

國立彰化師範大學資訊管理學系

資料庫實作筆記

學生：S1061103 許鉅偉

指導教授：翁政雄 教授

中華民國 112 年 9 月 ~ 113 年 1 月

# 目錄

9/12 第一週 .....	1
★課堂簡要說明 .....	1
★課程的評分標準(有 80 分可以請老師寫推薦信).....	1
●安裝 SQL server 2012 .....	1
●附加資料庫 .....	5
●加入資料庫圖表 ERD(加入主鍵) .....	5
●實際演練 .....	6
1.每個產品的銷售量 .....	6
2.每個產品、每年銷售量(產品編號、產品名稱、年、銷售量).....	6
3.每個產品、每年、每月的銷售量(產品編號、產品名稱、年、月、銷售量).....	7
※小考題目 .....	7
9/19 第二週 .....	8
●為何要做資料正規化 .....	8
●何謂資料正規化 .....	8
●實際操作與演練(如何建立檢視表(3 step)).....	8
●選取資料列 .....	9
.....	9
●EXCEL 操作 .....	9
●使用 EXCEL 和 SQL 對照 演練 .....	10
1.每個產品的銷售量 .....	10
2.每個產品的銷售金額 .....	11
3. 產品每年銷售量 .....	12
※小考題目 .....	13
9/26 第三週 .....	14
★課堂概要 .....	14
●正規化的重要性 .....	14
●SQL 語法的技巧 .....	14
●Excel 操作與 SQL 語法演練 .....	14
1. 產品的銷售量 .....	14
2.找產品「三合一麥片」的銷售量 .....	15
3.找產品「三合一麥片」、「小米」、「大甲蟹」、「山渣片」的銷售量 .....	16
4.找產品「三合一麥片」、「小米」、「大甲蟹」、「山渣片」的銷售量大於 500 的 .....	17
5. 產品的項目裡，肉類的總銷售量介於 500 至 1000 的 .....	17
6.公司產品資料表中，有幾項產品 .....	17
7.公司產品資料表裡，有銷售紀錄的有幾項 .....	18
8. 第 6 題和第 7 題的比較差異，產品資料表與銷售紀錄的差異品項是哪項 .....	18
9.產品年銷售量 .....	18
※小考題目 .....	20
10/3 第四週 .....	21

●Join 的種類介紹 .....	21
1. 甲 Inner Join 乙 .....	21
2. 甲 Left Join 乙 .....	22
3. 甲 Right Join 乙 .....	22
4. 甲 Full Join 乙 .....	22
5. 甲 Cross Join 乙 .....	23
●SQL 語法演練 .....	23
1. 盡量不要用「*」因為會吃記憶體 .....	23
2. 建立甲(產品資料)和乙(銷售資料)資料表 .....	24
3. 甲和乙資料表的差別 .....	24
4. 建立 1997 年和 1998 年資料表 .....	25
5. 1997 年和 1998 年資料表的差別(產品編號).....	26
6. 1997 年和 1998 年資料表的差別項目 .....	26
7. 建立 1997 年台中市和台北市的銷售紀錄 .....	27
8. 哪些產品 1997 年台中市有銷售紀錄，台北市沒有(產品編號).....	28
9. 哪些產品 1997 年台中市有銷售紀錄，台北市沒有(產品編號和項目).....	29
10. 哪些客戶在 1997 年有訂單，1998 年沒有(客戶編號).....	30
11. 哪些客戶在 1997 年有訂單，1998 年沒有(客戶編號和公司名稱(客戶)) .....	30
※小考 .....	31
10/10 第五週 雙十連假.....	31
10/16 第六週 .....	32
●建立 excel 檔案的資料(大雄進銷存系統).....	32
●在 SQL 裡，建立 NorthwindERP .....	32
●Lazarus 操作(基本建立).....	33
※小考(無小考) .....	37
10/24 第七週 .....	38
●SQL 演練 .....	38
1.員工業年營業額、排名、年分組(Rank).....	38
2.員工业年營業額 小計 group by(拆解 1).....	39
3.員工业營業額排名(總排名).....	39
4. 員工业營業額排名(總排名、年度排名).....	40
5.客戶消費金額 top3 的產品和購買量 .....	41
6.pivot 旋轉應用 .....	43
7.pivot 加總(產品年季銷售量 季 pivot 總銷售量) .....	45
※小考 .....	46
10/31 第八週 .....	47
●SQL 演練(DDL: create、alter、drop / DCL: grant、revoke).....	47
1.新增資料庫 MyDB .....	47
2.在資料庫(MyDB)，新增資料表 customer.....	47
3.在資料庫(MyDB)，新增資料表 students .....	48
4.在資料表(students)，新增欄位(SEmail)，資料型態 nvarchar(255) .....	48

5. 在資料表(students)，刪除欄位(SEmail).....	49
6. 刪除資料表、刪除資料庫 .....	49
7. 授權使用者(readuser)，具有權限(select) / 使用者(readuser)，移除權限 .....	49
●公佈期中考考題 .....	49
※小考 .....	50
11/7 第九週 期中考試.....	51
11/14 第十週.....	53
●課堂重點 .....	53
1.核對期中考成績 .....	53
2.檢討期中考 SQL.....	53
※正規化流程說明(縱向切割-物以類聚，橫向刪除-重複資料) .....	53
3.期末報告說明 .....	53
●課程資料匯入 SQL 操作(仿 POS 發票系統(期末報告)).....	54
※ Excel 操作(縱向切割，橫向刪除) 5(切割資料表)+1(原始資料)=6 個資料表.....	54
●SQL 匯入 課程資料表 .....	57
11/21 第十一週.....	59
●課程資料表→設計，設定主鍵 .....	59
1 教師 資料表 .....	59
2 課程 資料表 .....	59
3 課程教室 資料表 .....	59
4 學生 資料表 .....	59
5 學生成績 資料表 .....	59
●課程資料表→資料庫圖表 (ERD 圖) .....	60
●SQL(新增、刪除、修改).....	60
1.刪除資料(delete).....	60
2.新增資料(insert).....	60
3.修改資料(update) .....	61
※小考 .....	61
11/28 第十二週.....	62
●SQL 演練 .....	62
1.學生修課紀錄(All Data) .....	62
2.學生修課數總分總平均(學號、姓名、修課數、總分、總平均).....	63
●Stored Procedure .....	63
1. sp_Delete_AllData .....	63
2. sp_InsertAllData.....	64
3. sp_Delete_Insert_AllData .....	64
4. sp_Delete_Student_AllData(傳入參數) .....	65
5. sp_Delete_Student_AllData_Output(傳入參數、傳出參數).....	65
6. stored procedure 應用(1)刪除學號='F100001' .....	66
7. stored procedure 應用(2) 刪除學號='F100001' .....	66
※小考 .....	67

12/5 第十三週.....	68
●SQL 操作 .....	68
1.在員工資料表新增資料(設計).....	68
2.新增「員工營業績效」資料表(新增資料表).....	68
●修改 SQL 指令演練 .....	69
1.修改員工密碼(密碼=員工編號).....	69
2.修改員工密碼(密碼=員工編號+名).....	69
3.修改員工密碼(密碼=名+出生日期).....	70
4.修改員工密碼(密碼=員工編號+名+出生日期).....	71
5.修改員工職稱是“業務”的密碼(密碼=員工編號+名+內部分機號碼).....	72
#統整第 5 題重點 .....	73
6.修改員工資料表裡的營業額(5 step).....	73
※小考 .....	76
12/12 第十四週.....	78
●SQL 演練 .....	78
1.檢討上次小考 .....	78
2.員工 1998 年營業額的年終獎金(0.01).....	79
3.在員工資料表裡，新增績效獎金 .....	80
4.1997 年 12 月員工績效獎金(營業額*0.01).....	81
●Stored Procedure .....	82
1.sp_修改員工營業額.....	82
1-1 測試是否正確 .....	83
2.sp_修改員工營業額_參數.....	84
2-2 測試是否正確 .....	85
3.sp_修改員工營業額_參數 2.....	86
※小考 .....	87
1.將員工編號、每年、每月、營業額、績效獎金 insert(員工資料表)進去.....	87
12/19 第十五週.....	88
●SQL 演練 .....	88
1.新增「登入紀錄」資料表 .....	88
2.帳號輸入正確與否 .....	88
3.將第二題寫入 Stored Produce 需求(sp_login) .....	91
4.測試 sp_login.....	93
5.Stored Produce 增加回傳次數(sp_login) .....	94
6.測試回傳次數 .....	95
7.Stored Produce 加入「登入紀錄」(sp_login) .....	96
8.測試登入紀錄(sp_login) .....	97
※小考 .....	98
1. sp_InsertEmployee .....	98
●公佈期末考題目 .....	99
12/26 第十六週.....	100

●各組期末報告 .....	100
1.sp_修改訂貨主檔所有獎項、獎金.....	100
2.sp_修改獎項_2(另一種寫法).....	108
●期末考複習 .....	115
●補充資料 .....	118
2024/1/2 第十七週 .....	119
●期末考+各組期末報告 .....	119
2024/1/9 第十八週 .....	119
●總結及結算 .....	119

# 9/12 第一週

## ★課堂簡要說明

有證照和畢業證書沒什麼了不起，一個只是證明有讀過大學，一個只是證明你有繳費，把課程修完而已，我們需要的是解決問題的能力，所以用小考的方式驗證實力。

名言：

1. 能力決定勝負，能力決定你的薪水
2. 要學得好，上課要專心，要做筆記，筆記武林秘笈
3. 期末專題很重要，非常看重

## ★課程的評分標準(有 80 分可以請老師寫推薦信)

### ◎期中 30%

1. 紙筆測驗
2. 老師前一週會先給 60 分的題目與答案，剩下 40 分靠自身的能力和程度

### ◎期末 30%

1. 紙筆測驗
2. 老師前一週會先給 60 分的題目與答案，剩下 40 分靠自身的能力和程度

### ◎平時成績 40%

1. 期末報告 20%

#做專題(POS 發票系統，4~6 人一組，每人需收集 10 張發票)、書面和團體 10%、上台口頭報告 10%

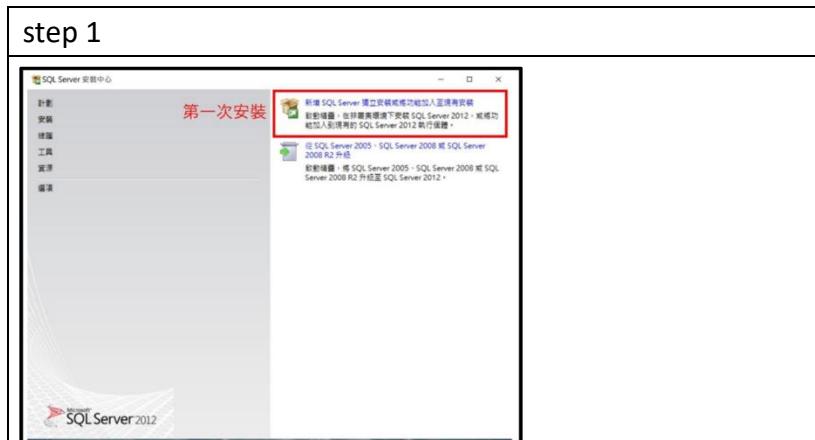
2. 小考和筆記 20%

#筆記 10%、小考+點名 10%

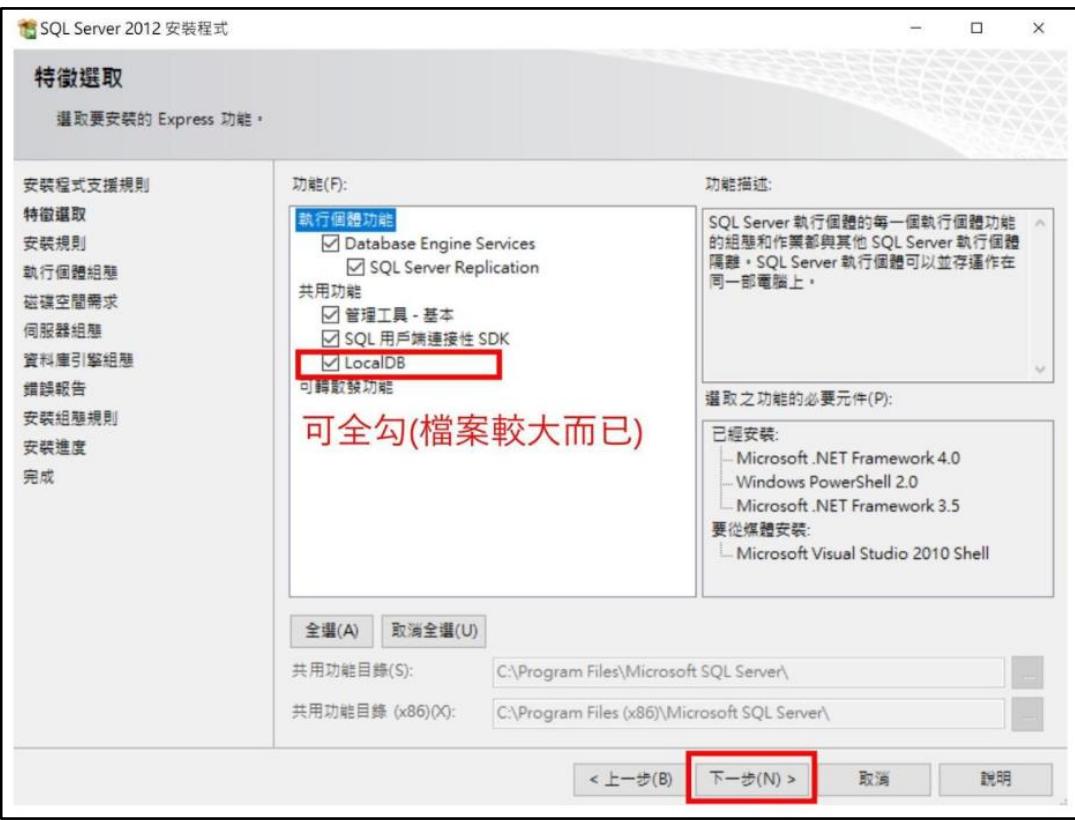
### ◎小考網址

<https://forms.gle/ZWuBEFWFUVmtsKgA8>

## ●安裝 SQL server 2012



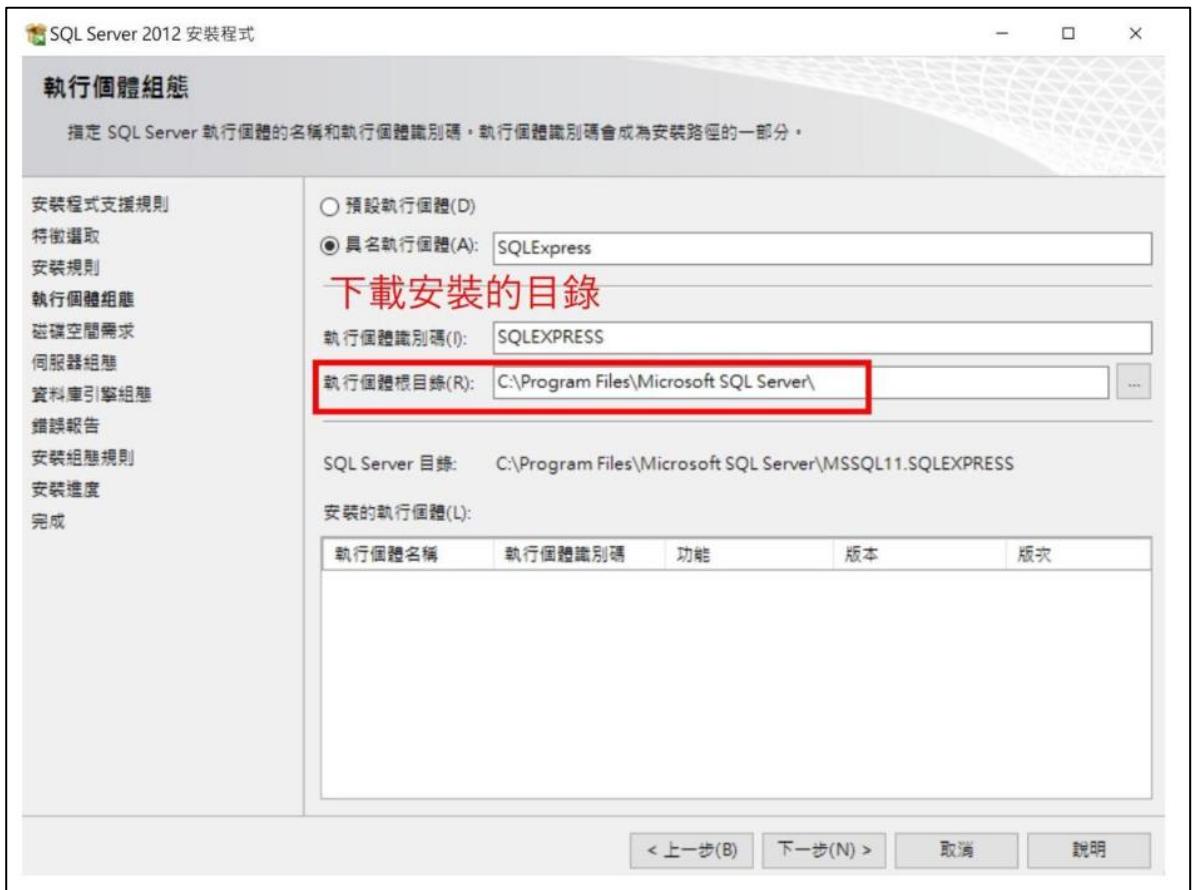
## step 2



## step3



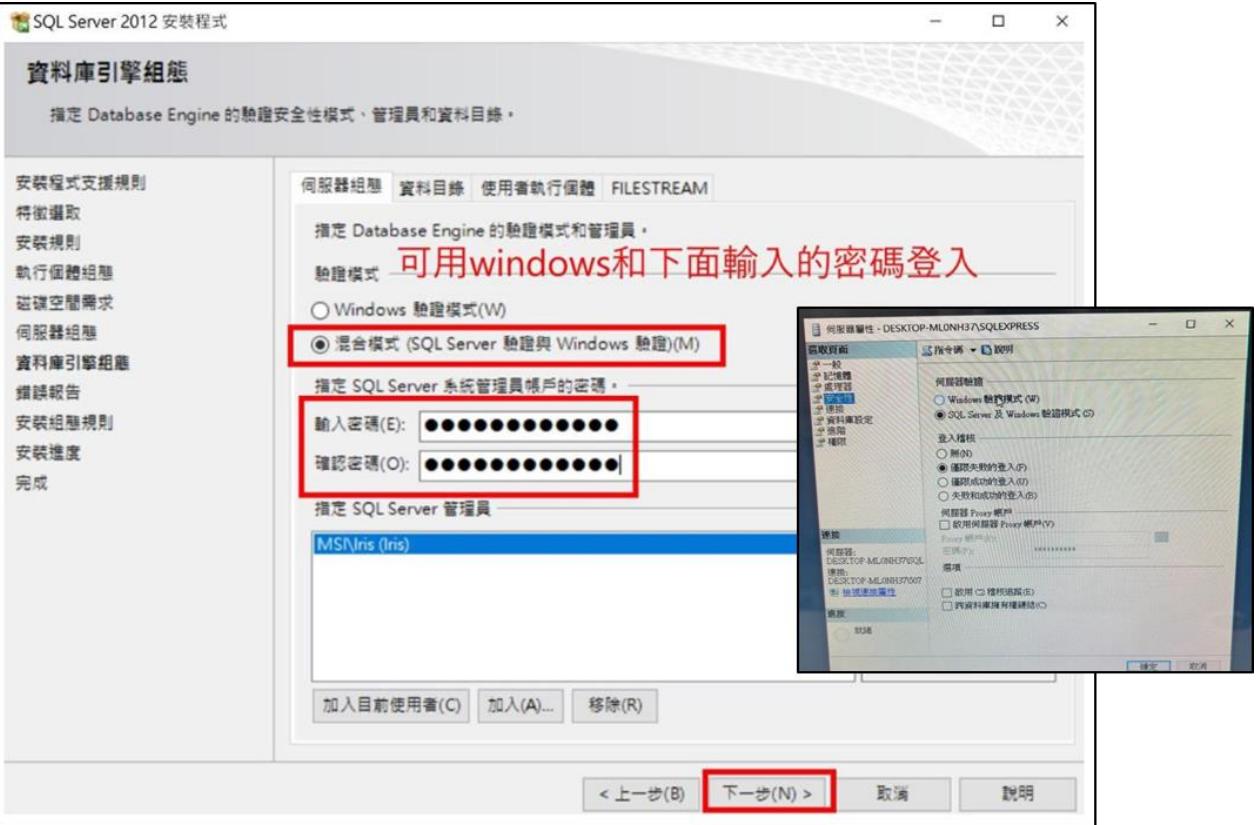
## step 4



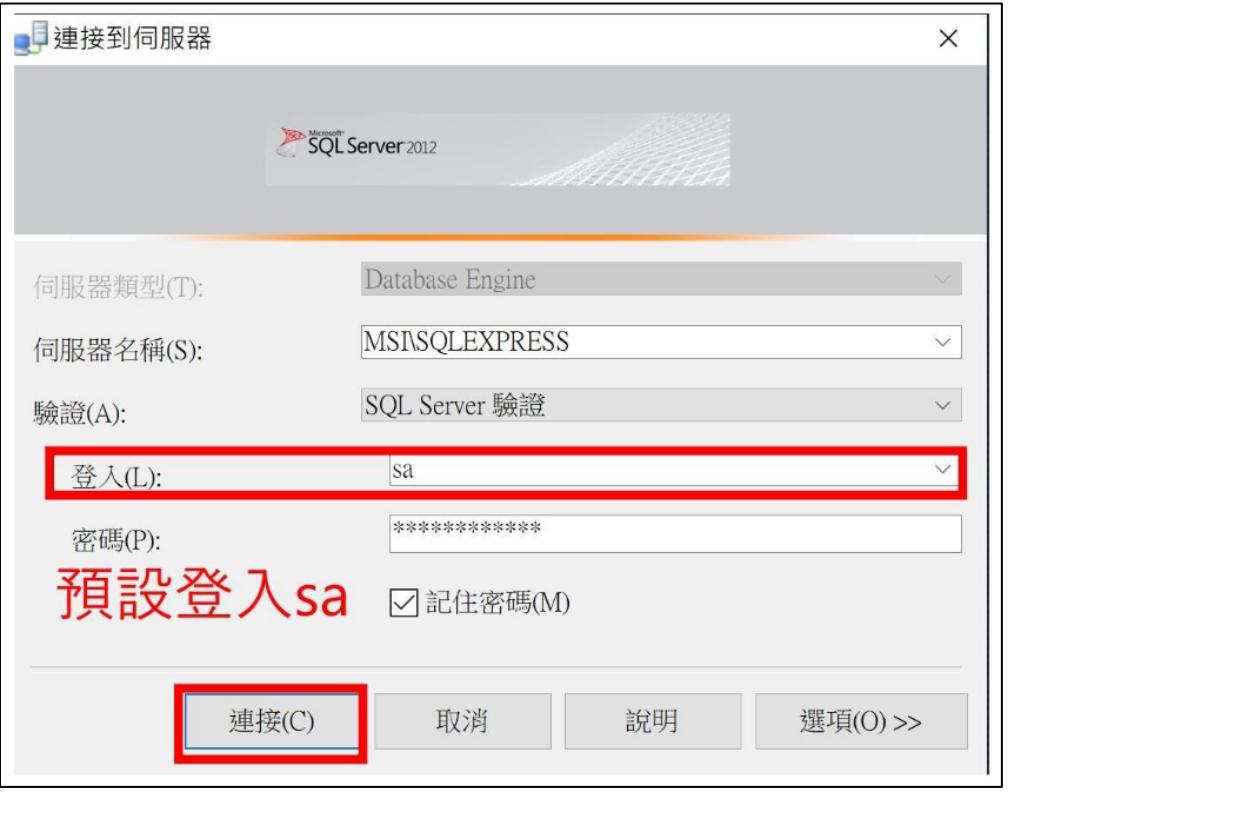
## step 5



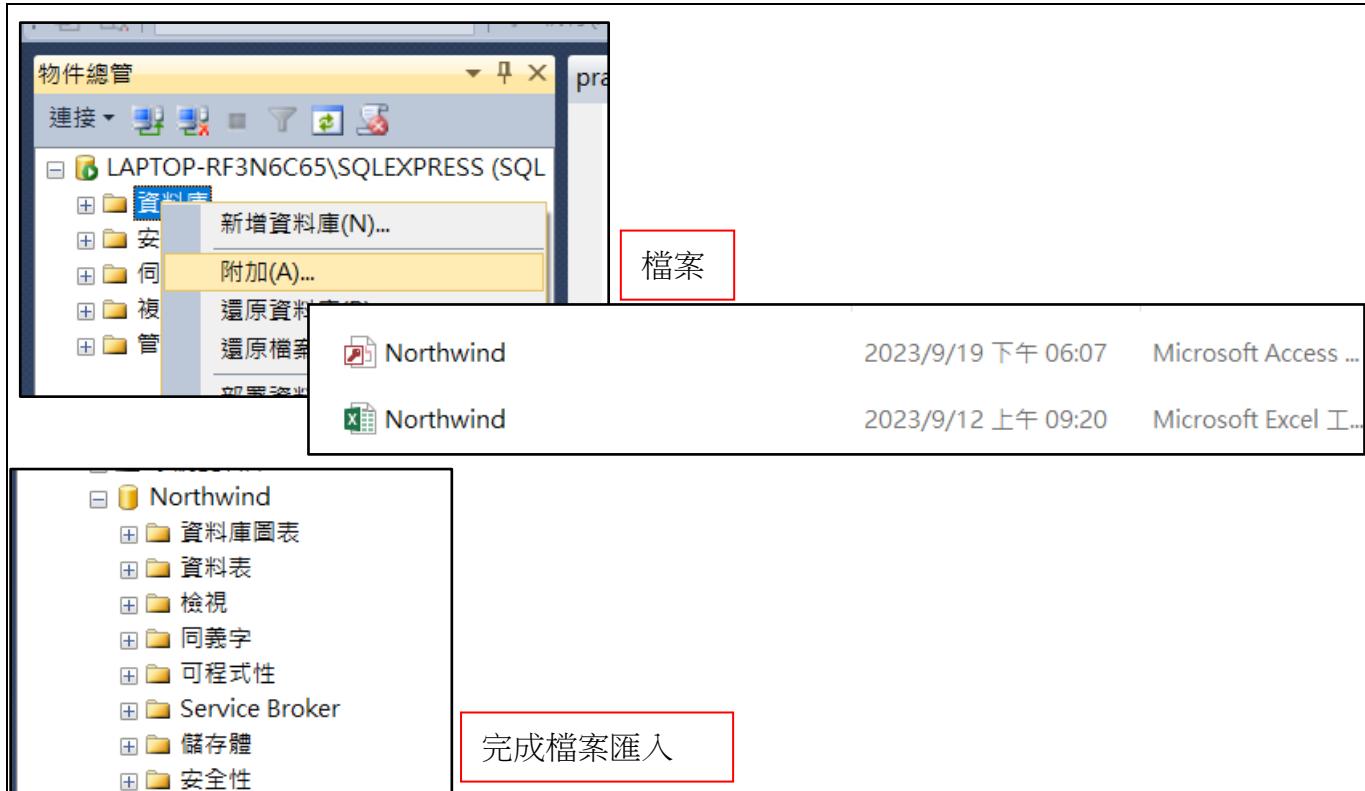
## step 6



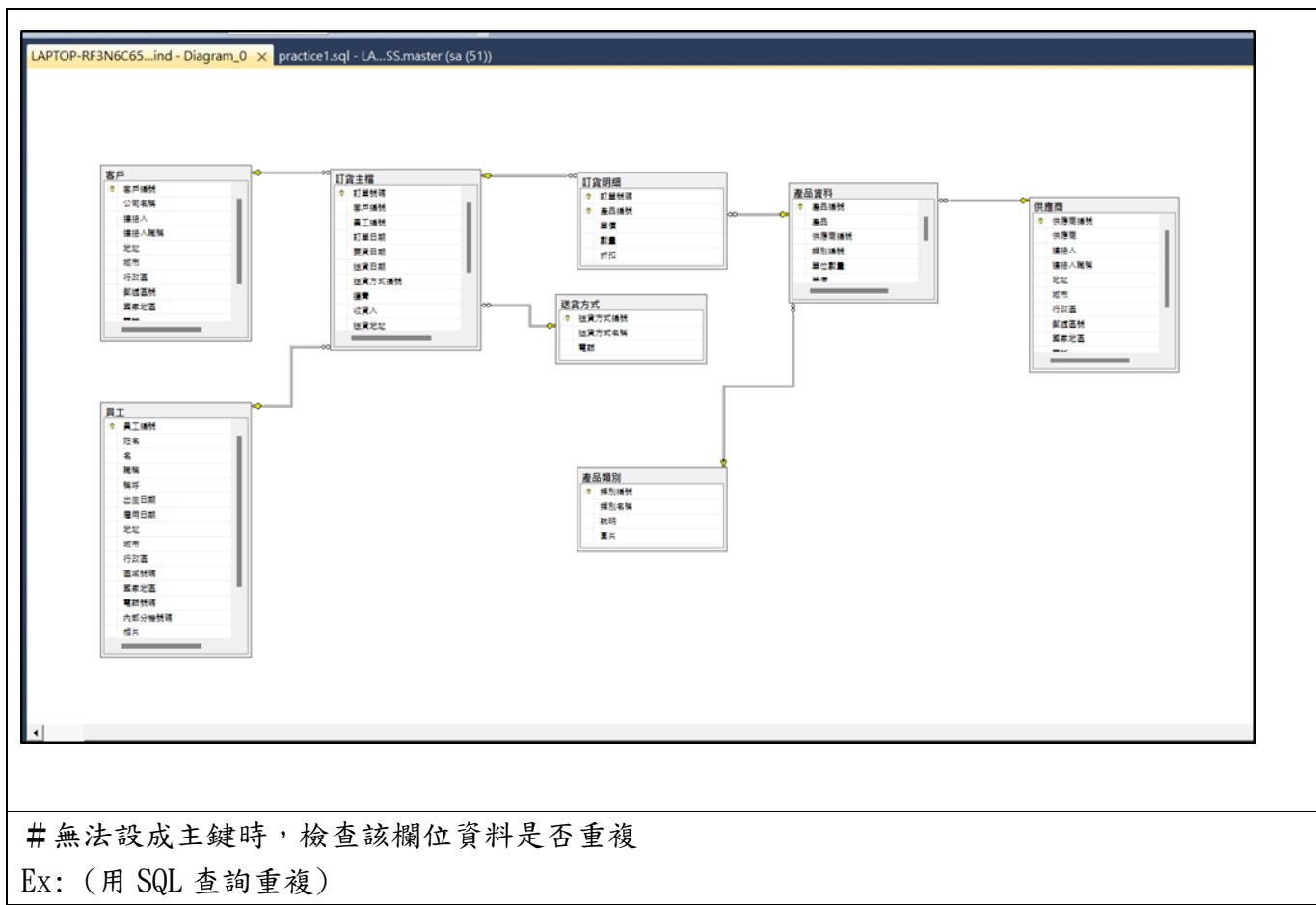
## step 7



## ●附加資料庫



## ●加入資料庫圖表 ERD(加入主鍵)



```

Select 供應商編號, count(供應商編號) as 筆數
From 供應商
Group by 供應商編號 having count(供應商編號) > 1

```

## ● 實際演練

### 1. 每個產品的銷售量

```

Select 產品資料.產品編號, 產品資料.產品, sum(訂貨明細.數量) as 銷售量
From 產品資料 inner join 訂貨明細 on 訂貨明細.產品編號=產品資料.產品編號
Group by 產品資料.產品編號, 產品資料.產品
Order by 產品資料.產品編號

```

結果 訊息

	產品編號	產品	銷售量
1	1	蘋果汁	783
2	2	牛奶	1058
3	3	蕃茄醬	328
4	4	鹽巴	453
5	5	麻油	298
6	6	醬油	301
7	7	海鮮粉	763
8	8	胡椒粉	372
9	9	讚油雞	95
10	10	大甲蟹	742
11	11	民眾起司	694
12	12	德國起司	344
13	13	龍蝦	891

### 2. 每個產品、每年銷售量(產品編號、產品名稱、年、銷售量)

```

Select 產品資料.產品編號, 產品資料.產品, year(訂貨主檔.訂單日期) as 年, sum(訂貨明細.數量) as 銷售量
From (產品資料 inner join 訂貨明細 on 訂貨明細.產品編號=產品資料.產品編號) inner
join 訂貨主檔 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
Group by 產品資料.產品編號, 產品資料.產品, year(訂貨主檔.訂單日期)
Order by 產品資料.產品編號, 產品資料.產品, year(訂貨主檔.訂單日期)

```

結果 訊息

	產品編號	產品	年	銷售量
1	1	蘋果汁	1996	80
2	1	蘋果汁	1997	304
3	1	蘋果汁	1998	399
4	2	牛奶	1996	226
5	2	牛奶	1997	435
6	2	牛奶	1998	397
7	3	蕃茄醬	1996	30
8	3	蕃茄醬	1997	190
9	3	蕃茄醬	1998	108
10	4	鹽巴	1996	107
11	4	鹽巴	1997	264
12	4	鹽巴	1998	82
13	5	麻油	1996	129

### 3. 每個產品、每年、每月的銷售量(產品編號、產品名稱、年、月、銷售量)

```
Select 產品資料.產品編號, 產品資料.產品, year(訂貨主檔.訂單日期) as 年, month(訂貨主檔.訂單日期) as 月, sum(訂貨明細.數量) as 銷售量  
From 產品資料 inner join 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼 on 產品資料.產品編號=訂貨明細.產品編號  
Group by 產品資料.產品編號, 產品資料.產品, year(訂貨主檔.訂單日期), month(訂貨主檔.訂單日期)
```

結果 訊息					
	產品編號	產品	年	月	銷售量
1	1	蘋果汁	1996	8	18
2	1	蘋果汁	1996	9	20
3	1	蘋果汁	1996	11	27
4	1	蘋果汁	1996	12	15
5	1	蘋果汁	1997	1	34
6	1	蘋果汁	1997	3	15
7	1	蘋果汁	1997	4	40
8	1	蘋果汁	1997	5	8
9	1	蘋果汁	1997	6	10
10	1	蘋果汁	1997	7	29
11	1	蘋果汁	1997	8	40
12	1	蘋果汁	1997	10	70
13	1	蘋果汁	1997	11	58

### ※小考題目

1. 今天上課學到哪些？
2. 上課有無任何問題？

# 9/19 第二週

## ●為何要做資料正規化

- 降低資料重複性(data redundancy)
- 避免資料更新異常(anomalies)

## ●何謂資料正規化

第一正規化	資料表中有主鍵，而其他所有的欄位都相依於主鍵
	每個欄位中都只儲存單一值
第二正規化	符合 1NF 的格式
	各欄位與主體間沒有“不分相依”的關係
第三正規化	符合 2NF 的格式
	非主鍵的各欄位間沒有“遞移相依”的關係

## ●實際操作與演練(如何建立檢視表(3 step))

The screenshot illustrates the process of creating a view in SQL Server Management Studio (SSMS) across three steps:

- Step 1:** A context menu is open over the "dbo.Diagram\_0" node in the Object Explorer. The "新增檢視(N...)" (New View) option is highlighted.
- Step 2:** The "LAPTOP-RF3N6C65...\dbo.vw\_AllData" view is being designed in the Diagram Designer. It shows five tables: 客戶 (Customer), 員工 (Employee), 訂貨主檔 (Order Header), 訂貨明細 (Order Details), and 產品資料 (Product). Relationships are established between them. A red box labeled "Step2" is overlaid on the diagram area.
- Step 3:** The "vw\_AllData" view is displayed in the Results pane. The results grid lists columns from various tables, including CustomerID, CompanyName, EmployeeID, FirstName, LastName, OrderID, ProductID, and Discount. A red box labeled "Step3" is overlaid on the results grid area.

## ●選取資料列

## ●EXCEL 操作

**複製貼上至 EXCEL**

客戶編號	公司名稱	員工編號	姓名	訂單日期	送貨城市	產品編號	產品	供應商編號	供應商
1	VINET	5	劉天王	1996-07-04 00:00:00.000	台北市	42	糙米	20	一心
2	VINET	5	劉天王	1996-07-04 00:00:00.000	台北市	72	麵起司	14	小陽堂
3	TOMSP	6	黎證明	1996-07-05 00:00:00.000	高雄市	14	沙茶	6	德基
4	TOMSP	6	黎證明	1996-07-05 00:00:00.000	高雄市	51	豬內乾	24	適合
5	HANAR	4	林美麗	1996-07-08 00:00:00.000	高雄市	41	蝦子	19	普三
6	HANAR	4	林美麗	1996-07-08 00:00:00.000	高雄市	51	豬肉乾	24	適合
7	HANAR	4	林美麗	1996-07-08 00:00:00.000	高雄市	65	海苔醬	2	光輝
8	VICTE	3	趙飛燕	1996-07-08 00:00:00.000	台北市	22	再來米	9	均花
9	VICTE	3	趙飛燕	1996-07-08 00:00:00.000	台北市	57	小米	26	宏仁
10	VICTE	3	趙飛燕	1996-07-08 00:00:00.000	台北市	65	海苔醬	2	光輝
11	SUPRD	4	林美麗	1996-07-09 00:00:00.000	桃園縣	20	豆乾	8	康寶
12	SUPRD	4	林美麗	1996-07-09 00:00:00.000	桃園縣	33	台中起司	15	德國
13	SUPRD	4	林美麗	1996-07-09 00:00:00.000	桃園縣	60	花生起司	28	玉成

## ●使用 EXCEL 和 SQL 對照 演練

### 1. 每個產品的銷售量

#### ●SQL :

```
Select 產品資料.產品, sum(訂貨明細.數量) as 銷售量  
From 產品資料 inner join 訂貨明細 on 產品資料.產品編號=訂貨明細.產品編號  
Group by 產品資料.產品
```

#### ●Excel 的做法：選小計，計算「產品銷售量」

The screenshot shows the Excel ribbon with the 'Data' tab selected. On the left, there is a table with columns '產品編號' and '產品'. A 'Sort & Filter' dialog is open over the table, showing 'A-Z' and 'Product' selected. In the center, the 'Outline' ribbon tab is selected, showing 'Grand Total' and 'Subtotal' buttons. On the right, a 'Subtotal' dialog box is open, with 'Category' set to 'Product' and 'Function' set to 'Sum'. The 'Subtotal' button is highlighted.

#### ◆比對

	產品	銷售量
1	墨魚	539
2	肉鬆	239
3	綠豆糕	799
4	龍蝦	891
5	甜辣醬	445
6	芭樂汁	883
7	玉米片	138
8	讚油雞	95
9	燕麥	580
10	蘇澳起司	1496
11	餅乾	1158
12	豆乾	313
13	白起司	297

E	F	G	H	I	J	K	L	M
日期	送貨城市	產品	供應商編號	供應商	單價	數量	折扣	
		三合一麥片 合計				500		
		大甲蟹 合計				742		
		小米 合計				434		
		山楂片 合計				1083		
		干貝 合計				125		
		牛奶 合計				1058		
		牛肉乾 合計				365		
		台中起司 合計				755		
		巧克力 合計				318		
		民眾起司 合計				694		
		玉米片 合計				138		
		玉米餅 合計				235		
		白米 合計				1263		
		白起司 合計				297		
		再来米 合計				348		
		肉鬆 合計				239		
		汽水 合計				1140		
		沙茶 合計				404		
		豆乾 合計				313		

## 2. 每個產品的銷售金額

### ●SQL:

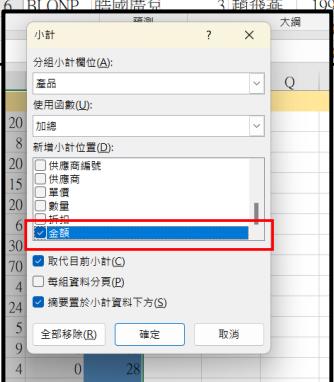
Select 產品資料.產品, sum(round(訂貨明細.單價\*訂貨明細.數量\*(1-訂貨明細.折扣),0))  
as 銷售金額

From 產品資料 inner join 訂貨明細 on 產品資料.產品編號=訂貨明細.產品編號  
Group by 產品資料.產品

### ●Excel 的作法：金額算法 = 單價\*數量\*(1-折扣)

公式=ROUND(K2\*L2\*(1-M2), 0) ROUND 函數為四捨五入

### ●選小計，計算「產品銷售金額」



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	客戶編號	公司名稱	員工編號	姓名	訂單日期	送貨城市	產品編號	產品	供應商編號	供應商				
2	PERIC	就業廣兒	7	郭國誠	1996/10/4	南投縣	52	三合一麥	24	涵合	5.6	20	0	112
3	VICTE	千固	8	蘇涵蘊	1996/10/21	台北市	52	三合一麥	24	涵合	5.6	8	0	45
4	SEVES	艾德高科	2	陳季暄	1996/12/19	台中市	52	三合一麥	24	涵合	5.6	20	0.2	90
5	TOMSP	東帝望	6	黎國明	1997/2/14	高雄市	52	三合一麥	24	涵合	5.6	15	0.1	76
6	BLONP	皓國廣兒	3	趙飛燕	1997/2/18	新竹市	52	三合一麥	24	涵合	5.6	20	0	112
					7/4/24	新竹市	52	三合一麥	24	涵合	7	6	0	42
					97/6/4	台北縣	52	三合一麥	24	涵合	7	30	0	210

### ●比對

H	I	J	K	L	M	N	O	P
編號	產品	供應商編號	供應商	單價	數量	折扣	金額	
三合一麥片	合計						3234	
大甲蟹	合計						20866	
小米	合計						7663	
山楂片	合計						47238	
干貝	合計						2688	
牛奶奶	合計						16377	
牛肉乾	合計						15101	
台中起司	合計						1652	
巧克力	合計						3705	
民衆起司	合計						12734	
玉米片	合計						1370	
玉米餅	合計						3439	
白米	合計						42548	
白起司	合計						8405	
再来不	合計						7125	
肉鬆	合計						3383	
汽水	合計						4749	
沙茶	合計						7000	
豆乾	合計						22564	
味素	合計						1785	
芭樂汁	合計						13645	
花生	合計						9104	



結果 訊息	
產品	銷售金額
1 墨魚	29173
2 肉鬆	3383
3 綠豆糕	8717
4 龍蝦	4960
5 甜辣醬	16702
6 芭樂汁	13645
7 玉米片	1370
8 讀油雞	7227
9 燕麥	4602
10 蘇澳起司	71160
11 餅乾	17215
12 豆乾	22564
13 白起司	8405

### 3. 產品每年銷售量

#### ●SQL:

產品每年銷售量

```
Select 產品資料.產品, year(訂貨主檔.訂單日期) as 年, sum(訂貨明細.數量) as 銷售量
From (產品資料 inner join 訂貨明細 on 產品資料.產品編號=訂貨明細.產品編號) inner
join 訂貨主檔
```

```
On 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
```

```
Group by 產品資料.產品, year(訂貨主檔.訂單日期)
```

```
Order by 產品資料.產品, year(訂貨主檔.訂單日期)
```

#### ●Excel 的作法：需先做樞紐分析表格(插入→樞紐分析表)（欄→年，列→產品，值→數量）

	1996	1997	1998	總計
三合一麥片	48	313	139	500
大甲蟹	85	346	311	742
小米	133	124	177	434
山渣片	250	482	351	1083
干貝	29	28	68	125
牛奶	226	435	397	1058
牛肉乾	40	260	65	365
台中起司	197	356	202	755
巧克力	71	152	95	318
民眾起司	98	374	222	694
玉米片		130	8	138
玉米餅	30	150	55	235
白米	96	971	196	1263
白起司	52	98	147	297
再來米	6	209	133	348
肉鬆	30	208	1	239
汽水	173	421	546	1140
沙茶	85	298	21	404
豆乾	106	96	111	313
味素	25	97		122

#### ●比對

	1996	1997	1998	總計
三合一麥片	48	313	139	500
大甲蟹	85	346	311	742
小米	133	124	177	434
山渣片	250	482	351	1083
干貝	29	28	68	125
牛奶	226	435	397	1058
牛肉乾	40	260	65	365
台中起司	197	356	202	755
巧克力	71	152	95	318
民眾起司	98	374	222	694
玉米片		130	8	138
玉米餅	30	150	55	235
白米	96	971	196	1263
白起司	52	98	147	297
再來米	6	209	133	348
肉鬆	30	208	1	239
汽水	173	421	546	1140
沙茶	85	298	21	404
豆乾	106	96	111	313
味素	25	97		122
芭樂汁	274	346	263	883
花生	55	610	351	1016

產品	年	銷售量
三合一麥片	1996	48
三合一麥片	1997	313
三合一麥片	1998	139
大甲蟹	1996	85
大甲蟹	1997	346
大甲蟹	1998	311
小米	1996	133
小米	1997	124
小米	1998	177
山渣片	1996	250
山渣片	1997	482
山渣片	1998	351
干貝	1996	29

已成功執行查詢。

## ※小考題目

1. 每個產品每年的銷售金額
2. 上課有無任何問題？

# 9/26 第三週

## ★課堂概要

1. 重新講解前兩個禮拜的觀念與整合
2. 再練習一次新的 excel 的資料庫檔案與 SQL 語法

## ●正規化的重要性

1. 把一張大資料拆成好幾張小資料
2. 刪除資料的重複性，避免資料庫在刪除或修改時異常
3. 相反的，查詢，則是把好幾張小資料表變成一張大資料

## ●SQL 語法的技巧

Select (選擇顯示欄位/欄位可運算/函數)

From (資料表/view/或是資料表與 view 的 join)

Where (篩選資料)(小計前的篩選) like(近似查詢)

Group by (小計欄位名稱)

Having (小計的篩選條件)(小計後的篩選)

%萬用字元，代表零個或一個以上的任意字元

Inner join 代表交集

## ●Excel 操作與 SQL 語法演練

### 1. 產品的銷售量

SQL:

```
Select 產品資料.產品, sum(訂貨明細.數量) as 銷售量
```

```
From 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號
```

```
Group by 產品資料.產品
```

## Excel 作法：

The screenshot shows the 'Summary' dialog box in Excel. In the 'Summary' dialog box, the 'Category' dropdown is set to 'Product' (产品). Under 'New Summary Position (D)', the 'Count' checkbox (數量) is selected. The 'Summary Below Detail (S)' checkbox is also checked. The 'OK' button is highlighted.

	H	I	J	K	L	M
台中起司				755		
巧克力 合計				318		
民衆起司				694		
玉米片 合計				138		
玉米餅 合計				235		
白米 合計				1263		
白起司 合計				297		
再來米 合計				348		
肉鬆 合計				239		
汽水 合計				1140		
沙茶 合計				404		
豆乾 合計				313		
味素 合計				122		
芭樂汁 合計				883		

## 對照：

H	I	J	K	L
產品編號	產品	單價	數量	折扣
三合一麥片			500	
大甲蟹	合計		742	
小米	合計		434	
山楂片	合計		1083	
干貝	合計		125	
牛奶	合計		1058	
牛肉乾	合計		365	
台中起司	合計		755	
巧克力	合計		318	
民衆起司	合計		694	
玉米片	合計		138	

	產品	銷售量
1	墨魚	539
2	肉鬆	239
3	綠豆糕	799
4	龍蝦	891
5	甜辣醬	445
6	芭樂汁	883
7	玉米片	138
8	讚油雞	95
9	燕麥	580
10	蘇澳起司	1496

## 2. 找產品「三合一麥片」的銷售量

SQL:

```
select 產品資料.產品, sum(訂貨明細.數量) as 銷售量
from 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號
where 產品資料.產品='三合一麥片'
group by 產品資料.產品
```

顯示:

	產品	銷售量
1	三合一麥片	500

### 3. 找產品「三合一麥片」、「小米」、「大甲蟹」、「山渣片」的銷售量

SQL:

```
select 產品資料.產品, sum(訂貨明細.數量) as 銷售量  
from 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號  
where 產品資料.產品='三合一麥片' or 產品資料.產品='小米' or 產品資料.產品='大甲蟹'  
or 產品資料.產品='山渣片'  
group by 產品資料.產品
```

顯示:

The screenshot shows a results grid from a SQL query execution. The grid has two columns: '產品' (Product) and '銷售量' (Sales Volume). There are four rows of data, each with an index number (1, 2, 3, 4) and a corresponding product name and its sales volume.

	產品	銷售量
1	三合一麥片	500
2	大甲蟹	742
3	小米	434
4	山渣片	1083

#### 4. 找產品「三合一麥片」、「小米」、「大甲蟹」、「山渣片」的銷售量大於 500 的

SQL:

```
select 產品資料.產品, sum(訂貨明細.數量) as 銷售量
from 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號
where 產品資料.產品='三合一麥片' or 產品資料.產品='小米' or 產品資料.產品='大甲蟹'
'or 產品資料.產品='山渣片'
group by 產品資料.產品
having sum(訂貨明細.數量) > 500
```

顯示:

結果	
	產品
1	大甲蟹
2	山渣片

#### 5. 產品的項目裡，肉類的總銷售量介於 500 至 1000 的

SQL:

```
select 產品資料.產品, sum(訂貨明細.數量) as 銷售量
from 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號
where 產品資料.產品 like '%肉%'
group by 產品資料.產品
having sum(訂貨明細.數量) between 500 and 1000
```

顯示:

結果	
	產品
1	烤肉醬
2	豬肉
3	豬肉乾
4	雞肉

#### 6. 公司產品資料表中，有幾項產品

SQL:

```
select 產品編號, 產品
from 產品資料
```

顯示:

結果	
	產品數
1	77

SQL:

```
select count(產品編號) as 產品數
from 產品資料
```

顯示:

結果	
1	77

--	--

## 7. 公司產品資料表裡，有銷售紀錄的有幾項

SQL:

#distinct 可過濾重複出現的紀錄值(唯一)

```
Select count(distinct 產品編號) as 產品數
From 訂貨明細
```

顯示:

結果	訊息
1	76

## 8. 第 6 題和第 7 題的比較差異，產品資料表與銷售紀錄的差異品項是哪項

SQL:(left join)

```
Select 產品資料.產品編號, 產品資料.產品, vw_銷售紀錄產品.產品編號
From 產品資料 left join (select distinct 訂貨明細.產品編號 from 訂貨明細)
Vw_銷售紀錄產品 on 產品資料.產品編號=vw_銷售紀錄產品.產品編號
Where vw_銷售紀錄產品.產品編號 is null
```

SQL:(right join)

```
Select 產品資料.產品編號, 產品資料.產品, vw_銷售紀錄產品.產品編號
From (select distinct 訂貨明細.產品編號 from 訂貨明細)
Vw_銷售紀錄產品 right join 產品資料 on 產品資料.產品編號=vw_銷售紀錄產品.產品編號
Where vw_銷售紀錄產品.產品編號 is null
```

顯示:

結果	訊息		
	產品編號	產品	產品編號
1	55	鴨肉	NULL

## 9. 產品年銷售量

SQL :

產品每年銷售量

```
Select 產品資料.產品, year(訂貨主檔.訂單日期) as 年, sum(訂貨明細.數量) as 銷售量
From (產品資料 inner join 訂貨明細 on 產品資料.產品編號=訂貨明細.產品編號) inner
join 訂貨主檔
On 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
Group by 產品資料.產品, year(訂貨主檔.訂單日期)
Order by 產品資料.產品, year(訂貨主檔.訂單日期)
```

## Excel 作法：

建立樞紐分析表

選擇您要分析的資料

選取表格或範圍(S)

表格/範圍(I): 產品年銷售量!\$L:\$L

使用外部資料來源(U)

選擇連線(C)...

連續名稱:

使用此活頁簿的資料模型(D)

選擇您要放置樞紐分析表的位置

新工作表(N)

已經存在的工作表(E)

位置(L): 產品年銷售量!\$Q\$18

選擇您是否要分析多個表格

新增此資料至資料模型(M)

確定 取消

新增「年」(YEAR(D2)) 建立樞紐分析表

顯示：

	產品	年	銷售量
1	三合一麥片	1996	48
2	三合一麥片	1997	313
3	三合一麥片	1998	139
4	大甲蟹	1996	85
5	大甲蟹	1997	346
6	大甲蟹	1998	311
7	小米	1996	133
8	小米	1997	124
9	小米	1998	177
10	山渣片	1996	250
11	山渣片	1997	482
12	山渣片	1998	351
13	干貝	1996	29

## ※小考題目

1. Join 種類有哪些，可以運用在哪些問題(情境)上？

Inner join (內部連接): 將兩個表公共都有的部分組成新表。

Left join (左外連接): 左表的全集及右表有的值，無值則為 Null。(outer join)

Right join (右外連接): 右表的全集及左表有的值，無值則為 Null。(outer join)

Full join (全部外部連接): 包含左右兩表的所有行，對應左右表沒有的都為 Null。

Cross join(交叉連接): 兩個資料表在結合時，不指定任何條件，即將兩個資料表中所有的可能排列組合出來

Natural join(自然連接): 自然連接有 NATURAL JOIN、NATURAL LEFT JOIN、NATURAL RIGHT JOIN，兩個表格在進行 JOIN 時，加上 NATURAL 這個關鍵字之後，兩資料表之間同名的欄位會被自動結合在一起。

2. 今天有無任何問題？

# 10/3 第四週

## ●Join 的種類介紹

已知有 2 張資料表(甲、乙)，皆有一個欄位/屬性(ID)，其資料型態為 nvarchar(2)。

其中，資料表(甲)有 3 筆資料(A, B, C)：資料表(乙)有 3 筆資料(B, C, D)。試問：

1. 甲 Inner Join 乙
2. 甲 Left Join 乙 (甲有乙沒有)(差異比較)
3. 甲 Right Join 乙 (乙有甲沒有)(差異比較)
4. 甲 Full Join 乙
5. 甲 Cross Join 乙

其結果為何？需顯示 SQL 語法及其結果！

#需先匯入甲和乙的資料表

	A	B
1	甲	乙
2	A	
3	B	B
4	C	C
5		D

### 1. 甲 Inner Join 乙

SQL:

```
select 甲.ID,乙.ID  
from 甲 inner join 乙 on 甲.ID=乙.ID
```

```
select 甲.ID,乙.ID  
from 甲,乙  
where 甲.ID=乙.ID
```

結果：

ID	ID
1	B
2	C

## 2. 甲 Left Join 乙

SQL:

```
select 甲.ID, 乙.ID  
from 甲 left join 乙 on 甲.ID=乙.ID
```

```
select 甲.ID, 乙.ID  
from 甲 left join 乙 on 甲.ID=乙.ID  
where 乙.ID is null
```

結果：

ID	ID
1	A
2	B
3	C

ID	ID
1	A

## 3. 甲 Right Join 乙

SQL:

```
select 甲.ID, 乙.ID  
from 甲 right join 乙 on 甲.ID=乙.ID
```

```
select 甲.ID, 乙.ID  
from 甲 right join 乙 on 甲.ID=乙.ID  
where 甲.ID is null
```

結果：

ID	ID
1	B
2	C
3	NULL

ID	ID
1	D

## 4. 甲 Full Join 乙

SQL:

```
select 甲.ID, 乙.ID  
from 甲 full join 乙 on 甲.ID=乙.ID
```

```
select 甲.ID, 乙.ID
```

```
from 甲 full join 乙 on 甲.ID=乙.ID  
where 甲.ID is null or 乙.ID is null
```

結果：

ID	ID
1	A
2	B
3	C
4	NULL

ID	ID
1	A
2	NULL

## 5. 甲 Cross Join 乙

SQL:

```
select 甲.ID,乙.ID  
from 甲 cross join 乙
```

```
select 甲.ID,乙.ID  
from 甲,乙
```

結果：

ID	ID
1	A B
2	B B
3	C B
4	A C
5	B C
6	C C
7	A D
8	B D
9	C D

## ●SQL 語法演練

### 1. 盡量不要用「\*」因為會吃記憶體

SQL :

```
select * from 產品資料
```

結果：

產品編號	產品	供應商編號	類別編號	單位數量	單價	庫存量	已訂購量	安全存量	不再銷售
1	蘋果汁	1	1	每箱24瓶	18.00	39	0	10	0
2	牛奶	1	1	每箱24瓶	19.00	17	40	25	0
3	蕃茄醬	1	2	每箱12瓶	10.00	13	70	25	0
4	鹽巴	2	2	每箱12瓶	22.00	53	0	0	0
5	麻油	2	2	每箱12瓶	21.35	0	0	0	1
6	醬油	3	2	每箱12瓶	25.00	120	0	25	0
7	海鮮粉	3	7	每箱30盒	30.00	15	0	10	0
8	胡椒粉	3	2	每箱30盒	40.00	6	0	0	0
9	味中之味	4	6	每袋500克	22.00	20	0	0	1

## 2. 建立甲(產品資料)和乙(銷售資料)資料表

甲 SQL:

```
select 產品編號, 產品 from 產品資料
```

甲 結果 :

	產品編號	產品
1	1	蘋果汁
2	2	牛奶
3	3	蕃茄醬
4	4	鹽巴
5	5	麻油
6	6	醬油
7	7	海鮮粉
8	8	胡椒粉
9	9	讚油雞
10	10	大甲蟹
11	11	民衆起司
12	12	德國起司
13	13	龍蝦

77 個資料列

乙 SQL:

```
select distinct 產品編號 from 訂貨明細
```

乙 結果 :

	產品編號
1	23
2	46
3	69
4	29
5	75
6	15
7	9
8	3
9	52
10	72
11	66
12	32
13	26

76 個資料列

## 3. 甲和乙資料表的差別

SQL :

```
select 產品資料.產品編號, 產品資料.產品, vw_銷售紀錄.產品編號  
from 產品資料 left join (select distinct 產品編號 from 訂貨明細) vw_銷售紀錄 on 產品資料.產品編號=vw_銷售紀錄.產品編號  
where vw_銷售紀錄.產品編號 is null
```

結果 :

	結果	訊息
	產品編號	產品
1	55	鴨肉
		NULL

#### 4. 建立 1997 年和 1998 年資料表

1997 年資料表 SQL :

```
Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1997
```

結果：

結果	
	產品編號
1	23
2	46
3	69
4	29
5	75
6	9
7	15
8	3
9	52
10	72

76 個資料列

1998 年資料表 SQL :

```
Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1998
```

結果：

結果	
	產品編號
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11

75 個資料列

## 5. 1997 年和 1998 年資料表的差別(產品編號)

SQL :

```
Select vw_1997.產品編號  
from (Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1997)vw_1997 left join  
(Select distinct 產品編號 from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂  
貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1998 )vw_1998 on vw_1997.產品編號=vw_1998.產品編號  
where vw_1998.產品編號 is null
```

結果：

結果	
訊息	
	產品編號
1	15

## 6. 1997 年和 1998 年資料表的差別項目

SQL :

```
Select 產品資料.產品編號,產品  
from 產品資料 inner join (Select vw_1997.產品編號  
from (Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1997)vw_1997 left join  
(Select distinct 產品編號 from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂  
貨明細.訂單號碼  
Where year(訂貨主檔.訂單日期)=1998 )vw_1998 on vw_1997.產品編號=vw_1998.產品編號  
where vw_1998.產品編號 is null)  
vw_A on 產品資料.產品編號=vw_A.產品編號
```

結果：

結果	
訊息	
產品編號	產品
1	15 味素

## 7. 建立 1997 年台中市和台北市的銷售紀錄

1997 年台中市 SQL :

```
Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where 送貨城市='台中市' and year(訂貨主檔.訂單日期)=1997
```

結果：

結果		訊息
	產品編號	
1	1	
2	2	
3	3	
4	4	
5	7	
6	9	
7	10	
8	11	
9	13	
10	14	
11	16	
		62 個資料列

1997 年台北市 SQL :

```
Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where 送貨城市='台北市' and year(訂貨主檔.訂單日期)=1997
```

結果：

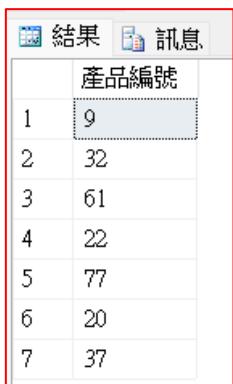
結果		訊息
	產品編號	
1	1	
2	2	
3	3	
4	4	
5	5	
6	7	
7	8	
8	10	
9	11	
10	12	
		66 個資料列

## 8. 哪些產品 1997 年台中市有銷售紀錄，台北市沒有(產品編號)

SQL :

```
Select vw_1997_台中市.產品編號
from (Select distinct 產品編號
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
Where 送貨城市='台中市' and year(訂貨主檔.訂單日期)=1997)vw_1997_台中市
left join
(Select distinct 產品編號
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
Where 送貨城市='台北市' and year(訂貨主檔.訂單日期)=1997)vw_1997_台北市 on vw_1997_
台中市.產品編號=vw_1997_台北市.產品編號
where vw_1997_台北市.產品編號 is null
```

結果：



The screenshot shows a database query results window. At the top, there are two tabs: '結果' (Results) and '訊息' (Messages). The '結果' tab is selected. Below the tabs, there is a table with a single column labeled '產品編號' (Product ID). The table contains seven rows, each with a number from 1 to 7. Row 1 contains the value '9', which is highlighted with a red rectangular selection box. The other rows contain values 32, 61, 22, 77, 20, and 37 respectively.

產品編號	
1	9
2	32
3	61
4	22
5	77
6	20
7	37

## 9. 哪些產品 1997 年台中市有銷售紀錄，台北市沒有(產品編號和項目)

SQL :

```
select 產品資料.產品編號, 產品資料.產品  
from 產品資料 inner join (select vw_1997_台中市.產品編號  
from (select distinct 產品編號 from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
where 送貨城市='台中市' and year(訂貨主檔.訂單日期)=1997)vw_1997_台中市 left join  
(Select distinct 產品編號  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
Where 送貨城市='台北市' and year(訂貨主檔.訂單日期)=1997)vw_1997_台北市 on vw_1997_  
台中市.產品編號=vw_1997_台北市.產品編號  
where vw_1997_台北市.產品編號 is null)vw_A on 產品資料.產品編號=vw_A.產品編號
```

結果：



The screenshot shows a database query results window. At the top, there are two tabs: '結果' (Results) and '訊息' (Messages). The '結果' tab is selected. Below the tabs, there is a table with two columns: '產品編號' (Product ID) and '產品' (Product Name). The table contains 7 rows of data:

	產品編號	產品
1	9	讚油雞
2	20	豆乾
3	22	再來米
4	32	白起司
5	37	千貝
6	61	海鮮醬
7	77	辣椒粉

## 10. 哪些客戶在 1997 年有訂單，1998 年沒有(客戶編號)

SQL :

```
select vw_客戶 1997.客戶編號  
from  
(select distinct 客戶編號 from 訂貨主檔 where year(訂單日期)=1998) vw_客戶 1998  
right join  
(select distinct 客戶編號 from 訂貨主檔 where year(訂單日期)=1997) vw_客戶 1997  
on vw_客戶 1997.客戶編號 = vw_客戶 1998.客戶編號  
where vw_客戶 1998.客戶編號 is null
```

結果：

客戶編號	
1	FAMIA
2	FOLIG
3	GROSR
4	HUNGC
5	LAZYK
6	MEREP
7	VINET

## 11. 哪些客戶在 1997 年有訂單，1998 年沒有(客戶編號和公司名稱(客戶))

SQL :

```
select 客戶.客戶編號,客戶.公司名稱  
from 客戶 inner join (select vw_客戶 1997.客戶編號  
from  
(select distinct 客戶編號 from 訂貨主檔 where year(訂單日期)=1998) vw_客戶 1998  
right join  
(select distinct 客戶編號 from 訂貨主檔 where year(訂單日期)=1997) vw_客戶 1997  
on vw_客戶 1997.客戶編號 = vw_客戶 1998.客戶編號  
where vw_客戶 1998.客戶編號 is null)  
vw_A on 客戶.客戶編號=vw_A.客戶編號
```

結果：

客戶編號		公司名稱
1	FAMIA	紅陽事業
2	FOLIG	路福村
3	GROSR	南華銀行
4	HUNGC	五金機械
5	LAZYK	春永建設
6	MEREP	資醫
7	VINET	山泰企業

## ※小考

1. 哪些客戶在 1997 年有訂單，1998 年沒有
2. 上課有無任何問題

10/10 第五週 雙十連假

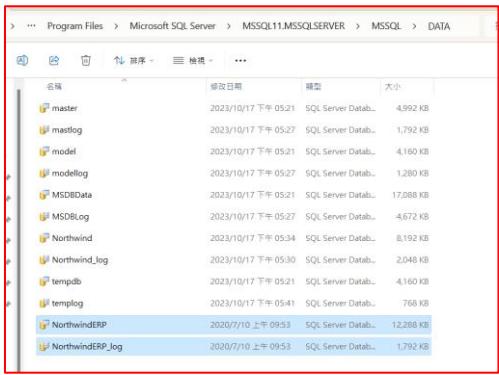
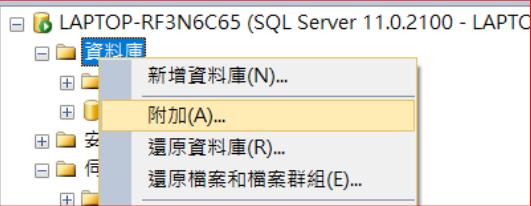
# 10/16 第六週

## ●建立 excel 檔案的資料(大雄進銷存系統)

A	B	C	D	E
系統(A)	資料(B)	訂單(C)	進貨(D)	銷貨(E)
登入(A01)	員工查詢(BE1)	訂單查詢(CO1)	進貨單查詢(DP1)	銷貨單查詢(ES1)
登出(A02)	員工管理(BE2)	訂單管理(CO2)	進貨單管理(DP2)	銷貨單管理(ES2)
	顧客查詢(BC1)			
	顧客管理(BC2)			
	產品查詢(BP1)			
	產品管理(BP2)			
	供應商查詢(BS1)			
	供應商管理(BS2)			

A	B	C
Form名稱	Form編號	功能說明
主功能表	Frm000	提供使用者功能選單及控制視窗畫面
登入(A01)	FrmA01	登入系統，使用者輸入帳密
登出(A02)	FrmA02	登出系統，使用離開系統。程式關閉
員工查詢(BE1)	FrmBE1	查詢員工基本資料。查詢條件：員工編號、員工姓名、部門
員工管理(BE2)	FrmBE2	新增、刪除、修改員工基本資料。須注意資料參考一致性(外來鍵)
顧客管理(BC1)	FrmBC1	
顧客管理(BC2)	FrmBC2	
產品查詢(BP1)	FrmBP1	
產品管理(BP2)	FrmBP2	
訂單查詢(CO1)	FrmCO1	
訂單管理(CO2)	FrmCO2	新增、刪除、修改員工基本資料。須注意資料參考一致性(外來鍵)

## ●在 SQL 裡，建立 NorthwindERP

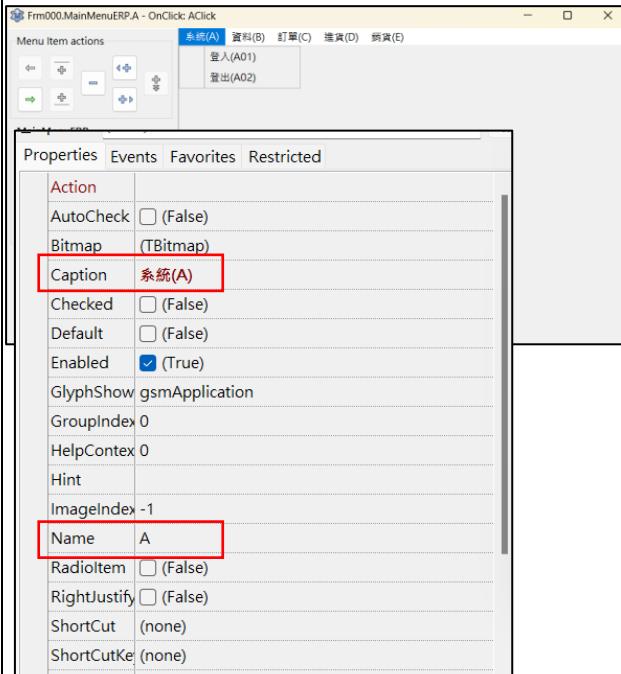
step 1 (放置檔案到 SQL 資料夾裡) (program files→SQL→data)	step 2(附加 northwindERP 資料庫)
	

step 3(刪除 erpuser)	step 4(新增登入)
	<p style="color: red; font-weight: bold;">密: erp2012</p>

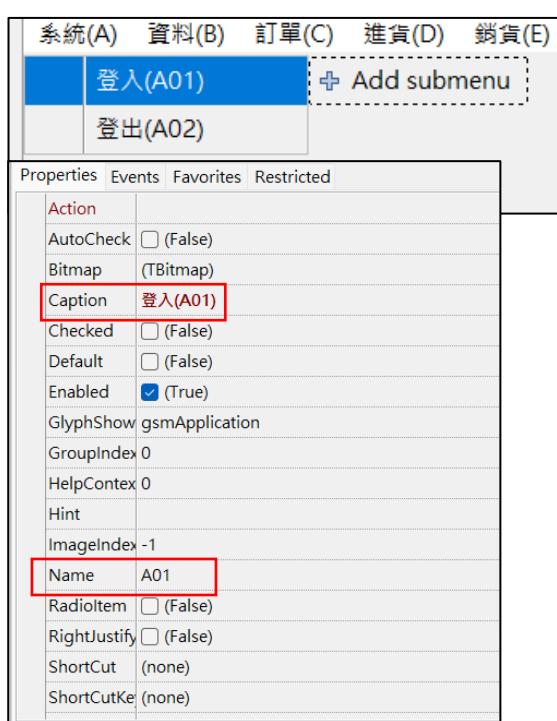
## ● Lazarus 操作(基本建立)

step 1(修改 form 名稱)	step 2(建立 MainMenuERP)

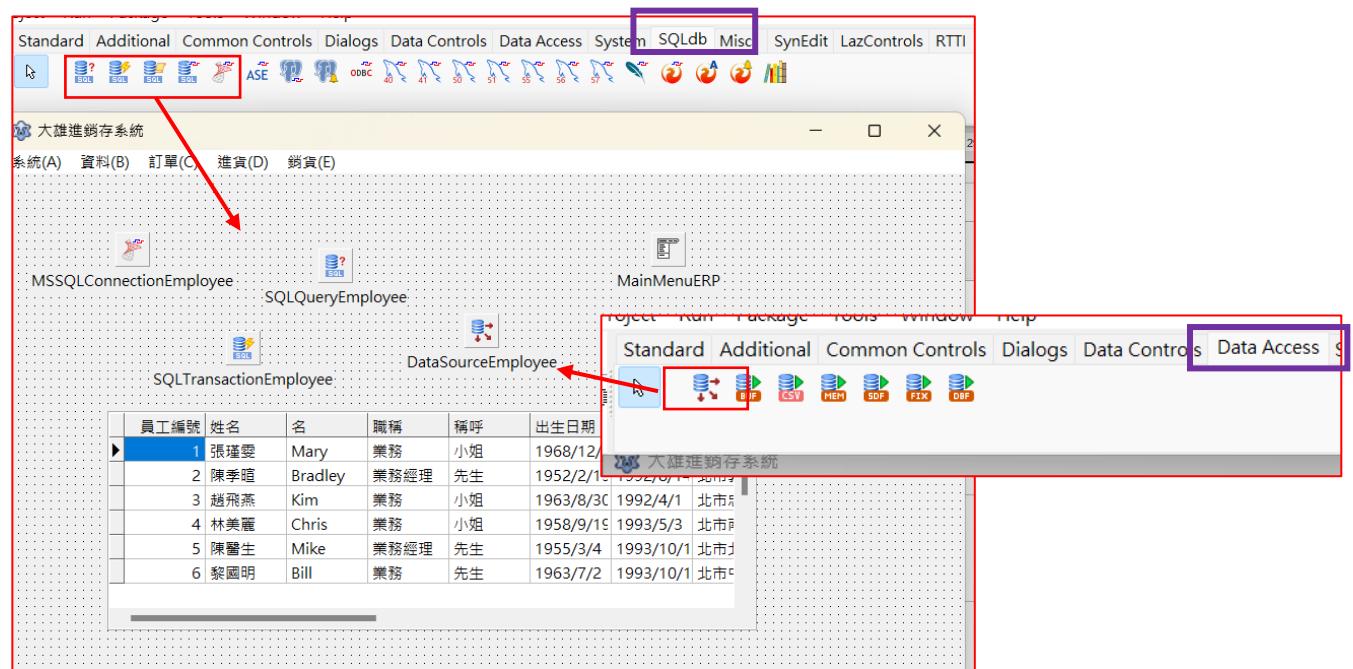
step 3(MainMenuERP 點擊，製作上方功能表)  
以此類推，做「系統」至「銷貨」



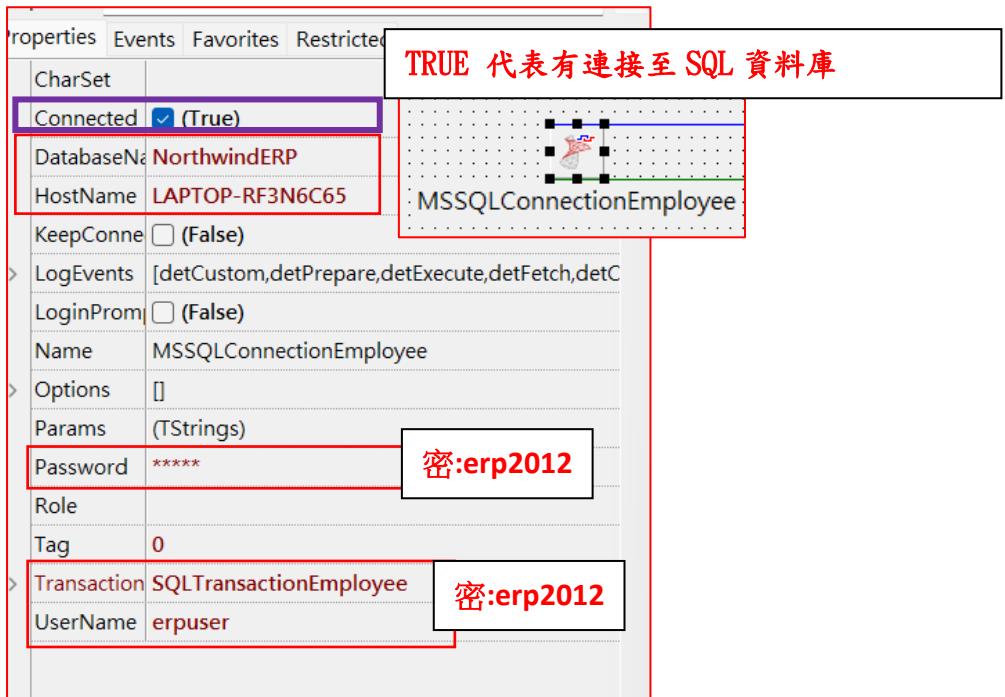
step 3(MainMenuERP 點擊，製作下方功能表)  
以此類推，做「系統」至「銷貨」



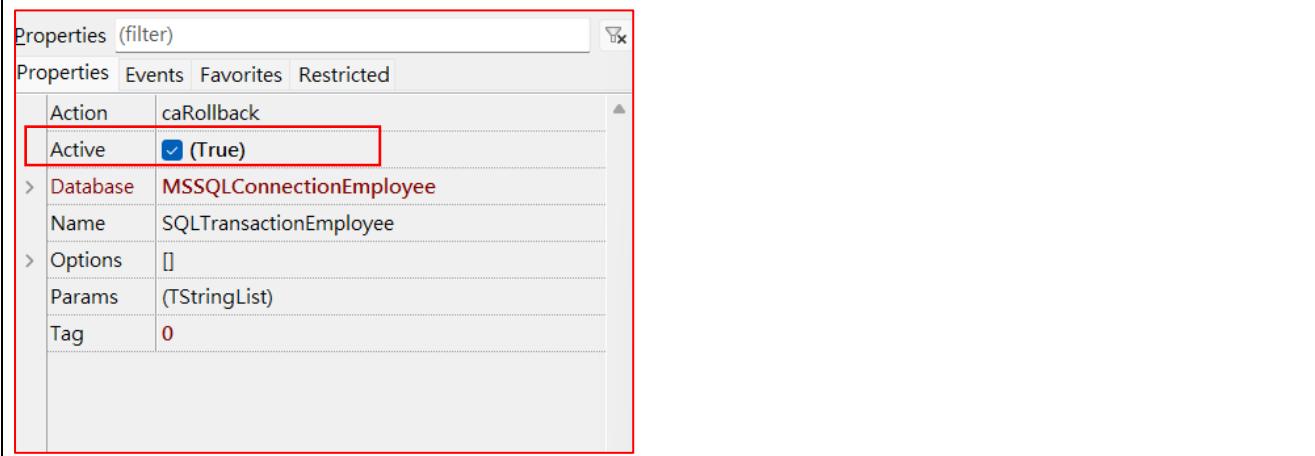
step 4(建立連接資料庫)



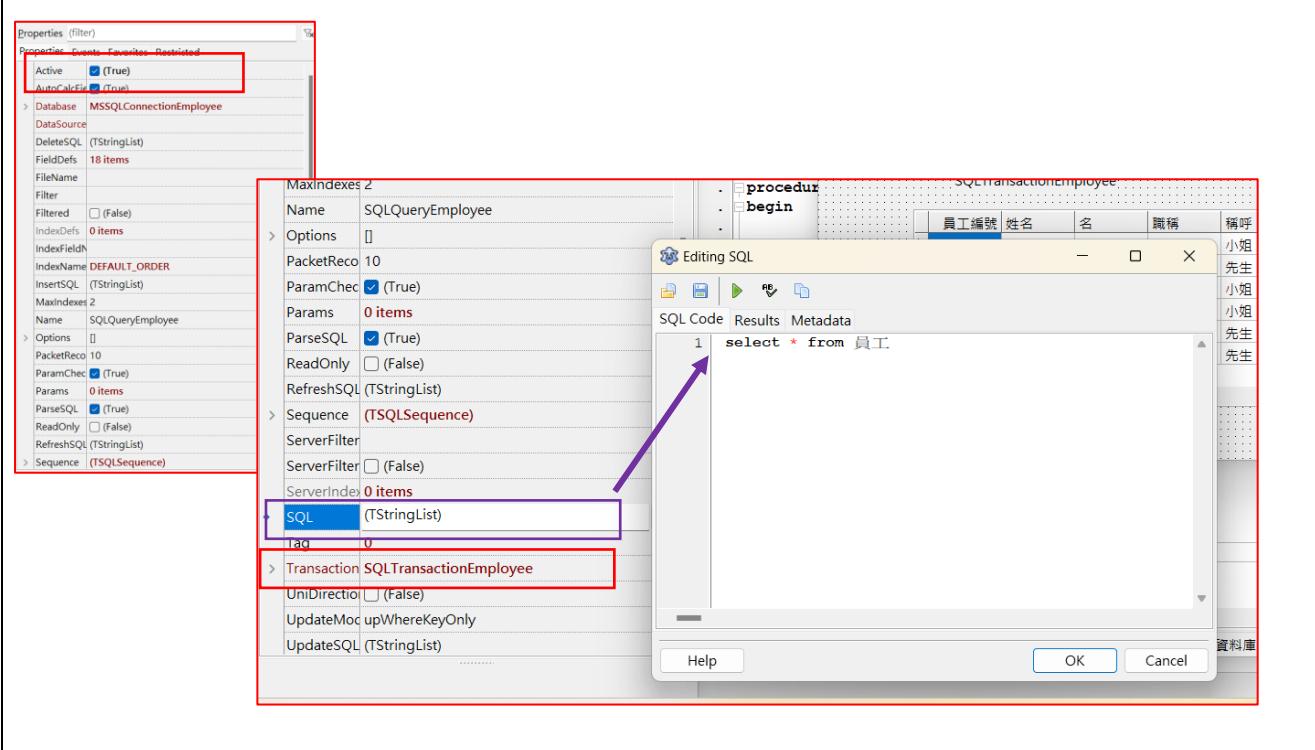
## step 5(connection 設定)(True 確認連接)



## step 6(Transaction)



## step 7(SQLQuery 設定)



## step 8(DataSource)(確認連接成功)

The screenshot shows the Visual Studio Properties window for a Data Source named 'DataSourceEmployee'. The 'AutoEdit' property is set to 'True' (checked). The 'DataSet' property is set to 'SQLQueryEmployee'. The 'Enabled' property is also set to 'True'. The 'Name' property is 'DataSourceEmployee' and the 'Tag' property is '0'. A red arrow points from the 'AutoEdit' property in the Properties window down to the DataGridView control below.

員工編號	姓名	名	職稱	稱呼	出生日期	雇用日期	地址
1	張瑾雲	Mary	業務	小姐	1968/12/8	1992/1/5	北市
2	陳季暄	Bradley	業務經理	先生	1952/2/19	1992/8/14	北市
3	趙飛燕	Kim	業務	小姐	1963/8/30	1992/4/1	北市
4	林美麗	Chris	業務	小姐	1958/9/19	1993/5/3	北市
5	陳醫生	Mike	業務經理	先生	1955/3/4	1993/10/1	北市
6	黎國明	Bill	業務	先生	1963/7/2	1993/10/1	北市

※小考(無小考)

# 10/24 第七週

## ●SQL 演練

### 1. 員工年營業額、排名、年分組(Rank)

SQL:

```
select 員工編號,姓名,年,營業額,  
rank() over (partition BY 年 ORDER BY 營業額 DESC) As 營業額年排名,  
rank() OVER (order BY 營業額 DESC) AS 營業額排名  
From  
(select 員工.員工編號,員工.姓名,DATEPART(yyyy,訂貨主檔.訂單日期) AS  
年,ROUND(SUM((訂貨明細.單價*訂貨明細.數量)*(1-訂貨明細.折扣)),0) AS 營業額  
FROM 員工 INNER JOIN 訂貨主檔 ON 員工.員工編號 = 訂貨主檔.員工編號 INNER JOIN  
訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼  
GROUP BY 員工.員工編號, 員工.姓名, DATEPART(yyyy,訂貨主檔.訂單日期)) vw
```

結果:

	員工編號	姓名	年	營業額	營業額年排名	營業額排名
1	4	林美麗	1997	128810	1	1
2	3	趙飛燕	1997	108026	2	2
3	1	張瑾雯	1997	93148	3	3
4	3	趙飛燕	1998	75629	1	4
5	2	陳季暄	1998	74337	2	5
6	2	陳季暄	1997	70444	4	6
7	1	張瑾雯	1998	63195	3	7
8	7	郭國誠	1997	60471	5	8
9	8	蘇涵蘊	1997	56033	6	9
10	4	林美麗	1998	54136	4	10
11	4	林美麗	1996	49945	1	11
12	7	郭國誠	1998	48865	5	12
13	8	蘇涵蘊	1998	48590	6	13

已成功執行查詢。

## 2. 員工年營業額 小計 group by(拆解 1)

SQL:

```
select 員工.員工編號, 員工.姓名, DATEPART(yyyy, 訂貨主檔.訂單日期) AS 年, ROUND(SUM((訂  
貨明細.單價*訂貨明細.數量)*(1-訂貨明細.折扣)),0) AS 營業額  
FROM 員工 INNER JOIN 訂貨主檔 ON 員工.員工編號 = 訂貨主檔.員工編號  
INNER JOIN  
訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼  
GROUP BY 員工.員工編號, 員工.姓名, DATEPART(yyyy, 訂貨主檔.訂單日期)
```

結果：

	員工編號	姓名	年	營業額
1	3	趙飛燕	1996	18224
2	9	孟庭亭	1998	41103
3	6	黎國明	1997	43126
4	7	郭國誠	1998	48865
5	1	張瑾雯	1997	93148
6	5	劉天王	1996	18216
7	2	陳季暄	1998	74337
8	6	黎國明	1998	14144
9	4	林美麗	1998	54136
10	8	蘇涵蘊	1997	56022

## 3. 員工營業額排名(總排名)

SQL:

```
select 員工編號, 姓名, 年, 營業額, rank() over (order by 營業額 desc) as 總排名  
from  
(  
select 員工.員工編號, 員工.姓名, DATEPART(yyyy, 訂貨主檔.訂單日期) AS 年, ROUND(SUM((訂  
貨明細.單價*訂貨明細.數量)*(1-訂貨明細.折扣)),0) AS 營業額  
FROM 員工 INNER JOIN 訂貨主檔 ON 員工.員工編號 = 訂貨主檔.員工編號 INNER JOIN  
訂貨明細 ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼  
GROUP BY 員工.員工編號, 員工.姓名, DATEPART(yyyy, 訂貨主檔.訂單日期)  
)_vw
```

結果：

	員工編號	姓名	年	營業額	總排名
1	4	林美麗	1997	128810	1
2	3	趙飛燕	1997	108026	2
3	1	張瑾雯	1997	93148	3
4	3	趙飛燕	1998	75629	4
5	2	陳季暄	1998	74337	5
6	2	陳季暄	1997	70444	6
7	1	張瑾雯	1998	63195	7
8	7	郭國誠	1997	60471	8
9	8	蘇涵蘊	1997	56033	9
10	4	林美麗	1998	54136	10
11	4	林美麗	1996	49945	11
12	7	郭國誠	1998	48865	12
13	8	蘇涵蘊	1998	48590	13

已成功執行查詢。

#### 4. 員工營業額排名(總排名、年度排名)

SQL:

```
select 員工編號,姓名,年,營業額,  
rank() over (order by 營業額 desc) as 總排名  
,rank () over (partition by 年 order by 營業額 desc) as 年度排名  
from(  
select 員工.員工編號,姓名,DATEPART(YYYY,訂貨主檔.訂單日期)as 年,  
Round(sum(訂貨明細.單價*數量*(1-折扣)),0) as 營業額 from 員工 inner join  
訂貨主檔 on 員工.員工編號=訂貨主檔.員工編號 inner join 訂貨明細 on 訂貨主檔.訂單  
號碼=訂貨明細.訂單號碼  
group by 員工.員工編號,姓名,DATEPART(YYYY,訂貨主檔.訂單日期)  
)_vw
```

結果:

	員工編號	姓名	年	營業額	總排名	年度排名
1	4	林美麗	1996	49945	11	1
2	1	張瑾雯	1996	35246	16	2
3	8	蘇涵蘊	1996	22240	19	3
4	2	陳季暄	1996	21757	20	4
5	3	趙飛燕	1996	18224	22	5
6	5	劉天王	1996	18216	23	6
7	6	黎國明	1996	16643	24	7
8	7	郭國誠	1996	15232	25	8
9	9	孟庭亭	1996	9895	27	9
10	4	林美麗	1997	128810	1	1
11	3	趙飛燕	1997	108025	2	2
12	1	張瑾雯	1997	93148	3	3
13	2	陳季暄	1997	70444	6	4

## 5. 客戶消費金額 top3 的產品和購買量

step 1 (所有客戶購買的產品和數量)vw\_B

SQL:

```
select 訂貨主檔.客戶編號, 產品資料.產品編號, 產品資料.產品, sum(訂貨明細.數量) as 數量
from (訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼)
inner join
產品資料 on 訂貨明細.產品編號 = 產品資料.產品編號
group by 訂貨主檔.客戶編號, 產品資料.產品編號, 產品資料.產品
```

結果:

	客戶編號	產品編號	產品	數量
1	BERGS	1	蘋果汁	35
2	BLONP	1	蘋果汁	25
3	BOTTM	1	蘋果汁	60
4	CHOPS	1	蘋果汁	15
5	DUMON	1	蘋果汁	3
6	EASTC	1	蘋果汁	25
7	GODOS	1	蘋果汁	10
8	GREAL	1	蘋果汁	8
9	HILAA	1	蘋果汁	20
10	HUNGO	1	蘋果汁	15
11	LAMAI	1	蘋果汁	24
12	LEHMS	1	蘋果汁	80
13	LINOD	1	蘋果汁	54

已成功執行查詢。

step 2(前三名客戶和消費金額)vw\_A

SQL:

```
select top 3 客戶.客戶編號, 客戶.公司名稱, sum((訂貨明細.數量*訂貨明細.單價)*(1-訂貨明細.折扣)) as 消費金額
from 客戶 inner join 訂貨主檔 on 客戶.客戶編號= 訂貨主檔.客戶編號
inner join 訂貨明細 on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼
group by 客戶.客戶編號, 客戶.公司名稱
order by 消費金額 desc
```

結果:

	客戶編號	公司名稱	消費金額
1	QUICK	高上補習班	109758.904953003
2	ERNSH	正人資源	104874.978713989
3	SAVEA	大鈺貿易	104361.949920654

## step finally (兩表和在一起)

SQL:

```
select vw_A.客戶編號, vw_A.公司名稱, vw_A.消費金額, vw_B.產品編號, vw_B.產品, vw_B.數量
from
(select top 3 客戶.客戶編號, 客戶.公司名稱, sum((訂貨明細.數量*訂貨明細.單價)*(1-訂貨明細.折扣)) as 消費金額
from 客戶 inner join 訂貨主檔 on 客戶.客戶編號= 訂貨主檔.客戶編號
inner join 訂貨明細 on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼
group by 客戶.客戶編號, 客戶.公司名稱
order by 消費金額 desc)vw_A
inner join
(
select 訂貨主檔.客戶編號, 產品資料.產品編號, 產品資料.產品, sum(訂貨明細.數量) as 數量
from (訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼) inner join
產品資料 on 訂貨明細.產品編號 = 產品資料.產品編號
group by 訂貨主檔.客戶編號, 產品資料.產品編號, 產品資料.產品 )vw_B
on vw_A.客戶編號=vw_B.客戶編號
```

結果:

結果						
	客戶編號	公司名稱	消費金額	產品編號	產品	數量
1	QUICK	高上補習班	109758.904953003	1	蘋果汁	30
2	SAVEA	大鈺貿易	104361.949920654	1	蘋果汁	130
3	ERNSH	正人資源	104874.978713989	2	牛奶	58
4	QUICK	高上補習班	109758.904953003	2	牛奶	121
5	SAVEA	大鈺貿易	104361.949920654	2	牛奶	203
6	ERNSH	正人資源	104874.978713989	3	蕃茄醬	45
7	QUICK	高上補習班	109758.904953003	3	蕃茄醬	60
8	QUICK	高上補習班	109758.904953003	4	鹽巴	50
9	ERNSH	正人資源	104874.978713989	5	麻油	97
10	SAVEA	大鈺貿易	104361.949920654	5	麻油	70
11	ERNSH	正人資源	104874.978713989	7	海鮮粉	18
12	QUICK	高上補習班	109758.904953003	7	海鮮粉	135
13	SAVEA	大鈺貿易	104361.949920654	7	海鮮粉	45

已成功執行查詢。

## 6. pivot 旋轉應用

### step 1 (產品年季銷售量 小計) vw\_A

```
select 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期) as 年,datepart([q],訂貨主檔.訂單日期) as 季,sum(訂貨明細.數量) as 銷售量  
from 訂貨主檔 inner join (訂貨明細 inner join 產品資料 on 訂貨明細.產品編號 = 產品資料.產品編號) on 訂貨主檔.訂單號碼= 訂貨明細.訂單號碼  
group by 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期),datepart([q],訂貨主檔.訂單日期)
```

結果：

	產品	年	季	銷售量
1	三合一麥片	1996	4	48
2	三合一麥片	1997	1	35
3	三合一麥片	1997	2	106
4	三合一麥片	1997	3	42
5	三合一麥片	1997	4	130
6	三合一麥片	1998	1	105
7	三合一麥片	1998	2	34
8	大甲蟹	1996	3	39
9	大甲蟹	1996	4	46
10	大甲蟹	1997	1	54
11	大甲蟹	1997	2	23
12	大甲蟹	1997	3	144
13	大甲蟹	1997	4	125

已成功執行查詢。

### step 2(產品年季銷售量 季 pivot) #[1],[2]...代表季

SQL:

```
select 產品,年,  
B.[1] as 第一季,  
B.[2] as 第二季,  
B.[3] as 第三季,  
B.[4] as 第四季  
from  
(select 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期) as 年,datepart([q],訂貨主檔.訂單日期) as 季, sum(訂貨明細.數量) as 銷售量  
from 訂貨主檔 inner join (訂貨明細 inner join 產品資料 on 訂貨明細.產品編號 = 產品資料.產品編號) on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼  
group by 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期),datepart([q],訂貨主檔.訂單日期)) as A  
pivot(sum(銷售量) for 季 in([1],[2],[3],[4]))as B
```

結果：

	產品	年	第一季	第二季	第三季	第四季
1	三合一麥片	1996	NULL	NULL	NULL	48
2	大甲蟹	1996	NULL	NULL	39	46
3	小米	1996	NULL	NULL	67	66
4	山楂片	1996	NULL	NULL	192	58
5	干貝	1996	NULL	NULL	1	28
6	牛奶	1996	NULL	NULL	145	81
7	牛肉乾	1996	NULL	NULL	40	NULL
8	台中起司	1996	NULL	NULL	129	68
9	巧克力	1996	NULL	NULL	NULL	71
10	民眾起司	1996	NULL	NULL	12	86
11	玉米餅	1996	NULL	NULL	NULL	30
12	白米	1996	NULL	NULL	26	70
13	白起司	1996	NULL	NULL	46	6

已成功執行查詢。

step 3 (產品年季銷售量 季 pivot) # [1],[2]... 代表季 / iff 判斷是否為 null=0 值

SQL:

```
select 產品,年,
iif(B.[1] is null,0,B.[1]) as 第一季,
iif(B.[2] is null,0,B.[2]) as 第二季,
iif(B.[3] is null,0,B.[3]) as 第三季,
iif(B.[4] is null,0,B.[4]) as 第四季
from
(select 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期) as 年,datepart([q],訂貨主檔.訂單日期) as 季, sum(訂貨明細.數量) as 銷售量
from 訂貨主檔 inner join (訂貨明細 inner join 產品資料 on 訂貨明細.產品編號 = 產品資料.產品編號) on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼
group by 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期),datepart([q],訂貨主檔.訂單日期)) as A
pivot(sum(銷售量) for 季 in([1],[2],[3],[4]))as B
```

結果：

	產品	年	第一季	第二季	第三季	第四季
1	三合一麥片	1996	0	0	0	48
2	大甲蟹	1996	0	0	39	46
3	小米	1996	0	0	67	66
4	山楂片	1996	0	0	192	58
5	干貝	1996	0	0	1	28
6	牛奶	1996	0	0	145	81
7	牛肉乾	1996	0	0	40	0
8	台中起司	1996	0	0	129	68
9	巧克力	1996	0	0	0	71
10	民眾起司	1996	0	0	12	86
11	玉米餅	1996	0	0	0	30
12	白米	1996	0	0	26	70
13	白起司	1996	0	0	46	6

已成功執行查詢。

## 7. pivot 加總(產品年季銷售量 季 pivot 總銷售量)

SOL:

```
select 產品,年,
iif(B.[1] is null,0,B.[1]) as 第一季,
iif(B.[2] is null,0,B.[2]) as 第二季,
iif(B.[3] is null,0,B.[3]) as 第三季,
iif(B.[4] is null,0,B.[4]) as 第四季,
(iif(B.[1] is null,0,B.[1])+iif(B.[2] is null,0,B.[2])+iif(B.[3] is
null,0,B.[3])+iif(B.[4] is null,0,B.[4])) as 總銷售量
from
(select 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期) as 年,datepart([q],訂貨主
檔.訂單日期) as 季, sum(訂貨明細.數量) as 銷售量
from 訂貨主檔 inner join (訂貨明細 inner join 產品資料 on 訂貨明細.產品編號 = 產品
資料.產品編號) on 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼
group by 產品資料.產品,datepart([yyyy],訂貨主檔.訂單日期),datepart([q],訂貨主檔.訂
單日期)) as A
pivot(sum(銷售量) for 季 in([1],[2],[3],[4]))as B
```

結果：

	產品	年	第一季	第二季	第三季	第四季	總銷售量
1	三合一麥片	1996	0	0	0	48	48
2	大甲蟹	1996	0	0	39	46	85
3	小米	1996	0	0	67	66	133
4	山渣片	1996	0	0	192	58	250
5	干貝	1996	0	0	1	28	29
6	牛奶	1996	0	0	145	81	226
7	牛肉乾	1996	0	0	40	0	40
8	台中起司	1996	0	0	129	68	197
9	巧克力	1996	0	0	0	71	71
10	民眾起司	1996	0	0	12	86	98
11	玉米餅	1996	0	0	0	30	30
12	白米	1996	0	0	26	70	96
13	白起司	1996	0	0	46	6	52

已成功執行查詢。

## ※小考

### 1. 每個員工每年每季的營業額(取整數)

SQL:

```
select 姓名,年,
iif(B.[1] is null,0,B.[1]) as 第一季,
iif(B.[2] is null,0,B.[2]) as 第二季,
iif(B.[3] is null,0,B.[3]) as 第三季,
iif(B.[4] is null,0,B.[4]) as 第四季
from
(select 員工.員工編號,姓名,DATEPART(YYYY,訂貨主檔.訂單日期)as 年,DATEPART(q,訂貨主檔.訂單日期)as 季,
Round(sum(訂貨明細.單價*數量*(1-折扣)),0) as 營業額
from 員工
inner join 訂貨主檔 on 員工.員工編號=訂貨主檔.員工編號 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
group by 員工.員工編號,姓名,DATEPART(YYYY,訂貨主檔.訂單日期),DATEPART(q,訂貨主檔.訂單日期)) as A
pivot(sum(營業額) for 季 in([1],[2],[3],[4]))as B
```

結果:

	姓名	年	第一季	第二季	第三季	第四季
1	張瑾雯	1996	0	0	13303	21943
2	張瑾雯	1997	14402	14824	32077	31845
3	張瑾雯	1998	44090	19105	0	0
4	陳季暄	1996	0	0	5941	15816
5	陳季暄	1997	7489	24374	17309	21272
6	陳季暄	1998	41416	32920	0	0
7	趙飛燕	1996	0	0	8177	10047
8	趙飛燕	1997	28793	33902	10469	34862
9	趙飛燕	1998	63605	12024	0	0
10	林美麗	1996	0	0	18816	31130
11	林美麗	1997	41089	24474	29948	33299
12	林美麗	1998	38187	15948	0	0
13	劉天王	1996	0	0	2891	15325

已成功執行查詢。

# 10/31 第八週

## ●SQL 演練(DDL: create、alter、drop / DCL: grant、revoke)

### 1. 新增資料庫 MyDB

SQL :

```
create database MyDB  
go
```

結果：



### 2. 在資料庫(MyDB)，新增資料表 customer

SQL :

```
USE [MyDB]  
Create table customer  
(  
CID nvarchar(5) not null,  
First_name nvarchar(50),  
Last_name nvarchar(50),  
Address nvarchar(50),  
City nvarchar(50),  
Country nvarchar(25),  
BDate datetime,  
Primary key (CID)  
);  
GO --GO 執行
```

結果：

A screenshot of the SQL Server Management Studio Object Explorer. It shows a table named 'customer' under the 'dbo' schema of the 'MyDB' database. The table has seven columns: CID, First\_name, Last\_name, Address, City, Country, and BDate. The 'CID' column is identified as the primary key. A red rectangular box highlights the entire table structure in the list.

資料行名稱	資料類型	允許 Null
CID	nvarchar(5)	<input type="checkbox"/>
First_name	nvarchar(50)	<input checked="" type="checkbox"/>
Last_name	nvarchar(50)	<input checked="" type="checkbox"/>
Address	nvarchar(50)	<input checked="" type="checkbox"/>
City	nvarchar(50)	<input checked="" type="checkbox"/>
Country	nvarchar(25)	<input checked="" type="checkbox"/>
BDate	datetime	<input checked="" type="checkbox"/>

### 3. 在資料庫(MyDB)，新增資料表 students

SQL :

```
USE [MyDB]
```

```
Create table students(
```

```
    SID int NOT NULL primary key, Sname nvarchar(255), STel nvarchar(255), SAdress  
    nvarchar(255)  
)
```

結果：

資料行名稱	資料類型	允許 Null
SID	int	<input type="checkbox"/>
Sname	nvarchar(255)	<input checked="" type="checkbox"/>
STel	nvarchar(255)	<input checked="" type="checkbox"/>
SAdress	nvarchar(255)	<input checked="" type="checkbox"/>

### 4. 在資料表(students)，新增欄位(SEmail)，資料型態 nvarchar(255)

SQL :

```
USE [MyDB]
```

```
ALTER TABLE students
```

```
ADD SEmail nvarchar(50)
```

```
go
```

```
USE [MyDB] – nvarchar(50)改 nvarchar(255)
```

```
ALTER TABLE students
```

```
ADD COLUMN SEmail nvarchar(255)
```

```
go
```

結果：

SID	int	<input type="checkbox"/>
Sname	nvarchar(255)	<input checked="" type="checkbox"/>
STel	nvarchar(255)	<input checked="" type="checkbox"/>
SAdress	nvarchar(255)	<input checked="" type="checkbox"/>
SEmail	nvarchar(255)	<input checked="" type="checkbox"/>

## 5. 在資料表(students)，刪除欄位(SEmail1)

SQL :

USE [MyDB]

ALTER TABLE students

DROP COLUMN SEmail1

go

## 6. 刪除資料表、刪除資料庫

SQL :

USE [MyDB]

DROP TABLE students

USE [MyDB]

DROP DATABASE MyDB

GO

## 7. 授權使用者(readuser)，具有權限(select) / 使用者(readuser)，移除權限

SQL :

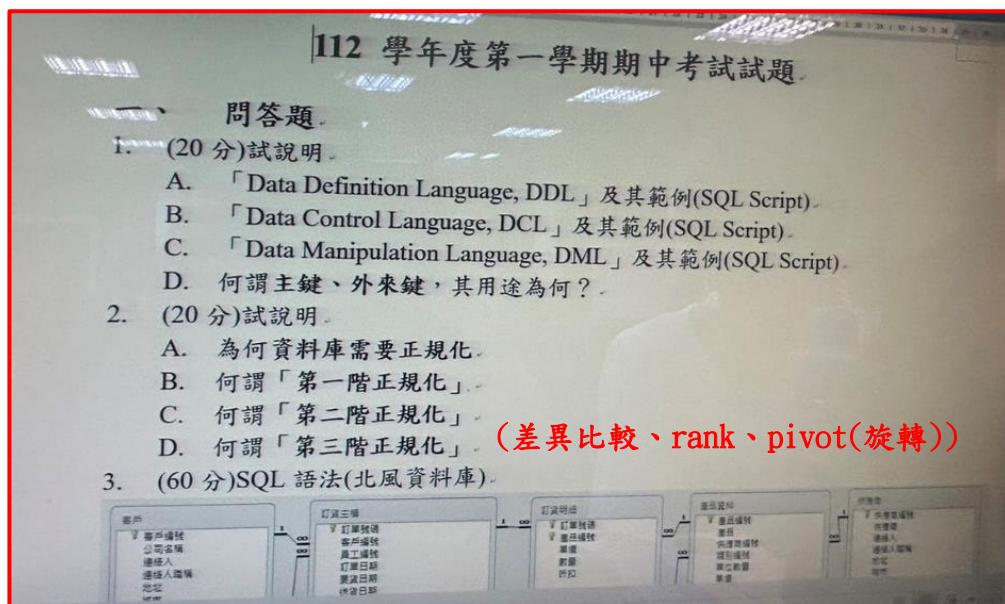
USE [MyDB]

GRANT SELECT to readuser

USE [MyDB]

REVOKE SELECT to readuser

## ●公佈期中考考題



## ※小考

1.

### A. 「Data Definition Language, DDL」及其範例(SQL Script)

定義資料庫結構與資料物件的語言

例：USE [master] CREATE DATABASE Northwind

### B. 「Data Control Language, DCL」及其範例(SQL Script)

設定使用權限與安全

例：USE [MyDB] GRANT SELECT to readuser

### C. 「Data Manipulation Language, DML」及其範例(SQL Script)

資料的存取：查詢(select)、新增(insert)、刪除(delete)、修改(update)

例：

```
Select 產品資料. 產品編號, 產品資料. 產品, sum(訂貨明細. 數量) as 銷貨量
```

```
From 產品資料 inner join 訂貨明細 on 訂貨明細. 產品編號=產品資料. 產品編號
```

```
Group by 產品資料. 產品編號, 產品資料. 產品
```

```
Order by 產品資料. 產品編號
```

### D. 何謂主鍵、外來鍵(外來的主鍵)，其用途為何？

主鍵是指用來識別記錄的唯一性，它不可以重複及空值(NULL)。資料表裡面的唯一性

外來鍵是指用來建立資料表之間的關係，其外鍵內含值必須要與另一個資料表的主鍵相同。

# 11/7 第九週 期中考試

## ●題目及答案

### 1. 試說明

#### a. 「Data Definition Language, DDL」及其範例(SQL Script)

定義資料庫結構與資料物件的語言 (create, alter, drop)

ex: USE [master] CREATE DATABASE Northwind

#### b. 「Data Control Language, DCL」及其範例(SQL Script)

設定使用權限與安全 (grant, revoke, deny, commit, rollback)

ex: USE [MyDB] GRANT SELECT to readuser

#### c. 「Data Manipulation Language, DML」及其範例(SQL Script)

資料的存取：查詢(select)、新增(insert)、刪除(delete)、修改(update)

ex: Select \* from 客戶

#### d. 何謂主鍵(primary key)、外來鍵(foreign key)(外來的主鍵)，其用途為何？

主鍵 是指用來識別記錄的唯一性，它不可以重複及空值(Null)。(資料表裡面的唯一性)

外來鍵 是指用來建立資料表之間的關係，其外鍵內含值必須要與另一個資料表的主鍵相同。

### 2. 試說明

#### a. 為何資料庫需要正規化

1. 降低資料重複性 (data redundancy)

2. 避免資料更新異常 (anomalies)

#### b. 何謂「第一階正規化」(1NF)

資料表中有主鍵，而其他所有的欄位都相依於主鍵，每個欄位中都只儲存單一值。

一個關聯表 R 的每個屬性都是單元值。

具體方式：將實體之中一對多的資料予以分割。

#### c. 何謂「第二階正規化」(2NF：確立實體的功能相依)

功能相依(functional dependency)是指實體與實體間的相依關係。

若某個實體中有兩個屬性 x 及 y，當 x 屬性值可以推導出 y 屬性值，稱功能相依，表達 y 屬性值相依於 x 屬性值。

若一關聯 R，其屬性 Y 功能相依於屬性 X，記做 R. X→R. Y。

符合 1NF 的格式

各欄位與主體間沒有“不分相依”的關係

具體方式：決定主鍵，保持 1NF 切割的各實體之間關聯性。

#### d. 何謂「第三階正規化」(3NF：消除遞移相依的情況)

遞移相依是指在一個實體中，所有屬性應該相依於主鍵，也就是 1NF 的功能相依主鍵 R. X→R. Y。

如果又存在某一屬性可以決定其餘屬性的值，就稱為遞移相依。

EX: R. X→R. Y 且 R. Y→R. Z 則 R. X→R. Z 成立

符合 2NF 的格式

非主鍵的各欄位間沒有“遞移相依”的關係

具體方式：自我相依的屬性必須再分割，並維持 2NF 的關聯性

### 3. SQL

#### (1) 哪一個產品沒有銷售紀錄(產品編號、產品)

```
select vw_a.產品編號, vw_a.產品  
from  
(select distinct 產品編號, 產品  
from 產品資料) vw_a  
left join  
(select distinct 訂貨明細.產品編號, 產品  
from 訂貨明細 inner join 產品資料 on 訂貨明細.產品編號=產品資料.產品編號) vw_b  
on vw_a.產品編號=vw_b.產品編號  
where vw_b.產品編號 is null
```

#### (2) 產品城市年銷售量排名(產品編號、產品、城市、年、銷售量、總排名、年排名)

```
select 產品編號, 產品, 送貨城市, 年, 銷售量,  
rank () over (partition by 年 order by 銷售量 desc) as 年排名,  
rank () over (order by 銷售量 desc) as 總排名  
from(  
select distinct 產品資料.產品編號, 產品資料.產品, 訂貨主檔.送貨城市, datepart(yyyy, 訂  
貨主檔.訂單日期) as 年, sum(訂貨明細.數量) as 銷售量  
from 產品資料 inner join 訂貨明細 on 訂貨明細.產品編號=產品資料.產品編號 inner join  
訂貨主檔 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
group by 產品資料.產品編號, 產品資料.產品, 訂貨主檔.送貨城市, datepart(yyyy, 訂貨主檔.  
訂單日期)) vw_a
```

#### (3) 客戶年季消費金額(客戶編號、客戶、年、季、消費金額)，季 pivot

```
select 客戶編號, 公司名稱, 年,  
B.[1] AS 第一季,  
B.[2] AS 第二季,  
B.[3] AS 第三季,  
B.[4] AS 第四季  
from (  
select distinct 客戶.客戶編號, 客戶.公司名稱, datepart(yyyy, 訂貨主檔.訂單日期) as 年,  
datepart(q, 訂貨主檔.訂單日期) as 季, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂  
貨明細.折扣)), 0) as 消費金額  
from 客戶 inner join 訂貨主檔 on 客戶.客戶編號=訂貨主檔.客戶編號 inner join 訂貨明  
細 on 訂貨明細.訂單號碼=訂貨主檔.訂單號碼  
group by 客戶.客戶編號, 客戶.公司名稱, datepart(yyyy, 訂貨主檔.訂單日期), datepart(q,  
訂貨主檔.訂單日期)) vw_a  
pivot(sum(消費金額) for 季 in ([1],[2],[3],[4])) as B
```

# 11/14 第十週

## ●課堂重點

### 1. 核對期中考成績

### 2. 檢討期中考 SQL

第三題另一種寫法

(3) 客戶年季消費金額(客戶編號、客戶、年、季、消費金額)，季 pivot

```
select 客戶.客戶編號, 客戶.客戶, 年.B.[台北市], B.[台中市], B.[高雄市],  
iif(B.[台北市] is null, 0, B.[台北市])+  
iif(B.[台中市] is null, 0, B.[台中市])+  
iif(B.[高雄市] is null, 0, B.[高雄市]) as 消費金額  
from  
(  
select 客戶.客戶編號, 客戶.公司名稱 as 客戶, datepart(yyyy, 訂貨主檔.訂單日期)as 年, 訂  
貨主檔.送貨城市, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)), 0) as  
消費金額  
from 客戶 inner join 訂貨主檔 on 客戶.客戶編號=訂貨主檔.客戶編號 inner join 訂貨明  
細 on 訂貨明細.訂單號碼=訂貨主檔.訂單號碼  
group by 客戶.客戶編號, 客戶.公司名稱, datepart(yyyy, 訂貨主檔.訂單日期), 訂貨主檔.送貨  
城市  
)vw_a  
pivot(sum(消費金額) for 送貨城市 in ([台北市], [台中市], [高雄市])) as B
```

※正規化流程說明(縱向切割-物以類聚，橫向刪除-重複資料)

**傳統資料庫正規化的案例解說 (2/4)**

主鍵(PK)

專家

專家編號	專家名稱	員工編號	員工姓名	工作類別	類別薪資	工作時數
P001	選課系統	E001	小叮噹	分析師	500	5
		E002	靜音	資料庫設計師	400	6

專家員工工作薪資表

專家	員工	工作類別	類別薪資	工作時數
P001	E001	分析師	500	5
P001	E002	資料庫設計師	400	6

部分相依

處理「相依性」欄位

- 專家資料
- 專家名稱
- 員工資料
- 工作類別
- 類別薪資
- 工作時數

處理「剩餘」欄位

- 工作時數

執行成果

運用Excel排序法

- 點選特定欄位
- 點選排序功能
- 觀察欄位之間的相依性
- 縱向切割(物以類聚)
- 橫向刪除(重複資料)

序號	姓名	年齡	性別
1	張三	25	男
2	李四	28	女
3	王五	30	男
4	陳六	26	女
5	林七	29	男
6	吳八	31	女
7	黃九	27	男
8	鄭十	32	女
9	范十一	24	男
10	周十二	26	女
11	洪十三	29	男
12	許十四	30	女
13	鄧十五	25	男
14	黃十六	27	女
15	林十七	31	男
16	陳十八	26	女
17	周十九	29	男
18	范二十	32	女
19	洪二十一	24	男
20	許二十二	26	女
21	鄧二十三	29	男
22	黃二十四	25	女
23	林二十五	31	男
24	陳二十六	26	女
25	周二十七	29	男
26	范二十八	32	女
27	洪二十九	24	男
28	許三十	26	女
29	鄧三十一	29	男
30	黃三十二	25	女
31	林三十三	31	男
32	陳三十四	26	女
33	周三十五	29	男
34	范三十六	32	女
35	洪三十七	24	男
36	許三十八	26	女
37	鄧三十九	29	男
38	黃四十	25	女
39	林四十一	31	男
40	陳四十二	26	女
41	周四十三	29	男
42	范四十四	32	女
43	洪四十五	24	男
44	許四十六	26	女
45	鄧四十七	29	男
46	黃四十八	25	女
47	林四十九	31	男
48	陳五十	26	女
49	周五十一	29	男
50	范五十二	32	女
51	洪五十三	24	男
52	許五十四	26	女
53	鄧五十五	29	男
54	黃五十六	25	女
55	林五十七	31	男
56	陳五十八	26	女
57	周五十九	29	男
58	范六十	32	女
59	洪六十一	24	男
60	許六十二	26	女
61	鄧六十三	29	男
62	黃六十四	25	女
63	林六十五	31	男
64	陳六十六	26	女
65	周六十七	29	男
66	范六十八	32	女
67	洪六十九	24	男
68	許七十	26	女
69	鄧七十一	29	男
70	黃七十二	25	女
71	林七十三	31	男
72	陳七十四	26	女
73	周七十五	29	男
74	范七十六	32	女
75	洪七十七	24	男
76	許七十八	26	女
77	鄧七十九	29	男
78	黃八十	25	女
79	林八十一	31	男
80	陳八十二	26	女
81	周八十三	29	男
82	范八十四	32	女
83	洪八十五	24	男
84	許八十六	26	女
85	鄧八十七	29	男
86	黃八十八	25	女
87	林八十九	31	男
88	陳九十	26	女
89	周九十一	29	男
90	范九十二	32	女
91	洪九十三	24	男
92	許九十四	26	女
93	鄧九十五	29	男
94	黃九十六	25	女
95	林九十七	31	男
96	陳九十八	26	女
97	周九十九	29	男
98	范一百	32	女
99	洪一百一	24	男
100	許一百二	26	女
101	鄧一百三	29	男
102	黃一百四	25	女
103	林一百五	31	男
104	陳一百六	26	女
105	周一百七	29	男
106	范一百八	32	女
107	洪一百九	24	男
108	許一百十	26	女
109	鄧一百十一	29	男
110	黃一百十二	25	女
111	林一百十三	31	男
112	陳一百十四	26	女
113	周一百十五	29	男
114	范一百十六	32	女
115	洪一百十七	24	男
116	許一百十八	26	女
117	鄧一百十九	29	男
118	黃一百二十	25	女
119	林一百二十一	31	男
120	陳一百二十二	26	女
121	周一百二十三	29	男
122	范一百二十四	32	女
123	洪一百二十五	24	男
124	許一百二十六	26	女
125	鄧一百二十七	29	男
126	黃一百二十八	25	女
127	林一百二十九	31	男
128	陳一百三十	26	女
129	周一百三十一	29	男
130	范一百三十二	32	女
131	洪一百三十三	24	男
132	許一百三十四	26	女
133	鄧一百三十五	29	男
134	黃一百三十六	25	女
135	林一百三十七	31	男
136	陳一百三十八	26	女
137	周一百三十九	29	男
138	范一百四十	32	女
139	洪一百四十一	24	男
140	許一百四十二	26	女
141	鄧一百四十三	29	男
142	黃一百四十四	25	女
143	林一百四十五	31	男
144	陳一百四十六	26	女
145	周一百四十七	29	男
146	范一百四十八	32	女
147	洪一百四十九	24	男
148	許一百五十	26	女
149	鄧一百五十一	29	男
150	黃一百五十二	25	女
151	林一百五十三	31	男
152	陳一百五十四	26	女
153	周一百五十五	29	男
154	范一百五十六	32	女
155	洪一百五十七	24	男
156	許一百五十八	26	女
157	鄧一百五十九	29	男
158	黃一百六十	25	女
159	林一百七十一	31	男
160	陳一百七十二	26	女
161	周一百七十三	29	男
162	范一百七十四	32	女
163	洪一百七十五	24	男
164	許一百七十六	26	女
165	鄧一百七十七	29	男
166	黃一百七十八	25	女
167	林一百七十九	31	男
168	陳一百八十	26	女
169	周一百八十一	29	男
170	范一百八十二	32	女
171	洪一百八十三	24	男
172	許一百八十四	26	女
173	鄧一百八十五	29	男
174	黃一百八十六	25	女
175	林一百八十七	31	男
176	陳一百八十八	26	女
177	周一百八十九	29	男
178	范一百九十	32	女
179	洪一百九十一	24	男
180	許一百九十二	26	女
181	鄧一百九十三	29	男
182	黃一百九十四	25	女
183	林一百九十五	31	男
184	陳一百九十六	26	女
185	周一百九十七	29	男
186	范一百九十八	32	女
187	洪一百九十九	24	男
188	許一百二十	26	女
189	鄧一百二十一	29	男
190	黃一百二十二	25	女
191	林一百二十三	31	男
192	陳一百二十四	26	女
193	周一百二十五	29	男
194	范一百二十六	32	女
195	洪一百二十七	24	男
196	許一百二十八	26	女
197	鄧一百二十九	29	男
198	黃一百三十	25	女
199	林一百三十一	31	男
200	陳一百三十二	26	女
201	周一百三十三	29	男
202	范一百三十四	32	女
203	洪一百三十五	24	男
204	許一百三十六	26	女
205	鄧一百三十七	29	男
206	黃一百三十八	25	女
207	林一百三十九	31	男
208	陳一百四十	26	女
209	周一百四十一	29	男
210	范一百四十二	32	女
211	洪一百四十三	24	男
212	許一百四十四	26	女
213	鄧一百四十五	29	男
214	黃一百四十六	25	女
215	林一百四十七	31	男
216	陳一百四十八	26	女
217	周一百四十九	29	男
218	范一百五十	32	女
219	洪一百五十一	24	男
220	許一百五十二	26	女
221	鄧一百五十三	29	男
222	黃一百五十四	25	女
223	林一百五十五	31	男
224	陳一百五十六	26	女
225	周一百五十七	29	男
226	范一百五十八	32	女
227	洪一百五十九	24	男
228	許一百六十	26	女
229	鄧一百六十一	29	男
230	黃一百六十二	25	女
231	林一百六十三	31	男
232	陳一百六十四	26	女
233	周一百六十五	29	男
234	范一百六十六	32	女
235	洪一百六十七	24	男
236	許一百六十八	26	女
237	鄧一百六十九	29	男
238	黃一百七十	25	女
239	林一百七十一	31	男
240	陳一百七十二	26	女
241	周一百七十三	29	男
242	范一百七十四	32	女
243	洪一百七十五	24	男
244	許一百七十六	26	女
245	鄧一百七十七	29	男
246	黃一百七十八	25	女
247	林一百七十九	31	男
248	陳一百八十	26	女
249	周一百八十一	29	男
250	范一百八十二	32	女
251	洪一百八十三	24	男
252	許一百八十四	26	女
253	鄧一百八十五	29	男
254	黃一百八十六	25	女
255	林一百八十七	31	男
256	陳一百八十八	26	女
257	周一百八十九	29	男
258	范一百九十	32	女
259	洪一百九十一	24	男
260	許一百九十二	26	女
261	鄧一百九十三	29	男
262	黃一百九十四	25	女
263	林一百九十五	31	男
264	陳一百九十六	26	女
265	周一百九十七	29	男
266	范一百九十八	32	女
267	洪一百九十九	24	男
268	許一百二十	26	女
269	鄧一百二十一	29	男
270	黃一百二十二	25	女
271	林一百二十三	31	男
272	陳一百二十四	26	女
273	周一百二十五	29	男
274	范一百二十六	32	女
275	洪一百二十七	24	男
276	許一百二十八	26	女
277	鄧一百二十九	29	男
278	黃一百三十	25	女
279	林一百三十一	31	男
280	陳一百三十二	26	女
281	周一百三十三	29	男
282	范一百三十四	32	女
283	洪一百三十五	24	男
284	許一百三十六	26	女
285	鄧一百三十七	29	男
286	黃一百三十八	25	女
287	林一百三十九	31	男
288	陳一百四十	26	女
289	周一百四十一	29	男
290	范一百四十二	32	女
291	洪一百四十三	24	男
292	許一百四十四	26	女
293	鄧一百四十五	29	男
294	黃一百四十六	25	女
295	林一百四十七	31	男
296	陳一百四十八		

### 3. 期末報告說明

#### ●課程資料匯入 SQL 操作(仿 POS 發票系統(期末報告))

※ Excel 操作(縱向切割，橫向刪除) 5(切割資料表)+1(原始資料)=6 個資料表

##### 1. 「學生」資料表

	A	B	C	D	E	F	G	H	I	J	K	L
1	學號	姓名	班級	座號	課程編號	課程名稱	教室	成績	教師編號	授課老師	職級	
2	F100001	張安麗	FM4A		1 FM2A03	系統分析		2302	80	107090	翁政雄	副教授
3	F100001	張安麗	FM4A		1 FM1A01	程式語言		2303	90	101001	李桂春	助理教授
4	F100001	張安麗	FM4A		1 FM2A02	資料庫		2301	70	101002	曾建銘	助理教授
5	F100001	張安麗	FM4A		1 FM3A01	資料探勘		2304	70	107090	翁政雄	副教授
6	F100002	許文玲	FM4A		2 FM2A03	系統分析		2302	80	107090	翁政雄	副教授
7	F100002	許文玲	FM4A		2 FM1A01	程式語言		2303				
8	F100002	許文玲	FM4A		2 FM2A02	資料庫		2301				
9	F100002	許文玲	FM4A		2 FM3A01	資料探勘		2304				
10	F100103	蚵藝又	FM4B		1 FM2A03	系統分析		2302				
11	F100103	蚵藝又	FM4B		1 FM1A01	程式語言		2303				
12	F100103	蚵藝又	FM4B		1 FM2A02	資料庫		2314				
13												

	A	B	C	D
1	學號	姓名	班級	座號
2	F100001	張安麗	FM4A	1
3	F100002	許文玲	FM4A	2
4	F100103	蚵藝又	FM4B	1
5				
6				

##### 2. 「課程」資料表

	A	B	C	D	E	F	G	H
學號	課程編號	課程名稱	教室	成績	教師編號	授課老師	職級	
F100001	FM1A01	程式語言		2303	90	101001	李桂春	助理教授
F100002	FM1A01	程式語言		2303	73	101001	李桂春	助理教授
F100103	FM1A01	程式語言		2303	90	101001	李桂春	助理教授
F100001	FM2A02	資料庫		2301	70	101002	曾建銘	
F100002	FM2A02	資料庫		2301	70	101002	曾建銘	
F100103	FM2A02	資料庫		2314	70	107090	翁政雄	
F100001	FM2A03	系統分析		2302	80	107090	翁政雄	
F100002	FM2A03	系統分析		2302	80	107090	翁政雄	
F100103	FM2A03	系統分析		2302	80	107090	翁政雄	
F100001	FM3A01	資料探勘		2304	70	107090	翁政雄	
F100002	FM3A01	資料探勘		2304	74	107090	翁政雄	

	A	B
1	課程編號	課程名稱
2	FM1A01	程式語言
3	FM2A02	資料庫
4	FM2A03	系統分析
5	FM3A01	資料探勘
6		

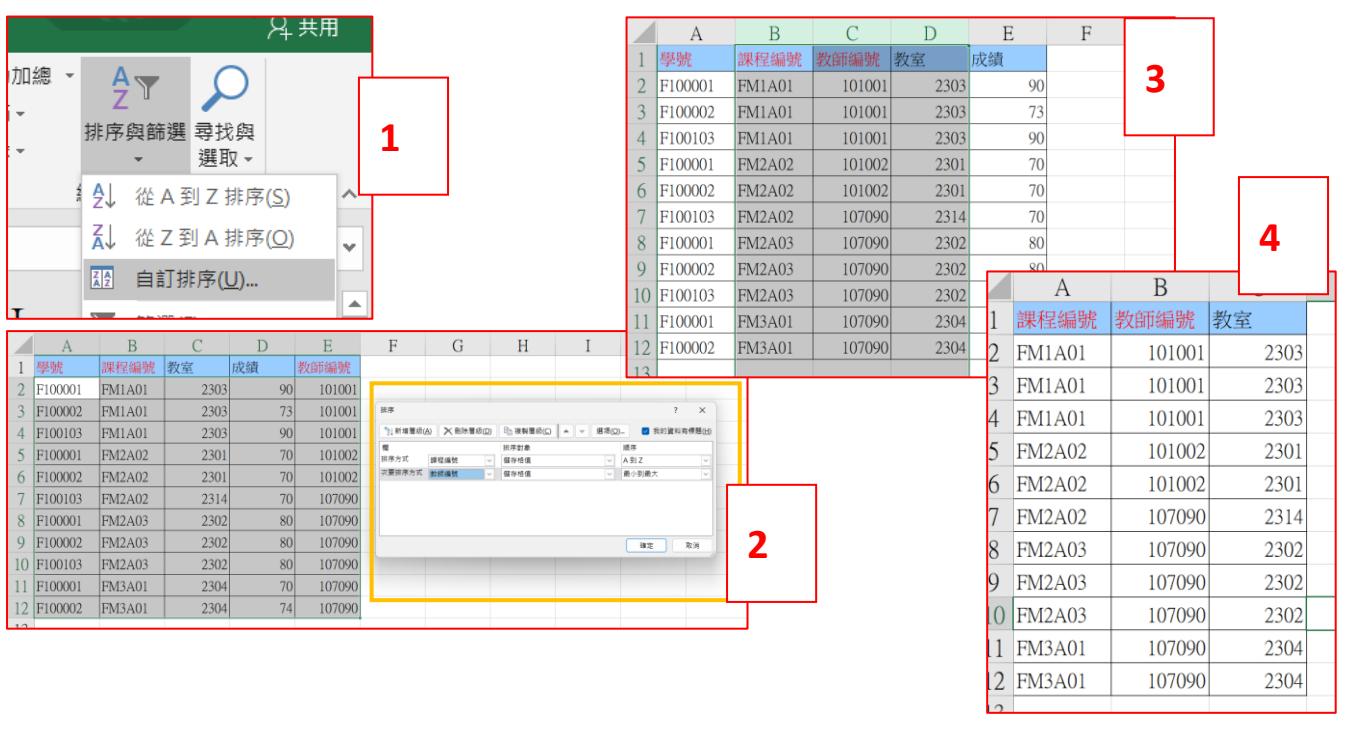
### 3. 「教師」資料表

	A	B	C	D	E	F	G
1	學號	課程編號	教室	成績	教師編號	授課老師	職級
2	F100001	FM1A01	2303	90	101001	李桂春	助理教授
3	F100002	FM1A01	2303	73	101001	李桂春	助理教授
4	F100103	FM1A01	2303	90	101001	李桂春	助理教授
5	F100001	FM2A02	2301	70	101002	曾建銘	助理教授
6	F100002	FM2A02	2301	70	101002	曾建銘	助理教授
7	F100103	FM2A02	2314	70	107090	翁政雄	副教授
8	F100001	FM2A03	2302	80	107090	翁政雄	副教授
9	F100002	FM2A03					
10	F100103	FM2A03					
11	F100001	FM3A01					
12	F100002	FM3A01					
13							

	A	B	C
1	教師編號	授課老師	職級
2	101001	李桂春	助理教授
3	101002	曾建銘	助理教授
4	107090	翁政雄	副教授
5			

### 4. 「課程教室」資料表



The screenshot shows the 'Course Classroom' data table with several annotations:

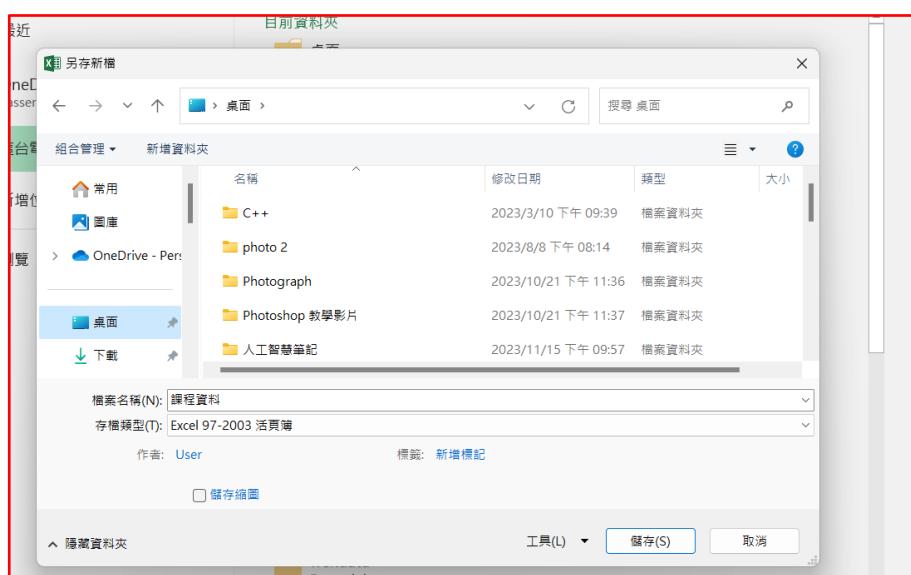
- Number 1:** Points to the 'Sort & Filter' button in the ribbon.
- Number 2:** Points to the 'Sort' dialog box, which is open and set to sort by 'Teacher ID' (教師編號) in ascending order (A to Z).
- Number 3:** Points to the 'Sort & Filter' button in the ribbon again, located above the first one.
- Number 4:** Points to the 'Sort' dialog box, which is also open and set to sort by 'Teacher ID' (教師編號) in ascending order (A to Z).

	A	B	C	D	E	F
1	學號	課程編號	教室	成績	教師編號	
2	F100001	FM1A01	2303	90	101001	
3	F100002	FM1A01	2303	73	101001	
4	F100103	FM1A01	2303	90	101001	
5	F100001	FM2A02	2301	70	101002	
6	F100002	FM2A02	2301	70	101002	
7	F100103	FM2A02	2314	70	107090	
8	F100001	FM2A03	2302	80	107090	
9	F100002	FM2A03	2302	80	107090	
10	F100103	FM2A03	2302	80	107090	
11	F100001	FM3A01	2304	70	107090	
12	F100002	FM3A01	2304	74	107090	
13						

## 5. 「學生成績」資料表(刪除其他欄位後)

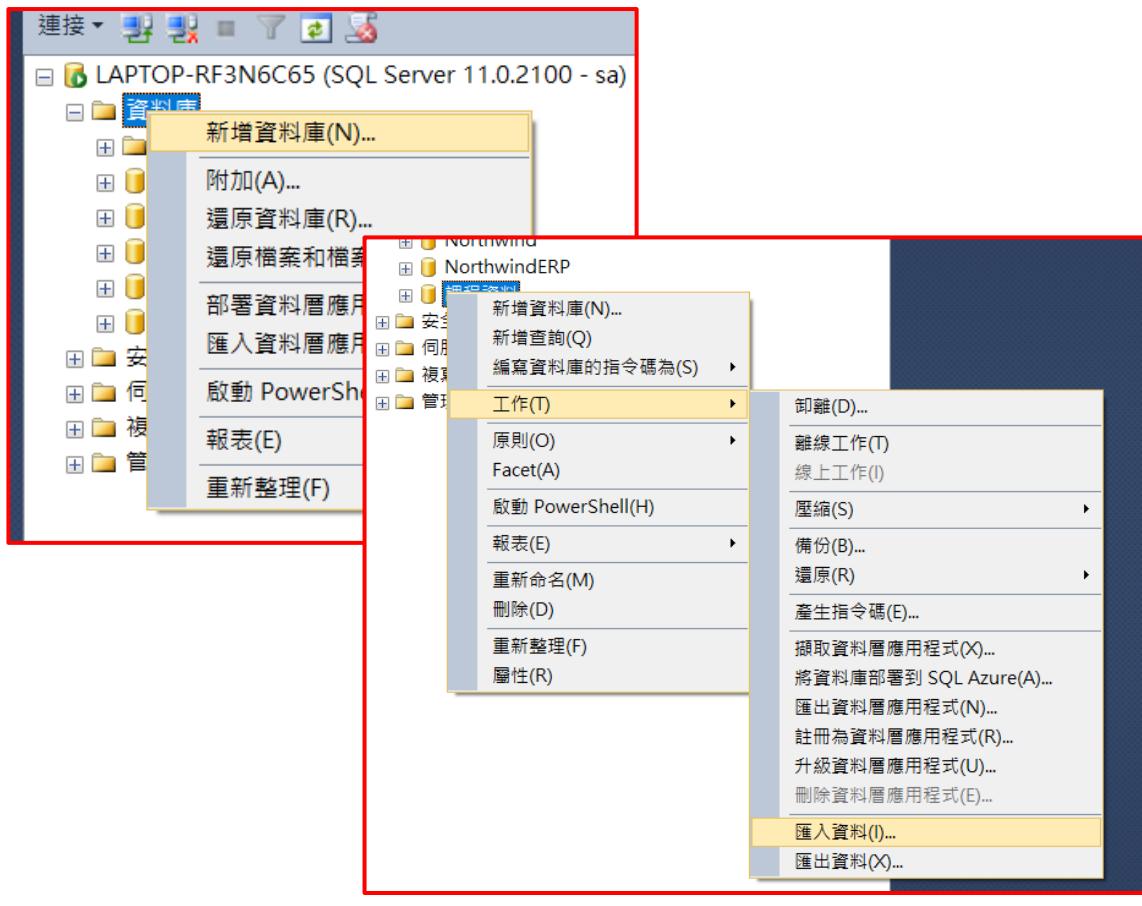
	A	B	C	D	
1	學號	課程編號	教師編號	成績	
2	F100001	FM1A01	101001	90	
3	F100002	FM1A01	101001	73	
4	F100103	FM1A01	101001	90	
5	F100001	FM2A02	101002	70	
6	F100002	FM2A02	101002	70	
7	F100103	FM2A02	107090	70	
8	F100001	FM2A03	107090	80	
9	F100002	FM2A03	107090	80	
10	F100103	FM2A03	107090	80	
11	F100001	FM3A01	107090	70	
12	F100002	FM3A01	107090	74	
13					

## 6. 儲存「存檔類型」(97-2003 活頁簿)

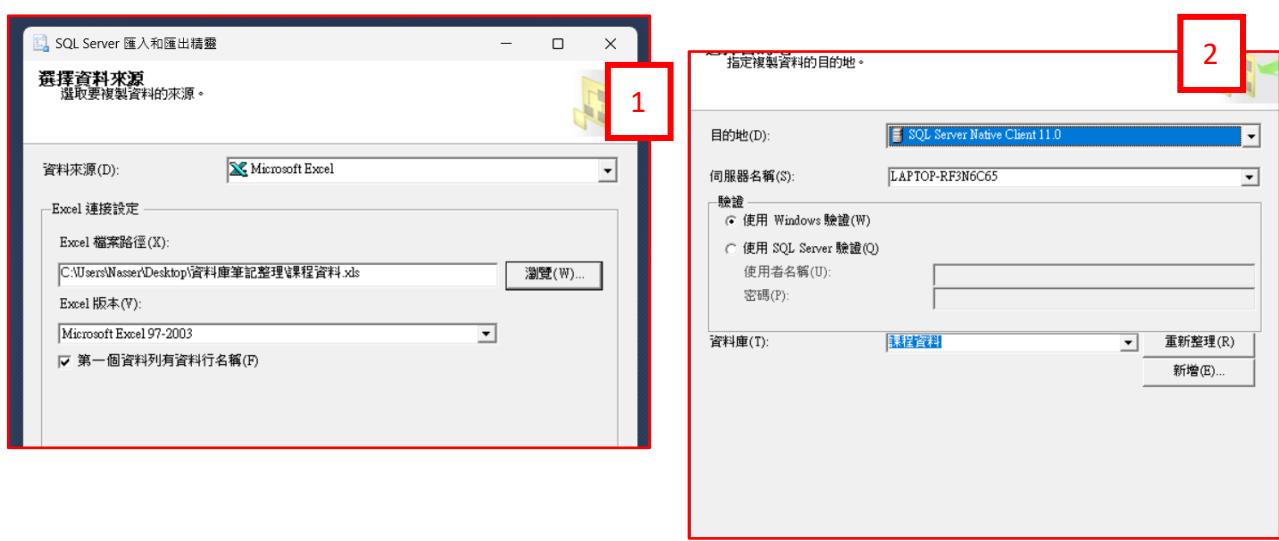


## ●SQL 匯入 課程資料表

### 1. 新增資料庫(課程資料)、匯入資料



### 2. 匯入資料



### 3. 匯入資料

**指定資料表複製或查詢**  
指定是否要從資料來源複製一個或多個的資料表/檢視，或複製查詢的結果。

**從一個或多個資料表或檢視表複製資料(C)**  
使用這個選項可以複製來源資料庫中現有資料表或檢視表的所有資料。

**寫入查詢來指定要傳送的資料(W)**  
使用這個選項可以寫入 SQL 查詢來操作或限制複製作業的來源資料。

**SQL Server 匯入和匯出精靈**  
請選擇一個或多個要複製的資料表和檢視。

**資料表和檢視(T):**

來源: C:\Users\Nasser\Desktop\資料庫筆...	目的地: LAPTOP-RF3N6C65
<input checked="" type="checkbox"/> '原始資料\$'	[dbo].[原始資料]
<input checked="" type="checkbox"/> '教師\$'	[dbo].[教師]
<input checked="" type="checkbox"/> '課程\$'	[dbo].[課程]
<input checked="" type="checkbox"/> '課程教室\$'	[dbo].[課程教室]
<input checked="" type="checkbox"/> '學生\$'	[dbo].[學生]
<input checked="" type="checkbox"/> '學生成績\$'	[dbo].[學生成績]

**選取來源資料表和檢視**  
請選擇一個或多個要複製的資料表和檢視。

來源: C:\Users\Nasser\Desktop\資料庫筆...	目的地: LAPTOP-RF3N6C65
<input checked="" type="checkbox"/> '原始資料\$'	[dbo].[原始資料]
<input checked="" type="checkbox"/> '教師\$'	[dbo].[教師]
<input checked="" type="checkbox"/> '課程\$'	[dbo].[課程]
<input checked="" type="checkbox"/> '課程教室\$'	[dbo].[課程教室]
<input checked="" type="checkbox"/> '學生\$'	[dbo].[學生]
<input checked="" type="checkbox"/> '學生成績\$'	[dbo].[學生成績]

### 4. 匯入成功

**SQL Server 匯入和匯出精靈**

**已成功執行**

動作	狀態	訊息
正在初始化資料流程工作	成功	
正在初始化連接	成功	
正在設定 SQL 命令	成功	
正在設定來源連接	成功	
正在設定目的地連接	成功	
正在驗證	警告	警告 0x80049304: ...
準備執行	成功	
執行前	成功	
正在執行	成功	
① 正在複製到 [dbo].[原始資料]	成功	已傳送 11 個資料列
① 正在複製到 [dbo].[教師]	成功	已傳送 3 個資料列
① 正在複製到 [dbo].[課程]	成功	已傳送 4 個資料列
① 正在複製到 [dbo].[課程教室]	成功	已傳送 11 個資料列
① 正在複製到 [dbo].[學生]	成功	已傳送 3 個資料列
① 正在複製到 [dbo].[學生成績]	成功	已傳送 11 個資料列
執行後	成功	

**停止(S)** **報表(R)** **關閉**

# 11/21 第十一週

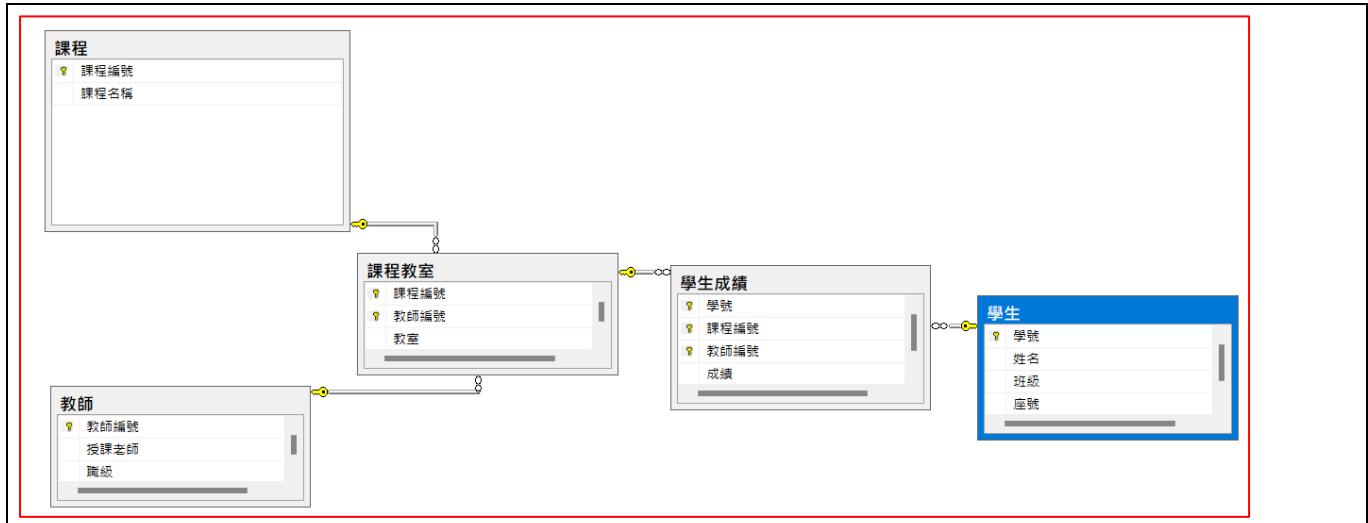
## ●課程資料表→設計，設定主鍵

1 教師 資料表	2 課程 資料表																					
<table border="1"><thead><tr><th>資料行名稱</th><th>資料類型</th><th>允許 Null</th></tr></thead><tbody><tr><td>教師編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>授課老師</td><td>nvarchar(255)</td><td><input checked="" type="checkbox"/></td></tr><tr><td>職級</td><td>nvarchar(255)</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>	資料行名稱	資料類型	允許 Null	教師編號	nvarchar(50)	<input type="checkbox"/>	授課老師	nvarchar(255)	<input checked="" type="checkbox"/>	職級	nvarchar(255)	<input checked="" type="checkbox"/>	<table border="1"><thead><tr><th>資料行名稱</th><th>資料類型</th><th>允許 Null</th></tr></thead><tbody><tr><td>課程編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>課程名稱</td><td>nvarchar(255)</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>	資料行名稱	資料類型	允許 Null	課程編號	nvarchar(50)	<input type="checkbox"/>	課程名稱	nvarchar(255)	<input checked="" type="checkbox"/>
資料行名稱	資料類型	允許 Null																				
教師編號	nvarchar(50)	<input type="checkbox"/>																				
授課老師	nvarchar(255)	<input checked="" type="checkbox"/>																				
職級	nvarchar(255)	<input checked="" type="checkbox"/>																				
資料行名稱	資料類型	允許 Null																				
課程編號	nvarchar(50)	<input type="checkbox"/>																				
課程名稱	nvarchar(255)	<input checked="" type="checkbox"/>																				

3 課程教室 資料表	4 學生 資料表																											
<table border="1"><thead><tr><th>資料行名稱</th><th>資料類型</th><th>允許 Null</th></tr></thead><tbody><tr><td>課程編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>教師編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>教室</td><td>float</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>	資料行名稱	資料類型	允許 Null	課程編號	nvarchar(50)	<input type="checkbox"/>	教師編號	nvarchar(50)	<input type="checkbox"/>	教室	float	<input checked="" type="checkbox"/>	<table border="1"><thead><tr><th>資料行名稱</th><th>資料類型</th><th>允許 Null</th></tr></thead><tbody><tr><td>學號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>姓名</td><td>nvarchar(50)</td><td><input checked="" type="checkbox"/></td></tr><tr><td>班級</td><td>nvarchar(50)</td><td><input checked="" type="checkbox"/></td></tr><tr><td>座號</td><td>float</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>	資料行名稱	資料類型	允許 Null	學號	nvarchar(50)	<input type="checkbox"/>	姓名	nvarchar(50)	<input checked="" type="checkbox"/>	班級	nvarchar(50)	<input checked="" type="checkbox"/>	座號	float	<input checked="" type="checkbox"/>
資料行名稱	資料類型	允許 Null																										
課程編號	nvarchar(50)	<input type="checkbox"/>																										
教師編號	nvarchar(50)	<input type="checkbox"/>																										
教室	float	<input checked="" type="checkbox"/>																										
資料行名稱	資料類型	允許 Null																										
學號	nvarchar(50)	<input type="checkbox"/>																										
姓名	nvarchar(50)	<input checked="" type="checkbox"/>																										
班級	nvarchar(50)	<input checked="" type="checkbox"/>																										
座號	float	<input checked="" type="checkbox"/>																										

5 學生成績 資料表															
<table border="1"><thead><tr><th>資料行名稱</th><th>資料類型</th><th>允許 Null</th></tr></thead><tbody><tr><td>學號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>課程編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>教師編號</td><td>nvarchar(50)</td><td><input type="checkbox"/></td></tr><tr><td>成績</td><td>float</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>	資料行名稱	資料類型	允許 Null	學號	nvarchar(50)	<input type="checkbox"/>	課程編號	nvarchar(50)	<input type="checkbox"/>	教師編號	nvarchar(50)	<input type="checkbox"/>	成績	float	<input checked="" type="checkbox"/>
資料行名稱	資料類型	允許 Null													
學號	nvarchar(50)	<input type="checkbox"/>													
課程編號	nvarchar(50)	<input type="checkbox"/>													
教師編號	nvarchar(50)	<input type="checkbox"/>													
成績	float	<input checked="" type="checkbox"/>													

## ●課程資料表→資料庫圖表 (ERD 圖)



## ●SQL(新增、刪除、修改)

### 1. 刪除資料(delete)

```
delete from 課程教室
```

```
delete from 學生成績
```

```
delete from 學生
```

```
delete from 課程
```

```
delete from 教師
```

### 2. 新增資料(insert)

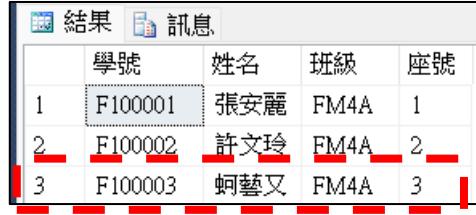
```
insert into 學生  
select distinct 學號, 班級, 姓名, 座號 from 原始資料
```

```
insert into 課程  
select distinct 課程編號, 課程名稱 from 原始資料
```

```
insert into 教師  
select distinct 教師編號, 授課老師, 職級 from 原始資料
```

```
insert into 課程教室  
select distinct 課程編號, 教師編號, 教室 from 原始資料
```

```
insert into 學生成績  
select distinct 學號, 課程編號, 教師編號, 成績 from 原始資料
```

3. 修改資料(update)		查詢確認																																								
update 學生 set 班級='FM4A',座號='3' where 學號='F100003'		select * from 學生																																								
修改前：		修改後：																																								
 <table border="1"> <thead> <tr> <th></th> <th>學號</th> <th>姓名</th> <th>班級</th> <th>座號</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>F100001</td> <td>張安麗</td> <td>FM4A</td> <td>1</td> </tr> <tr> <td>2</td> <td>F100002</td> <td>許文玲</td> <td>FM4A</td> <td>2</td> </tr> <tr> <td>3</td> <td>F100003</td> <td>蛔藝又</td> <td>FM4B</td> <td>1</td> </tr> </tbody> </table>			學號	姓名	班級	座號	1	F100001	張安麗	FM4A	1	2	F100002	許文玲	FM4A	2	3	F100003	蛔藝又	FM4B	1	 <table border="1"> <thead> <tr> <th></th> <th>學號</th> <th>姓名</th> <th>班級</th> <th>座號</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>F100001</td> <td>張安麗</td> <td>FM4A</td> <td>1</td> </tr> <tr> <td>2</td> <td>F100002</td> <td>許文玲</td> <td>FM4A</td> <td>2</td> </tr> <tr> <td>3</td> <td>F100003</td> <td>蛔藝又</td> <td>FM4A</td> <td>3</td> </tr> </tbody> </table>		學號	姓名	班級	座號	1	F100001	張安麗	FM4A	1	2	F100002	許文玲	FM4A	2	3	F100003	蛔藝又	FM4A	3
	學號	姓名	班級	座號																																						
1	F100001	張安麗	FM4A	1																																						
2	F100002	許文玲	FM4A	2																																						
3	F100003	蛔藝又	FM4B	1																																						
	學號	姓名	班級	座號																																						
1	F100001	張安麗	FM4A	1																																						
2	F100002	許文玲	FM4A	2																																						
3	F100003	蛔藝又	FM4A	3																																						

## ※小考

1. 新增自我個人資料

```
insert into 學生 values('S1061103', '許鉅偉', '資管三', '10')
```

2. 今天學習到什麼

3. 今天上課有無任何問題

# 11/28 第十二週

## ●SQL 演練

※正規化正確的話，拆得下來，組得回去

### 1. 學生修課紀錄(A11 Data)

```
SELECT dbo.教師.教師編號, dbo.教師.授課老師, dbo.教師.職級, dbo.課程.課程編號, dbo.課程.課程名稱, dbo.課程教室.教室, dbo.學生.學號, dbo.學生.姓名, dbo.學生.班級, dbo.學生.座號, dbo.學生成績.成績
FROM dbo.教師 INNER JOIN
    dbo.課程教室 ON dbo.教師.教師編號 = dbo.課程教室.教師編號 INNER JOIN
    dbo.課程 ON dbo.課程教室.課程編號 = dbo.課程.課程編號 INNER JOIN
    dbo.學生成績 ON dbo.課程教室.課程編號 = dbo.學生成績.課程編號 AND
    dbo.課程教室.教師編號 = dbo.學生成績.教師編號 INNER JOIN
    dbo.學生 ON dbo.學生成績.學號 = dbo.學生.學號
```

SQL :

```
SELECT dbo.教師.教師編號, dbo.教師.授課老師, dbo.教師.職級, dbo.課程.課程編號,
dbo.課程.課程名稱, dbo.課程教室.教室,
        dbo.學生.學號, dbo.學生.姓名, dbo.學生.班級, dbo.學生.座號, dbo.
學生成績.成績
FROM      dbo.教師 INNER JOIN
                dbo.課程教室 ON dbo.教師.教師編號 = dbo.課程教室.教師編號 INNER JOIN
                dbo.課程 ON dbo.課程教室.課程編號 = dbo.課程.課程編號 INNER JOIN
                dbo.學生成績 ON dbo.課程教室.課程編號 = dbo.學生成績.課程編號
AND
                dbo.課程教室.教師編號 = dbo.學生成績.教師編號 INNER JOIN
                dbo.學生 ON dbo.學生成績.學號 = dbo.學生.學號
```

結果 :

	教師編號	授課老師	職級	課程編號	課程名稱	教室	學號	姓名	班級	座號	成績
1	101001	李桂春	助理教授	FM1A01	程式語言	2303	F100001	FM4A	張安麗	1	90
2	101002	曾建銘	助理教授	FM2A02	資料庫	2301	F100001	FM4A	張安麗	1	70
3	107090	翁政雄	副教授	FM2A03	系統分析	2302	F100001	FM4A	張安麗	1	80
4	107090	翁政雄	副教授	FM3A01	資料探勘	2304	F100001	FM4A	張安麗	1	70
5	101001	李桂春	助理教授	FM1A01	程式語言	2303	F100002	FM4A	許文玲	2	73
6	101002	曾建銘	助理教授	FM2A02	資料庫	2301	F100002	FM4A	許文玲	2	70
7	107090	翁政雄	副教授	FM2A03	系統分析	2302	F100002	FM4A	許文玲	2	80
8	107090	翁政雄	副教授	FM3A01	資料探勘	2304	F100002	FM4A	許文玲	2	74
9	101001	李桂春	助理教授	FM1A01	程式語言	2303	F100103	FM4B	飼藝又	1	90
10	107090	翁政雄	副教授	FM2A02	資料庫	2314	F100103	FM4B	飼藝又	1	70
11	107090	翁政雄	副教授	FM2A03	系統分析	2302	F100103	FM4B	飼藝又	1	80

## 2. 學生修課數總分總平均(學號、姓名、修課數、總分、總平均)

SQL:

```
Select 學生.學號,學生.姓名,count(學生成績.成績) as 修課數,sum(學生成績.成績) as 總分,avg(學生成績.成績) as 總平均  
from 學生 inner join 學生成績 on 學生.學號=學生成績.學號  
group by 學生.學號,學生.姓名
```

結果:

	學號	姓名	修課數	總分	總平均
1	F100001	FM4A	4	310	77.5
2	F100002	FM4A	4	297	74.25
3	F100103	FM4B	3	240	80

## ● Stored Procedure

### 1. sp\_Delete\_AllData

```
sp_Delete_AllData
```

```
delete from 學生成績  
delete from 課程教室  
delete from 教師  
delete from 課程  
delete from 學生
```

```
--=====  
ALTER PROCEDURE [dbo].[sp_Delete_AllData]  
-- Add the parameters for the stored procedure here  
--沒參數，可省略  
AS  
BEGIN  
-- SET NOCOUNT ON added to prevent extra result sets from  
-- interfering with SELECT statements.  
SET NOCOUNT ON;  
  
-- Insert statements for procedure here  
--delete all data,關聯(多)的資料表先刪除，關聯(一)的資料表後刪除  
delete from 學生成績  
delete from 課程教室  
delete from 教師  
delete from 課程  
delete from 學生  
  
END
```

## 2. sp\_InsertAllData

```
sp_InsertAllData
insert into 學生 select distinct 學號,姓名,班級,座號 from 原始資料
insert into 課程 select distinct 課程編號,課程名稱 from 原始資料
insert into 教師 select distinct 教師編號,授課老師,職級 from 原始資料
insert into 課程教室 select distinct 課程編號,教師編號,教室 from 原始資料
insert into 學生成績 select distinct 學號,課程編號,教師編號,成績 from 原始資料
```

```
ALTER PROCEDURE [dbo].[sp_InsertAllData]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    --insert all data,關聯(一)的資料表先新增，關聯(多)的資料表後新增
    insert into 學生 select distinct 學號,姓名,班級,座號 from 原始資料
    insert into 課程 select distinct 課程編號,課程名稱 from 原始資料
    insert into 教師 select distinct 教師編號,授課老師,職級 from 原始資料
    insert into 課程教室 select distinct 課程編號,教師編號,教室 from 原始資料
    insert into 學生成績 select distinct 學號,課程編號,教師編號,成績 from 原始資料

END
```

## 3. sp\_Delete\_Insert\_AllData

```
sp_Delete_Insert_AllData
--呼叫/執行 現有 stored procedure
exec sp_DeleteAllData
exec sp_InsertAllData
```

```
ALTER PROCEDURE [dbo].[sp_Delete_Insert_AllData]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    --呼叫/執行 先有 stored pre
    exec sp_DeleteAllData
    exec sp_InsertAllData

END
```

#### 4. sp\_Delete\_Student\_AllData(傳入參數)

```
sp_Delete_Student_AllData  
@學號 nvarchar(10)  
    delete from 學生成績 where 學號 = @學號  
    delete from 學生 where 學號 = @學號
```

```
ALTER PROCEDURE [dbo].[sp_Delete_Student_AllData]  
    -- Add the parameters for the stored procedure here  
    @學號 nvarchar(10)  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    -- Insert statements for procedure here  
    delete from 學生成績 where 學號 = @學號  
    delete from 學生 where 學號 = @學號  
  
END
```

#### 5. sp\_Delete\_Student\_AllData\_Output(傳入參數、傳出參數)

```
sp_Delete_Student_AllData_Output  
@學號 nvarchar(10),  
@更新筆數 int=0 output  
    delete from 學生成績 where 學號 = @學號  
    delete from 學生 where 學號 = @學號  
    set @更新筆數 = @@ROWCOUNT
```

```
ALTER PROCEDURE [dbo].[sp_Delete_Student_AllData_Output]  
    -- Add the parameters for the stored procedure here  
    @學號 nvarchar(10),  
    @更新筆數 int=0 output  
  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    -- Insert statements for procedure here  
    delete from 學生成績 where 學號 = @學號  
    delete from 學生 where 學號 = @學號  
    set @更新筆數 = @@ROWCOUNT  
  
END
```

## 6. stored procedure 應用(1)刪除學號='F100001'

--執行前

```
select * from 學生  
select * from 學生成績
```

--宣告變數

```
declare @學號 nvarchar(10)  
declare @更新筆數 int
```

--設定參數值

```
set @學號='F100001'
```

--執行

```
exec sp_Delete_Student_AllData @學號
```

	學號	姓名	班級	座號
1	F100002	FM4A 許文玲	2	
2	F100103	FM4B 虹藝又	1	

	學號	課程編號	教師編號	成績
1	F100002	FM1A01	101001	73
2	F100002	FM2A02	101002	70
3	F100002	FM2A03	107090	80
4	F100002	FM3A01	107090	74
5	F100103	FM1A01	101001	90
6	F100103	FM2A02	107090	70
7	F100103	FM2A03	107090	80

## 7. stored procedure 應用(2) 刪除學號='F100001'

--執行前

```
select * from 學生  
select * from 學生成績
```

--宣告變數

```
declare @學號 nvarchar(10)  
declare @更新筆數 int
```

--設定參數值

```
set @學號='F100001'
```

--執行

```
--exec sp_Delete_Student_AllData @學號  
exec sp_Delete_Student_AllData_Output @學號 ,@更新筆數 output
```

--執行後

```
select * from 學生  
select * from 學生成績
```

```
select @學號 as 學號 ,@更新筆數 as 更新筆數
```



The screenshot shows two separate SQL query results side-by-side, each enclosed in a red border.

**Result 1 (Left):**

	學號	姓名	班級	座號
1	F100001	張安麗	FM4A	1
2	F100002	許文玲	FM4A	2
3	F100103	蚵藝又	FM4B	1

**Result 2 (Right):**

	學號	姓名	班級	座號
1	F100002	許文玲	FM4A	2
2	F100103	蚵藝又	FM4B	1

	學號	課程編號	教師編號	成績
1	F100001	FM1A01	101001	90
2	F100001	FM2A02	101002	70
3	F100001	FM2A03	107090	80
4	F100001	FM3A01	107090	70
5	F100002	FM1A01	101001	73
6	F100002	FM2A02	101002	70
7	F100002	FM2A03	107090	80
8	F100002	FM3A01	107090	74

	學號	更新筆數
1	F100001	1

## ※小考

今日無小考

# 12/5 第十三週

## ●SQL 操作

### 1. 在員工資料表新增資料(設計)

資料行名稱	資料類型	允許 Null
雇用日期	datetime	<input checked="" type="checkbox"/>
地址	nvarchar(60)	<input checked="" type="checkbox"/>
城市	nvarchar(15)	<input checked="" type="checkbox"/>
行政區	nvarchar(15)	<input checked="" type="checkbox"/>
區域號碼	nvarchar(10)	<input checked="" type="checkbox"/>
國家地區	nvarchar(15)	<input checked="" type="checkbox"/>
電話號碼	nvarchar(24)	<input checked="" type="checkbox"/>
內部分機號碼	nvarchar(4)	<input checked="" type="checkbox"/>
相片	nvarchar(255)	<input checked="" type="checkbox"/>
附註	nvarchar(MAX)	<input checked="" type="checkbox"/>
主管	int	<input checked="" type="checkbox"/>
密碼	nvarchar(255)	<input checked="" type="checkbox"/>
次數	smallint	<input checked="" type="checkbox"/>
營業額	money	<input checked="" type="checkbox"/>
年終獎金	money	<input checked="" type="checkbox"/>

### 2. 新增「員工營業績效」資料表(新增資料表)

資料行名稱	資料類型	允許 Null
員工編號	int	<input type="checkbox"/>
年	smallint	<input type="checkbox"/>
月	smallint	<input type="checkbox"/>
營業額	money	<input checked="" type="checkbox"/>
績效獎金	money	<input checked="" type="checkbox"/>

## ●修改 SQL 指令演練

```
(--update 員工 set 密碼=null  
--update 員工 set 營業額=null)
```

### 1. 修改員工密碼(密碼=員工編號)

SQL:

--修改前

```
select 員工編號,密碼 from 員工
```

--修改

```
update 員工 set 密碼=員工編號
```

--修改後

```
select 員工編號,密碼 from 員工
```

結果:

結果 訊息		
	員工編號	密碼
1	1	NULL
2	2	NULL
3	3	NULL
4	4	NULL
5	5	NULL
...	...	...

	員工編號	密碼
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7

### 2. 修改員工密碼(密碼=員工編號+名)

SQL:

--修改前

```
select 員工編號,名,密碼 from 員工
```

--修改

```
update 員工
```

```
set 密碼=convert(nvarchar(10),員工編號)+名
```

--字串相加，資料型態必須相同

--convert(資料型態,變數)

--修改後

```
select 員工編號,名,密碼 from 員工
```

結果：

結果 訊息			
	員工編號	名	密碼
1	1	Mary	NULL
2	2	Bradley	NULL
3	3	Kim	NULL
4	4	Chris	NULL
5	5	Mike	NULL

	員工編號	名	密碼
1	1	Mary	1Mary
2	2	Bradley	2Bradley
3	3	Kim	3Kim
4	4	Chris	4Chris
5	5	Mike	5Mike
6	6	Bill	6Bill
7	7	Steven	7Steven

### 3. 修改員工密碼(密碼=名+出生日期)

<https://www.sqlservertutorial.net/sql-server-system-functions/convert-datetime-to-string/> (convert(資料型態, 變數[style]);112=yyyymmdd 參考資料)

SQL:

--修改前

```
select 員工編號,名,出生日期,密碼 from 員工
```

--修改

```
update 員工
```

```
set 密碼=名+convert(nvarchar(8),出生日期,112)
```

--字串相加，資料型態必須相同

--convert(資料型態, 變數[style]);112=yyyymmdd

--修改後

```
select 員工編號,名,出生日期,密碼 from 員工
```

結果：

結果 訊息			
	員工編號	名	出生日期
1	1	Mary	1968-12-08 00:00:00.000
2	2	Bradley	1952-02-19 00:00:00.000
3	3	Kim	1963-08-30 00:00:00.000
4	4	Chris	1958-09-19 00:00:00.000
5	5	Mike	1955-03-04 00:00:00.000
6	6	Bill	1963-07-02 00:00:00.000
7	7	Steven	1960-05-30 00:00:00.000

	員工編號	名	出生日期	密碼
1	1	Mary	1968-12-08 00:00:00.000	Mary19681208
2	2	Bradley	1952-02-19 00:00:00.000	Bradley19520219
3	3	Kim	1963-08-30 00:00:00.000	Kim19630830
4	4	Chris	1958-09-19 00:00:00.000	Chris19580919
5	5	Mike	1955-03-04 00:00:00.000	Mike19550304
6	6	Bill	1963-07-02 00:00:00.000	Bill19630702
7	7	Steven	1960-05-30 00:00:00.000	Steven19600530

#### 4. 修改員工密碼(密碼=員工編號+名+出生日期)

SQL:

--修改前

```
select 員工編號,名,出生日期,密碼 from 員工
```

--修改

```
update 員工
```

```
set 密碼=convert(nvarchar(10),員工編號)+名+convert(nvarchar(8),出生日期,112)
```

--修改後

```
select 員工編號,名,出生日期,密碼 from 員工
```

結果：

結果 訊息			
	員工編號	名	出生日期
1	1	Mary	1968-12-08 00:00:00.000
2	2	Bradley	1952-02-19 00:00:00.000
3	3	Kim	1963-08-30 00:00:00.000
4	4	Chris	1958-09-19 00:00:00.000
5	5	Mike	1955-03-04 00:00:00.000

	員工編號	名	出生日期	密碼
1	1	Mary	1968-12-08 00:00:00.000	1Mary19681208
2	2	Bradley	1952-02-19 00:00:00.000	2Bradley19520219
3	3	Kim	1963-08-30 00:00:00.000	3Kim19630830
4	4	Chris	1958-09-19 00:00:00.000	4Chris19580919
5	5	Mike	1955-03-04 00:00:00.000	5Mike19550304
6	6	Bill	1963-07-02 00:00:00.000	6Bill19630702
7	7	Steven	1960-05-29 00:00:00.000	7Steven19600529

已成功執行查詢。

## 5. 修改員工職稱是“業務”的密碼(密碼=員工編號+名+內部分機號碼)

SQL:

--修改前

```
select 員工編號,名,職稱,內部分機號碼,密碼  
from 員工  
where 職稱='業務'
```

--修改

```
update 員工  
set 密碼=convert(nvarchar(10),員工編號)+名+內部分機號碼  
where 員工.職稱='業務'
```

--修改後

```
select 員工編號,名,職稱,內部分機號碼,密碼  
from 員工  
where 職稱='業務'
```

結果:

結果 訊息				
	員工編號	名	職稱	內部分機號碼
1	1	Mary	業務	5467
2	3	Kim	業務	3355
3	4	Chris	業務	5176
4	6	Bill	業務	4281
5	7	Steven	業務	4651
6	9	Linda	業務	4521

	員工編號	名	職稱	內部分機號碼	密碼
1	1	Mary	業務	5467	1Mary5467
2	3	Kim	業務	3355	3Kim3355
3	4	Chris	業務	5176	4Chris5176
4	6	Bill	業務	4281	6Bill4281
5	7	Steven	業務	4651	7Steven4651
6	9	Linda	業務	4521	9Linda4521

## # 統整第 5 題重點

要做資料的異動之前有兩大重點

1. 透過查詢，篩選條件是否正確

2. 在 select 確定修改規則是對的

最後再放修改規則在 update 語法上

--修改前

```
select 員工編號,名,職稱,內部分機號碼,密碼,convert(nvarchar(10),員工編號)+名+內部分  
機號碼 as 新密碼  
from 員工  
where 職稱= '業務'
```

修改的內容

--修改 修改的資料/筆數

```
update 員工  
set 密碼=convert(nvarchar(10),員工編號)+名+內部分機號碼  
where 員工.職稱='業務'
```

## 6. 修改員工資料表裡的營業額(5 step)

step1

--(A)vw\_員工營業額(確認營業額)

SQL:

```
select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折  
扣)),0) as 營業額  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
group by 訂貨主檔.員工編號
```

結果:

	員工編號	營業額
1	9	77308
2	3	201879
3	6	73913
4	7	124568
5	1	191589
6	4	232891
7	13	19
8	5	68624
9	2	166538
10	8	126862

## step2

--(B)員工資料表裡的營業額 inner join vw\_員工營業額

SQL:

```
select 員工.員工編號,員工.營業額,vw_員工營業額.員工編號 as vw 員工編號,vw_員工營業額.營業額 as vw 营業額
from 員工 inner join
(
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

結果:



	員工編號	營業額	vw員工編號	vw營業額
1	9	NULL	9	77308
2	3	NULL	3	201879
3	6	NULL	6	73913
4	7	NULL	7	124568
5	1	NULL	1	191589
6	4	NULL	4	232891
7	13	NULL	13	19
8	5	NULL	5	68624
9	2	NULL	2	166538
10	8	NULL	8	126862

## step3

--(C)修改到員工資料表裡的營業額

SQL:

```
update 員工
set 營業額 = vw_員工營業額.營業額
from 員工 inner join
(
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

結果:



訊息  
(10 個資料列受到影響)

#### step4

--(D)顯示員工資料表營業額

SQL:

```
select 員工編號,營業額  
from 員工
```

結果:

	員工編號	營業額
1	1	191589.00
2	2	166538.00
3	3	201879.00
4	4	232891.00
5	5	68624.00
6	6	73913.00
7	7	124568.00
8	8	126862.00
9	9	77308.00
10	12	NULL
11	13	19.00
12	14	NULL

#### step5 兩個資料表相互確認

SQL:

```
select 員工.員工編號,員工.營業額,vw_員工營業額.員工編號 as vw_員工編號,vw_員工營業  
額.營業額 as vw_營業額  
from 員工 inner join  
(  
select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折  
扣)),0) as 營業額  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
group by 訂貨主檔.員工編號  
)  
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

結果:

	員工編號	營業額	vw_員工編號	vw_營業額
1	9	77308.00	9	77308
2	3	201879.00	3	201879
3	6	73913.00	6	73913
4	7	124568.00	7	124568
5	1	191589.00	1	191589
6	4	232891.00	4	232891
7	13	19.00	13	19
8	5	68624.00	5	68624
9	2	166538.00	2	166538
10	8	126862.00	8	126862

## ※小考

1. 修改員工年終獎金(營業額\*0.01)(-- update 員工 set 年終獎金=null)

### --(a)vw\_員工年終獎金

```
select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0)*0.01 as 年終獎金
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
group by 訂貨主檔.員工編號
```

### --(b)員工資料表裡的年終獎金 inner join vw\_員工年終獎金

```
select 員工.員工編號, 員工.年終獎金, vw_員工年終獎金.員工編號 as vw_員工編號, vw_員工年終獎金.營業額 as vw_年終獎金
from 員工 inner join
(
    select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0)*0.01 as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號
)
vw_員工年終獎金 on 員工.員工編號=vw_員工年終獎金.員工編號
```

### --(c)修改到員工資料表裡的年終獎金 方法 1

```
update 員工
set 年終獎金 = vw_員工年終獎金.年終獎金
from 員工 inner join
(
    select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0)*0.01 as 年終獎金
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號
)
vw_員工年終獎金 on 員工.員工編號=vw_員工年終獎金.員工編號
```

### --(c)修改到員工資料表裡的年終獎金 方法 2

```
update 員工
set 營業額 = vw_員工營業額.營業額,
    年終獎金 = vw_員工營業額.營業額*0.01
from 員工 inner join
(
    select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
)
```

```
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
group by 訂貨主檔.員工編號  
)  
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

--(d)

```
select 員工編號,年終獎金  
from 員工
```

	員工編號	年終獎金
1	1	1915.89
2	2	1665.38
3	3	2018.79
4	4	2328.91
5	5	686.24
6	6	739.13
7	7	1245.68
8	8	1268.62
9	9	773.08
10	12	NULL
11	13	0.19
12	14	NULL

# 12/12 第十四週

## ●SQL 演練

### 1. 檢討上次小考

SQL:

```
update 員工
set 營業額 = vw_員工營業額.營業額,
    年終獎金 = vw_員工營業額.營業額*0.01
from 員工 inner join
(
select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

結果:

	員工編號	年終獎金
1	1	1915.89
2	2	1665.38
3	3	2018.79
4	4	2328.91
5	5	686.24
6	6	739.13
7	7	1245.68
8	8	1268.62
9	9	773.08
10	12	NULL
11	13	0.19
12	14	NULL

## 2. 員工 1998 年營業額的年終獎金(0.01)

SQL:

--(A)員工 1998 營業額

```
select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
where datepart(yyyy,訂貨主檔.訂單日期)=1998
group by 訂貨主檔.員工編號
```

--(B)員工(1998 年)資料表裡的營業額 inner join vw\_營業額

```
select 員工.員工編號, 員工.營業額, vw_員工營業額.員工編號 as vw_員工編號, vw_員工營業額.營業額 as vw_營業額
from 員工 inner join
(
    select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    where datepart(yyyy,訂貨主檔.訂單日期)=1998
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

--(C)修改到員工(1998 年)資料表裡

```
update 員工
set 營業額 = vw_員工營業額.營業額,
    年終獎金 = vw_員工營業額.營業額*0.01
from 員工 inner join
(
    select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    where datepart(yyyy,訂貨主檔.訂單日期)=1998
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

--(D)查詢確認

```
select 員工編號, 營業額, 年終獎金
from 員工
```

**(A)**

	員工編號	營業額
1	1	63195
2	2	74337
3	3	75629
4	4	54136
5	5	19692
6	6	14144
7	7	48865
8	8	48590
9	9	41103
10	13	19

**(B)**

	員工編號	營業額	vw員工編號	vw營業額
1	1	NULL	1	63195
2	2	NULL	2	74337
3	3	NULL	3	75629
4	4	NULL	4	54136
5	5	NULL	5	19692
6	6	NULL	6	14144
7	7	NULL	7	48865
8	8	NULL	8	48590
9	9	NULL	9	41103
10	13	NULL	13	19

已成功執行查詢。

**(D)**

	員工編號	營業額	年終獎金
1	1	63195.00	631.95
2	2	74337.00	743.37
3	3	75629.00	756.29
4	4	54136.00	541.36
5	5	19692.00	196.92
6	6	14144.00	141.44
7	7	48865.00	488.65
8	8	48590.00	485.90
9	9	41103.00	411.03
10	12	NULL	NULL
11	13	19.00	0.19
12	14	NULL	NULL

**(C)**

(10 個資料列受到影響)

### 3. 在員工資料表裡，新增績效獎金

The screenshot shows the '員工' table structure in Object Explorer. A new column '績效獎金' (Performance Bonus) has been added, highlighted with a red box. The column is defined as 'money' type.

欄位	型別	備註
密碼	nvarchar(255)	
次數	smallint	
營業額	money	
年終獎金	money	
績效獎金	money	

#### 4. 1997 年 12 月員工績效獎金(營業額\*0.01)

SQL:

--(A)

```
select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
where datepart(yyyy,訂貨主檔.訂單日期)=1997 and datepart(mm,訂貨主檔.訂單日期)=12  
group by 訂貨主檔.員工編號
```

--(B)

```
select 員工.員工編號, 員工.營業額, vw_員工營業額.員工編號 as vw_員工編號, vw_員工營業額.營業額 as vw_營業額  
from 員工 inner join  
(  
select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
where datepart(yyyy,訂貨主檔.訂單日期)=1997 and datepart(mm,訂貨主檔.訂單日期)=12  
group by 訂貨主檔.員工編號  
)  
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

--(C)

```
update 員工  
set 營業額 = vw_員工營業額.營業額,  
    績效獎金 = vw_員工營業額.營業額*0.01  
from 員工 inner join  
(  
select 訂貨主檔.員工編號, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額  
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼  
where datepart(yyyy,訂貨主檔.訂單日期)=1997 and datepart(mm,訂貨主檔.訂單日期)=12  
group by 訂貨主檔.員工編號  
)  
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

--(D)

```
select 員工編號, 營業額, 績效獎金, 年終獎金  
from 員工
```

The screenshot shows four tables labeled (A) through (D) in separate windows:

- (A)**: A table with columns '員工編號' and '營業額'. Data rows: 1 (1, 15340), 2 (2, 7540), 3 (3, 17637), 4 (4, 16235), 5 (5, 507), 6 (6, 7431), 7 (7, 872), 8 (8, 3925), 9 (9, 1910).
- (B)**: A table with columns '員工編號', '營業額', 'vw員工編號', and 'vw營業額'. Data rows: 1 (1, NULL, 1, 15340), 2 (2, NULL, 2, 7540), 3 (3, NULL, 3, 17637), 4 (4, NULL, 4, 16235), 5 (5, NULL, 5, 507), 6 (6, NULL, 6, 7431), 7 (7, NULL, 7, 872), 8 (8, NULL, 8, 3925), 9 (9, NULL, 9, 1910).
- (C)**: A message box stating '(9 個資料列受到影響)' (9 rows affected).
- (D)**: A table with columns '員工編號', '營業額', and '績效獎金'. Data rows: 1 (1, 15340.00, 153.40), 2 (2, 7540.00, 75.40), 3 (3, 17637.00, 176.37), 4 (4, 16235.00, 162.35), 5 (5, 507.00, 5.07), 6 (6, 7431.00, 74.31), 7 (7, 872.00, 8.72), 8 (8, 3925.00, 39.25), 9 (9, 1910.00, 19.10), 10 (12, NULL, NULL), 11 (13, NULL, NULL), 12 (14, NULL, NULL).

## ● Stored Procedure

### 1. sp\_修改員工營業額

```
update 員工
set 營業額 = vw_員工營業額.營業額
from 員工 inner join
(
select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
```

```

CREATE PROCEDURE sp_修改員工營業額
    -- Add the parameters for the stored procedure here

    AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    update 員工
    set 營業額 = vw_員工營業額.營業額
    from 員工 inner join
    (
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號
    )
    vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號
END
GO

```

## 1-1 測試是否正確

```
select 員工編號,營業額  
from 員工
```

```
exec sp_修改員工營業額
```

```
select 員工編號,營業額  
from 員工
```

結果 訊息		
	員工編號	營業額
1	1	NULL
2	2	NULL
3	3	NULL
4	4	NULL
5	5	NULL
6	6	NULL
7	7	NULL
8	8	NULL
9	9	NULL
10	12	NULL
11	13	NULL
12	14	NULL

訊息  
命令已順利完成。

結果 訊息		
	員工編號	營業額
1	1	191589.00
2	2	166538.00
3	3	201879.00
4	4	232891.00
5	5	68624.00
6	6	73913.00
7	7	124568.00
8	8	126862.00
9	9	77308.00
10	12	NULL
11	13	19.00
12	14	NULL

## 2. sp\_修改員工營業額\_參數

```
@year smallint,
@更新筆數 int output
update 員工
set 營業額=0, 年終獎金 = 0
update 員工
set 營業額 = vw_員工營業額.營業額
from 員工 inner join
(
select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
where datepart(yyyy,訂貨主檔.訂單日期)= @year
group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號

set @更新筆數=@@ROWCOUNT
```

```
-- =====
--CREATE PROCEDURE sp_修改員工營業額_參數
--  Add the parameters for the stored procedure here
--@year smallint,
--@更新筆數 int output
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    update 員工
    set 營業額 = vw_員工營業額.營業額
    from 員工 inner join
    (
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    where datepart(yyyy,訂貨主檔.訂單日期)= @year
    group by 訂貨主檔.員工編號
    )
    vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號

    set @更新筆數=@@ROWCOUNT
END
GO
```

## 2-2 測試是否正確

```

declare @year smallint
declare @更新筆數 int
set @year =1998
set @更新筆數 =0

select 員工.員工編號,員工.營業額,vw_員工營業額.員工編號 as vw_員工編號,vw_員工營業額.營業額 as vw_營業額
from 員工 inner join
(
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    where datepart(yyyy,訂貨主檔.訂單日期)= @year
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號 ----- (A)

exec sp_修改員工營業額_參數 @year,@更新筆數 output ----- (B)

select 員工編號,營業額 ----- (C)
from 員工

```

結果 訊息			
	員工編號	營業額	vw員工編號
1	1	NULL	1
2	2	NULL	2
3	3	NULL	3
4	4	NULL	4
5	5	NULL	5
6	6	NULL	6
7	7	NULL	7
8	8	NULL	8
9	9	NULL	9
10	13	NULL	13

(A)

結果 訊息			
	員工編號	營業額	vw員工編號
1	1	63195.00	1
2	2	74337.00	2
3	3	75629.00	3
4	4	54136.00	4
5	5	19692.00	5
6	6	14144.00	6
7	7	48865.00	7
8	8	48590.00	8
9	9	41103.00	9
10	13	19.00	13

(B)

結果 訊息		
	員工編號	營業額
1	1	63195.00
2	2	74337.00
3	3	75629.00
4	4	54136.00
5	5	19692.00
6	6	14144.00
7	7	48865.00
8	8	48590.00
9	9	41103.00
10	12	NULL
11	13	19.00
12	14	NULL

(C)

### 3. sp\_修改員工營業額\_參數 2

```
@year smallint,
@更新筆數 int output

--先將營業額=0,年終獎金=0
update 員工 set 營業額 = 0, 年終獎金=0
--@year=''，則計算現在日期的前一年，例如:2021/5/11，計算 2020 年
if @year = '' begin
    set @year = year(getdate())-1
    select @year as 'year'
end

update 員工
set 營業額 = vw_員工營業額.營業額,
    年終獎金 = vw_員工營業額.營業額*0.01
from 員工 inner join
(
    select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    where datepart(yyyy,訂貨主檔.訂單日期)= @year
    group by 訂貨主檔.員工編號
)
vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號

set @更新筆數=@@ROWCOUNT
```

```
CREATE PROCEDURE sp_修改員工營業額_參數2
-- Add the parameters for the stored procedure here
@year smallint,
@更新筆數 int output

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    --先將營業額=0,年終獎金=0
    update 員工 set 營業額 = 0, 年終獎金=0
    --@year=''，則計算現在日期的前一年，例如:2021/5/11，計算2020年
    if @year = '' begin
        set @year = year(getdate())-1
        select @year as 'year'
    end
    update 員工
    set 營業額 = vw_員工營業額.營業額,
        年終獎金 = vw_員工營業額.營業額*0.01
    from 員工 inner join
    (
        select 訂貨主檔.員工編號,round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)),0) as 營業額
        from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
        where datepart(yyyy,訂貨主檔.訂單日期)= @year
        group by 訂貨主檔.員工編號
    )
    vw_員工營業額 on 員工.員工編號=vw_員工營業額.員工編號

    set @更新筆數=@@ROWCOUNT
```

## ※小考

先新增員工營業績效資料表

資料行名稱	資料類型	允許 Null
員工編號	int	<input type="checkbox"/>
年	smallint	<input type="checkbox"/>
月	smallint	<input type="checkbox"/>
營業額	money	<input checked="" type="checkbox"/>
績效獎金	money	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

### 1. 將員工編號、每年、每月、營業額、績效獎金 insert(員工資料表)進去

```
insert into 員工營業績效
select distinct 員工.員工編號, 年, 月, vw.營業額 as 營業額, (vw.營業額)*0.01 as 繢效獎金
from 員工 inner join
(
    select 訂貨主檔.員工編號, datepart(yyyy, 訂貨主檔.訂單日期) as 年, datepart(mm, 訂貨主檔.訂單日期) as 月, round(sum((訂貨明細.單價)*(訂貨明細.數量)*(1-訂貨明細.折扣)), 0)
    as 營業額
    from 訂貨主檔 inner join 訂貨明細 on 訂貨主檔.訂單號碼=訂貨明細.訂單號碼
    group by 訂貨主檔.員工編號, datepart(yyyy, 訂貨主檔.訂單日期), datepart(mm, 訂貨主檔.訂單日期)
)vw
on 員工.員工編號 = vw.員工編號
```

確認是否正確

```
select * from 員工營業績效
```

	員工編號	年	月	營業額	績效獎金
1	1	1996	7	1615.00	16.15
2	1	1996	8	5037.00	50.37
3	1	1996	9	6651.00	66.51
4	1	1996	10	3933.00	39.33
5	1	1996	11	9563.00	95.63
6	1	1996	12	8447.00	84.47
7	1	1997	1	7332.00	73.32
8	1	1997	2	1946.00	19.46
9	1	1997	3	5124.00	51.24
10	1	1997	4	240.00	2.40
11	1	1997	5	9116.00	91.16
12	1	1997	6	5468.00	54.68
13	1	1997	7	19531.00	195.31
14	1	1997	8	5105.00	51.05
15	1	1997	9	7442.00	74.42
16	1	1997	10	12414.00	124.14

已成功執行查詢。

# 12/19 第十五週

## ●SQL 演練

### 1. 新增「登入紀錄」資料表

資料行名稱	資料類型	允許 Null
員工編號	nvarchar(50)	<input type="checkbox"/>
登入時間	datetime	<input type="checkbox"/>
密碼	nvarchar(50)	<input checked="" type="checkbox"/>
登入結果	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

### 2. 帳號輸入正確與否

SQL:

STEP1

--宣告變數

```
declare @EID nvarchar(255)=''
declare @Password nvarchar(255)=''
declare @Name nvarchar(255)=''
declare @Degree nvarchar(255)=''
declare @LoginResult nvarchar(255)=''
```

--設定帳號密碼

```
set @EID=1 set @Password='1'
```

--(A)帳密正確，回傳姓名,職稱,登入成功

```
update 員工
```

```
set @Name=姓名,
```

```
    @Degree= 職稱,
```

```
    @LoginResult='帳密正確，登入成功'
```

```
from 員工
```

```
where 員工編號=@EID and 密碼=@Password
```

```
select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職  
稱,@LoginResult as 登入結果
```

	員工編號	密碼	姓名	職稱	登入結果
1	1	1	張瑾雯	業務	帳密正確，登入成功

STEP2

```
declare @EID nvarchar(255)=''
```

```
declare @Password nvarchar(255)=''
```

```

declare @Name nvarchar(255)=''
declare @Degree nvarchar(255)=''
declare @LoginResult nvarchar(255)=''

set @EID=1 set @Password='2'
--(A)帳密正確，回傳姓名,職稱,登入成功
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳密正確，登入成功'
from 員工
where 員工編號=@EID and 密碼=@Password

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職
稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗
if (@LoginResult='') begin
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳號正確，密碼錯誤，登入失敗'
from 員工
where 員工編號=@EID
select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職
稱,@LoginResult as 登入結果
end --if(@LOGINRESULT='')begin

```

	員工編號	密碼	姓名	職稱	登入結果
1	1	2	張瑾雯	業務	帳號正確，密碼錯誤，登入失敗

### STEP3

```

declare @EID nvarchar(255)=''
declare @Password nvarchar(255)=''
declare @Name nvarchar(255)=''
declare @Degree nvarchar(255)=''
declare @LoginResult nvarchar(255)=''

set @EID=10 set @Password='2'

```

--(A)帳密正確，回傳姓名,職稱,登入成功

update 員工

set @Name=姓名,

@Degree= 職稱,

@LoginResult='帳密正確，登入成功'

from 員工

where 員工編號=@EID and 密碼=@Password

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗

if (@LoginResult='') begin

update 員工

set @Name=姓名,

@Degree= 職稱,

@LoginResult='帳號正確，密碼錯誤，登入失敗'

from 員工

where 員工編號=@EID

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

end --if(@LOGINRESULT='')begin

--(C)無此帳號

if (@LoginResult='') begin

update 員工

set @LoginResult='帳號不存在'

end --(@LoginResult='')begin

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

員工編號	密碼	姓名	職稱	登入結果
1	10	2		帳號不存在

### 3. 將第二題寫入 Stored Produce 需求(sp\_login)

```
sp_login
    @EID nvarchar(255)='',
    @Password nvarchar(255)='',
    @Name nvarchar(255)='output',
    @Degree nvarchar(255)='output',
    @LoginResult nvarchar(255)='output'

--(A)帳密正確，回傳姓名,職稱,登入成功
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳密正確，登入成功'
from 員工
where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職
稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗
if (@LoginResult='') begin
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳號正確，密碼錯誤，登入失敗'
from 員工
where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職
稱,@LoginResult as 登入結果
end --if(@LOGINRESULT='')begin

--(C)無此帳號
if (@LoginResult='') begin
update 員工
set @LoginResult='帳號不存在'
end --(@LoginResult='')begin
```

## 顯示：

```
SQLQuery5.sql - L...orthwind (sa (55))* X SQLQuery2.sql - L...orthwind (sa (53))* SQLQuery1.sql - L...orthwind (sa (52))*
-- Author:      <Author,Name>
-- Create date: <Create Date,.>
-- Description: <Description,.>
-- 
CREATE PROCEDURE sp_login
    -- Add the parameters for the stored procedure here
    @EID nvarchar(255)='',
    @Password nvarchar(255)= '',
    @Name nvarchar(255)='output',
    @Degree nvarchar(255)='output',
    @LoginResult nvarchar(255)='output'
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
--(A)帳密正確，回傳姓名,職稱,登入成功
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳密正確，登入成功'
from 員工
where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗
if (@LoginResult='') begin
update 員工
set @Name=姓名,
    @Degree= 職稱,
    @LoginResult='帳號正確，密碼錯誤，登入失敗'
from 員工
where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果
end --if(@LOGINRESULT='')begin

--(C)無此帳號
if (@LoginResult='') begin
update 員工
set @LoginResult='帳號不存在'
end --(@LoginResult='')begin

END
GO
```

#### 4. 測試 sp\_login

```
declare @EID nvarchar(255)=' '
declare @Password nvarchar(255)=' '
declare @Name nvarchar(255)=' '
declare @Degree nvarchar(255)=' '
declare @LoginResult nvarchar(255)=' '

set @EID = 1 set @Password='1'
exec sp_login @EID,@Password,@Name output,@Degree,@LoginResult output
```

```
select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果
```

三種不同結果顯示：

員工編號 密碼 姓名 職稱 登入結果				
1	1	1	張瑾雯	帳密正確，登入成功

員工編號 密碼 姓名 職稱 登入結果				
1	2	1	陳季暄	帳號正確，密碼錯誤，登入失敗

員工編號 密碼 姓名 職稱 登入結果				
1	10	1		帳號不存在

## 5. Stored Produce 增加回傳次數(sp\_login)

--(A)帳密正確，回傳姓名,職稱,登入成功

update 員工

set @Name=姓名,

    @Degree= 職稱,

**次數=0,**

    @LoginResult='帳密正確，登入成功'

from 員工

where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗

if (@LoginResult="") begin

update 員工

set @Name=姓名,

    @Degree= 職稱,

**次數=次數+1,**

    @LoginResult='帳號正確，密碼錯誤，登入失敗'

from 員工

where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

end --if(@LOGINRESULT="")begin

--(C)無此帳號

if (@LoginResult="") begin

update 員工

set @LoginResult='帳號不存在'

end --(@LoginResult="")begin

END

```

--(A)帳密正確，回傳姓名,職稱,登入成功
update 員工
set @Name=姓名,
    @Degree=職稱,
    次數=0,
    @LoginResult='帳號正確，登入成功'
from 員工
where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗
if (@LoginResult='') begin
update 員工
set @Name=姓名,
    @Degree=職稱,
    次數=次數+1,
    @LoginResult='帳號正確，密碼錯誤，登入失敗'
from 員工
where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果
end --if(@LOGINRESULT='')

--(C)無此帳號
if (@LoginResult='') begin
update 員工
set @LoginResult='帳號不存在'
end --(@LoginResult='')
END

```

## 6. 測試回傳次數

```

declare @EID nvarchar(255)=''
declare @Password nvarchar(255)=''
declare @Name nvarchar(255)=''
declare @Degree nvarchar(255)=''
declare @LoginResult nvarchar(255)=''

--update 員工 set 次數 = 0

set @EID = 1 set @Password='12'
exec sp_login @EID,@Password,@Name output,@Degree,@LoginResult output

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

select 員工編號,次數 from 員工 where 員工編號=@EID

```

結果 訊息				
	員工編號	密碼	姓名	職稱
1	1	12	張瑾雯	帳號正確，密碼錯誤，登

結果 訊息					
	員工編號	密碼	姓名	職稱	登入結果
1	2	12	陳季暄		帳號正確，密碼錯誤，登入失敗

結果 訊息		
	員工編號	次數
1	1	1

結果 訊息		
	員工編號	次數
1	2	3

## 7. Stored Produce 加入「登入紀錄」(sp\_login)

--(A)帳密正確，回傳姓名,職稱,登入成功

update 員工

set @Name=姓名,

@Degree= 職稱,

次數=0,

@LoginResult='帳密正確，登入成功'

from 員工

where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗

if (@LoginResult='') begin

update 員工

set @Name=姓名,

@Degree= 職稱,

次數 = 次數+1,

--@LoginResult='帳號正確，密碼錯誤，登入失敗'

@LoginResult ='登入失敗次數:'+convert(nvarchar(10),次數+1)

from 員工

where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

end --if(@LOGINRESULT='')begin

--(C)無此帳號

if (@LoginResult='') begin

update 員工

set @LoginResult='帳號不存在'

end --(@LoginResult='')begin

--寫入登入紀錄

insert 登入紀錄 values(@EID,GETDATE(),@Password,@LoginResult)

END

```

--(A)帳密正確，回傳姓名,職稱,登入成功
update 員工
set @Name=姓名,
@Degree= 職稱,
次數=0,
@LoginResult='帳密正確，登入成功'
from 員工
where 員工編號=@EID and 密碼=@Password

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

--(B)帳號正確，密碼錯誤，回傳姓名,職稱,登入失敗
if (@LoginResult='') begin
update 員工
set @Name=姓名,
@Degree= 職稱,
次數 = 次數+1,
--@LoginResult='帳號正確，密碼錯誤，登入失敗'
@LoginResult ='登入失敗次數:' +convert(nvarchar(10),次數+1)
from 員工
where 員工編號=@EID

--select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果
end --if(@LOGINRESULT='')begin

--(C)無此帳號
if (@LoginResult='') begin
update 員工
set @LoginResult='帳號不存在'
end --if(@LoginResult='')begin

--寫入登入紀錄
insert 登入紀錄 values(@EID,GETDATE(),@Password,@LoginResult)
END

```

## 8. 測試登入紀錄(sp\_login)

```

declare @EID nvarchar(255)=''
declare @Password nvarchar(255)=''
declare @Name nvarchar(255)=''
declare @Degree nvarchar(255)=''
declare @LoginResult nvarchar(255)=''

--update 員工 set 次數 = 0

set @EID = 2 set @Password='12'
exec sp_login @EID,@Password,@Name output,@Degree output,@LoginResult output

select @EID as 員工編號,@Password as 密碼,@Name as 姓名,@Degree as 職稱,@LoginResult as 登入結果

select 員工編號,次數 from 員工 where 員工編號=@EID

select * from 登入紀錄

```

結果					訊息
	員工編號	密碼	姓名	職稱	登入結果
1	2	12	陳季暄	業務經理	登入失敗次數:2
員工編號		次數			
1	2	2			
員工編號		登入時間		密碼	登入結果
1	2	2023-12-20 17:44:19.653		12	登入失敗次數:2

## ※小考

### 1. sp\_InsertEmployee

```

@EID nvarchar(255)='',
@Password nvarchar(255)='',
@Name nvarchar(255)='',
@Degree nvarchar(255)='',
@InsertResult nvarchar(255)='' output

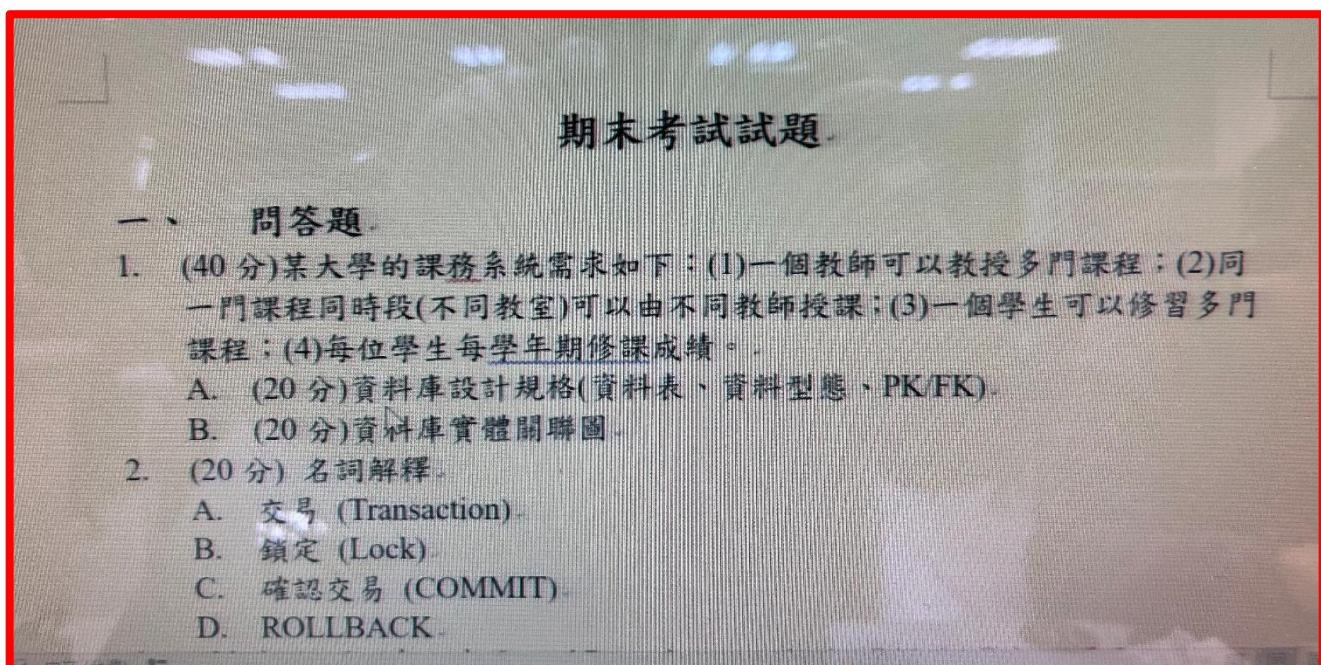
--(A)帳號不存在，新增成功
if NOT EXISTS(SELECT * FROM 員工 WHERE 員工編號=@EID)
begin
    insert into 員工(員工編號, 密碼, 姓名,名, 職稱) values
(@EID,@Password,@Name,' ',@Degree)
    set @InsertResult='帳號不存在，新增成功'
end
--(B)帳號存在，新增失敗
else
begin
    set @InsertResult='帳號存在，新增失敗'
end

```

```
ALTER PROCEDURE sp_InsertEmployee
-- Add the parameters for the stored procedure here
@EID nvarchar(255)='',
@Password nvarchar(255)='',
@Name nvarchar(255)='',
@Degree nvarchar(255)='',
@InsertResult nvarchar(255)=''' output
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
--(A)帳號不存在，新增成功
if NOT EXISTS(SELECT * FROM 員工 WHERE 員工編號=@EID)
begin
    insert into 員工(員工編號, 密碼, 姓名,名, 職稱) values (@EID,@Password,@Name, '',@Degree)
    set @InsertResult='帳號不存在，新增成功'
end
--(B)帳號存在，新增失敗
else
begin
    set @InsertResult='帳號存在，新增失敗'
end
END
```

## ●公佈期末考題目



# 12/26 第十六週

## ●各組期末報告

我做的預存程序報告部分：

### 1. sp\_修改訂貨主檔所有獎項、獎金

STEP 1 在訂貨主檔資料表裡新增「獎項」「獎金」

資料行名稱	資料類型	允許 Null
訂單編號	nvarchar(50)	<input type="checkbox"/>
客戶編號	nvarchar(50)	<input checked="" type="checkbox"/>
門市編號	nvarchar(50)	<input checked="" type="checkbox"/>
發票號碼	nvarchar(255)	<input checked="" type="checkbox"/>
日期	date	<input checked="" type="checkbox"/>
隨機碼	float	<input checked="" type="checkbox"/>
獎項	nvarchar(50)	<input checked="" type="checkbox"/>
獎金	nvarchar(50)	<input checked="" type="checkbox"/>

STEP 2 新增「統一發票兌獎號碼」資料表，並在裡面新增「中獎號碼」「獎項」「獎金」及新增中獎號碼相關資料(修改訂貨主檔發票號碼，以利結果顯示)

資料行名稱	資料類型	允許 Null
中獎號碼	nvarchar(50)	<input checked="" type="checkbox"/>
獎項	nvarchar(50)	<input checked="" type="checkbox"/>
獎金	nvarchar(50)	<input checked="" type="checkbox"/>

	中獎號碼	獎項	獎金
▶	07546773	特別獎	1000萬
	09251201	特獎	200萬
	04398272	頭獎	20萬

STEP 3 新增預存程序

USE [POS\_發票系統]

GO

\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_所有中獎號碼] Script Date: 2023/12/25

下午 09:26:13 \*\*\*\*\*

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =====

-- Author: <Author, ,Name>

```

-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[sp_所有中獎號碼]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
--獎項=0 獎金=0
update 訂貨主檔
set 獎項 = null,獎金=0

--update 特別獎--
update 訂貨主檔
set 獎項 = '特別獎',獎金='1000 萬'
from 訂貨主檔 inner join
(
--1.特別獎--
select vw_b.發票號碼,vw_b.獎項
from
(select 中獎號碼,獎項
from 統一發票兌獎號碼
where 獎項='特別獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,8)as 中獎號碼,獎項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
-----
--2.update 特獎--
update 訂貨主檔
set 獎項 = '特獎',獎金='200 萬'
from 訂貨主檔 inner join
(

```

--特獎--

```
select vw_b.發票號碼, vw_b.獎項
from
(select 中獎號碼, 奬項
from 統一發票兌獎號碼
where 奬項='特獎'
)vw_a
inner join
(select 發票號碼, right(發票號碼,8)as 中獎號碼, 奬項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
```

--3.update 頭獎--

```
update 訂貨主檔
set 奬項 = '頭獎', 奬金='20 萬'
from 訂貨主檔 inner join
(
--頭獎--
select vw_b.發票號碼, vw_b.獎項
from
(select 中獎號碼, 奬項
from 統一發票兌獎號碼
where 奬項='頭獎'
)vw_a
inner join
(select 發票號碼, right(發票號碼,8) as 中獎號碼, 奬項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
```

--4.update 二獎--

```
update 訂貨主檔
set 奬項 = '二獎', 奬金='4 萬'
from 訂貨主檔 inner join
(
--二獎--
select VW_B.發票號碼, VW_B.獎項
from
```

```

(
select vw_b.發票號碼, vw_b.獎項
from
(select 中獎號碼, 奬項
from 統一發票兌獎號碼
where 奬項='頭獎'
)vw_a
inner join
(select 發票號碼, right(發票號碼,8)as 中獎號碼, 奬項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)VW_A right join
(
select vw_b.發票號碼, vw_b.獎項
from
(select right(中獎號碼,7) as 二獎, 奬項
from 統一發票兌獎號碼
where 奬項='頭獎'
)vw_a
inner join
(select 發票號碼, right(發票號碼,7) as 二獎, 奬項
from 訂貨主檔
)vw_b on vw_a.二獎=vw_b.二獎
)VW_B
on VW_A.發票號碼=VW_B.發票號碼
where VW_A.發票號碼 is null
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
-----
--5.update 三獎--
update 訂貨主檔
set 奬項 = '三獎', 奬金='1 萬'
from 訂貨主檔 inner join
(
--三獎--
select VW_B.發票號碼, VW_B.獎項
from
(
select vw_b.發票號碼, vw_b.獎項
from
(select right(中獎號碼,7) as 中獎號碼 ,獎項

```

```

from 統一發票兌獎號碼
where 嘉獎項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,7) as 中獎號碼,獎項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)VW_A right join
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,6) as 三獎,獎項
from 統一發票兌獎號碼
where 嘉獎項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,6) as 三獎,獎項
from 訂貨主檔
)vw_b on vw_a.三獎=vw_b.三獎
)VW_B
on VW_A.發票號碼=VW_B.發票號碼
where VW_A.發票號碼 is null
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
-----
--6.update 四獎--
update 訂貨主檔
set 嘉獎項 = '四獎',獎金='4 千'
from 訂貨主檔 inner join
(
--四獎--
select VW_B.發票號碼,VW_B.獎項
from
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,6) as 中獎號碼 ,獎項
from 統一發票兌獎號碼
where 嘉獎項='頭獎'
)vw_a
inner join

```

```

(select 發票號碼,right(發票號碼,6) as 中獎號碼,獎項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)VW_A right join
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,5) as 四獎,獎項
from 統一發票兌獎號碼
where 獎項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,5) as 四獎,獎項
from 訂貨主檔
)vw_b on vw_a.四獎=vw_b.四獎
)VW_B
on VW_A.發票號碼=VW_B.發票號碼
where VW_A.發票號碼 is null
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼
-----
--7.update 五獎--
update 訂貨主檔
set 獎項 = '五獎',獎金='1 千'
from 訂貨主檔 inner join
(
--五獎--
select VW_B.發票號碼,VW_B.獎項
from
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,5) as 中獎號碼 ,獎項
from 統一發票兌獎號碼
where 獎項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,5) as 中獎號碼,獎項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)VW_A right join

```

```

(
select vw_b.發票號碼, vw_b.獎項
from
(select right(中獎號碼,4) as 五獎 ,獎項
from 統一發票兌獎號碼
where 嘍項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,4)as 五獎,獎項
from 訂貨主檔
)vw_b on vw_a.五獎=vw_b.五獎
)VW_B
on VW_A.發票號碼=VW_B.發票號碼
where VW_A.發票號碼 is null
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼

-----
--8.update 六獎--
update 訂貨主檔
set 嘍項 = '六獎',獎金='2百'
from 訂貨主檔 inner join
(
--六獎--
select VW_B.發票號碼,VW_B.獎項
from
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,4) as 中獎號碼 ,獎項
from 統一發票兌獎號碼
where 嘍項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,4) as 中獎號碼,獎項
from 訂貨主檔
)vw_b on vw_a.中獎號碼=vw_b.中獎號碼
)VW_A right join
(
select vw_b.發票號碼,vw_b.獎項
from
(select right(中獎號碼,3) as 六獎,獎項

```

```
from 統一發票兌獎號碼
where 獎項='頭獎'
)vw_a
inner join
(select 發票號碼,right(發票號碼,3) as 六獎,獎項
from 訂貨主檔
)vw_b on vw_a.六獎=vw_b.六獎
)VW_B
on VW_A.發票號碼=VW_B.發票號碼
where VW_A.發票號碼 is null
)vwC
on vwC.發票號碼=訂貨主檔.發票號碼

END
```

#### STEP 4 測試預存程序是否正確

exec sp\_所有中獎號碼

```
select 訂貨主檔.發票號碼,訂貨主檔.獎項,訂貨主檔.獎金  
from 訂貨主檔  
where 嘉獎項 is not null
```

## STEP 5 前後對比(修改前)

## 前後對比(修改後)

結果		訊息	
	發票號碼	獎項	獎金
1	QN-09251201	特獎	200萬
2	SX-07398272	三獎	1萬
3	TJ-98398272	三獎	1萬
4	SX-07546773	特別獎	1000萬
5	RN-63698272	四獎	4千
6	NX-54398272	二獎	4萬
7	NX-04398272	頭獎	20萬
8	NX-64412272	六獎	2百
9	QF-40448272	五獎	1千

## 2. sp\_修改獎項\_2(另一種寫法)

### 1. 預存程序

```
-- =====  
-- Template generated from Template Explorer using:  
-- Create Procedure (New Menu).SQL  
  
--  
-- Use the Specify Values for Template Parameters  
-- command (Ctrl-Shift-M) to fill in the parameter  
-- values below.  
  
--  
-- This block of comments will not be included in  
-- the definition of the procedure.  
-- =====  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
-- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
CREATE PROCEDURE sp_修改獎項_2  
    -- Add the parameters for the stored procedure here  
  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    -- Insert statements for procedure here  
--六獎--  
update 訂貨主檔  
set 獎項=vw.獎項  
from 訂貨主檔 inner join  
(  
select vw_a.發票號碼,vw_b.六獎 as 統一編號六獎,'六獎' as 獎項  
from  
(select 發票號碼,right(發票號碼,3) as 六獎,獎項 from 訂貨主檔)vw_a  
inner join
```

```
(select right(中獎號碼,3) as 六獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b  
on vw_a.六獎=vw_b.六獎  
)vw  
on vw.發票號碼=訂貨主檔.發票號碼
```

--五獎--

```
update 訂貨主檔  
set 嘉獎項=vw.獎項  
from 訂貨主檔 inner join  
(  
select vw_a.發票號碼, vw_b.五獎 as 統一編號五獎,'五獎' as 嘉獎項  
from  
(select 發票號碼,right(發票號碼,4) as 五獎,獎項 from 訂貨主檔)vw_a  
inner join  
(select right(中獎號碼,4) as 五獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b  
on vw_a.五獎=vw_b.五獎  
)vw  
on vw.發票號碼=訂貨主檔.發票號碼
```

--四獎--

```
update 訂貨主檔  
set 嘉獎項=vw.獎項  
from 訂貨主檔 inner join  
(  
select vw_a.發票號碼, vw_b.四獎 as 統一編號四獎,'四獎' as 嘉獎項  
from  
(select 發票號碼,right(發票號碼,5) as 四獎,獎項 from 訂貨主檔)vw_a  
inner join  
(select right(中獎號碼,5) as 四獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b  
on vw_a.四獎=vw_b.四獎  
)vw  
on vw.發票號碼=訂貨主檔.發票號碼
```

--三獎--

```
update 訂貨主檔  
set 嘉獎項=vw.獎項  
from 訂貨主檔 inner join  
(  
select vw_a.發票號碼, vw_b.三獎 as 統一編號三獎,'三獎' as 嘉獎項  
from  
(select 發票號碼,right(發票號碼,6) as 三獎,獎項 from 訂貨主檔)vw_a
```

```
inner join
(select right(中獎號碼,6) as 三獎 from 統一發票兌獎號碼 where 嘉項='頭獎')vw_b
on vw_a.三獎=vw_b.三獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--二獎--

```
update 訂貨主檔
set 嘉項=vw.嘉項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.二獎 as 統一編號二獎, '二獎' as 嘉項
from
(select 發票號碼,right(發票號碼,7) as 二獎,嘉項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,7) as 二獎 from 統一發票兌獎號碼 where 嘉項='頭獎')vw_b
on vw_a.二獎=vw_b.二獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--頭獎--

```
update 訂貨主檔
set 嘉項=vw.嘉項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.頭獎 as 統一編號頭獎, '頭獎' as 嘉項
from
(select 發票號碼,right(發票號碼,8) as 頭獎,嘉項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 頭獎 from 統一發票兌獎號碼 where 嘉項='頭獎')vw_b
on vw_a.頭獎=vw_b.頭獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--特獎--

```
update 訂貨主檔
set 嘉項=vw.嘉項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.特獎 as 統一編號特獎, '特獎' as 嘉項
from
```

```

(select 發票號碼,right(發票號碼,8) as 特獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 特獎 from 統一發票兌獎號碼 where 嘉獎='特獎')vw_b
on vw_a.特獎=vw_b.特獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

--特別獎--
update 訂貨主檔
set 嘉獎=vw.獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.特別獎 as 統一編號特別獎,'特別獎' as 嘉獎
from
(select 發票號碼,right(發票號碼,8) as 特別獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 特別獎 from 統一發票兌獎號碼 where 嘉獎='特別獎')vw_b
on vw_a.特別獎=vw_b.特別獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

END
GO

```

## 2. 修改程式碼

```

--六獎--
update 訂貨主檔
set 嘉獎=vw.獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.六獎 as 統一編號六獎,'六獎' as 嘉獎
from
(select 發票號碼,right(發票號碼,3) as 六獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,3) as 六獎 from 統一發票兌獎號碼 where 嘉獎='頭獎')vw_b
on vw_a.六獎=vw_b.六獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼


```

```

--五獎--
update 訂貨主檔
set 嘉獎=vw.獎項

```

```
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.五獎 as 統一編號五獎,'五獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,4) as 五獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,4) as 五獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b
on vw_a.五獎=vw_b.五獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--四獎--

```
update 訂貨主檔
set 嘉獎項=vw.嘉獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.四獎 as 統一編號四獎,'四獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,5) as 四獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,5) as 四獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b
on vw_a.四獎=vw_b.四獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--三獎--

```
update 訂貨主檔
set 嘉獎項=vw.嘉獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.三獎 as 統一編號三獎,'三獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,6) as 三獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,6) as 三獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b
on vw_a.三獎=vw_b.三獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼
```

--二獎--

```
update 訂貨主檔
```

```

set 嘉獎項=vw.獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.二獎 as 統一編號二獎,'二獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,7) as 二獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,7) as 二獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b
on vw_a.二獎=vw_b.二獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

--頭獎--
update 訂貨主檔
set 嘉獎項=vw.獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.頭獎 as 統一編號頭獎,'頭獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,8) as 頭獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 頭獎 from 統一發票兌獎號碼 where 嘉獎項='頭獎')vw_b
on vw_a.頭獎=vw_b.頭獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

--特獎--
update 訂貨主檔
set 嘉獎項=vw.獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼, vw_b.特獎 as 統一編號特獎,'特獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,8) as 特獎,獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 特獎 from 統一發票兌獎號碼 where 嘉獎項='特獎')vw_b
on vw_a.特獎=vw_b.特獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

--特別獎--

```

```

update 訂貨主檔
set 嘉獎項=vw.嘉獎項
from 訂貨主檔 inner join
(
select vw_a.發票號碼,vw_b.特別獎 as 統一編號特別獎,'特別獎' as 嘉獎項
from
(select 發票號碼,right(發票號碼,8) as 特別獎,嘉獎項 from 訂貨主檔)vw_a
inner join
(select right(中獎號碼,8) as 特別獎 from 統一發票兌獎號碼 where 嘉獎項='特別獎')vw_b
on vw_a.特別獎=vw_b.特別獎
)vw
on vw.發票號碼=訂貨主檔.發票號碼

```

結果顯示：(一筆一筆更新)

```

select 訂貨主檔.發票號碼,訂貨主檔.嘉獎項
from 訂貨主檔
where 嘉獎項 is not null

```

(7 個資料列受到影響)

(6 個資料列受到影響)

(5 個資料列受到影響)

(4 個資料列受到影響)

(2 個資料列受到影響)

(1 個資料列受到影響)

(1 個資料列受到影響)

(1 個資料列受到影響)

	發票號碼	嘉獎項
1	QN-09251201	特獎
2	SX-07398272	三獎
3	TJ-98398272	三獎
4	SX-07546773	特別獎
5	RN-63698272	四獎
6	NX-54398272	二獎
7	NX-04398272	頭獎
8	NX-64412272	六獎
9	QF-40448272	五獎

# 資料庫實作 期末考

## ●期末考複習

名詞解釋：

1. 「交易」(Transaction)：設定多個連續的資料操作必須全部執行成功，否則即回復到未執行任何操作的狀態

2. 交易可保證：對資料庫的多項修改只有兩種結果

(1)全部修改完成

(2)一項也不修改

3. 更改成功並確定交易(COMMIT)幾個要點

交易通常是由 BEGIN TRAN 敘述開始

(1)一直執行到 COMMIT TRAN(成功)或 ROLLBACK TRAN(失敗)敘述時才結束

(2)若沒有任何錯誤：執行 COMMIT TRAN 表示確認交易

(3)有任何的錯誤：執行 ROLLBACK TRAN 取消交易，並回復至交易執行前的狀況

4. ROLLBACK

(1)利用一個暫時的交易紀錄檔來回復異動資料

(2)在交易開始時，SQL Server 會將要更改的相關資料一一鎖定並進行更改，同時也會建立一個暫時的交易紀錄檔，來存放交易中更改資料的過程及內容

5. 交易的 4 大特性(ACID)

(1)單元性(Atomicity)：整個交易中的敘述會視為一個執行單位，要就全部成功，不然就全部取消

(2)一致性(Consistency)：在交易完成後，資料庫的內容必須全部更新妥當(包括各個資料表、索引等均處於一致狀態)，而且仍然具備正確性及完整性(例如要符合資料表的 CHECK、FOREIGN KEY 等各項條件約束)

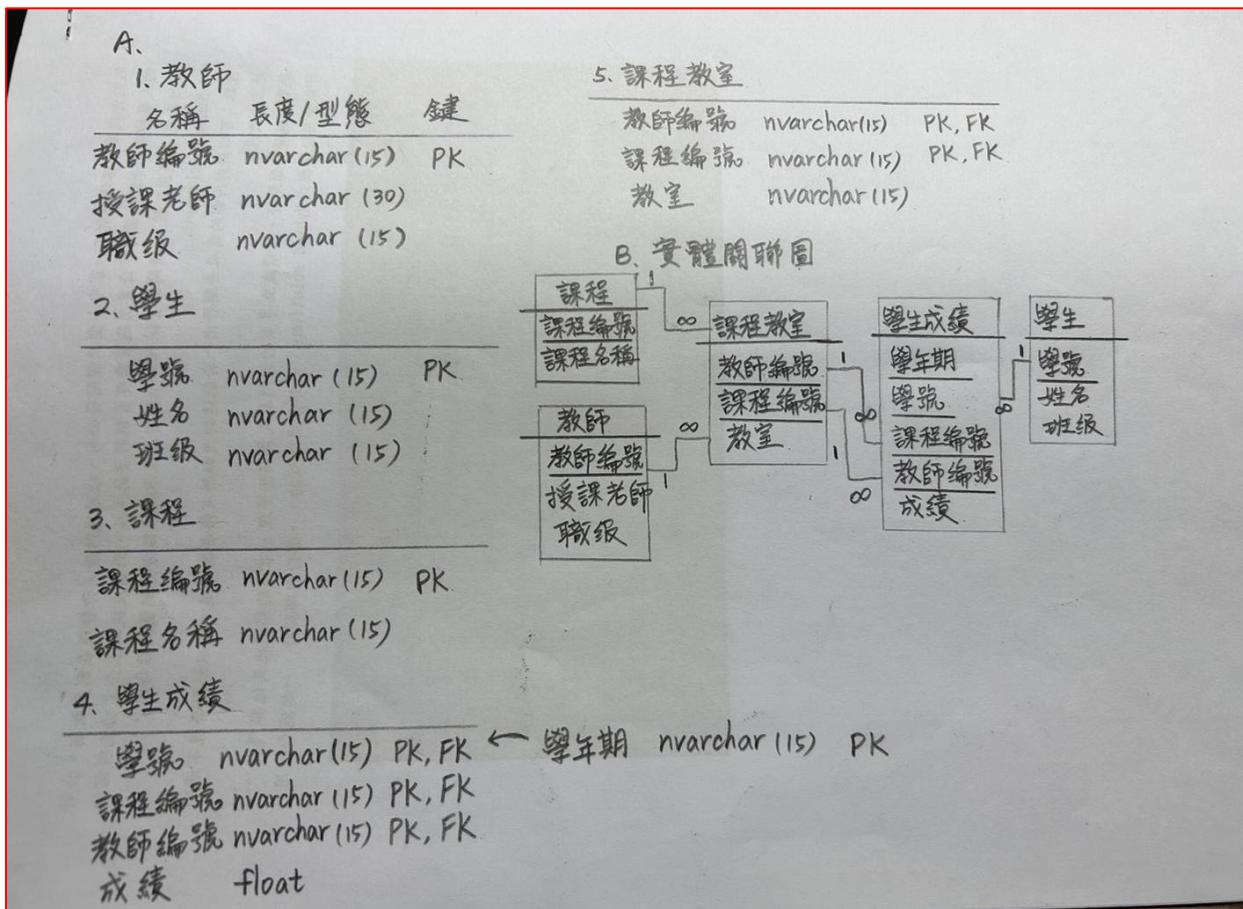
(3)隔離性(Isolation)：在交易中所使用到的資料，必須與其他同時在進行的交易做適度隔離，在 SQL Server 是用鎖定(Lock)資料的方式來隔離交易

(4)永久性(Durability)：交易一旦確認之後，其所做的資料修改將視為永久性的，無法再用 ROLLBACK 回復了。即使發生連線中斷、系統當機等狀況，其更改也不會自動回復。

# 資料庫實作 期末考

1. 某大學的課務系統需求如下：

- (1) 一個教師可以教授多門課程
  - (2) 同一門課程同時段(不同教室)可以由不同教師授課
  - (3) 一個學生可以修習多門課程
  - (4) 每位學生每學年期修課成績
- a. 資料庫設計規格(資料表、資料型態、PK/FK)
  - b. 資料庫實體關聯圖



## 資料庫實作 期末考

### 2. 名詞解釋

#### (1) 交易(Transaction)

設定多個連續的資料操作必須全部執行成功，否則即回復到未執行任何操作的狀態。

#### (2) 鎖定(Lock)

將指定的資料暫時鎖起來供我們使用，以防止該資料被別人修改或讀取。

#### (3) 確認交易(COMMIT)

交易通常是由 BEGIN TRAN 敘述開始

(1) 一直執行到 COMMIT TRAN(成功)或 ROLLBACK TRAN(失敗)敘述時才結束

(2) 若沒有任何錯誤：執行 COMMIT TRAN 表示確認交易

(3) 有任何的錯誤：執行 ROLLBACK TRAN 取消交易，並回復至交易執行前的狀況

#### (4) ROLLBACK

利用一個暫時的交易紀錄檔來回復異動資料。

在交易開始時，SQL Server 會將要更改的相關資料**一一鎖定**並進行更改，同時也會**建立一個暫時的交易紀錄檔**，來存放交易中更改資料的過程及內容。

```
begin tran
    -- 資料操作(新增、刪除、修改)
    -- 甲=甲-100
    -- 乙=乙+100
    if @@ERROR>0
        begin
            rollback tran
            set @執行結果=@@ERROR
        end      --of tran
    else
        begin
            commit tran      --of tran
            set @執行結果='執行成功'
        end      --of tran
```

# 資料庫實作 期末考

## ●補充資料

**交易的隔離等級**

- 隔離等級由低到高共分為 5 等

1. Read uncommitted
  - 完全沒有隔離效果，即使要讀取的資料已被其他交易使用且尚未 COMMIT 也沒關係。
  - 讀取到的資料隨時都可能被別人更改或刪除(機率很高)，而且也很可能讀到尚未全部修改完成的資料
2. Read committed
  - 不允許讀取尚未 COMMIT 的資料，因此該資料後來被更動的機率就比較小，而且也不會讀取到交易尚未完成的資料。
  - 讀取完資料後就和 Read uncommitted 一樣，不會在乎該資料是否還會被別人更改。因此在同一個交易中，每次讀取到的資料可能會不相同

**交易的隔離等級**

3. Repeatable read
  - 在交易中所讀取到的資料將不允許別人更改或刪除，以保證在交易中每次都可以讀取到相同的內容。但別人仍然可以在該資料表中新增記錄，所以在交易中重複讀取資料時，可能會突然多出一、二筆記錄來
4. Snapshot
  - 自 SQL Server 2005 開始才新增的等級
  - 交易進行前會先建立資料快照 (Snapshot)，而在交易期間所讀取的資料則均來自快照，因此即使實際資料又被別人異動過(新增/修改/刪除)，也不影響該交易中讀取資料的一致性(每次都會讀取到相同的內容)
  - 雖然從 Snapshot 中讀取的資料可能不是最新的，但至少是前後一致的，也不會像 Repeatable read 在重複讀取資

**交易的隔離等級**

5. Serializable
  - 此等級會將要使用的資料表全部鎖定，不允許別人來修改、刪除、或新增資料。由於必須等交易完成後，其他交易才能使用這些資料表
  - Serializable 的並行性最低，所有要使用到這些資料的交易必須先排隊，然後一個個循序地進行

**交易的隔離等級**

- 在下表中，我們針對上述 5 種不同程度的交易需求來做比較：

隔離等級	保證不會讀取到別人交易中的資料	保證已讀取的資料不被更改	保證使用到的資料表不會被更改
Read uncommitted	X	X	X
Read committed	○	X	X
Repeatable read	○	○	X
Snapshot	○	○(限快照中)	○(限快照中)
Serializable	○	○	○

## 資料庫實作 期末考

2024/1/2 第十七週

●期末考+各組期末報告

2024/1/9 第十八週

●總結及結算