

資料庫期末 (老師版)

第四次練習 1. --試試主鍵，例：jobs中job_id

--建立主鍵，例：jobs中job_id
/* 2 */ ALTER TABLE dbo.jobs
ADD PRIMARY KEY (job_id)

刪除主鍵，例：jobs中job_id
/* 1 */ ALTER TABLE dbo.jobs
drop [PK__jobs__6E32B6A58D0E0F1C]

建立主鍵，例：jobs中job_id
/* 2 */ ALTER TABLE dbo.jobs
ADD PRIMARY KEY (job_id)

2. 建立外來鍵，例：employee中pub_id
ALTER TABLE [dbo].[employee]
add CONSTRAINT fk_pubs_restrict
FOREIGN KEY ([pub_id])
REFERENCES [dbo].[publishers]([pub_id])

移除外來鍵，例：employee中pub_id
--移除外來鍵，例：employee中pub_id
/* 4 */ ALTER TABLE [dbo].[employee]
drop CONSTRAINT fk_pubs_restrict

3. --試試外來鍵
建立外來鍵，例：employee中job_id
/* 3 */ ALTER TABLE [dbo].[employee]
add CONSTRAINT fk_job_restrict
FOREIGN KEY ([job_id])
REFERENCES [dbo].[jobs]([job_id])
移除外來鍵，例：employee中job_id
/* 4 */ ALTER TABLE [dbo].[employee]
drop CONSTRAINT fk_job_restrict

4. 改變資料型態
例5：publishers中Country char(30) 變 char(20)
ALTER TABLE [dbo].[publishers]
ALTER COLUMN [country] varchar(20) NULL
before

After

5. --試試改變資料型態
例：jobs中job_desc (50) 變 char(40)
ALTER TABLE [dbo].[jobs]
ALTER COLUMN [job_desc]
varchar(40) NULL
Before

After

6. 新增欄位

例6：因為要報稅，employee 中增加一欄ssn char(10)
/* 6 */ ALTER TABLE [dbo].[employee]
ADD ssn char(10) null
Before

after 7. 加入限制

例7：增加ssn 身分證字號限制
/* 7 */ ALTER TABLE [dbo].[employee]
add CONSTRAINT check_ssn
CHECK (SSN like '[A-Z][12][0-9][0-9][0-9][0-9]
[0-9][0-9][0-9][0-9]')
before
after
Try

8. 刪除限制

例8：刪除ssn 身分證字號限制
/* 8 */ ALTER TABLE [dbo].[employee]
drop CONSTRAINT [check_ssn]

9. 刪除欄位

例9：刪除ssn
/* 9 */ ALTER TABLE [dbo].[employee]
drop column ssn

10. --試試欄位

-- employee 中增加一欄連絡人職稱char(30)
/* 6 */ ALTER TABLE [dbo].[employee]
ADD 連絡人職稱 char(30) null
Before
After

--加入限制連絡人職稱在('業務','業務經理','助理')
中
/* 7 */ ALTER TABLE [dbo].[employee]
add CONSTRAINT check_連絡人職稱
CHECK (連絡人職稱 IN ('業務','業務經理','助理'))

11. 增加預設值

例10：估價缺少資料數量欄位的預設值為1
--估價缺少資料數量欄位的預設值為1
/* 10 */ ALTER TABLE [dbo].[估價]
ADD CONSTRAINT default_qty
DEFAULT 1 FOR [數量]
Before

After

12. --試試預設

例：估價缺少資料新增估價日期的預設值為今天
/* 10 */ ALTER TABLE [dbo].[估價]
add 估價日期 datetime

資料庫期末（老師版）

DEFAULT(getdate())

輸入資料試試

13. 刪除預設

--刪除數量欄位的預設值為1的預設

/* 11 */ALTER TABLE [dbo].[估價]

drop CONSTRAINT default_qty

14. 建立規則

--估價增加數量>=1

/* 12 */ALTER TABLE [dbo].[估價]

ADD CONSTRAINT rule_qty

check ([數量]>= 1)

2. --更新的欄位不只一個請使用逗號分隔

UPDATE dbo.估價 SET 數量=4, 日期= getdate()

where 數量 = 3

Before

After

3. --將資料表符合條件的記錄刪除掉

DELETE FROM dbo.估價 WHERE 日期 is null

Before

After

P8-19~p8-27

1. 使用「*」符號代表所有欄位

<ex>查詢 所有客戶資料 *

SELECT 客戶.*

FROM 客戶;

第五次練習

1. --新增記錄 單筆輸入 整批輸入 p7-20

INSERT INTO dbo.估價

VALUES (7,10,'2018/11/29')

2. SELECT column1, column2 FROM table WHERE conditions

<ex>查詢 所有客戶的名稱、聯絡人、聯絡人職稱和電話資料

SELECT 客戶.公司名稱, 客戶.聯絡人, 客戶.聯絡人職稱, 客戶.電話

FROM 客戶;

3. 欄位（單一欄位）中的資料重複，擁有相同值，在SELECT指令只需加上DISTINCT指令，就只會顯示其中一筆記錄資料。

<ex>查詢 產品的成本有多少種

SELECT DISTINCT 產品資料.單價

FROM 產品資料;

4. WHERE條件子句

WHERE條件的欄位如果是字串需要加上單引號

日期/時間字串需要使用「#」符號括起

Like

<ex> 查詢產品名稱擁有子字串"起司"

SELECT * FROM 產品資訊

WHERE 產品 LIKE '%起司%'<ex> 查詢產品名稱是以任何字串開頭，結尾為"起司"

SELECT DISTINCT 產品資料.*

FROM 產品資料

WHERE (((產品資料.[產品]) Like '%起司'));<ex>查詢產品的單位數量是以子字串"12"

SELECT * FROM 產品

WHERE單位數量 LIKE '%12%'<ex>查詢單位數量不是以子字串"12" SELECT * FROM 產品資料

WHERE 單位數量 not LIKE '*12'*5. AND與OR多條件查詢

<ex>查詢產品資訊的單位數量是以子字串"12"且產品名稱有"油"子字串

SELECT 產品資料.產品, *

FROM 產品資料

WHERE (((產品資料.[單位數量]) Like '%12%') AND ((產品資料.產品) Like "%油%"))6. NOT運算子可以搭配前述子句，

取得與條件相反的查詢結果（優先順序以括號為優先）

查詢產品資訊的產品名稱有"Nano"子字串或入庫日期有"7"子字串，這2個條件使用括號括起，而且定價需要大於等於5000的記錄

SELECT * FROM 產品資訊

資料庫期末（老師版）

WHERE (產品名稱 LIKE '*Nano*' OR 入庫日期 LIKE '*7*') AND 定價>=5000

7. ORDER BY子句指定排序欄位，就可以由小到大進行排序

<ex>查詢定價大於等於3500元的記錄，並且使用定價欄位由小到大進行排序

SELECT * FROM 產品資訊

WHERE 定價>=3500 ORDER BY 定價 ASC (升冪)-- DESC (降冪)

8. BETWEEN AND子句可以定義SELECT指令WHERE條件子句的範圍，範圍值可以使用文字、數值或日期/時間

<ex> 查詢1992年1月1日到12月31日雇用的員工資料

SELECT 員工.*

FROM 員工

WHERE (((員工.雇用日期) Between #1/1/1992# And #12/31/1992#));

9. IN運算子可以設定一串文字或數值清單

<ex>查詢單價(成本)為38, 43.9, 和123.79的產品資料

SELECT * FROM 產品資料

WHERE 單價 IN (38,43.9,123.79)問題

【a】 產品單價有多少種?欄位不要重複

SELECT DISTINCT 產品資料.單價

FROM 產品資料

ORDER BY 產品資料.單價;

【b】 查詢所有客戶的名稱、聯絡人和電話資料，其中連絡人職稱為董事長或業務

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話

FROM 客戶

WHERE (((客戶.連絡人職稱)="董事長" Or (客戶.連絡人職稱)="業務"));

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話

FROM 客戶

WHERE (客戶.連絡人職稱) in ("董事長","業務"); 【c】 查詢所有客戶的名稱、聯絡人和電話資料，其中公司名稱有”中”的資料

SELECT 客戶.公司名稱, 客戶.連絡人, 客戶.電話, 客戶.公司名稱

FROM 客戶

where 公司名稱 like "*中*"重要

在Pubs中想知道哪些已簽約的作家和他們所寫的書籍名稱

select * from [dbo].[authors] where [contract] =1select [au_lname],[au_fname], [contract],[title]

from [dbo].[authors],[dbo].[titles], [dbo].[titleauthor]

where [dbo].[authors].[au_id]=[dbo].[titleauthor].au_id and [dbo].[titles].title_id=[dbo].[titleauthor].title_id and

[contract] =1

聚合函數

【d】 查詢貢獻最好的顧客

SELECT 客戶.公司名稱, Sum([單價]*[數量]) AS total

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.公司名稱

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;

或

SELECT 客戶.公司名稱, Sum([單價]*[數量]) AS total into temp

FROM 客戶,訂貨明細,訂貨主檔

where (客戶.客戶編號 = 訂貨主檔.客戶編號) and (訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼)

GROUP BY 客戶.公司名稱

ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;將顧客分 A B C 等級

SELECT 公司名稱, total, rank=

case

when total>58000 then 'a_class'

資料庫期末（老師版）

```
when total<58000 and total>28000 then 'b_class'
```

```
else 'c_class'
```

```
end
```

```
FROM temp
```

【e】查詢顧客所購買最多的產品

```
SELECT 客戶.公司名稱, 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM 產品資料 INNER JOIN ((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) ON 產品資料.產品編號 = 訂貨明細.產品編號
```

```
GROUP BY 客戶.公司名稱, 產品資料.產品
```

```
ORDER BY 客戶.公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;select [公司名稱],[送貨地址]
```

```
from [dbo].[客戶] left join [dbo].[訂貨主檔]
```

```
on [dbo].[客戶].[客戶編號] =[dbo].[訂貨主檔].[客戶編號]
```

```
where [公司名稱]='ppp'
```

【f】查詢販賣最好的產品

```
SELECT 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
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
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC; 【g】 1995/1/1~1995/5/31銷售最好產品
```

```
SELECT 產品資料.產品, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM 訂貨主檔 INNER JOIN (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) ON 訂貨主檔.訂單號碼 = 訂貨明細.訂單號碼
```

```
WHERE (((訂貨主檔.訂單日期)>=#1/1/1995#) AND ((訂貨主檔.訂單日期)<=#5/31/1995#))
```

```
GROUP BY 產品資料.產品
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC; 【h】查詢業績最好的員工
```

```
SELECT 員工.姓名, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM 員工 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 員工.員工編號 = 訂貨主檔.員工編號
```

```
GROUP BY 員工.姓名
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC; 【i】查詢販賣最好的類別
```

```
SELECT 產品類別.類別名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM 產品類別 INNER JOIN (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) ON 產品類別.類別編號 = 產品資料.類別編號
```

```
GROUP BY 產品類別.類別名稱
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC; 【j】查詢販賣最好產品的供應商
```

```
SELECT 供應商.供應商, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM 供應商 INNER JOIN (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) ON 供應商.供應商編號 = 產品資料.供應商編號
```

```
GROUP BY 供應商.供應商
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;
```

【k】查詢販賣最好產品的貨運公司

```
SELECT 貨運公司.貨運公司名稱, Sum([訂貨明細].[單價]*[訂貨明細].[數量]) AS total
```

```
FROM (貨運公司 INNER JOIN 訂貨主檔 ON 貨運公司.貨運公司編號 = 訂貨主檔.送貨方式) 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 貨運公司.貨運公司名稱
```

```
ORDER BY Sum([訂貨明細].[單價]*[訂貨明細].[數量]) DESC;作業 2 十個查詢 自訂問題 下達SQL查詢語言 結果
```

第六次練習 p10-2~10-18

chap 10 視界與資料庫程式設計

建立檢視表

```
CREATE VIEW [View_2] AS
```

資料庫期末（老師版）

```
(SELECT 學號, 姓名, YEAR(GETDATE()) - YEAR([出生日期]) AS 歲
FROM [dbo].[students])
```

--欄子集視界(垂直取)

--取作家姓名、電話和簽約

```
create view view1 as
```

```
SELECT au_lname, au_fname, phone, contract
FROM dbo.authors
```

--列欄子集視界(平行取)

--取沒有簽約作家所有資料

```
create view view2 as
```

```
SELECT *
FROM dbo.authors
where [contract] <> 1
```

--合併子集視界

--取作家所編著的書本，其中顯示姓名、電話、簽約和書本名稱資料

```
create view view3 as
```

```
SELECT au_lname, au_fname, phone, contract, title
FROM dbo.authors, [dbo].[titles],[dbo].[titleauthor]
where [dbo].[authors].au_id=[dbo].[titleauthor].au_id
and [dbo].[titleauthor].title_id=[dbo].[titles].title_id
```

--統計子集視界

--顯示作家姓名與所編著書本的本數

```
create view view4 as
```

```
SELECT au_lname, au_fname, count(titles.title_id) '本數'
FROM dbo.authors, [dbo].[titles],[dbo].[titleauthor]
where [dbo].[authors].au_id=[dbo].[titleauthor].au_id
and [dbo].[titleauthor].title_id=[dbo].[titles].title_id
group by au_lname, au_fname
```

總表

view	欄子集視界	列子集視界	合併視界	統計視界				
	含主鍵Ex1	不含主鍵Ex2	含主鍵Ex3	不含主鍵Ex4	含主鍵Ex5	不含主鍵Ex6	含主鍵Ex7	不含主鍵Ex8
insert	○	X	○	x	x	x	X	X
update	○	主X	○	○	○	○	x	X
delete	○	主X	○	○	○	x	x	X

--Ex1欄子集視界含主鍵

```
create view ex1 as
```

```
SELECT au_id, au_lname, au_fname, phone, contract
FROM dbo.authors
```

--新增

```
insert ex1(au_id, au_lname, au_fname, phone, contract)
values ('899-46-2037','Tsao','Wen-Yu', '801 826-0752', 0)
```

--更新

```
update ex1 set phone ='801 826-0753'
```

```
where au_lname ='Tsao'
```

--更新主鍵

```
update ex1 set au_id ='899-46-2036'
```

```
where au_lname ='Tsao'
```

--刪除

```
DELETE FROM ex1 WHERE au_lname='Tsao'--Ex2欄子集視界不含主鍵
```

資料庫期末（老師版）

```
create view ex2 as
SELECT au_lname, au_fname, phone, contract
FROM dbo.authors
--新增
insert ex2(au_lname, au_fname, phone, contract)
values ('Tsao','Wen-Yu', '801 826-0752', 0)
--更新
update ex2 set phone ='801 826-9999'
where au_lname ='White'
--更新主鍵xx
update ex2 set au_id ='899-46-9999'
where au_lname ='white'
--刪除
DELETE FROM ex2 WHERE au_id='899-46-2037'(x)
DELETE FROM ex2 WHERE au_lname='Tsao'([?])
--Ex3列子集視界含主鍵
create view ex3 as
SELECT *
FROM dbo.authors
where [contract] <> 1
--新增
insert ex3(au_id, au_lname, au_fname, phone, contract)
values ('899-46-2037','Tsao','Wen-Yu', '801 826-0752', 0)
--更新
update ex3 set phone ='801 826-0753'
where au_lname ='Tsao'
--更新主鍵
update ex3 set au_id ='899-46-2036'
where au_lname ='Tsao'
--刪除
DELETE FROM ex3 WHERE au_lname='Tsao'--Ex4列子集視界不含主鍵
create view ex4 as
SELECT au_lname, au_fname, phone, contract
FROM dbo.authors
where [contract] <> 1
--新增
insert ex4(au_lname, au_fname, phone, contract)
values ('Tsao','Wen-Yu', '801 826-0752', 0)
--更新
update ex4 set phone ='801 826-0753'
where au_lname ='Tsao'
--更新主鍵
update ex4 set au_id ='899-46-2036'
where au_lname ='Tsao'
--刪除
DELETE FROM ex4 WHERE au_lname='Tsao'--Ex5 合併子集視界含主鍵
create view ex5 as
SELECT authors.au_id as 'authors.au_id', titleauthor.au_id as 'titleauthor.au_id', dbo.titles.title_id, au_lname,
au_fname, phone, contract, title
FROM dbo.authors, [dbo].[titles],[dbo].[titleauthor]
where [dbo].[authors].au_id=[dbo].[titleauthor].au_id
and [dbo].[titleauthor].title_id=[dbo].[titles].title_id
```

資料庫期末（老師版）

--新增

```
insert ex5--(authors.au_id, titleauthor.au_id, au_lname, au_fname, phone, contract)
values ('899-46-2037', '899-46-2037', 'BU1032', 'Tsao', 'Wen-Yu', '801 826-0752', 0, 'The Busy Executives
Database Guide')
```

--更新

```
update ex5 set phone = '801 826-0753'
where au_lname = 'Green'
```

--更新主鍵

```
update ex5 set au_id = '899-46-2036'
where au_lname = 'Tsao'
```

--刪除

```
DELETE FROM ex5 WHERE au_lname='Tsao'--Ex6 合併子集視界不含主鍵
```

create view ex6 as

```
SELECT au_lname, au_fname, phone, contract, title
FROM dbo.authors, [dbo].[titles],[dbo].[titleauthor]
where [dbo].[authors].au_id=[dbo].[titleauthor].au_id
and [dbo].[titleauthor].title_id=[dbo].[titles].title_id
```

--新增

```
insert ex6(au_id, au_lname, au_fname, phone, contract, title)
values ('899-46-2037', 'Tsao', 'Wen-Yu', '801 826-0752', 0, 'Is Anger the Enemy?')
```

--更新

```
update ex6 set phone = '801 826-0753'
where au_lname = 'Tsao'
```

--刪除

```
DELETE FROM ex6 WHERE au_lname='Tsao' use pubs
```

--

CREATE VIEW titles_view

AS

```
SELECT title, type, price, pubdate
FROM titles
```

select * from titles_view--列欄子集視界(平行取)

--在Students基底關聯表建立學生成績GPA欄位大於等於.0的GPA_View視界：

CREATE VIEW GPA_View AS

SELECT * FROM Students

WHERE GPA >= 3.0

select * from GPA_View select * from column_View

where not au_lname = 'Smith' CREATE VIEW row_View AS

SELECT * FROM authors where contract=1

select * from row_view--列欄子集視界

--在Students基底關聯表建立學生成績GPA欄位大於等於.0，而且只有sid、name和GPA三個欄位的Score_View視界：

CREATE VIEW Score_View AS

SELECT sid, name, GPA FROM Students

WHERE GPA >= 3.0 use pubs

CREATE VIEW con_row_View AS

SELECT au_lname, phone, address FROM dbo.authors where contract=1--合併視界

--建立合併視界Std_Class_View，可以顯示學生的選課資料：

CREATE VIEW Std_Class_View AS

SELECT Classes.sid, Students.name, Classes.eid,

Instructors.name AS professor,

Classes.c_no, Courses.title, Classes.room, time

FROM Students, Courses, Instructors, Classes

WHERE Students.sid = Classes.sid

資料庫期末（老師版）

```
and Courses.c_no = Classes.c_no
and Instructors.eid = Classes.eid
select name,professor,title,room,time from dbo.Std_Class_View說明錯誤
--Ex5合併視界含主鍵
create view ex5 as
SELECT authors.au_id,au_lname, au_fname, phone, contract,title
FROM  dbo.authors, dbo.titles, dbo.titleauthor
where dbo.authors.au_id= dbo.titleauthor.au_id
and dbo.titles.title_id=dbo.titleauthor.title_id
錯誤原因是authors 中作家的編號重複 而不是合併視界含主鍵create view view4 as
SELECT  au_lname, au_fname, count(titles.title_id) '本數'
FROM  dbo.authors, [dbo].[titles],[dbo].[titleauthor]
where [dbo].[authors].au_id=[dbo].[titleauthor].au_id
and [dbo].[titleauthor].title_id=[dbo].[titles].title_id
group by au_lname, au_fname
```

第七次練習流程控制

chap 11 視界與資料庫程式設計

BEGIN...END

<ex1>抓取172-32-1176號作家姓名

Declare @customer varchar(60)

select @customer =[au_lname]+' ' +[au_fname]

from [dbo].[authors]

where [au_id]='172-32-1176'

select @customer

顯示員工姓名,性別和工作年資

select [fname],[lname],right([emp_id],1) 性別, 年資=year(getdate())-year([hire_date])

from [dbo].[employee]

<ex2>--進階顯示員工姓名,性別和工作年資

select [fname],[lname], 性別=case right([emp_id],1) when 'M' then N'男' else N'女' end, year(getdate())-
year([hire_date]) as 年資

from [employee]

IF...ELSE 條件判斷

抓取編號大於409-56-7008最後一位作家姓名

Declare @customer varchar(60),@id varchar(11)

set @id ='409-56-7008'

if @id >='409-56-7008'

begin

select @id=au_id, @customer =[au_lname]+' ' +[au_fname]

from [dbo].[authors]

order by au_id

end

select @id,@customer

進階

顯示編號大於409-56-7008作家姓名

select au_id, [au_lname]+' ' +[au_fname]

from [dbo].[authors]

where au_id > '409-56-7008'

order by au_idCASE

資料庫期末（老師版）

```
<ex3>select DATEPART ( dw, getdate())-1
SELECT CASE DATEPART ( dw, getdate())-1
        WHEN 1 THEN N'星期一'
        WHEN 2 THEN '星期二'
        WHEN 3 THEN '星期三'
        WHEN 4 THEN '星期四'
        WHEN 5 THEN '星期五'
        WHEN 6 THEN '星期六'
        WHEN 7 THEN '星期天'

        End
datepart      縮寫
year    yy, yyyy
quarter    qq, q
monthmm, m
dayofyear    dy, y
day    dd, d
week    wk, ww
--將書籍庫存管理系統中書籍英文分類呈現中文分類
use pubs
SELECT title ,type,'<' +
        CASE type
            WHEN 'business' THEN N'商業'
            WHEN 'mod_cook' THEN N'現代烹飪'
            WHEN 'popular_comp' THEN N'暢銷電腦'
            WHEN 'psychology' THEN N'心理學'
            when 'trad_cook' then N'傳統烹飪'
            ELSE N'未分'
        END + N'類>' AS 類別
FROM titles
ORDER BY 類別Ex4-7 顯示銷售客戶排行榜銷售金額
<ex4>7進階分為高中低的公司名稱和銷售金額的資料
select  CompanyName,sum(UnitPrice*Quantity) as total, '<此顧客為' +
        case
            when sum(UnitPrice*Quantity)>30000 then '高等'
            when sum(UnitPrice*Quantity)<10000 then '低等'
            else '中等'
        end +'>' as 等級
from dbo.Customers,dbo.[Order Details],dbo.Orders
where dbo.Customers.CustomerID=dbo.Orders.CustomerID
and dbo.Orders.OrderID=dbo.[Order Details].OrderID
group by CompanyName
--order by sum(UnitPrice*Quantity) descselect [公司名稱],sum([單價]*[數量]) as total, '<此顧客為' +
        case
            when sum([單價]*[數量])>30000 then 'A等'
            when sum([單價]*[數量])<10000 then 'C等'
            else 'B等'
        end +'>' as 等級 into temp
from [dbo].[客戶],[dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[客戶].[客戶編號]=[dbo].[訂貨主檔].客戶編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].[訂單號碼]
group by [公司名稱]select [公司名稱], substring([等級],6,2) rank  from tempuse northwind
SELECT CategoryName,'<' +
```

資料庫期末（老師版）

```
CASE CategoryName
  WHEN 'Beverages' THEN N'飲料'
  WHEN 'Condiments' THEN N'佐料'
  WHEN 'Confections' THEN N'甜食'
  WHEN 'Dairy Products' THEN N'日常品'
  when 'Grains/Cereals' then N'穀粒'
  when 'Meat/Poultry' then N'肉'
  when 'Produce' then N'農產品'
  when 'Seafood' then N'海產'
  ELSE N'未分'
END + N'類>' AS 類別 into temp
FROM Categories
ORDER BY 類別 desc
select [產品],[類別名稱], '<' +
  CASE [類別名稱]
    WHEN '飲料' THEN N'Beverages'
    WHEN '調味品' THEN N'Condiments'
    WHEN '點心' THEN N'Confections'
    WHEN '日用品' THEN N'Dairy Products'
    when '穀類/麥片' then N'Grains/Cereals'
    when '肉/家禽' then N'Meat/Poultry'
    when '特製品' then N'Produce'
    when '海鮮' then N'Seafood'
    ELSE N'未分'
  END + N'類>' AS 類別
from [dbo].[產品資料],[dbo].[產品類別]
where [dbo].[產品資料].類別編號=[dbo].[產品類別].類別編號<ex5>--顯示員工姓名,性別和工作年資
declare @temp table (id char(50),fname char(50),lname char(50), 性別 nchar(2), 年資 int)
insert @temp select emp_id, [fname],[lname],case right([emp_id],1) when 'M' then '男' else '女' end,
year(getdate())-year([hire_date])
from [dbo].[employee]
select * from @tempWHILE 迴圈
--while 顯示Inside WHILE LOOP on TechOnTheNet.com 10次
DECLARE @site_value INT;
SET @site_value = 1;
WHILE @site_value <= 10
BEGIN
  PRINT convert(char(3),@site_value)+'Hello world.';
  SET @site_value = @site_value + 1;
END;
PRINT 'Done WHILE LOOP on Hello world.'--northwind 產品編號為單數的產品名稱
DECLARE @id int, @name varchar(50)
SET @id = 0
WHILE @id < 500
  BEGIN
    SET @id = @id + 1
    IF @id % 6 = 1 PRINT '.....'
    IF @id % 2 = 0 CONTINUE
    SELECT @name =ProductName
      FROM dbo.Products WHERE ProductID = @id
    IF @@ROWCOUNT = 0
      BEGIN
```

資料庫期末（老師版）

```
PRINT '*** The End ***'
BREAK
END
PRINT CAST(@id AS CHAR(2)) + ' -- ' + @name
END GOTO 跳躍控制
```

第八次練習Sp and trigger

建立SP p10-35-10-40

<ex6> 在 Pubs 中寫一副程式 當主程式輸入0 顯示沒有簽約作家所有資料

use pubs

--副程式參數

CREATE PROCEDURE myproc1 @para bit AS

select * from authors where contract = @para

go

--主程式

exec myproc1 1

exec myproc1 0

CREATE PROCEDURE myproc1 @para char(14) AS

select * from temp where [等級]= @para

更改sp

若想找高等級顧客

ALTER PROCEDURE [dbo].[myproc1] @para char(4) AS

select * from temp where [等級] like '%' + @para + '%myproc1 '高等'

如何動態的找到顧客等級資料

Sol1

1. Select 存成temp
2. 利用 SP 找到高等顧客
3. 若資料庫異動 則無法跟著動 Sol2

1. 建立view
2. 利用 SP 找到高等顧客
3. 若資料庫異動 則跟著動Sol1

try!!

Step 1 利用 SP 找到高等顧客

ALTER PROCEDURE [dbo].[myproc1] @para char(3) AS

select * from temp where [等級] like '%' + @para + '%'

<例如> 原本加美留學中心的訂購資料

```
select [公司名稱],[單價],[數量],[單價]*[數量] total
from [dbo].[客戶],[dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[客戶].[客戶編號]=[dbo].[訂貨主檔].客戶編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].[訂單號碼]
and [公司名稱]='加美留學中心'
```

Step 2 更新資料加美留學中心

1. 知道加美留學中心的客戶編號
select * from [dbo].[客戶] where [公司名稱]='加美留學中心'
2. 知道加美留學中心的訂單號碼
3. select [dbo].[訂貨明細].[訂單號碼],[數量] from [dbo].[訂貨主檔],[dbo].[訂貨明細] where [dbo].[訂貨主檔].[訂單號碼]=[dbo].[訂貨明細].訂單號碼 and [客戶編號]='QUEEN'

資料庫期末（老師版）

41. 改訂單號碼10253的數量

Step 3

```
update [dbo].[訂貨明細] set [數量]=1 where [訂單號碼]=10372
myproc1 '高等'
```

合作金庫 34101.15 <此顧客為高等>

加美留學中心 30226.1 <此顧客為高等>解

Create view v1 as

```
select [公司名稱],sum([單價]*[數量]) as total, '<此顧客為' +
    case
        when sum([單價]*[數量])>30000 then 'A等'
        when sum([單價]*[數量])<10000 then 'C等'
        else 'B等'
    end +'>' as 等級
from [dbo].[客戶],[dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[客戶].[客戶編號]=[dbo].[訂貨主檔].客戶編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].[訂單號碼]
group by [公司名稱]select * from v1
order by [公司名稱]create procedure myproc2 @para char(4) AS
select * from v1 where [等級] like '%' + @para + '%myproc2 '高等'
```

加美留學中心不再是高等 找找看中等

sp加密

<ex7> --在 Pubs 中寫一加密副程式插入一筆新書資料

```
create PROCEDURE MyProc2
    @param1 char(6), @param2 char(10),@param3 char(10) , @para4 datetime
WITH ENCRYPTION /*加密*/
AS INSERT titles (title_id, title, [type], [pubdate])
VALUES (@param1, @param2, @param3, @para4)
GO
select * from titles
--main
EXEC MyProc2 'aaa','aaaaaaaaaa', 'business','2017/12/25'
EXEC MyProc2 'bbb','bbbb', 'mod_cook','2015/4/4'
```

使用output

<ex8>在 Pubs 中如何使用output寫一副程式找出某本書售出總本數和總金額

```
create PROCEDURE MyProc3
@param1 char(80), @param2 money output, @param3 money output
AS
SELECT @param2 = SUM(qty),@param3 =sum(price*qty)
FROM titles, sales
where titles.title_id = sales.title_id and title= @param1
GO
--main
DECLARE @sum int ,@tot money
EXEC MyProc3 'Is Anger the Enemy?',@sum OUTPUT, @tot output
print 'Is Anger the Enemy? '+'的數量='+cast(@sum as char(10))+ '本 總金額= '+cast(@tot as char(10))
<ex9>在 Pubs 中寫一副程式輸入作家所住的州，取得作家地址
CREATE PROCEDURE proc4
```

資料庫期末（老師版）

```
@state char(2), @地址 varchar(100) OUTPUT
AS SELECT @地址=address +' '+city+ ' ' +state
FROM authors
WHERE state = @state
IF @@rowcount > 0
    RETURN 0 /* 如果查詢到則傳回0 */
ELSE
    RETURN 1 /* 沒有查到就傳回1 */
GO
--主程式
DECLARE @ret int, @地址 varchar(100)
EXEC @ret=proc4 'CA', @地址 OUTPUT
/* 用@ret 接收傳回值*/
IF @ret = 0
    PRINT @地址
ELSE
    PRINT '找不到！'
```

宣告區域表格變數

<try>在 Pubs 中寫一副程式取得所有作家地址

create PROCEDURE proc4

@state char(2)

AS

declare @all table (au_lname varchar(50), au_fname varchar(50), address varchar(50), city varchar(50), state varchar(2))

insert @all SELECT au_lname, au_fname, address, city, state

FROM authors

WHERE state = @state

select * from @all

IF @@rowcount > 0

RETURN 0 /* 如果查詢到則傳回*/

ELSE

RETURN 1 /* 沒有查到就傳回*/

GO

--主程式

DECLARE @ret int

declare @all table (au_lname varchar(50), au_fname varchar(50), address varchar(50), city varchar(50), state varchar(2))

EXEC @ret=proc4 'CA'

/* 用@ret 接收傳回值*/

select @ret

IF @ret = 0

PRINT '找到！'

ELSE

PRINT '找不到！'

呼叫另一副程式

<ex10>如何呼叫另一副程式

CREATE PROCEDURE proc3

AS PRINT 'Proc3: at level ' + CAST(@@NESTLEVEL AS CHAR)

GO CREATE PROCEDURE proc2

AS PRINT 'Proc2 start: at level ' + CAST(@@NESTLEVEL AS CHAR)

EXEC proc3

資料庫期末（老師版）

```
PRINT 'Proc2 end: at level ' + CAST(@@NESTLEVEL AS CHAR)
GO CREATE PROCEDURE proc1
AS PRINT 'Proc1 start: at level ' + CAST(@@NESTLEVEL AS CHAR)
EXEC proc2
PRINT 'Proc1 end: at level ' + CAST(@@NESTLEVEL AS CHAR)
GO
```

--主程式

```
EXEC proc1
```

Trigger p10-35

觸發程序（Trigger）：一種特殊用途的預存程序，是主動執行的程序，當資料表操作符合指定的條件時，就會自動執行觸發程序。Use bob

use bob

--step 1在資料表中建立trigger

```
Create Trigger monitor_bookstores
```

```
On Bookstores
```

```
For Delete
```

```
As delete from [Orders]
```

```
Where Orders.no in (select no from deleted)
```

--step 2 去掉關聯

--step 3 刪除bookstores資料表中no = 'f'

```
delete bookstores where no='f'
```

```
go
```

--看看bookstores and orders資料

```
select * from bookstores
```

```
select * from ordersCreate Trigger monitor_books
```

```
On Books
```

```
For Delete
```

```
As delete from Orders
```

```
Where Orders.id in (select id from deleted)
```

```
Create Trigger monitor_titles
```

```
On titles
```

```
For Delete
```

```
As delete from Orders
```

```
Where Orders.no in (select no from deleted)Create Trigger monitor_books
```

```
On Books
```

```
For Delete
```

```
As delete from Orders
```

```
Where Orders.id in (select id from deleted)Pivot 更進階
```

暖身

```
SELECT LastName + ' ' + FirstName as 員工姓名,
```

```
case month(BirthDate)
```

```
when 01 then N'一月'
```

```
when 02 then N'二月'
```

```
when 03 then N'三月'
```

```
when 04 then N'四月'
```

```
when 05 then N'五月'
```

```
when 06 then N'六月'
```

```
when 07 then N'七月'
```

```
when 08 then N'八月'
```

```
when 09 then N'九月'
```

```
when 10 then N'十月'
```

資料庫期末（老師版）

```
when 11 then N'十一月'
when 12 then N'十二月'
end AS 月份
FROM dbo.Employees
Pivot 員工use northwind
--step 1 根據員工和年月份加總銷售金額
select [dbo].[員工].[員工編號], year([訂單日期]) as 年, month([訂單日期]) as 月,sum([單價]*[數量]) as 銷售金額 into
temp_new
from [dbo].[員工], [dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[員工].員工編號=[dbo].[訂貨主檔].員工編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].訂單號碼
group by [dbo].[員工].員工編號,year([訂單日期]), month([訂單日期])--step 2 將要的格式存在temp_all
select [姓名] as 員工姓名, 年,
case 月
when 01 then N'一月'
when 02 then N'二月'
when 03 then N'三月'
when 04 then N'四月'
when 05 then N'五月'
when 06 then N'六月'
when 07 then N'七月'
when 08 then N'八月'
when 09 then N'九月'
when 10 then N'十月'
when 11 then N'十一月'
when 12 then N'十二月'
end AS 月份 , cast(format(銷售金額, '##,##0.00') as money) 銷售金額 into temp_all
from [dbo].[員工],temp_new
where [dbo].[員工].員工編號=[dbo].[temp_new].[員工編號]合併step1 and step2
select [dbo].[員工].姓名, year([訂單日期]) as 年,case month([訂單日期]) when 01 then N'一月'
when 02 then N'二月'
when 03 then N'三月'
when 04 then N'四月'
when 05 then N'五月'
when 06 then N'六月'
when 07 then N'七月'
when 08 then N'八月'
when 09 then N'九月'
when 10 then N'十月'
when 11 then N'十一月'
when 12 then N'十二月'
end AS 月份 ,sum([單價]*[數量]) as 銷售金額 into temp_new
from [dbo].[員工], [dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[員工].員工編號=[dbo].[訂貨主檔].員工編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].訂單號碼
group by [dbo].[員工].姓名,year([訂單日期]),month([訂單日期])
--step 3 建立pivot
CREATE TABLE all_sheet (員工姓名 char(32), 一月 money, 二月 money, 三月 money, 四月 money, 五月 money, 六月
money,
七月 money, 八月 money, 九月 money, 十月 money, 十一月 money, 十二月 money)
--
合併step 1, step 2 and step 3
```

資料庫期末（老師版）

```
SELECT 姓名,年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月
FROM (select [dbo].[員工].姓名, year([訂單日期]) as 年,case month([訂單日期]) when 01 then N'一月'
            when 02 then N'二月'
            when 03 then N'三月'
            when 04 then N'四月'
            when 05 then N'五月'
            when 06 then N'六月'
            when 07 then N'七月'
            when 08 then N'八月'
            when 09 then N'九月'
            when 10 then N'十月'
            when 11 then N'十一月'
            when 12 then N'十二月'
        end AS 月份 ,sum([單價]*[數量]) as 銷售金額
from [dbo].[員工], [dbo].[訂貨主檔],[dbo].[訂貨明細]
where [dbo].[員工].員工編號=[dbo].[訂貨主檔].員工編號
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].訂單號碼
group by [dbo].[員工].姓名,year([訂單日期]),month([訂單日期]))
AS p
PIVOT
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
AS pt--設定資料庫等級
ALTER DATABASE Northwind
SET compatibility_level = 110;
sp_dbcmtlevel northwind, 90
-- step 4 根據轉置欄位資料產生欄位
SELECT 員工姓名,年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月
-- 2.1 欲進行轉置的資料來源
FROM (SELECT 員工姓名,年, 銷售金額,月份 FROM dbo.temp_all)
-- 2.2 GroupCol (群組欄位) [?] Employee欄位
-- AggregationCol (彙總欄位) [?] Hours欄位
-- PivotCol (轉置欄位) [?] Kind欄位 )
AS p
PIVOT
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
AS pt

--最後寫sp
create procedure [dbo].[sp_cmd] @year int AS
-- 根據轉置欄位資料產生欄位
SELECT 員工姓名,年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月
-- 2.1 欲進行轉置的資料來源
FROM (SELECT 員工姓名,年, 銷售金額,月份 FROM dbo.temp_all)
-- 2.2 GroupCol (群組欄位) == Employee欄位
-- AggregationCol (彙總欄位) == Hours欄位
-- PivotCol (轉置欄位) == Kind欄位 )
AS p
PIVOT
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
AS pt
where 年=@year
--執行
```


資料庫期末（老師版）

```
exec sp_cmd 1997, '張瑾雯 Mary','林美麗 Chris','郭國城 Steven'
```

員工姓名	年	一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
張瑾雯 Mary	1997	7331.6		2504.6		5493.9		240	9168.25		6112.65		19997.88
		8461.5	12920.15	4106.7	16077.25								
郭國城 Steven	1997	13703.4		3891	3867.2		5707.35		6041.25		2082	6144.6	7853.05
		13839.29	642	1990	928	try							

```
select *,iif(一月 is null, 0,一月)+ iif(二月 is null, 0, 二月)+ iif(三月 is null, 0, 三月)+ iif(四月 is null, 0, 四月)+ iif(五月 is null, 0, 五月)+ iif(六月 is null,0, 六月) +
iif(七月 is null, 0, 七月)+ iif(八月 is null, 0, 八月)+ iif(九月 is null, 0, 九月)+ iif(十月 is null, 0, 十月)+
iif(十一月 is null, 0, 十一月)+ iif(十二月 is null, 0, 十二月) as total from finally create procedure final @got int as
select *,iif(一月 is null, 0,一月)+ iif(二月 is null, 0, 二月)+ iif(三月 is null, 0, 三月)+ iif(四月 is null, 0, 四月)+ iif(五月 is null, 0, 五月)+ iif(六月 is null,0, 六月) +
iif(七月 is null, 0, 七月)+ iif(八月 is null, 0, 八月)+ iif(九月 is null, 0, 九月)+ iif(十月 is null, 0, 十月)+
iif(十一月 is null, 0, 十一月)+ iif(十二月 is null, 0, 十二月) as total from finally
where [年] =@got
```

Pivot 客戶

--step 1 根據員工和年月份加總銷售金額

```
select [dbo].[客戶].客戶編號, year([訂單日期]) as 年, month([訂單日期]) as 月,sum([單價]*[數量]) as 銷售金額 into
temp_new_c
```

```
from [dbo].[客戶],[dbo].[訂貨主檔],[dbo].[訂貨明細]
```

```
where [dbo].[客戶].客戶編號=[訂貨主檔].客戶編號
```

```
and [dbo].[訂貨主檔].訂單號碼=[dbo].[訂貨明細].訂單號碼
```

```
group by [dbo].[客戶].客戶編號,year([訂單日期]), month([訂單日期])--step 2 將要的格式存在temp_all_c
```

```
select [公司名稱], 年,
```

```
case 月
```

```
when 01 then N'一月'
```

```
when 02 then N'二月'
```

```
when 03 then N'三月'
```

```
when 04 then N'四月'
```

```
when 05 then N'五月'
```

```
when 06 then N'六月'
```

```
when 07 then N'七月'
```

```
when 08 then N'八月'
```

```
when 09 then N'九月'
```

```
when 10 then N'十月'
```

```
when 11 then N'十一月'
```

```
when 12 then N'十二月'
```

```
end AS 月份 , cast(format(銷售金額, '##,##0.00') as money) 銷售金額 into temp_all_c
```

```
from [dbo].[客戶],temp_new_c
```

```
where [dbo].[客戶].客戶編號=[dbo].[temp_new_c].[客戶編號]-- step 3 根據轉置欄位資料產生欄位
```

```
SELECT [公司名稱],年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月
```

-- 2.1 欲進行轉置的資料來源

```
FROM (SELECT [公司名稱],年, 銷售金額,月份 FROM dbo.temp_all_c)
```

-- 2.2 GroupCol (群組欄位) [?] Employee欄位

-- AggregationCol (彙總欄位) [?] Hours欄位

-- PivotCol (轉置欄位) [?] Kind欄位)

AS p

PIVOT

```
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
```

```
AS ptcreate procedure [dbo].[sp_cmd_c] @year int AS
```

```
SELECT [公司名稱],年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月
```

資料庫期末（老師版）

```
-- 2.1 欲進行轉置的資料來源
FROM (SELECT [公司名稱],年, 銷售金額,月份 FROM dbo.temp_all_c)
-- 2.2 GroupCol (群組欄位) [?] Employee欄位
-- AggregationCol (彙總欄位) [?] Hours欄位
-- PivotCol (轉置欄位) [?] Kind欄位 )
AS p
PIVOT
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
AS pt
where 年=@year
Pivot 客戶更進階create procedure [dbo].[sp_cmd_c_fin] @year int AS
SELECT [公司名稱],年, 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月, iif(一月 is null, 0,一
月)+ iif(二月 is null, 0, 二月)+ iif(三月 is null, 0, 三月)+ iif(四月 is null, 0, 四月)+ iif(五月 is null, 0, 五月)+ iif(六月 is
null,0, 六月) +
iif(七月 is null, 0, 七月)+ iif(八月 is null, 0, 八月)+ iif(九月 is null, 0, 九月)+ iif(十月 is null, 0, 十月)+
iif(十一月 is null, 0, 十一月)+ iif(十二月 is null, 0, 十二月) as total
```

```
-- 2.1 欲進行轉置的資料來源
FROM (SELECT [公司名稱],年, 銷售金額,月份 FROM dbo.temp_all_c)
-- 2.2 GroupCol (群組欄位) [?] Employee欄位
-- AggregationCol (彙總欄位) [?] Hours欄位
-- PivotCol (轉置欄位) [?] Kind欄位 )
AS p
PIVOT
(sum(銷售金額) FOR 月份 IN ( 一月, 二月, 三月, 四月, 五月, 六月, 七月, 八月, 九月, 十月, 十一月, 十二月) )
AS pt
where 年=@year Pivot 成績更進階
SELECT *,(計算構概論*3+ 線性代數, 離散數學, 程式語言, 網頁程式設計, 物件導向程式設計, 資料庫管理系統, 物件導向
分析)/ FROM ( select [姓名],[名稱], [分數]
from [dbo].[學生],[dbo].[課程], [dbo].[成績]
where [dbo].[學生].[學號]=[dbo].[成績].[學號] and [dbo].[課程].[課程編號]=[dbo].[成績].[課程編號]) AS p
PIVOT
(sum([分數]) FOR [名稱] IN (計算構概論, 線性代數, 離散數學, 程式語言, 網頁程式設計, 物件導向程式設計, 資料庫管理
系統, 物件導向分析)) AS pt
```

第九次練習 function

```
-- 找出年訂購最多的書籍名稱
use pubs
CREATE FUNCTION GetToptitle
(@year char(5))
RETURNS char(80)
BEGIN
    DECLARE @name char(80)
    SELECT TOP 1 @name=title
    FROM sales JOIN titles
        ON sales.title_id = titles.title_id
    WHERE DATEPART (YYYY, ord_date) = @year
    GROUP BY title
    ORDER BY SUM(qty) DESC
    IF @@ROWCOUNT = 0
        RETURN 0
    RETURN @name
END
```

```
GO--使用
SELECT dbo.GetToptitle('1993') AS '書籍'--依售價
範圍查詢書籍
CREATE FUNCTION 依售價範圍查詢書籍
(@由 money, @到 money)
RETURNS TABLE
RETURN (SELECT title, price
        FROM dbo.titles
        WHERE price >=@由 AND price <= @到)
GOSELECT *
FROM 依售價範圍查詢書籍(10, 15)
ORDER BY price
```