

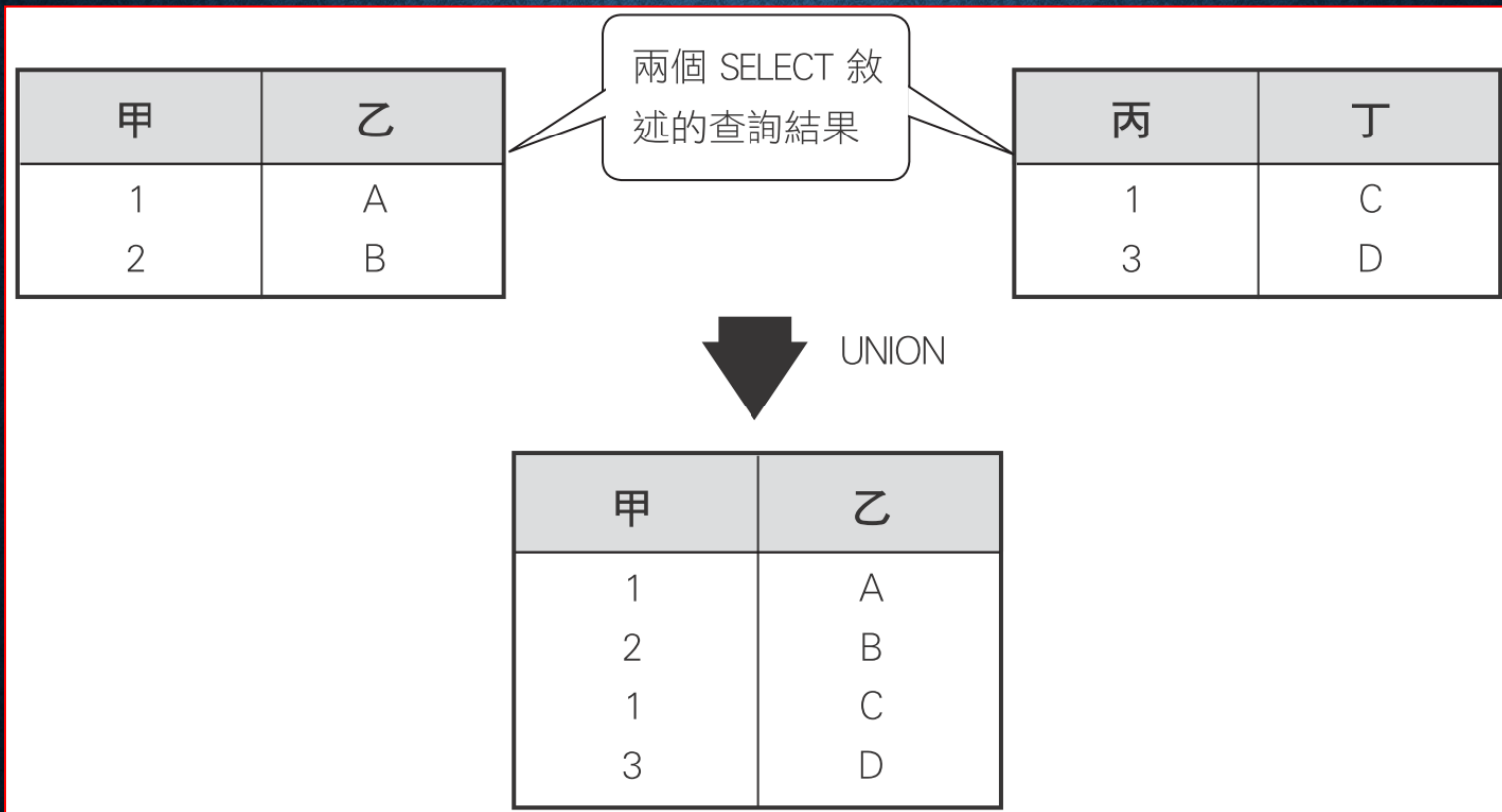
CHAP 10 更多的查詢技巧

- **10-1** 用 **UNION** 合併多個查詢結果
- **10-2** 子查詢 **Subquery**
- **10-3** 使用 **SSMS** 設計 **SQL** 查詢
- **10-4** **T-SQL** 的常數
- **10-5** 隱含式型別轉換
- **10-6** **T-SQL** 的運算子

CHAP 10 更多的查詢技巧

- 10-7 運算子的優先順序
- 10-8 處理欄位中的 **NULL** 值
- 10-9 邏輯函數：**IIF()**、**CHOOSE()**
- 10-10 排序函數：**ROW_NUMBER()**、**RANK()** 與 **DENSE_RANK()**

10-1 用 UNION 合併多個查詢結果



用 **UNION** 合併多個查詢結果

- **UNION** 的條件與結果
- **UNION** 的語法
- 應用範例

UNION 的條件與結果

- 欲合併的查詢結果，其欄位數必須相同
- 欲合併的查詢結果，其對應的欄位
 - 合併結果的欄位名稱會以第一個查詢結果的欄位名稱為名，其他查詢結果的欄位名稱則會被忽略掉。
 - 合併時，若對應的欄位具備不同的資料型別，則 **SQL Server** 會進行相容性的型別轉換，轉換的原則是以"可容納較多資料的型別為主"。

UNION 的語法

```
select_statement UNION [ALL] select_statement  
                    [ UNION [ALL] select_statement ] [...n]  
[ ORDER BY ... ]  
[ COMPUTE ... ]
```

UNION 的語法

- **select_statement**
 - **GROUP BY** 和 **HAVING** 子句只能用在個別的 **select_statement** 中，不可用於整個 **UNION** 敘述的最後。
 - **ORDER BY** 及 **COMPUTE** 子句則只能用在整個敘述的最後，針對最後的合併結果做排序或計算，不能用在個別的 **select_statement** 中。
 - 只有第一個 **select_statement** 可以設定 **INTO** 子句。
- **ALL**
- 小括弧()**:** 改變合併的優先順序。

應用範例

	編號	廠商名稱	聯絡人	性別	地址	電話
1	1	匯宏網路	陳韻琴	女	新北市永和區竹林路160號	0286312748
2	2	天天書局	方永正	男	台北市忠孝東路一段30號	0225467887
3	3	愛普動	尹育璋	男	台北市信義路109號	0226158887
4	4	千瑞百貨	白琳中	女	台北市福德路72號'	0224382456
5	5	大雄書局	孟廷亭	女	台北市南京東路三段34號	0227896457

合作廠商資料表

	客戶編號	客戶名稱	聯絡人	地址	電話
1	1	十全書局	陳小草	台北市仁愛路56號	0223219845
2	2	大發書店	陳季瑄	台北市敦化南路一段1號	0223478158
3	3	天天書局	方永正	台北市忠孝東路一段30號	0225467887
4	4	大雄書局	孟庭亭	台北市南京東路三段34號	0227896456
5	5	愚人書店	王發財	台北市重慶南路一段57號	0227465808
6	6	新新書店	黎國民	台北市中山北路六段88號	0225576635
7	7	旗竿書局	王平立	台北市師大路67號	0223468970
8	8	聰明書店	蘇小小	台北市羅斯福路四段80號	0226753499
9	9	Flags Book Store	John Potter	270 Bayside Parkway Fremont, CA 94538, USA	+1.5106877123

客戶資料表


```
SELECT 聯絡人 AS 邀請名單, 地址
FROM 合作廠商
UNION
SELECT 聯絡人, 地址
FROM 客戶
ORDER BY 聯絡人
```



	邀請名單	地址
1	John Potter	270 Bayside Parkway Fremont, CA 94538, USA
2	尹育璋	台北市信義路109號
3	方永正	台北市忠孝東路一段30號
4	王平立	台北市師大路67號
5	王發財	台北市重慶南路一段57號
6	白琳中	台北市福德路72號'
7	孟廷亭	台北市南京東路三段34號
8	孟庭亭	台北市南京東路三段34號
9	陳小草	台北市仁愛路56號
10	陳季瑄	台北市敦化南路一段1號
11	陳韻琴	新北市永和區竹林路160號
12	黎國民	台北市中山北路六段88號
13	蘇小小	台北市羅斯福路四段80號

應用範例

- 不刪除資料表合併後重複的資料

```
SELECT 聯絡人 AS 邀請名單, 地址  
FROM 合作廠商
```

```
UNION ALL
```

← 加上 ALL 參數, 則即使是重複的資料也會顯示出來

```
SELECT 聯絡人, 地址  
FROM 客戶  
ORDER BY 聯絡人
```



應用範例

	邀請名單	地址
1	John Potter	270 Bayside Parkway Fremont, CA 94538, USA
2	尹育璋	台北市信義路109號
3	方永正	台北市忠孝東路一段30號
4	方永正	台北市忠孝東路一段30號
5	王平立	台北市師大路67號
6	王發財	台北市重慶南路一段57號
7	白琳中	台北市福德路72號'
8	孟廷亭	台北市南京東路三段34號
9	孟庭亭	台北市南京東路三段34號
10	陳小草	台北市仁愛路56號
11	陳季瑄	台北市敦化南路一段1號
12	陳韻琴	新北市永和區竹林路160號
13	黎國民	台北市中山北路六段88號
14	蘇小小	台北市羅斯福路四段80號

重復的資料

應用範例

- 加入臨時的資料

```
SELECT 聯絡人 AS 邀請名單, 地址  
FROM 合作廠商
```

```
UNION
```

```
SELECT 聯絡人, 地址  
FROM 客戶
```

```
UNION
```

```
SELECT '王大砲' , '台北市南京東路三段 34 號 5 樓'
```

```
ORDER BY 聯絡人
```

運用 UNION 加入
一筆自訂資料

10-2 子查詢 SUBQUERY

- 子查詢的語法與範例
- 子查詢的類型與處理方式
- 獨立子查詢與關聯子查詢

子查詢的語法與範例

- 整個子查詢敘述需用小括弧 () 括住。
- 子查詢中不能使用 **INTO** 子句。
- 若子查詢中有用到 "**SELECT TOP n ...**"，才可設定 **ORDER BY** 子句來排序。

```
SELECT 訂單編號, 下單日期,  
       總數量 = (SELECT SUM(數量) ← "總數量 = XXX" 相當於 "XXX AS  
                 FROM 訂購項目      總數量", 都可用來定義欄位別名  
                 WHERE 訂單編號 = 訂單.訂單編號)  
FROM 訂單
```

子查詢的類型與處理方式

- **方法 1**：直接取值－直接使用子查詢的傳回值，例如用 =、>、< 做比較，或進行加減乘除等運算。
- **方法 2**：比對清單－使用 **IN**、**ALL**、或 **ANY (SOME)** 運算子判斷某個值是否存在於傳回清單中，其比對結果為 **True** 或 **False**。
- **方法 3**：測試存在－使用 **EXISTS** 運算子判斷是否有傳回資料，其測試結果亦為 **True** 或 **False**。

子查詢的類型

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

子查詢的類型

- 直接取值的子查詢

```
SELECT 產品名稱,  
       百分比 = 價格 * 100 / ( SELECT SUM(價格) FROM 標標公司 )  
FROM 標標公司
```



	產品名稱	百分比
1	Windows 使用手冊	30.0751
2	Linux 架站實務	36.8421
3	SQL 指令寶典	33.0827

```
SELECT 產品名稱, 價格  
FROM 旗旗公司  
WHERE 價格 > ( SELECT MAX(價格)  
                FROM 標標公司 )
```



	產品名稱	價格
1	Linux 架站實務	500.00

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

子查詢的類型與處理方式

- 比對清單的子查詢
 - **IN**

```
SELECT *  
FROM 標標公司  
WHERE 產品名稱 IN ( SELECT 產品名稱  
                     FROM 旗旗公司 )
```



	產品名稱	價格
1	Windows 使用手冊	400.00
2	Linux 架站實務	490.00

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

子查詢的類型與處理方式

- **ALL**

```
SELECT 價格
FROM 標標公司
WHERE 價格 <= ALL ( SELECT 價格
                    FROM 旗旗公司
                    WHERE 價格 > 410)
```

← 子查詢可得到價格為 420 與 500 兩筆記錄

↓

	價格
1	400.00

← 查詢結果必須小於或等於子查詢中所有的值, 因此只列出價格為 400 的一筆記錄

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

子查詢的類型與處理方式

- **ANY、SOME**

```
SELECT 價格
FROM 標標公司
WHERE 價格 <= ANY ( SELECT 價格
                    FROM 旗旗公司
                    WHERE 價格 > 410 )
```

← 子查詢中可得到價格為 420 與 500 兩筆記錄



	價格
1	400.00
2	490.00
3	440.00

由於只要小於或等於子查詢結果的任何一個值即可, 因此 3 筆都符合

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

子查詢的類型與處理方式

- 測試存在的子查詢

```
SELECT *
FROM 標標公司
WHERE EXISTS ( SELECT *
                FROM 旗旗公司
                WHERE 產品名稱 = 標標公司.產品名稱
                AND 價格 > 495)
```

用 EXISTS 測試存在時, 子查詢的 SELECT 子句不用指定欄位名稱, 一律用 * 即可

↓

	產品名稱	價格
1	Linux 架站實務	490.00

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

相同功能的不同查詢方式

```
SELECT 標標公司.*  
FROM 標標公司 JOIN 旗旗公司  
      ON 標標公司.產品名稱 = 旗旗公司.產品名稱  
WHERE 旗旗公司.價格 > 495
```

```
SELECT *  
FROM 標標公司  
WHERE 產品名稱 IN ( SELECT 產品名稱  
                     FROM 旗旗公司  
                     WHERE 價格 > 495)
```

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

獨立子查詢與關聯子查詢

- 獨立子查詢

是指可以脫離主查詢，單獨執行的子查詢。

```
SELECT *  
FROM 標標公司  
WHERE 產品名稱 IN ( SELECT 產品名稱  
                     FROM 旗旗公司)
```

← 這個子查詢可以單獨執行

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

獨立子查詢與關聯子查詢

- 關聯子查詢

是指無法單獨存在的子查詢。

```
SELECT *  
FROM 旗旗公司  
WHERE 產品名稱 IN ( SELECT 產品名稱  
                     FROM 標標公司  
                     WHERE 旗旗公司.價格 > 標標公司.價格)
```

← 旗旗公司資料表不在子查詢的 FROM 子句中, 所以此查詢不可單獨執行

產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	490
SQL 指令寶典	440

標標公司

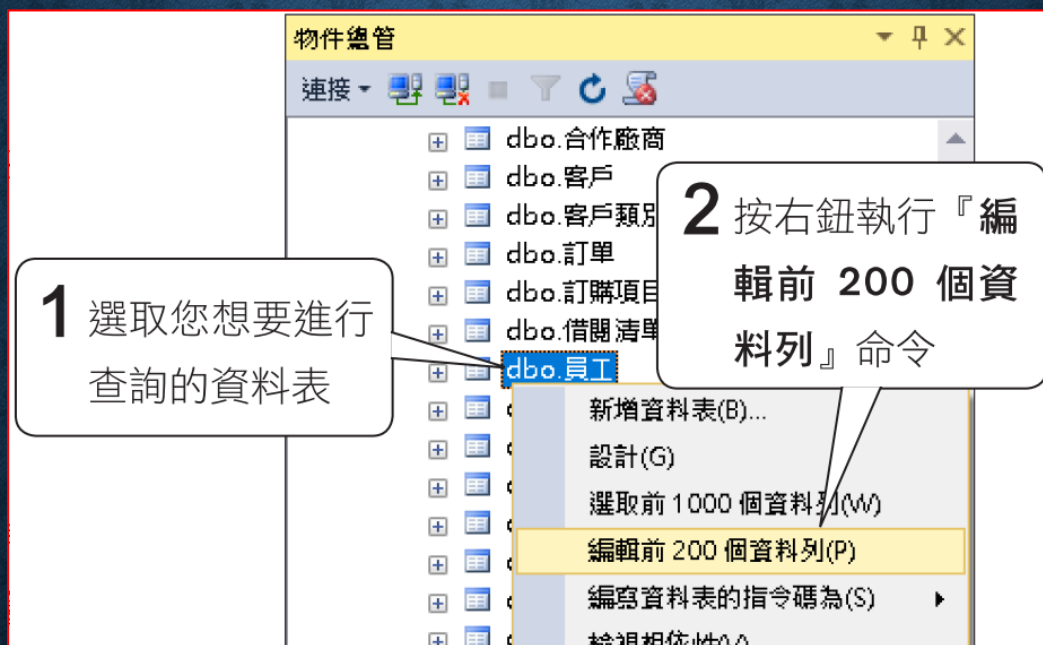
產品名稱	價格
Windows 使用手冊	400
Linux 架站實務	500
JAVA 程式語言	420

旗旗公司

10-3 使用 SSMS 設計 SQL 查詢

- 查詢設計工具
- 設計查詢的技巧 - 各窗格的操作
- 設定群組欄位
- 整個查詢敘述的屬性設定

查詢設計工具



查詢設計工具

此即 SQL Server Management Studio 所提供的各類 "視覺化" SQL 設計工具, 稍後將會分別介紹

3 這些鈕可開啟或關閉右側圖表、準則、SQL 及結果窗格

圖表窗格

準則窗格

SQL 窗格





結果窗格

資料行	別名	資料表	輸出	排序類型	排序次序
編號		員工	<input checked="" type="checkbox"/>		
姓名		員工	<input checked="" type="checkbox"/>		
性別		員工	<input checked="" type="checkbox"/>		

```
SELECT TOP (200) 編號, 姓名, 性別, 地址, 電話, 主管編號, 職位
FROM 員工
```

編號	姓名	性別	地址	電話
1	陳國園	女	台北市仁愛路...	0223219845
2	劉敏敏	女	台北市敦化北...	0223447782
3	趙清燕	女	台北市忠孝東...	0225983290
4	李淑娟	女	台北市南京東...	0227661222

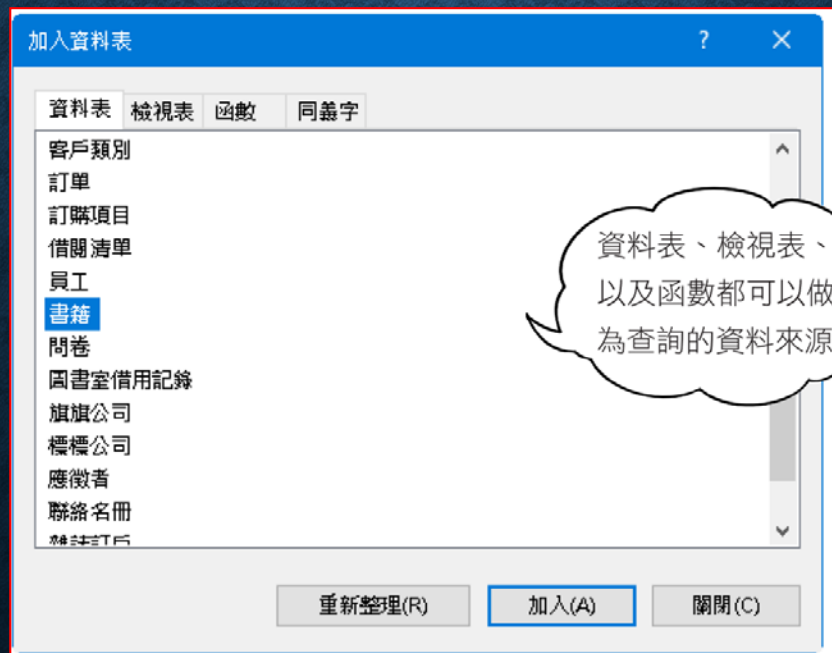
查詢設計工具

按鈕	說明
	顯示/隱藏圖表窗格
	顯示/隱藏準則窗格
	顯示/隱藏 SQL 窗格
	顯示/隱藏結果窗格

設計查詢的技巧 - 各窗格的操作

- 圖表窗格

- 載入 / 移除資料來源
- **Step 1**：按加入資料表鈕 ，或按右鍵執行『加入資料表』

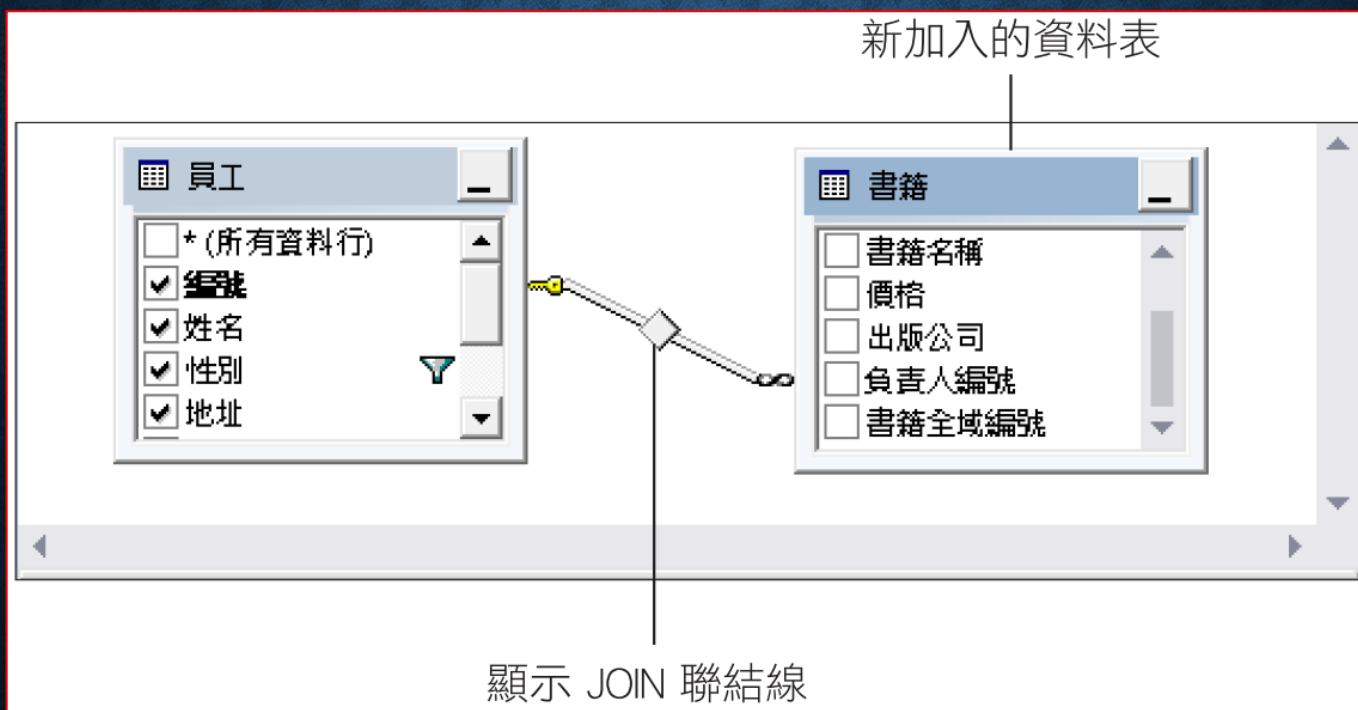


設計查詢的技巧 - 各窗格的操作

- **Step 2**
- 在列示窗中雙按您要加入的資料來源，或選取資料來源再按**加入**鈕，即可將資料來源載入**圖表窗格**中。

設計查詢的技巧 - 各窗格的操作

- Step 3



設定資料表別名



在此即可輸入
資料表的別名

設計查詢的技巧 - 各窗格的操作

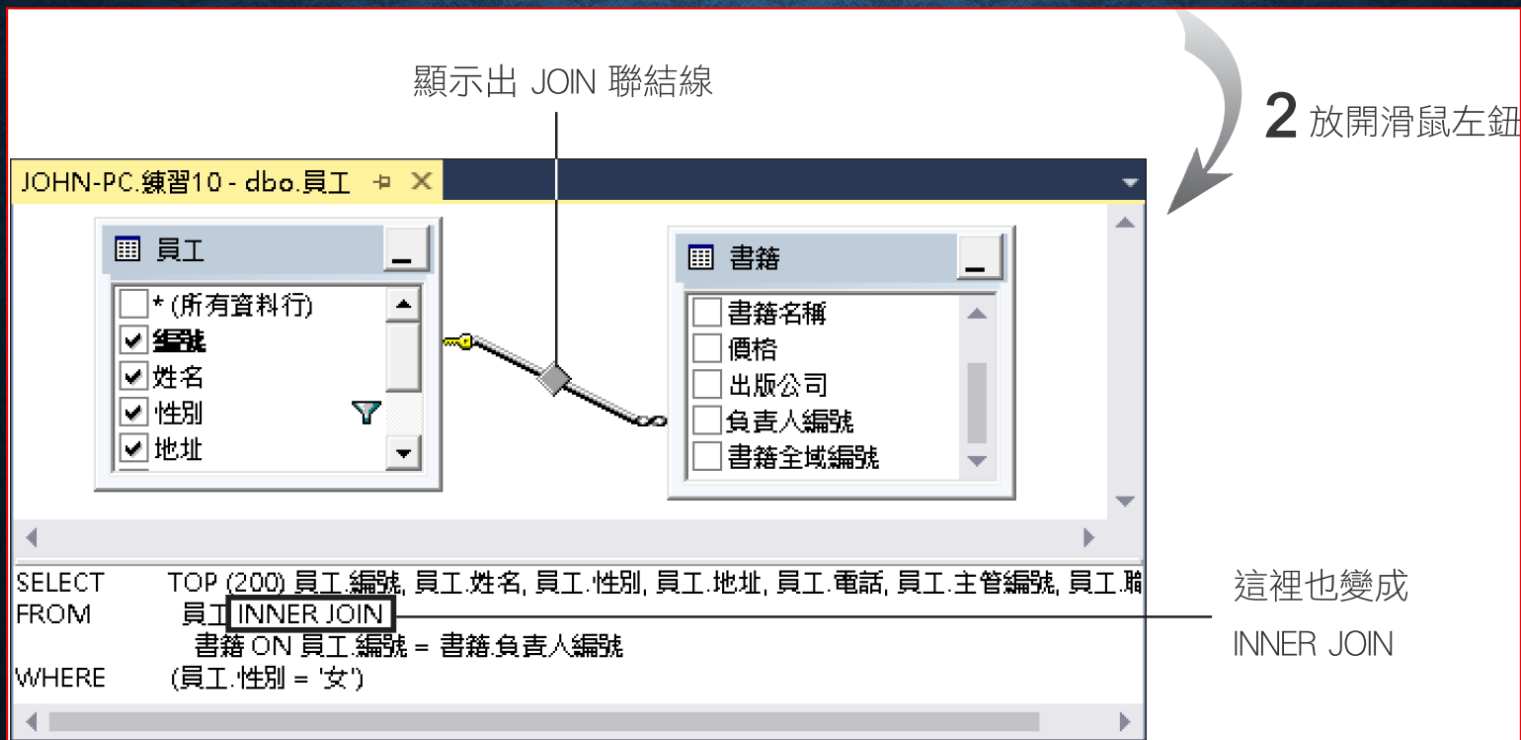
- **JOIN** 資料表

1 從員工資料表拉曳
編號欄位到書籍資
料表的負責人編號
欄位上，這個步驟
可設定 JOIN 的條件



設計查詢的技巧 - 各窗格的操作

顯示出 JOIN 連結線



2 放開滑鼠左鈕

這裡也變成 INNER JOIN

```
SELECT TOP (200) 員工.編號, 員工.姓名, 員工.性別, 員工.地址, 員工.電話, 員工.主管編號, 員工.聯  
FROM 員工 INNER JOIN  
書籍 ON 員工.編號 = 書籍.負責人編號  
WHERE (員工.性別 = '女')
```

設計查詢的技巧 - 各窗格的操作

JOHN-PC.練習10 - dbo.員工

員工

- *(所有資料行)
- 編號
- 姓名
- 性別
- 地址

書籍

- 書籍名稱
- 價格
- 出版公司

SELECT TOP (200) 員工.編號, 員工.姓名, 員工.性別
FROM 員工 INNER JOIN 書籍 ON 員工.編號 = 書籍.負責人編號
WHERE (員工.性別 = '女')

只勾選此命令為 LEFT JOIN

只勾選此命令為 RIGHT JOIN

兩者都勾選為 FULL JOIN,
兩者都不勾選則為 INNER JOIN

設計查詢的技巧 - 各窗格的操作

1 在 JOIN 聯結線上按滑鼠左鈕

The screenshot displays the SQL Server Enterprise Designer interface. On the left, the '員工' (Employee) table is selected, showing fields: 編號 (checked), 姓名 (checked), 性別 (checked), and 地址 (checked). On the right, the '書籍' (Books) table is selected, showing fields: 書籍名稱, 價格, 出版公司, 負責人編號, and 書籍全域編號. A join line connects the '員工' table to the '書籍' table. A mouse cursor is clicking on this join line. The right-hand '屬性' (Properties) pane is open, showing the '查詢設計工具' (Query Design Tools) section. The '聯結條件及型別' (Join Conditions and Types) property is selected, and its value is '內聯' (Inner). The 'WHERE' clause in the SQL statement below the grid is '(員工.性別 = '女')'. The SQL statement in the bottom pane is:

```
SELECT TOP (200) 員工.編號, 員工.姓名, 員工.性別, 員工.地址, 員工.電話, 員工.主管編號, 員.  
FROM 員工 INNER JOIN  
    書籍 ON 員工.編號 = 書籍.負責人編號  
WHERE (員工.性別 = '女')
```

2 按此鈕設定 JOIN 條件

設計查詢的技巧 - 各窗格的操作

可變更 JOIN 條件的比較方式

聯結

資料表:

員工

書籍

資料行:

=

員工編號

書籍負責人編號

包括資料列:

員工 中所有的資料列(A)

書籍 中所有的資料列(R)

確定 取消

在此可變更 JOIN 類型,
都不勾選為 INNER JOIN,
都勾選為 FULL JOIN

設計查詢的技巧 - 各窗格的操作

JOIN 圖示會隨著設定而改變

LECT TOP (200) 員工.編號, 員工.姓名, 員工.性別, 員工.地址, 員工.電話, 員工.主管編號, 員工.職位
OM 員工 RIGHT OUTER JOIN
書籍 ON 員工.編號 = 書籍.負責人編號
HERE (員工.性別 = '女')

變成 RIGHT JOIN
(RIGHT OUTER
JOIN) 了

設計查詢的技巧 - 各窗格的操作

打勾表示要顯示的欄位

排序欄位會出現排序符號

資料行	別名	資料表	輸出	排序類型	排序次序	篩選
姓名		員工	<input checked="" type="checkbox"/>			
書籍編號		書籍	<input checked="" type="checkbox"/>	遞增	1	
書籍名稱		書籍	<input checked="" type="checkbox"/>			

```
SELECT TOP (200) 員工.姓名, 書籍.書籍編號, 書籍.書籍名稱
FROM 員工 RIGHT OUTER JOIN
    書籍 ON 員工.編號 = 書籍.負責人編號
ORDER BY 書籍.書籍編號
```

準則窗格和 SQL 窗格會自動進行同步，反應出最新設定

設計查詢的技巧 - 各窗格的操作

- 準則窗格

設定欄位別名, 若您想讓輸出的資料保持原本的欄位名稱, 請刪除自動產生的欄位別名

設定排序方式

設定 WHERE 條件, 同一直行中的條件先以 AND 結合, 各行的條件再以 OR 組合

資料行	別名	資料表	輸出	排序類型	排序次序	篩選	或...	或...	或...
姓名		員工	<input checked="" type="checkbox"/>						
書籍編號		書籍	<input checked="" type="checkbox"/>	遞增	1				
書籍名稱		書籍	<input checked="" type="checkbox"/>						

拉下資料行列示窗可直接選取要顯示或要設定條件的欄位

勾選表示要顯示該欄位, 空白則表示不顯示

若設定多個排序欄位, 可在此欄設定優先順序

設計查詢的技巧 - 各窗格的操作

有設定 WHERE 條件的欄位會加上漏斗符號, 將滑鼠指在符號上還有條件提示

資料行	別名	資料表	輸出	排序類型	排序次序	篩選	或...	或...
姓名		員工	<input checked="" type="checkbox"/>					
書籍編號		書籍	<input checked="" type="checkbox"/>					
書籍名稱		書籍	<input checked="" type="checkbox"/>					
性別		員工	<input type="checkbox"/>			= '女'		
價格		書籍	<input type="checkbox"/>			> 350		

```
SELECT TOP (200) 員工.姓名, 書籍.書籍編號, 書籍.書籍名稱
FROM 員工 RIGHT OUTER JOIN
    書籍 ON 員工.編號 = 書籍.負責人編號
WHERE (員工.性別 = '女') AND 書籍.價格 > 350
```

WHERE 條件的組合方式 2 框內空白, 表示不顯示該欄位, 僅用來設定條件 1 自行輸入條件

設計查詢的技巧 - 各窗格的操作

資料行	別名	資料表	輸出	排序類型	排序次序	篩選	或...	或...
姓名		員工	<input checked="" type="checkbox"/>					
書籍編號		書籍	<input checked="" type="checkbox"/>					
書籍名稱		書籍	<input checked="" type="checkbox"/>					
▶ 性別		員工	<input type="checkbox"/>			= '女'		
價格		書籍	<input type="checkbox"/>			> 350		

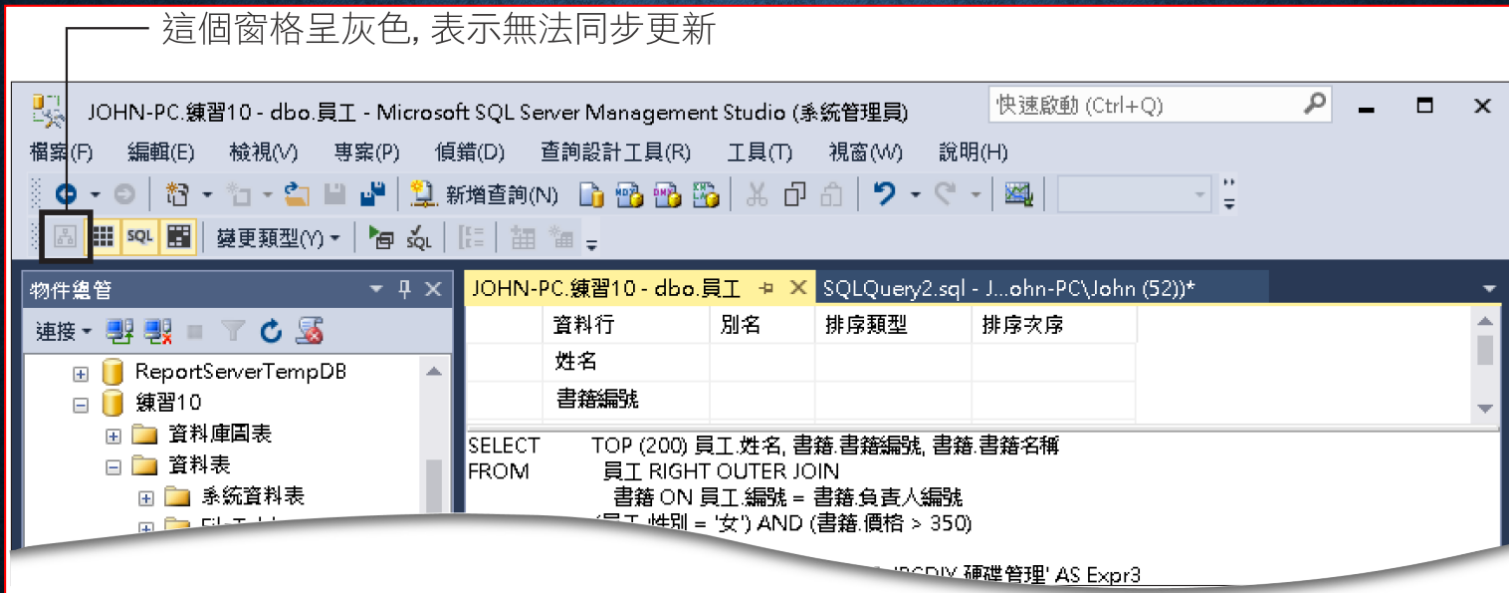
在左側灰色方塊上選取要移除的欄位 (按住

Shift 鍵即可選取多個), 然後按 **Delete** 鍵即可

設計查詢的技巧 - 各窗格的操作

- SQL 窗格

這個窗格呈灰色, 表示無法同步更新



The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The main window is titled "JOHN-PC.練習10 - dbo.員工 - Microsoft SQL Server Management Studio (系統管理員)". The SQL window is highlighted in grey, indicating it is not up-to-date. The query in the SQL window is:

```
SELECT TOP (200) 員工.姓名, 書籍.書籍編號, 書籍.書籍名稱
FROM 員工 RIGHT OUTER JOIN
    書籍 ON 員工.編號 = 書籍.負責人編號
WHERE (員工.性別 = '女') AND (書籍.價格 > 350)
```

The interface includes a menu bar (檔案(F), 編輯(E), 檢視(V), 專案(P), 偵錯(D), 查詢設計工具(R), 工具(T), 視窗(W), 說明(H)), a toolbar, and a "物件總管" (Object Explorer) on the left showing the database structure for "練習10".

設計查詢的技巧 - 各窗格的操作

- 結果窗格
- 結果窗格顧名思義就是顯示查詢結果的地方。在設計好查詢敘述後，按下執行鈕 ，查詢結果就會出現在此窗格中。

設定群組欄位

1 先設好要查詢的資料表及顯示欄位

2 按此鈕在方格窗格中顯示群組
依據欄位, 以便進行設定

物件選取

JOHN-PC.練習10 - dbo.員工 SQL 加入群組依據 JOHN-PC\John (52)*

資料行	別名	資料表	輸出	排序類型	排序次序	篩選	或...	或...	或...
姓名	Expr1	員工	<input checked="" type="checkbox"/>						
價格	Expr2	書籍	<input checked="" type="checkbox"/>						
*			<input checked="" type="checkbox"/>						

```
SELECT TOP (200) 員工.姓名 AS Expr1, 書籍.價格 AS Expr2, *  
FROM 員工 RIGHT OUTER JOIN  
書籍 ON 員工.編號 = 書籍.負責人編號
```

輸出

設定群組欄位

物件總管

JOHN-PC.練習10 - dbo.員工 SQLQuery2.sql - J...ohn-PC\John (52)*

員工

- *(所有資料)
- 編號
- 姓名
- 性別
- 地址

書籍

- *(所有資料)
- 書籍編號
- 書籍名稱
- 價格
- 出版公司

姓名欄已設為群組欄位(群組依據 = GROUP BY)

由於 * 不能用來分組, 所以自動設為 Count

資料行	別名	資料表	輸出	排序類型	排序次序	群組依據	篩選	或...	或...
姓名	Expr1	員工	<input checked="" type="checkbox"/>			群組依據			
*	Expr3		<input checked="" type="checkbox"/>			Count			
價格	Expr2	書籍	<input checked="" type="checkbox"/>						

```
SELECT TOP (200) 員工.姓名 AS Expr1, COUNT(*) AS Expr3, 書籍.價格 AS Expr2
FROM 員工 RIGHT OUTER JOIN
    書籍 ON 員工.編號 = 書籍.負責人編號
GROUP BY 員工.姓名, 書籍.價格
```

輸出

就緒

3 拉下列示窗選取要使用的彙總函數

可在彙總函數右方設定 HAVING 條件

設定群組欄位

設為群組的欄位
會加上群組符號

5 按此鈕執行查詢

執行 SQL (Ctrl+R)

使用彙總函數的欄位會加上此符號

資料行	別名	資料表	輸出	排序類型	排序次序	群組依據	篩選	或...	或...
姓名	編輯	員工	<input checked="" type="checkbox"/>			群組依據			
價格	金額	書籍	<input checked="" type="checkbox"/>			Sum			
	本數		<input checked="" type="checkbox"/>			Count			

```
SELECT TOP (200) 員工.姓名 AS 編輯, SUM(書籍.價格) AS 金額, COUNT(*) AS 本數
FROM 員工 RIGHT OUTER JOIN 書籍 ON 員工.編號 = 書籍.負責人編號
GROUP BY 員工.姓名
```

Expr1	Expr2	Expr3
孟庭詞	350.0000	1
陳國園	930.0000	2
楊咩咩	350.0000	1
趙飛燕	450.0000	1

查詢結果

輸出

就緒

整個查詢敘述的屬性設定

若有設定群組，還可在此設定群組的選項

此處可設定顯示前幾筆或前百分之幾的記錄 (參見 8-1 節)

選擇此項會在 SELECT 的欄位列表中加一 "*" (顯示全部資料表的全部欄位)

選擇此項表示不顯示重複的記錄

資料行	別名	資料表	輸出	排序類型	排序次序	群組依據
姓名	編輯	員工	<input checked="" type="checkbox"/>			群組依據
價格	金額	書籍	<input checked="" type="checkbox"/>			Sum
*	本數		<input checked="" type="checkbox"/>			Count

```
SELECT TOP (200) 員工.姓名 AS 編輯, SUM(書籍.價格) AS 金額, COUNT(*) AS 本數
FROM 員工 RIGHT OUTER JOIN 書籍 ON 員工.編號 = 書籍.負責人編號
GROUP BY 員工.姓名
```

編輯	金額	本數
孟廣輝	350.0000	1
陳國興	930.0000	2
楊祥祥	350.0000	1
...

在一般的查詢窗格中 使用查詢設計工具



在一般的 **SQLQuery** 窗格中按右鈕，執行『**在編輯器中設計查詢**』命令

在一般的查詢窗格中 使用查詢設計工具

按此鈕執行查詢

在查詢設計工具設計好的 SQL
敘述會自動輸入 **SQLQuery** 窗格

SQLQuery7.sql - J...ohn-PC\John (53))* X SQLQuery2.sql - J...ohn-PC\John (52))*

```
SELECT 員工.姓名 AS 編輯, SUM(書籍.價格) AS 金額, COUNT(*) AS 本數
FROM 員工 RIGHT OUTER JOIN
      書籍 ON 員工.編號 = 書籍.負責人編號
GROUP BY 員工.姓名
```

	編輯	金額	本數
1	孟庭訶	350.00	1
2	陳圓圓	930.00	2
3	楊咩咩	350.00	1
4	趙飛燕	450.00	1
5	劉天王	850.00	2
6	劉敏敏	960.00	2
7	蘇意涵	680.00	2

查詢結果

在一般的查詢窗格中 使用查詢設計工具

1 選取整個 SQL 敘述

The screenshot shows a SQL query window with the following text:

```
SELECT 員工.姓名 AS 編輯, SUM(書籍.價格) AS 金額, COUNT(*) AS 本數
FROM 員工 RIGHT OUTER JOIN 書籍
GROUP BY 員工.姓名
```

The context menu includes the following items:

- 剪下(T) Ctrl+X
- 複製(Y) Ctrl+C
- 貼上(P) Ctrl+V
- 插入程式碼片段(I)...
- 範圍陳述式(S)...
- 連接(C)
- 在物件總管中開啟伺服器(O) Alt+F8
- 執行(O) F5
- 顯示估計執行計畫(P) Ctrl+L
- IntelliSense 已啟用(I) Ctrl+Q, Ctrl+I
- 在 SQL Server Profiler 中追蹤查詢(T) Ctrl+Alt+P
- 在 Database Engine Tuning Advisor 中分析查詢(G)
- 在編輯器中設計查詢(D)... Ctrl+Shift+Q**
- 包括實際執行計畫(U) Ctrl+M

The results pane shows the following data:

	編輯	金額	本數
1	孟庭訶	350.00	1
2	陳圓圓	930.00	2
3	楊咩咩	350.00	1
4	趙飛燕	450.00	1
5	劉天王	850.00	2
6	劉敏敏	960.00	2
7	蘇意涵	680.00	2

Output: 已成功執行查詢。

2 按右鈕執行此命令即可
重新開啟查詢設計工具

10-4 T-SQL 的常數

- 字串常數 (**Character string constants**)

必須以單引號括起來，若字串內容本身即含有單引號時，可用連續兩個單引號來表示。

```
'Abcd'
```

```
'1234'
```

```
'SQL 伺服器 2016'
```

```
''
```

← 空字串

```
'I don''t know! '
```

← 字串中內含單引號, 即用連續兩個單引號來表示

T-SQL 的常數

- **Unicode 字串常數 (Unicode string)**

和字串常數表示法相同，但必須在字串最前面加上一個大寫的 **N**。

```
N'Abcd'
```

```
N'SQL 伺服器 2016'
```

T-SQL 的常數

- 二元碼常數 (**Binary constants**)

必須以 **0x** 開頭的 **16** 進位數值來表示，前後不必加引號。

0xAE

0x12Ef

0x69048AEFDD010E

0x

← 空的二元碼常數, 長度為 0

T-SQL 的常數

- 位元常數 (**bit constants**)

只有 0 與 1 兩種值。

- 日期時間常數 (**datetime constants**)

必須用單引號括起來的日期或時間字串。

有文字的，例如 'April15, 2016 ' 或 'April15 , 16'

分隔式的，例如 '4/15/2016'、'04-15-2016' 或 '4/15/16'、'04-15-16'

無分隔的，例如 '20161215' 或 '161215'

更改伺服器預設語言

伺服器屬性 - John-PC

選取頁面

- 一般
- 記憶體
- 處理器
- 安全性
- 連線
- 資料庫設定
- 進階
- 權限

選此項

指令碼 說明

平行處理原則的最大程度	0
查詢等候	-1
鎖定	0
其他	
已封鎖的處理序臨界值	0
允許觸發程序引發其他觸發程序	True
文字複寫大小上限	65536
全文檢索升級選項	匯入
兩位數年份的截止	2049
針對特定工作負載最佳化	False
掃描啟動程序	False
資料指標臨界值	-1
預設全文檢索語言	1028
預設語言	Traditional Chinese
網路	

可在此設定新增帳戶時的預設語言

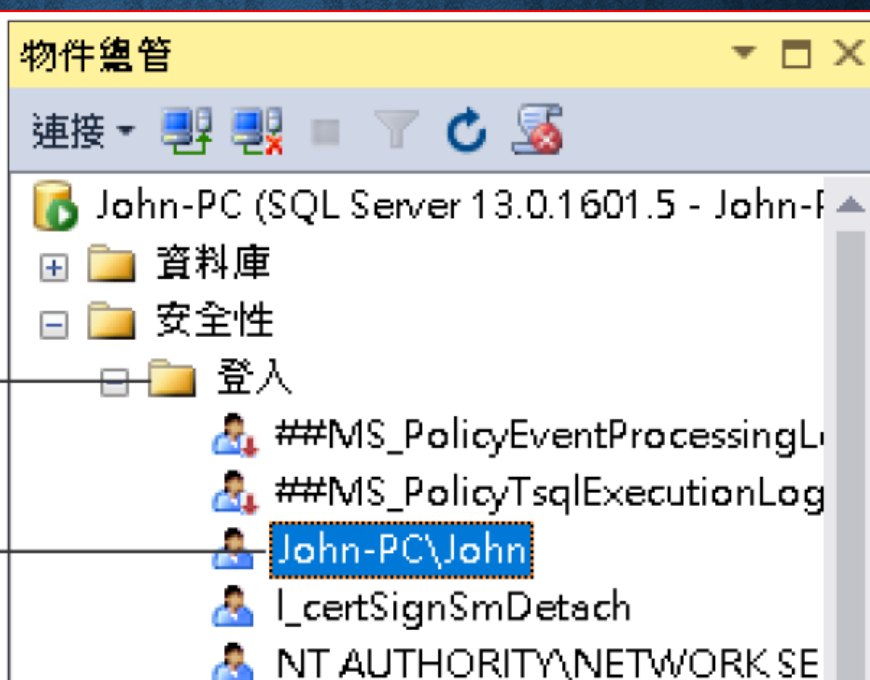
伺服器: John-PC

連接:

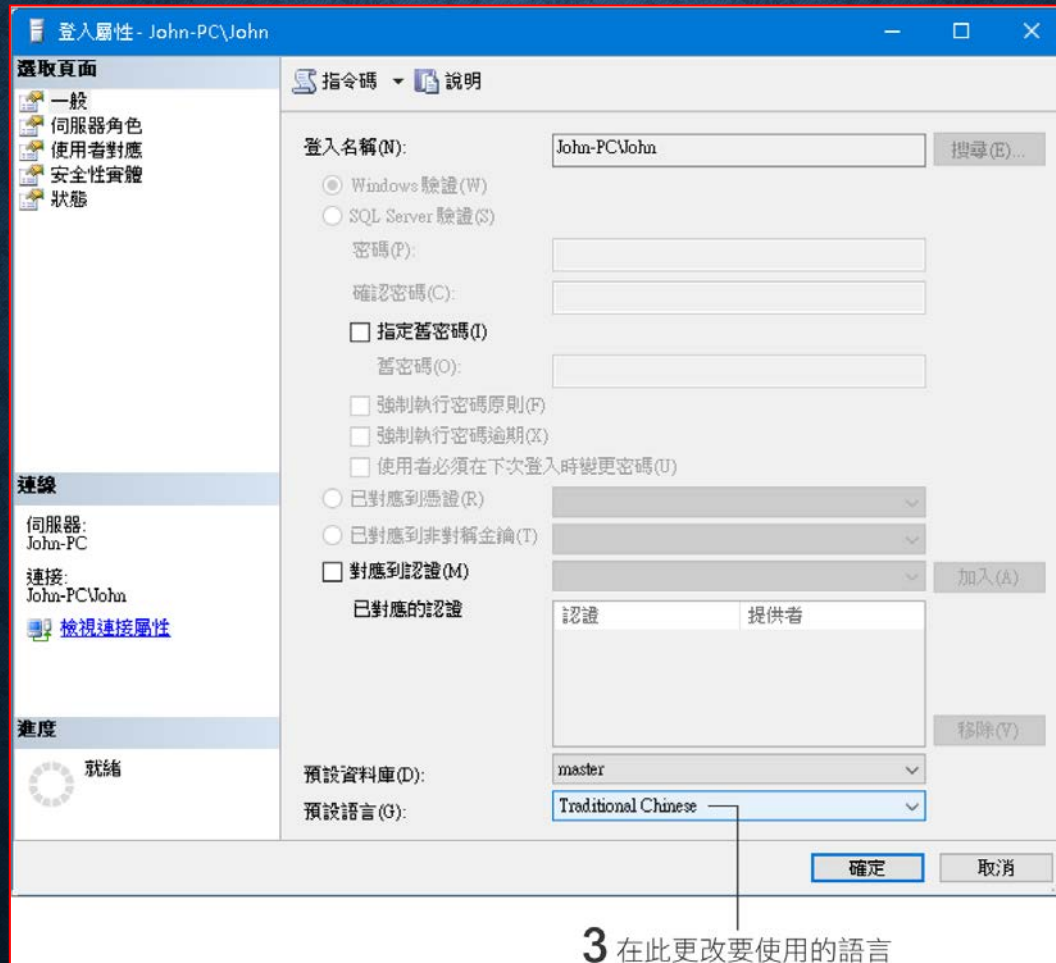
更改個人登入預設語言

1 切換到此處

2 雙按要變更的帳戶



更改個人登入預設語言



SQL SERVER 自動判斷年月日的順序

日期：

'1999/3/8'

'April 15, 1999'

'15 April, 2002'

'980415'

'04/15/08' ← 2008 年

時間：

' 14:30:24'

'04:24 PM'

'02am'

日期 + 時間：

'2013-10-10 15:30'

'5/22/133:10:14 pm'

統一的日期表示法

日期+ 時間 : { ts 'yyyy-mm-dd hh:mm:ss [.fff]' }
如 { ts '2016-12-21 10:02:20' }

日期 : { d 'yyyy-mm-dd' } 如 { d '2012-12-21' }

時間 : { t ' hh :mm: ss ' } 如 { t '10:02:20' }

```
SELECT CONVERT(DATETIME, '7/19/2016', 101)
```

統一的日期表示法

style (yy)	style (yyyy)	符合標準	格式
-	0 或 100	預設值	mon dd yyyy hh:miAM(PM)
1	101	USA	mm/dd/yy
2	102	ANSI	yy.mm.dd
3	103	British/French	dd/mm/yy
4	104	German	dd.mm.yy
5	105	Italian	dd-mm-yy
6	106	-	dd mon yy
7	107	-	mon dd, yy
8	108	-	hh:mm:ss
-	9 或 109	預設值+ 毫秒	mon dd yyyy hh:mi:ss:mmmAM(PM)
11	111	JAPAN	yy/mm/dd
10	110	USA	mm-dd-yy
12	112	ISO	yyymmdd
-	13 或 113	Europe + 毫秒	dd mon yyyy hh:mm:ss:mmm(24h)
14	114	-	hh:mi:ss:mmm(24h)

接下頁

統一的日期表示法

style (yy)	style (yyyy)	符合標準	格式
-	20 或 120	ODBC	yyyy-mm-dd hh:mi:ss(24h)
-	21 或 121	ODBC + 毫秒	yyyy-mm-dd hh:mi:ss.mmm(24h)
-	126	ISO8601	yyyy-mm-dd Thh:mm:ss:mmm
-	127	ISO8601 加時區 (Z) 標記	yyyy-mm-ddThh:mi:ss.mmmZ
-	130	Hijri	dd mon yyyy hh:mi:ss:mmmAM
-	131	Hijri	dd/mm/yy hh:mi:ss:mmmAM

T-SQL 的常數

- 整數常數 (**Integer constants**)
- 精確位數常數 (**Decimal constants**)
是指 **numeric** 或 **decimal** 型別的資料，以含有小數點的數值表示。
- 浮點常數 (**Float and Real constants**)
是指 **float** 或 **real** 型別的資料，以科學記號表示。
- 貨幣常數 (**Money constants**)
是指 **money** 或 **smallmoney** 型別的資料，以 \$ 開頭的數值表示。
- 標記常數 (**Uniqueidentifier constants**)
是 **uniqueidentifier** 型別的資料，可以用字串或二元碼常數表示。

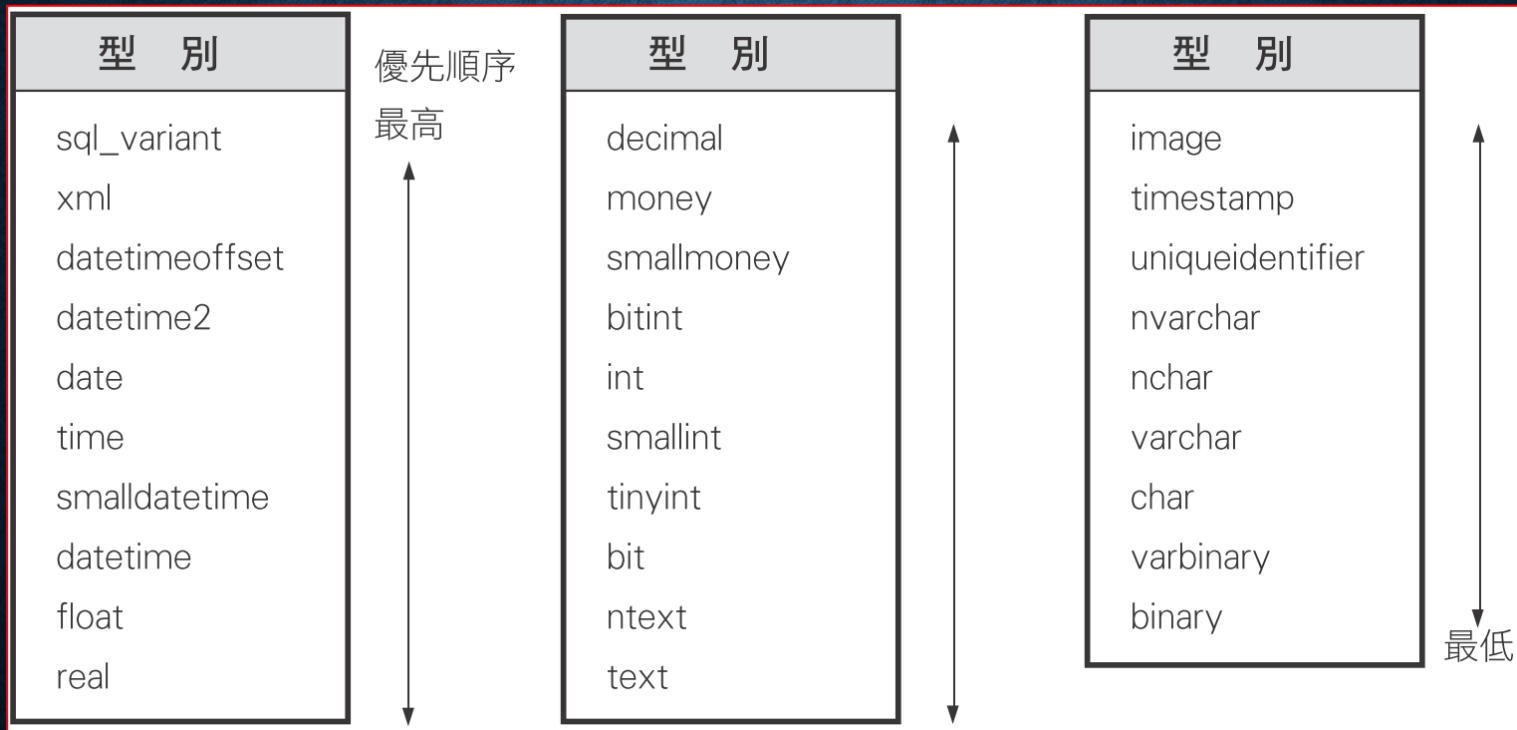
10-5 隱含式型別轉換

- 型別轉換的兩種類型
- 轉換型別的優先順序
- 型別轉換的限制

型別轉換的兩種類型

- 資料型別的轉換分成兩種類型，像剛才 **smallint** 和 **int** 的資料進行運算，**SQL Server** 會自動將 **smallint** 資料轉換成 **int** 型別，這樣的“自動型別轉換”就稱為隱含式型別轉換 (**Implicit conversion**)；而需由我們主動以 **CAST** 或 **CONVERT** 函數來轉換型別時，則稱為強迫式型別轉換 (**Explicit conversion**) 或稱明確轉換。

轉換型別的優先順序



型別轉換的限制

- 當然，並不是每種型別都可以相互轉換的，例如 **nchar** 型別是無法轉換成 **image** 的；而有些型別只能用強迫式型別轉換而不允許隱含式型別轉換，例如 **nchar** 轉換成 **binary**。

10-6 T-SQL 的運算子

- 指定運算子
- 算數運算子
- 比較運算子
- 邏輯運算子
- 位元運算子
- 字串連結運算子
- 單一運算元運算子
- 複合運算子

指定運算子

- 將數值或字串等資料指定給欄位或變數。

```
UPDATE 書籍  
SET 價格 = 400 ← 將 400 指定給價格欄位  
WHERE 書籍名稱 = 'Windows Server 系統實務'
```

```
DECLARE @number int  
DECLARE @string char(20) ]— 宣告變數
```

```
SET @number = 100 ← 將數值 100 指定給變數 @number  
SET @string = ' 天天書局' ← 將字串 '天天書局' 指定給 @string
```

```
SELECT @number AS 數字, @string AS 字串
```



	數字	字串
1	100	天天書局

算數運算子

- 括 + (加)、- (減)、* (乘)、/ (除) 與 % (整數相除的餘數)。

```
SELECT 書籍名稱, 價格 * 0.75 AS 特惠價 ← 使用 * 乘法運算  
FROM 書籍
```



	書籍名稱	特惠價
1	Windows Server 系統實務	300.000000
2	Outlook 快學快用	262.500000
3	AutoCAD 電腦繪圖與圖學	337.500000
4	Word 使用手冊	225.000000
5	抓住你的 Photoshop 中文版	337.500000
6	Linux 架站實務	375.000000
7	EXCEL 快速入門	262.500000
8	PHP 程式語言	345.000000
9	XOOPS 架站王	285.000000
10	防火牆架設實務	360.000000
11	Linux 系統管理實務	262.500000

算數運算子

```
PRINT CAST('2/20/2016' AS DATETIME) - 1    ← 向前推 1 天  
PRINT CAST('2/20/2016' AS DATETIME) + 3.25 ← 向後加 3.25 天
```



```
02 19 2016 12:00AM ← 未指定時間時, 預設為 12:00AM  
02 23 2016 6:00AM  ← 注意時間也改變了
```

比較運算子

- 比較數字的大小，或是字串的差異。

運算子	說明	運算子	說明	運算子	說明
=	等於	>	大於	<	小於
>=	大於或等於	<=	小於或等於	<>	不等於
!=	不等於	!<	不小於	!>	不大於

比較運算子

```
SELECT 書籍名稱, 價格
```

```
FROM 書籍
```

```
WHERE 價格 >= 390
```

← 將價格大於或等於 390 元的書籍列出



	書籍名稱	價格
1	Windows Server 系統實務	400.00
2	AutoCAD 電腦繪圖與圖學	450.00
3	抓住你的 Photoshop 中文版	450.00
4	Linux 架站實務	500.00
5	PHP 程式語言	460.00
6	防火牆架設實務	480.00

比較運算子

```
SELECT 書籍名稱, 價格  
FROM 書籍  
WHERE 書籍名稱 = 'AutoCAD 電腦繪圖與圖學' ← 字串的比較
```



	書籍名稱	價格
1	AutoCAD 電腦繪圖與圖學	450.00

邏輯運算子

- **AND**、**OR**：是做為兩個陳述式的邏輯判斷之用。

```
SELECT *  
FROM 旗旗公司  
WHERE (價格 > 450 AND 價格 < 500) OR 價格 < 430
```



	產品名稱	價格
1	Windows 使用手冊	400.00
2	JAVA 程式語言	420.00

邏輯運算子

- **BETWEEN**：表示在兩者之間，因此只要是在給定條件之間的資料都符合要求。

```
SELECT *  
FROM 旗旗公司  
WHERE 價格 BETWEEN 420 AND 510
```



	產品名稱	價格
1	Linux 架站實務	500.00
2	JAVA 程式語言	420.00

邏輯運算子

- **IN**：用來判斷給定的值是否在指定的項目列表或是子查詢中。

```
SELECT *  
FROM 標標公司  
WHERE 產品名稱 IN ( 'SQL 指令寶典' , 'AutoCAD 教學' , 'Linux 手冊' )
```



	產品名稱	價格
1	SQL 指令寶典	440.00

邏輯運算子

- **LIKE**：用指定的字串來找尋記錄。

```
SELECT *  
FROM 標標公司  
WHERE 產品名稱 LIKE '%SQL%'
```

萬用字元	代表意義
%	不確定有幾個字元時使用。例如 '鋼%' 可找尋以 "鋼" 開頭的所有記錄 (鋼筆)；'%鋼%' 可找尋有 "鋼" 這個字的記錄 (鋼筆、大鋼筆)。
_ (底線)	表示一個不確定的字元 (一個中文字也算一個字元)。例如：產品名稱 LIKE '_ _筆'，則可找出 '原子筆'、'螢光筆' 這些記錄。
[]	指定可以選用的字元項目或字元的範圍。例如：'{S-W}ea' 可找到 Sea、Tea；或是 '{STU}ea' 只找以 S、T、U 開頭的字。在使用中文時，一個中文即代表一個字元，例如 '{鋼鉛原子}筆' 可找到 '鋼筆' 與 '鉛筆'，但無法找到 '原子筆'，因為 '原子' 是兩個字元，而 [] 只能代表一個字元。
[^]	與 [] 的作用相反，凡是被列入其中的字元都被排除，例如 '[^S-W]ea' 會將 S 到 W 之間的字母排除，而得到 Gea、Pea 之類的記錄。

邏輯運算子

- **NOT**：可將邏輯運算元的值反向。

```
SELECT *  
FROM 標標公司  
WHERE NOT EXISTS ( SELECT *  
                    FROM 旗旗公司  
                    WHERE 產品名稱 = 標標公司.產品名稱 )
```

位元運算子

- **&**：運算子前後的兩個運算元都為 **1** 的時候，結果為 **1**。只要有一個不是 **1**，則結果為 **0**。
- **|**：運算子前後的兩個運算元只要有一個是 **1**，則結果就是 **1**，只有當兩個都是 **0** 的時候才會是 **0**。
- **^**：為互斥運算子，當兩個運算元的值不一樣的時候才會是 **1**，否則為 **0**。

位元運算子

```
PRINT 59 & 12    ← 00111011 & 00001100
PRINT 59 | 12    ← 00111011 | 00001100
PRINT 59 ^ 12    ← 00111011 ^ 00001100
  ↓
8                ← 00001000
63               ← 00111111
55               ← 00110111
```

字串連結運算子

```
SELECT 'Linux 架站實務的價格是' + CONVERT(varchar , 價格) + '元'  
FROM 標標公司  
WHERE 產品名稱 = 'Linux 架站實務'
```



	(沒有資料行名稱)
1	Linux 架站實務的價格是 490.00 元

單一運算元運算子

SELECT - 價格 ← 取負值

FROM 標標公司

WHERE 產品名稱 = 'Linux 架站實務'



-490.00

PRINT ~ CAST(1 AS tinyint) ← 補數運算 00000001 變成 11111110



254

複合運算子

運算子	使用例	等同於
+=	價格 += 100	價格 = 價格 + 100
-=	價格 -= 100	價格 = 價格 - 100
*=	價格 *= 100	價格 = 價格 * 100
/=	價格 /= 100	價格 = 價格 / 100
%=	價格 %= 100	價格 = 價格 % 100
&=	x &= 12	x = x & 12
=	x = 12	x = x 12
^=	x ^= 12	x = x ^ 12

UPDATE 書籍

SET 價格 += 100

← 等同於 『SET 價格 = 價格 + 100 』

WHERE 書籍名稱 = 'Windows Server 系統實務'

10-7 運算子的優先順序

運算子的優先順序 (由高到低)

+ (正號), - (負號), ~ (位元 NOT)

*, /, %

+ (加), + (字元串接), - (減)

^, &, | (位元運算)

+=, -=, *=, /=, %=, &=, |=, ^= (複合運算子)

=, >, <, >=, <=, <>, !=, !>, !< (比較運算子)

NOT

AND

ALL, ANY, BETWEEN, IN, LIKE, OR, SOME

= (指定運算子)

10-8 處理欄位中的 **NULL** 值

- **NULL** 值的運算
- **ISNULL()** 函數
- 檢查是否為 **NULL** 值

NULL 值的運算

```
SET ANSI_NULLS OFF ← 將 ANSI_NULLS 設定為 "OFF"  
SELECT *  
FROM 員工  
WHERE 主管編號 = NULL ← 可以用等號做比較
```



	編號	姓名	性別	地址	電話	主管編號	職位
1	2	劉敏敏	女	台北市敦化北路一段302號10樓	0223447782	NULL	經理

ISNULL() 函數

```
ISNULL ( che ck_expression, replacement_value )
```

```
SELECT 姓名,  
       ISNULL(CAST(主管編號 AS VARCHAR), ' 無') AS 主管  
FROM 員工
```



	姓名	主管
1	陳圓圓	2
2	劉敏敏	無
3	趙飛燕	2
4	孟庭訶	3
5	劉天王	1
6	楊咩咩	1
7	郭國成	3
8	蘇意涵	1

此員工的主管編號原為 NULL, 現在被 ISNULL() 函數替換為 '無'

檢查是否為 **NULL** 值

```
SELECT *
```

```
FROM 員工
```

```
WHERE 主管編號 IS NULL
```

← 找出主管編號欄是 NULL 值的記錄



	編號	姓名	性別	地址	電話	主管編號	職位
1	2	劉敏敏	女	台北市敦化北路一段302號10樓	0223447782	NULL	經理

檢查是否為 NULL 值

```
UPDATE 員工  
SET 主管編號 = 0  
WHERE 主管編號 IS NULL
```

← 將主管編號欄為 NULL 的記錄, 都改設為 0

```
SELECT *  
FROM 員工  
WHERE 主管編號 IS NOT NULL
```

← 將主管編號欄不是 NULL 的記錄都列出



	編號	姓名	性別	地址	電話	主管編號	職位
1	1	陳圓圓	女	台北市仁愛路二段56號4樓	0223219845	2	主任
2	2	劉敏敏	女	台北市敦化北路一段302號10樓	0223447782	0	經理
3	3	趙飛燕	女	台北市忠孝東路三段240號4樓	0225983290	2	主任
4	4	孟庭訶	女	台北市南京東路三段215號4樓	0227651332	3	辦事員
5	5	劉天王	男	台北市和平東路一段37號6樓	0227554980	1	辦事員
6	6	楊咩咩	男	台北市木柵路56號1樓	0223947860	1	辦事員
7	7	郭國成	男	台北市師大路87號	0223658790	3	辦事員
8	8	蘇意涵	女	台北市紹興南路89號	0223964355	1	辦事員

10-9 邏輯函數：IIF()、CHOOSE()

IIF (boolean_expression, true_value, false_value)

	問卷編號	性別	滿意度
1	1	1	3
2	2	0	3
3	3	0	2
4	4	1	1
5	5	0	2
6	6	1	2
7	7	0	2
8	8	0	1
9	9	1	3
10	10	1	2

性別用 1 代表男生、0 代表女生

滿意度用 3 代表滿意、2 代表尚可、1 代表差勁

邏輯函數：IIF()、CHOOSE()

```
SELECT IIF(性別=0, '女生', '男生') AS 性別, 滿意度, COUNT(*) AS 人數  
FROM 問卷  
GROUP BY 性別, 滿意度 ← 依性別及滿意度分組計數  
ORDER BY 性別, 滿意度
```



	性別	滿意度	人數
1	女生	1	1
2	女生	2	3
3	女生	3	1
4	男生	1	1
5	男生	2	2
6	男生	3	2

邏輯函數：IIF()、CHOOSE()

```
SELECT IIF(滿意度=3, '滿意', IIF(滿意度=2, '尚可', '差勁')) 評價, COUNT(*) 人數  
FROM 問卷  
GROUP BY 滿意度 ← 依滿意度分組計數  
ORDER BY 滿意度 DESC
```



	評價	人數
1	滿意	3
2	尚可	5
3	差勁	2

使用 **CHOOSE()** 比較簡單

```
CHOOSE( index, va 1_1, va 1_2 [ , va 1_n ] )
```

邏輯函數：IIF()、CHOOSE()

```
SELECT CHOOSE(滿意度, '差勁', '尚可', '滿意') 評價, COUNT(*) 人數  
FROM 問卷  
GROUP BY 滿意度 ← 依滿意度分組計數  
ORDER BY 滿意度 DESC
```



	評價	人數
1	滿意	3
2	尚可	5
3	差勁	2

10-10 排序函數：**ROW_NUMBER()** 、**RANK()** 與 **DENSE_RANK()**

- 針對資料表的欄位進行排序
- 指定查詢範圍
- 分組排名

針對資料表的欄位進行排序

```
SELECT 書籍編號, 書籍名稱, 價格, 出版公司,  
       ROW_NUMBER() OVER(ORDER BY 價格) AS 價格排名  
FROM 書籍
```

	書籍編號	書籍名稱	價格	出版公司	價格排名
1	4	Word 使用手冊	300.00	標標工作室	1
2	2	Outlook 快學快用	350.00	威威出版社	2
3	7	EXCEL 快速入門	350.00	立立出版社	3
4	11	Linux 系統管理實務	350.00	威威出版社	4
5	9	XOOPS 架站王	380.00	施施研究室	5
6	3	AutoCAD 電腦繪圖與圖學	450.00	施施研究室	6
7	5	抓住你的 Photoshop 中文版	450.00	立立出版社	7
8	8	PHP 程式語言	460.00	標標工作室	8
9	10	防火牆架設實務	480.00	威威出版社	9
10	6	Linux 架站實務	500.00	威威出版社	10
11	1	Windows Server 系統實務	500.00	施施研究室	11

針對資料表的欄位進行排序

```
SELECT 書籍編號, 書籍名稱, 價格, 出版公司,  
       RANK() OVER(ORDER BY 價格) AS 價格排名  
FROM 書籍
```

	書籍編號	書籍名稱	價格	出版公司	價格排名
1	4	Word 使用手冊	300.00	標標工作室	1
2	2	Outlook 快學快用	350.00	威威出版社	2
3	7	EXCEL 快速入門	350.00	立立出版社	2
4	11	Linux 系統管理實務	350.00	威威出版社	2
5	9	XOOPS 架站王	380.00	施施研究室	5
6	3	AutoCAD 電腦繪圖與圖學	450.00	施施研究室	6
7	5	抓住你的 Photoshop 中文版	450.00	立立出版社	6
8	8	PHP 程式語言	460.00	標標工作室	8
9	10	防火牆架設實務	480.00	威威出版社	9
10	6	Linux 架站實務	500.00	威威出版社	10
11	1	Windows Server 系統實務	500.00	施施研究室	10

因有三個第 2
名, 故此處由
第 5 名開始

針對資料表的欄位進行排序

```
SELECT 書籍編號, 書籍名稱, 價格, 出版公司,  
       DENSE_RANK() OVER(ORDER BY 價格) AS 價格排名  
FROM 書籍
```



	書籍編號	書籍名稱	價格	出版公司	價格排名
1	4	Word 使用手冊	300.00	標標工作室	1
2	2	Outlook 快學快用	350.00	威威出版社	2
3	7	EXCEL 快速入門	350.00	立立出版社	2
4	11	Linux 系統管理實務	350.00	威威出版社	2
5	9	XOOPS 架站王	380.00	施施研究室	3
6	3	AutoCAD 電腦繪圖與圖學	450.00	施施研究室	4
7	5	抓住你的 Photoshop 中文版	450.00	立立出版社	4
8	8	PHP 程式語言	460.00	標標工作室	5
9	10	防火牆架設實務	480.00	威威出版社	6
10	6	Linux 架站實務	500.00	威威出版社	7
11	1	Windows Server 系統實務	500.00	施施研究室	7

雖然有三個第 2 名,
但是下一個排名仍
然由第 3 名開始

指定查詢範圍

```
SELECT 書籍編號, 書籍名稱, 價格, 出版公司,  
ROW_NUMBER() OVER(ORDER BY 價格) AS 價格排名  
FROM 書籍  
ORDER BY 價格排名  
OFFSET 4 ROWS FETCH NEXT 4 ROWS ONLY
```

← 使用 ORDER BY...OFFSET...
指定傳回第 5~8 筆記錄



	書籍編號	書籍名稱	價格	出版公司	價格排名
1	9	XOOPS 架站王	380.00	施施研究室	5
2	3	AutoCAD 電腦繪圖與圖學	450.00	施施研究室	6
3	5	抓住你的 Photoshop 中文版	450.00	立立出版社	7
4	8	PHP 程式語言	460.00	標標工作室	8

分組排名

排序函數() OVER (PARTITION BY 分組欄位 ORDER BY 排名欄位) AS 別名

```
SELECT 書籍編號, 書籍名稱, 價格, 出版公司,  
ROW_NUMBER() OVER(PARTITION BY 出版公司 ORDER BY 價格) AS 價格排名  
FROM 書籍
```



	書籍編號	書籍名稱	價格	出版公司	價格排名
1	7	EXCEL 快速入門	350.00	立立出版社	1
2	5	抓住你的 Photoshop 中文版	450.00	立立出版社	2
3	2	Outlook 快學快用	350.00	威威出版社	1
4	11	Linux 系統管理實務	350.00	威威出版社	2
5	10	防火牆架設實務	480.00	威威出版社	3
6	6	Linux 架站實務	500.00	威威出版社	4
7	9	XOOPS 架站王	380.00	施施研究室	1
8	3	AutoCAD 電腦繪圖與圖學	450.00	施施研究室	2
9	1	Windows Server 系統實務	500.00	施施研究室	3
10	4	Word 使用手冊	300.00	標標工作室	1
11	8	PHP 程式語言	460.00	標標工作室	2

先依出版社分組，然後進行組內的排名