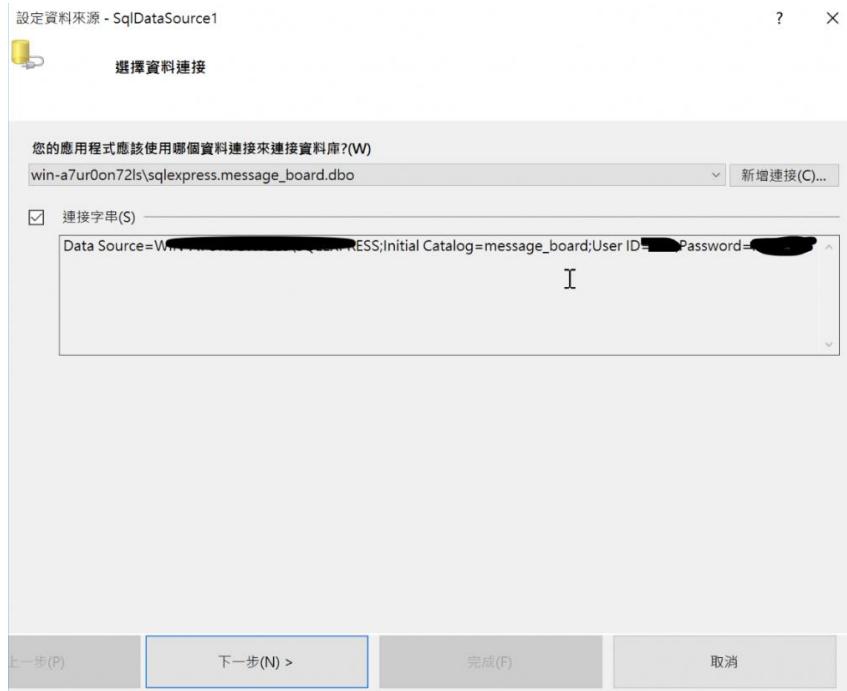
```
<connectionStrings>
<add name="DbConnectionString" connectionString="data source=資料庫來源;initial
catalog=資料庫名稱;integrated security=true;" providerName="System.Data.SqlClient" />
</connectionStrings>
```

- connectionString:**資料庫來源
- AttachDBFilename** 要連結的資料庫完整檔案路徑
- Initial Catalog:**要連接資料庫名稱
- User ID:**帳號
- Password:**密碼
- providerName:**Framework資料提供者的名稱

ASP.NET & SQL SERVER

使用SqlDataSource控制項連接

- 如果不會手動建立，也可以利用控制項的SqlDataSource產生：



- 從工具箱拉一個SqlDataSource出來後，新增連接字串，按一按伺服器名稱以及驗證方式就可以產稱連接字串了

ASP.NET & SQL SERVER

DataReader

- 一個資料流，以「順向」且「唯讀」的查看方式來讀取裡面的資料，就像個指標一樣在資料庫裡讀取資料直到連線關閉，與DataSet相比不會占用大量記憶體較省資源。
- DataReader 範例：

```
1 //創一個變數存放從config內的資訊，其實也可不用創立這變數，直接放進SqlConnection內即可。
2 string s_data = System.Web.Configuration.WebConfigurationManager.ConnectionStrings["config中connectionString的name"].ConnectionString;
3
4 //new一個SqlConnection物件，是與資料庫連結的通道(其名為Connection)，以s_data內的連接字串連接所對應的資料庫。
5 SqlConnection connection = new SqlConnection(s_data);
6
7 //new一個SqlCommand告訴這個物件準備要執行什麼SQL指令
8 SqlCommand Command = new SqlCommand($"SQL語法指令", connection);
9
10 //與資料庫連接的通道開啟
11 connection.Open();
12
13 //new一個DataReader接取Execute所回傳的資料。
14 SqlDataReader Reader = Command.ExecuteReader();
15
16 //檢查是否有資料列
17 if (Reader.HasRows){
18     //使用Read方法把資料讀進Reader，讓Reader一筆一筆順向指向資料列，並回傳是否成功。
19     if (Reader.Read()){
20         //DataReader讀出欄位內資料的方式，通常也可寫Reader[0]、[1]...[N]代表第一個欄位到N個欄位。
21         TextBox.Text = Reader["欄位名稱"].ToString();
22     }
23 }
24
25 //關閉與資料庫連接的通道
26 connection.Close();
```

ASP.NET & SQL SERVER

DataReader

1. Connection：與資料庫建立起連接的通道。
2. Command：下達SQL語法指令。要對資料庫下的SQL語法，必須是要一個物件才能執行，不能直接用SQL語法，所以才要把SQL丟進sqlcommand 這個物件
3. Open：開啟與資料庫連接的通道。
4. Execute：通知資料庫執行Command所下達的SQL語法指令，並回傳資料。可以通過這個物件來查詢結果，它提供了游水式的執行方式，即從結果中讀取一列之後，指向到下一列。
5. Read：使DataReader開始一行一行的讀取資料列。呼叫Read()方法之後，DataReader物件才會移動到資料集的第一列，同時此方法也返回一個Bool值，表明下一列是否可用，回傳True則可用，回傳False則到達資料集末尾。
6. Reader[欄位名稱 or 0~N]：DataReader讀出欄位內資料的方式，可寫Reader[0]、[1]...[N]代表第一個欄位到N個欄位，也可直接寫欄位的名稱。
7. Close：關閉與資料庫之間的連結管道。
 - if & while:
 - 使用Reader.Read()時，通常一筆資料時使用if;大於一筆資料時使用while判斷式。
 - 若讀取失敗(代表沒資料)，會傳回false 讓if或是while判斷失敗而不執行。

ASP.NET & SQL SERVER

DataSet

- DataSet一旦連接到資料來源並透過DataAdapter取得資料後，就會立刻離線，並且把獲得的資料放在主機的記憶體裡面，等待日後應用。因為是獨立出來的資料集，所以做任何操作都不會影響到真正的資料庫內的資料。

- **DataSet 範例：**

```
1 string s_data = System.Web.Configuration.WebConfigurationManager.ConnectionStrings["config中connectionString的name"].ConnectionString;
2
3 SqlConnection connection = new SqlConnection(s_data);
4
5 SqlCommand Command = new SqlCommand($"SQL語法指令", connection);
6
7 SqlDataAdapter dataAdapter = new SqlDataAdapter(Command); //從Command取得資料存入dataAdapter
8
9 DataSet dataset = new DataSet(); //創一個dataset的記憶體資料集
10
11 dataAdapter.Fill(dataset); //將dataAdapter資料存入dataset
12
13 foreach (DataRow datarow in dataset.Tables["選擇的資料表名稱"].Rows){
14     //每個資料列的內容
15 }
16
17
18 //新增一筆資料
19 DataRow newRow = dataset.Tables["資料表名稱"].NewRow();
20 newRow["id"] = 999;
21 newRow["header"] = 111;
22 dataset.Tables["資料表名稱"].Rows.Add(newRow);
23
24 //移除一筆資料
25 dataset.Tables["資料表名稱"].Rows.Remove(dataset.Tables["資料表名稱"].Rows[1]);
26
27 //修改一筆資料
28
29 DataRow[] rows = dataset.Tables["資料表名稱"].Select("id=1");
30 if (rows.Length > 0){
31     rows[0]["id"] = 2;
32 }
33
34 GridView1.DataSource = dataset; //告訴GridView1的DataSource資料來源是dataset
35 GridView1.DataBind(); //資料繫結
```

ASP.NET & SQL SERVER

DataSet

1. Connection : 建立與資料庫連接的通道。
2. Command : 下達SQL語法指令。要對資料庫下的SQL語法，必須是要一個物件才能執行，不能直接用SQL語法，所以才要把SQL丟進sqlcommand 這個物件
3. DataAdapter : 取得資料庫資料。
4. DataSet : 建立準備存放資料的記憶體資料集。
5. dataAdapter.Fill : 將取得的資料存入記憶體資料集中。
 - 就可以使用dataset來做很多種操作了，其實也可以建立DataTable來做操作，就不用dataset.Tables了

```
DataTable dataTable =new DataTable();
```

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 先建立一個里長得獎名單的資料庫，並建好資料表的欄位，要用來存資料的。
- 資料庫名稱：prizeLi、資料表名稱：Li、欄位如下：

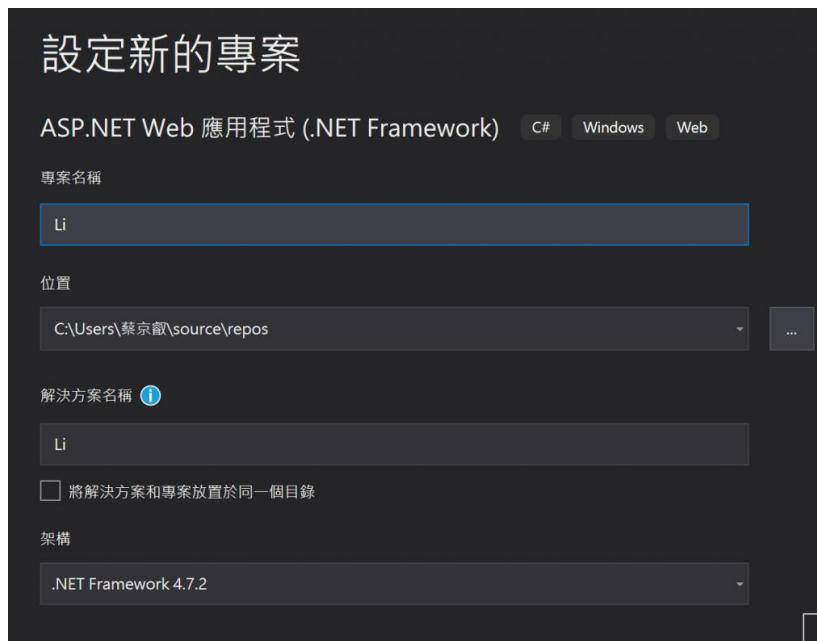
資料行名稱	資料類型	允許 Null
id	int	<input type="checkbox"/>
年度	nvarchar(50)	<input checked="" type="checkbox"/>
得獎類別	nvarchar(50)	<input checked="" type="checkbox"/>
區別	nvarchar(50)	<input checked="" type="checkbox"/>
里別	nvarchar(50)	<input checked="" type="checkbox"/>
姓名	nvarchar(50)	<input checked="" type="checkbox"/>
initDate	datetime	<input checked="" type="checkbox"/>

108,特優,鹽埕區,壽星里,黃強
108,特優,鹽埕區,沙地里,蔡鎮興
108,特優,鼓山區,內惟里,林進發
108,特優,鼓山區,裕豐里,周恆功
108,特優,鼓山區,山下里,蔡連成
108,特優,鼓山區,新民里,洪宗郁
108,特優,左營區,永清里,郭鎮國
108,特優,左營區,路東里,黃秀綢
108,特優,左營區,廊南里,陳美惠
108,特優,左營區,果峰里,曾瑞水
108,特優,楠梓區,惠楠里,曾馨慧
108,特優,楠梓區,惠豐里,張翼麟
108,特優,楠梓區,宏昌里,劉秀桃
108,特優,楠梓區,仁昌里,王永坤
108,特優,三民區,鼎盛里,謝有清
108,特優,三民區,寶獅里,林昆明
108,特優,三民區,寶華里,葉護謹
108,特優,三民區,灣勝里,陳金櫃
108,特優,三民區,正興里,沈秀玲
108,特優,三民區,同德里,許元成
108,特優,三民區,德行里,郭麗香
108,特優,三民區,博惠里,黃玉鄰
108,特優,新興區,振成里,李源登
108,特優,新興區,德生里,吳三雄
108,特優,新興區,南港里,陳福明
108,特優,前金區,三川里,莊晉勳
108,特優,前金區,文東里,黃崑光
108,特優,苓雅區,鼓中里,郭明鴻
108,特優,苓雅區,林華里,林有定
108,特優,苓雅區,林泉里,翁炯訓
108,特優,苓雅區,尚義里,曾蔡玉燕
108,特優,苓雅區,林榮里,王文祿
108,特優,苓雅區,朝陽里,丁文瑞
108,特優,苓雅區,正文里,莊文忠
108,特優,前鎮區,明正里,王春茂

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

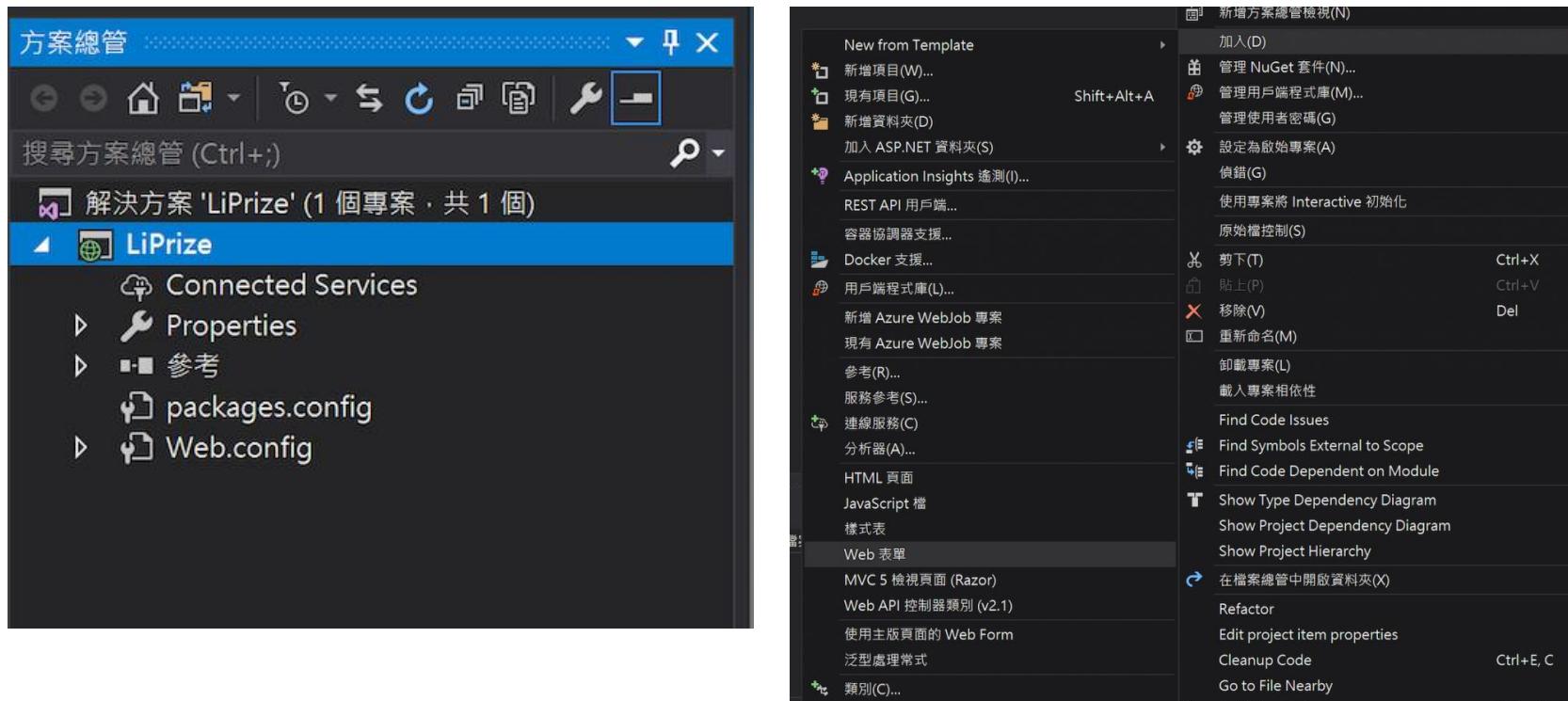
- 建好資料庫及資料表後，接下來就是開啟ASP.NET並且開啟一個Web Form專案，專案名稱就叫做LiPrize。



ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

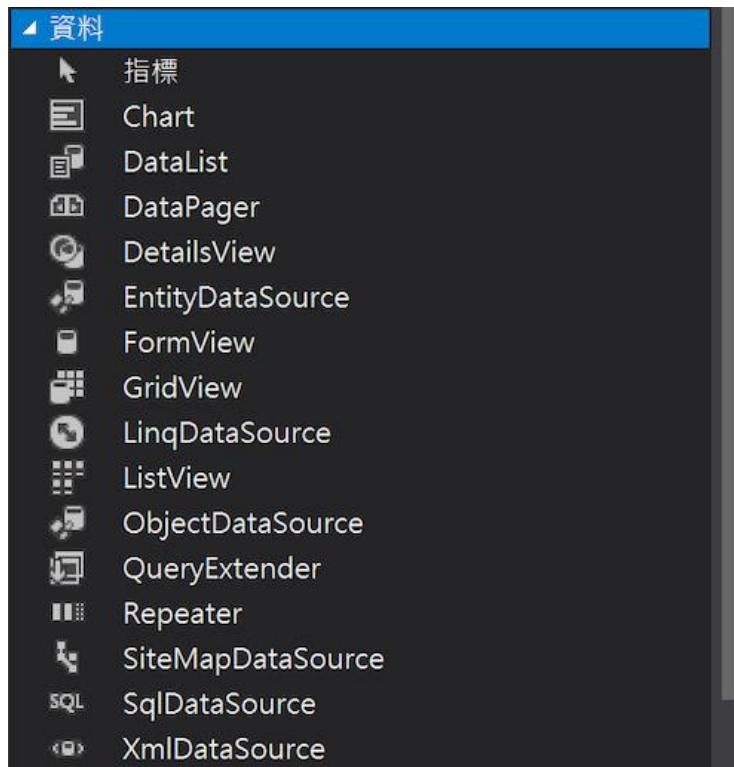
- 建立好Web Form後，看到方案總管，對方案按右鍵→加入→Web表單，名字就用預設的WebForm1
- 建立後可以按看看執行，執行結果會一個空白的頁面，到時候完成後，一進網頁就會顯示得獎名單的表格了。



ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 可以到工具箱裡的資料，直接用游標把GridView(表格)及SqlDataSource(用來與資料庫做連接用的控制項)，還有到標準拉一個Button按鈕，記得要拖曳到form內！



```
<body>
  <form id="form1" runat="server">
    <div>
    </div>
    <asp:GridView ID="GridView1" runat="server"></asp:GridView>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"></asp:SqlDataSource>
    <asp:Button ID="Button1" runat="server" Text="Button" />
  </form>
</body>
</html>
```

Column0	Column1	Column2
abc	abc	abc
SqlDataSource - SqlDataSource1		
Button		

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 現在要使用SqlDataSource1與資料庫做連接在Web.config產生連線字串。對SqlDataSource1角落箭頭按一下，設定資料來源，然後新增連接。

The screenshot shows a web form with a grid view containing four rows of data. Below the grid view is a SqlDataSource control labeled "SqlDataSource - SqlDataSource1". A tooltip "設定資料來源..." is displayed over the control. The status bar at the bottom says "設定您的資料來源的設定。"

Column0	Column1	Column2
abc	abc	abc

asp:SqlDataSource#SqlDataSource1

SqlDataSource - SqlDataSource1

設定資料來源...

設定您的資料來源的設定。



ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

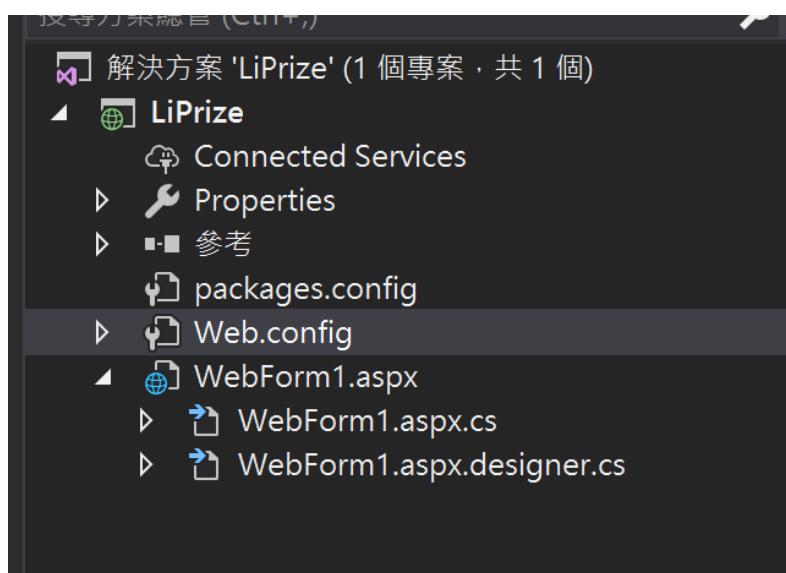
- 在加入連接頁面寫好資料庫資訊選好資料庫prizeLi的，接下來就確定勾。伺服器名稱就是SQL Server的伺服器名稱，可以從登入那裡複製。
- 然後按下一步，是否將連接字串存到應用程式主態檔(就是自動將連接字串加到Web.config內)，打勾勾後，再下一步，然後先取得所有欄位，所以直接勾「*」接下來按下一步，可以按看看測試查詢看有沒有取得資料庫，確認後按完成就可以了。



ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 然後可以在方案總管點開Web.config確認一下，connectionStrings有沒有連接字串。

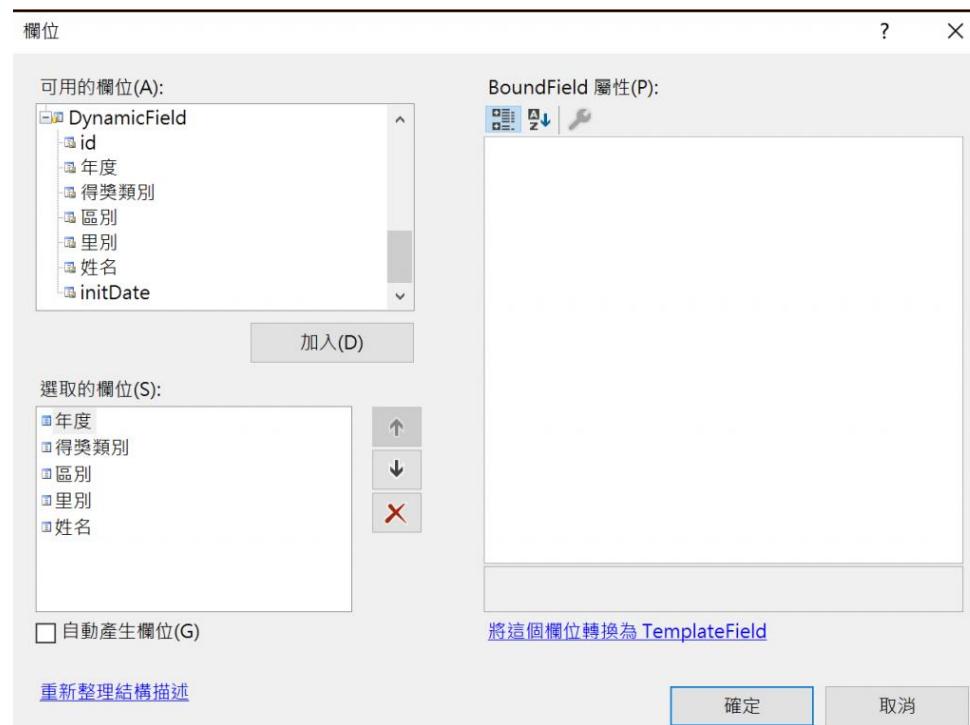


```
<connectionStrings>
    <add name="prizeLiConnectionString" connectionString="Data Source=WIN-A7UR
0ON72LS\SQLEXPRESS;Initial Catalog=prizeLi;User ID=帳號;Password=密碼"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 接下來對GridView1的右上角箭頭按一下選擇資料來源，選擇SqlDataSource1，就會看到GridView1自動加入了prizeLi的資料表的欄位了，代表已經與資料庫做連接了，接下來一樣對GridView1右上角箭頭的編輯資料行將id及initDate的資料行做取消(選到欄位案紅色X就可以刪除了)，按下確定後兩個欄位再GridView1都不見了。



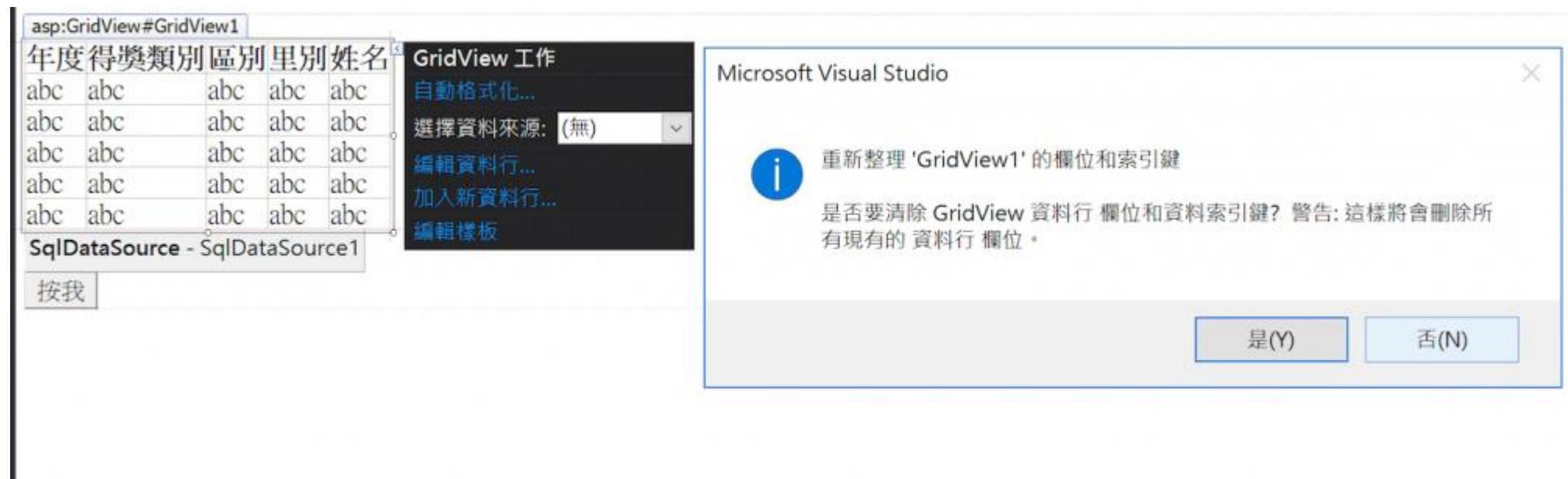
The screenshot shows a web page with a 'asp:GridView#GridView1' control. The grid displays five columns with the header '年度 得獎類別 區別 里別 姓名'. Each row contains the value 'abc' for all five columns. Below the grid, a 'SqlDataSource - SqlDataSource1' control is shown. At the bottom, there is a button labeled '按我'.

年度	得獎類別	區別	里別	姓名
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 接下來對GridView1的右上角箭頭按一下選擇資料來源，選則「無」，然後按下「否」，不要清除已經有的欄位。
- 然後就可以把SqlDataSource1控制項給刪掉了。



ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 接下來在設計模式對Button點擊兩下，會自動跳到aspx.cs的頁面，就開始設計點擊按鈕後的動作。
- 現在要寫點擊按鈕後會將csv檔的資料存入prizeLi的Li資料表內。

```
1 string file1 = System.IO.File.ReadAllText("C:\\\\儲存檔案的路徑\\\\檔名.csv").Trim(); //取得檔案位置將資料存成字串。
2 string[] f_array = file1.Split('\\n');//將換行做字串切割存到陣列內
3
4 //從config找到資料庫位置[]內放的是Web.config的connectionStrings.name。
5 string s_data = System.Web.Configuration.WebConfigurationManager.ConnectionStrings["prizeLiConnectionString"].ConnectionString;
6 SqlConnection connection=new SqlConnection(s_data); //建立與資料庫建立起連接的通道，以s_data內的連接字串連接所對應的資料庫。
7
8 string sql = "";//先創一個等下用來存SQL語法的空字串
9
10 connection.Open(); //開啟通道
11
12 foreach (var a in f_array) { //使用foreach將f_array陣列的每一個陣列都跑一遍。
13
14     string[] f_array2 = a.Split(',')//將每一次跑的陣列都再以逗號進行一次分割。
15
16     //存入插入多筆資料的SQL語法;將f_array2陣列的資料存進Li資料表欄位中。VALUES先使用@參數來取代直接給值，以防SQL Injection 程式碼。
17     sql = $"INSERT INTO [Li](年度, 得獎類別, 區別, 里別, 姓名)VALUES (@year, @prize, @Chitype, @Litype, @name)";
18
19     SqlCommand command = new SqlCommand(sql, connection); //要對SQL Server下什麼SQL指令。
20
21     command.Parameters.AddWithValue("@year", SqlDbType.NVarChar); //給@參數加入資料型態
22     command.Parameters["@year"].Value = f_array2[0]; //@參數的值是f_array2[0]
23     command.Parameters.AddWithValue("@prize", SqlDbType.NVarChar);
24     command.Parameters["@prize"].Value = f_array2[1];
25     command.Parameters.AddWithValue("@Chitype", SqlDbType.NVarChar);
26     command.Parameters["@Chitype"].Value = f_array2[2];
27     command.Parameters.AddWithValue("@Litype", SqlDbType.NVarChar);
28     command.Parameters["@Litype"].Value = f_array2[3];
29     command.Parameters.AddWithValue("@name", SqlDbType.NVarChar);
30     command.Parameters["@name"].Value = f_array2[4];
31
32     command.ExecuteNonQuery(); //執行command的SQL語法，回傳受影響的資料數目。
33
34 }
```

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

1. File: 開啟單一檔案的靜態方法。
2. ReadAllText: 開啟文字檔，讀取所有行，並關閉；()裡面放檔案路徑。
3. split字串切割:()內放的是要以什麼字串也可以放脫逸字元來做切割。可看這篇最常用的陣列!處理字串的一百種方法~(組合、切割、取代、擷取、尋找、修改)。
4. ConnectionStrings: 取得連接字串config內的資訊。
5. Connection: 與資料庫建立起連接的通道。
6. Command: 下達SQL語法指令。要對資料庫下的SQL語法，必須是要一個物件才能執行，不能直接用SQL語法，所以才要把SQL丟進sqlcommand 這個物件
7. Open: 開啟與資料庫連接的通道。
8. ExecuteNonQuery: 通知資料庫執行Command所下達的SQL語法指令，並回傳受影響的資料數目。
9. foreach: 通常配合陣列使用，指的是執行陣列的每一個物件，陣列有10個就跑10次，執行次數=陣列長度。
10. INSERT INTO: SQL新增資料的語法。
11. Parameters: 為了防止SQL Injection 程式碼攻擊，所以寫入SQL時，要使用參數「@參數名稱」來代替直接給的值，而給予參數的資料型態要使用Add。

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 讀取文字資料還有另一種方法StreamReader

```
1 String content;
2
3 StreamReader sr = new StreamReader("C:\\\\Users\\\\alvin\\\\source\\\\repos\\\\sqlserver1\\\\sqlserver1\\\\108.csv"); //使用StreamReader讀取文字檔
4
5 content = sr.ReadLine(); //對StreamReader做ReadLine()一行一行讀取，並存入字串變數內。
6
7 //從config找到資料庫位置[]內放的是Web.config的connectionStrings的name。
8 string s_data = System.Web.Configuration.WebConfigurationManager.ConnectionStrings["prizeLiConnectionString"].ConnectionString;
9
10 SqlConnection connection = new SqlConnection(s_data); //建立與資料庫建立起連接的通道，以s_data內的連接字串連接所對應的資料庫。
11
12 connection.Open(); //開啟通道
13
14 while (content != null) { //ReadLine()會一行一行讀取，直到讀到null時跳出迴圈
15     content = sr.ReadLine(); //StreamReader讀下一行
16     if (!string.IsNullOrEmpty(content)) { //當字串不是空值或是Null時
17         string[] f_array2 = content.Split(','); //將陣列字串分割
18         //存入插入多筆資料的SQL語法：將f_array2陣列的資料存進Li資料表欄位中。
19         sql = $"INSERT INTO [Li](年度, 得獎類別, 區別, 里別, 姓名)VALUES (@year,@prize,@Chitype, @Litype, @name)";
20         SqlCommand command = new SqlCommand(sql, connection); //要對SQL Server下什麼SQL指令。
21
22         command.Parameters.Add("@year", SqlDbType.NVarChar);
23         command.Parameters["@year"].Value = f_array2[0];
24         command.Parameters.Add("@prize", SqlDbType.NVarChar);
25         command.Parameters["@prize"].Value = f_array2[1];
26         command.Parameters.Add("@Chitype", SqlDbType.NVarChar);
27         command.Parameters["@Chitype"].Value = f_array2[2];
28         command.Parameters.Add("@Litype", SqlDbType.NVarChar);
29         command.Parameters["@Litype"].Value = f_array2[3];
30         command.Parameters.Add("@name", SqlDbType.NVarChar);
31         command.Parameters["@name"].Value = f_array2[4];
32     }
33     command.ExecuteNonQuery(); //執行command的SQL語法，回傳受影響的資料數目。
34 }
35
36 }
37
38 connection.Close(); //關閉與資料庫連線
39
40 sr.Close(); //關閉StreamReader
41
42 view(); //執行副程式
```

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 建立一個顯示整個表格GridView1的副程式

```
1 private void veiw() {//副程式名稱為:veiw()
2     //從config找到資料庫位置[]內放的是Web.config的connectionStrings的name。
3     string s_data = System.Web.Configuration.WebConfigurationManager.ConnectionStrings["prizeLiConnectionString"].ConnectionString;
4
5     SqlConnection connection = new SqlConnection(s_data); //建立與資料庫建立起連接的通道
6
7     SqlCommand command = new SqlCommand(@"SELECT id,年度,得獎類別,區別,里別,姓名 FROM [Li]", connection); //要對SQL Server下什麼SQL指令。
8
9     connection.Open();
10
11    SqlDataReader goDataReader = command.ExecuteReader(); //new一個DataReader接取ExecuteReader所執行SQL語法回傳的資料。
12
13    GridView1.DataSource = goDataReader; //告訴GridView1的資料來源是從goDataReader
14    GridView1.DataBind(); //將資料來源與GridView1做繫結
15    connection.Close();
16 }
```

1. SqlDataReader：一種資料流，以「順向」且「唯讀」的查看方式來讀取裡面的資料。
2. ExecuteReader：通知資料庫執行Command所下達的SQL語法指令，並回傳資料。可以通過這個物件來查詢結果，從結果中讀取一列之後，指向到下一列。
3. DataBind()：將資料來源繫結至GridView控制項。

ASP.NET & SQL SERVER

使用ASP.NET把文字檔案的資料存進資料庫(2019年高雄得獎里長.csv)

- 執行結果

按我

- 按下「按我」就傳上資料庫並顯示了。

年度	得獎類別	區別	里別	姓名
108	特優	鹽埕區	壽星里	黃強
108	特優	鹽埕區	沙地里	蔡鎮興
108	特優	鼓山區	內惟里	林進發
108	特優	鼓山區	裕豐里	周恆功
108	特優	鼓山區	山下里	蔡連成
108	特優	鼓山區	新民里	洪宗郁
108	特優	左營區	永清里	郭鎮國
108	特優	左營區	路東里	黃秀綱
108	特優	左營區	廈南里	陳美惠
108	特優	左營區	果峰里	曾瑞水
108	特優	楠梓區	惠楠里	曾馨慧
108	特優	楠梓區	惠豐里	張翼麟
108	特優	楠梓區	宏昌里	劉秀桃
108	特優	楠梓區	仁昌里	王永坤
108	特優	三民區	鼎盛里	謝有清
108	特優	三民區	寶獅里	林昆明
108	特優	三民區	寶華里	葉護謹
108	特優	三民區	灣勝里	陳金樞
108	特優	三民區	正興里	沈秀玲
108	特優	三民區	同德里	許元成
108	特優	三民區	德行里	郭麗香
108	特優	三民區	博惠里	黃玉鄰
108	特優	新興區	振成里	李源登
108	特優	新興區	德生里	吳三雄
108	特優	新興區	南港里	陳福明
108	特優	前金區	三川里	莊晉勲
108	特優	前金區	文東里	黃崑光
108	特優	苓雅區	鼓中里	郭明鴻
108	特優	苓雅區	林華里	林有定
108	特優	苓雅區	林泉里	翁炯訓
108	特優	苓雅區	尚義里	曾蔡玉燕
108	特優	苓雅區	林榮里	王文祿
108	特優	苓雅區	順陽里	丁文儒

WIN-A7UR0ON72LS\prizeLi - dbo.Li						
id	年度	得獎...	區別	里別	姓名	initD...
54	108	特優	林園區	溪州里	曾春枝	2019-...
55	108	特優	林園區	林家里	林明田	2019-...
56	108	特優	大寮區	中庄里	吳玉訓	2019-...
57	108	特優	大寮區	義和里	劉志陽	2019-...
58	108	特優	大樹區	模腳里	孫國憲	2019-...
59	108	特優	大社區	神農里	葉東火	2019-...
60	108	特優	仁武區	五和里	許有長	2019-...
61	108	特優	仁武區	赤山里	吳永宗	2019-...
62	108	特優	岡山區	維仁里	戴清福	2019-...
63	108	特優	岡山區	劉厝里	劉松森	2019-...
64	108	特優	岡山區	灣裡里	林仁傑	2019-...
65	108	特優	橋頭區	甲北里	李林...	2019-...
66	108	特優	橋頭區	三德里	柯明泉	2019-...
67	108	特優	燕巢區	鳳雄里	陳明發	2019-...
68	108	特優	田寮區	三和里	林正春	2019-...
69	108	特優	阿蓮區	復安里	何榮華	2019-...
70	108	特優	路竹區	竹瀧里	劉麗芳	2019-...
71	108	特優	路竹區	新達里	蔡桐滿	2019-...
72	108	特優	湖內區	劉家里	劉宏添	2019-...
73	108	特優	茄萣區	光定里	林義郎	2019-...
74	108	特優	茄萣區	嘉樂里	曾泰平	2019-...
75	108	特優	永安區	保寧里	莊守義	2019-...
76	108	特優	彌陀區	文安里	李泰山	2019-...
77	108	特優	梓官區	梓義里	陳榮富	2019-...
78	108	特優	梓官區	智蚜里	黃文讚	2019-...
79	108	特優	旗山區	中正里	郭舜明	2019-...
80	108	特優	旗山區	東平里	石正忠	2019-...
81	108	特優	美濃區	德興里	黃建雄	2019-...
82	108	特優	美濃區	獅山里	鍾明福	2019-...
83	108	特優	六龜區	新威里	邱國治	2019-...
84	108	特優	甲仙區	東安里	游新儀	2019-...
85	108	特優	杉林區	木梓里	林同德	2019-...
86	108	特優	內門區	永興里	陳順良	2019-...
87	108	特優	內門區	瑞山里	謝振城	2019-...

Q & A

