

## 合併理論與實作

# 合併 ( JOIN )

- 定義：是指將兩關聯表  $R_1$  與  $R_2$  依合併條件合併成一個新的關聯表  $R_3$ 。

表示符號▶▶  $\bowtie$

假設▶▶ 假設  $P$  為合併條件，以  $R_1 \bowtie_p R_2$  表示此合併運算。

關聯式代數▶▶  $R_3 = R_1 \bowtie_p R_2$

SQL語法▶▶

```
SELECT *
FROM A表格,B表格
WHERE 條件P
```

## 概念圖

- 由兩個或兩個以上的關聯（表格），透過某一欄位的共同值域所組合而成的，以建立出一個新的資料表

R1

A	B	C
A1	B1	C1
A2	B2	C1
A3	B3	C2

(a)

R2

C	D	E
C1	D1	E1
C2	D2	E2

(b)

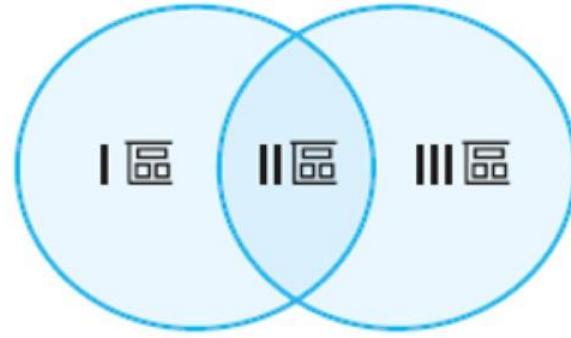
$R1 \bowtie R2 = R3$

R1.A	R1.B	R1.C	R2.D	R2.E
A1	B1	C1	D1	E1
A2	B2	C1	D1	E1
A3	B3	C2	D2	E2

(c)



## 合併的分類-來源合併： ( 需要F.K.→P.K. )



- 1. Inner Join ( 內部合併 )
  - 如果查詢目前老師有開設的課程，則會使用到「內部合併」。如圖的II區。
- 2. Outer Join ( 外部合併 )
  - (1) 如果要查詢尚未開課的老師，則會使用到「左外部合併」。如圖的I區。
  - (2) 如果查詢有哪些課程尚未被老師開課，則會使用到「右外部合併」。如圖的III區。
- 3. Join Itself ( 自我合併 )

## 內部合併 ( INNER JOIN )

- 定義：

- 內部合併 ( Inner Join ) 又稱為「條件式合併 ( Condition Join ) 」
- 也就是將「卡氏積」展開後的結果，在兩個資料表之間加上「限制條件」，亦即在兩個資料表之間找到「對應值組」才行
- 而外部合併 ( Outer Join ) 則無此規定。
- 這裡所指的「限制條件」是指兩個資料表之間的某一欄位值的「關係比較」。如表13-2所示：

◆表13-2 資料表關係比較運算子

運算子	條件式說明
= (等於)	學生成績單.成績=60
<> (不等於)	學生成績單.成績<>60
< (小於)	學生成績單.成績<60
<= (小於等於)	學生成績單.成績<=60
> (大於)	學生成績單.成績>60
>= (大於等於)	學生成績單.成績>=60

## 作法»

- 透過SELECT指令WHERE部分的等式，即對等合併（Equi-Join）。

```
From A ,B
```

```
Where (A.c=B.c)
```

- 透過SELECT指令FROM部分的INNER JOIN。即自然合併（Natural Join）；又稱為內部合併（Inner Join）。

```
From A INNER JOIN B
```

```
ON A.c=B.c
```

## 範例

- 假設有兩個資料表，分別是「學生表」與「課程表」，現在欲將這兩個資料表進行「內部合併」
- 因此，我們必須要透過相同的欄位值來進行關聯，亦即「學生表」的「課號」參考到「課程表」的「課號」

學生表

	<u>學號</u>	姓名	<u>課號</u>
#1	S0001	張三	C001
#2	S0002	李四	C002

課程表

<u>課號</u>	課名	學分數
C001	手機	3
C002	機器人	3
C003	物聯網	2

## 解答一：分析

從圖13-13中，我們就可以將此條關聯線條寫成：

**學生表.課號=課程表.課號**

- 因此，我們將這兩個資料表進行「卡氏積」運算，其結果如下圖所示
- 接下來，從展開後的紀錄中，找尋哪幾筆紀錄具有符合「學生表.課號=課程表.課號」的條件
- 亦即「學生表」的「課號」等於「課程表」的「課號」

「學生表」的屬性

「課程表」的屬性

	學號	姓名	學生表.課號	課程表.課號	課名	學分數
#1	S0001	張三	C001	C001	手機	3
#2	S0001	張三	C001	C002	機器人	3
#3	S0001	張三	C001	C003	物聯網	2
#4	S0002	李四	C002	C001	手機	3
#5	S0002	李四	C002	C002	機器人	3
#6	S0002	李四	C002	C003	物聯網	2

## 解答二：撰寫「SQL」程式碼

1. 第一種做法：(Equi-Join最常用)

```
Select 學號,姓名,課程表.課號,課程名稱,學分數
```

```
From 學生表,課程表
```

```
Where 學生表.課號=課程表.課號
```

2. 第二種做法：INNER JOIN

```
SELECT 學號,姓名,課程表.課號,課程名稱,學分數
```

```
FROM 學生表 INNER JOIN 課程表
```

```
ON 學生表.課號=課程表.課號
```

## 執行結果

結果 #1 (5×2)				
學號	姓名	課號	課名	學分數
S0001	張三	C001	手機程式	3
S0002	李四	C002	機器人程式	3

### 解答三：綜合分析

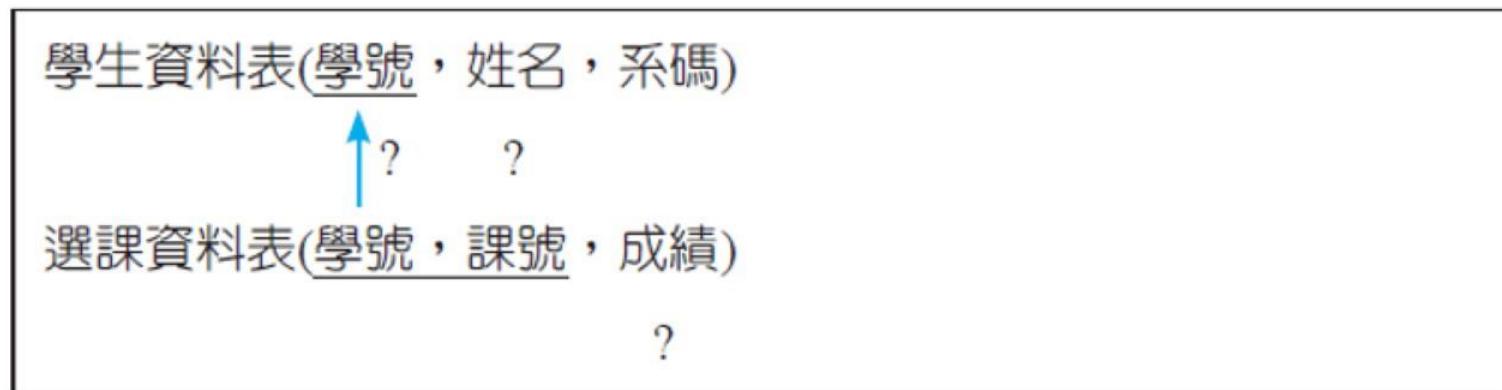
- 當我們欲查詢的欄位名稱是來源於兩個或兩個以上的資料表時，如下表所示：

學生資料表			選課資料表		
學生資料表 (3×4)			選課資料表 (3×5)		
學號	姓名	系碼	學號	課號	成績
S0001	張三	D001	S0001	C001	67
S0002	李四	D001	S0001	C002	85
S0003	王五	D002	S0001	C003	100
S0004	李安	D003	S0002	C004	89
			S0003	C002	90



## 步驟1：辨識「目標屬性」及「相關表格」。

- 1. 目標屬性：學號，姓名，課號，平均成績
- 2. 相關表格：學生資料表，選課資料表



## 步驟2：將相關表格進行「卡氏積」。

```
use ch13_db;  
SELECT *  
FROM 學生資料表 AS A, 選課資料表 AS B
```

- 執行結果：總共產生20筆記錄及6個欄位數。

👉 學號	姓名	系碼	👉 學號	👉 課號	成績
S0001	張三	D001	S0001	C001	67
S0002	李四	D001	S0001	C001	67
S0003	王五	D002	S0001	C001	67
S0004	李安	D003	S0001	C001	67
S0001	張三	D001	S0001	C002	85
S0002	李四	D001	S0001	C002	85
S0003	王五	D002	S0001	C002	85
S0004	李安	D003	S0001	C002	85
S0001	張三	D001	S0001	C003	100
S0002	李四	D001	S0001	C003	100
S0003	王五	D002	S0001	C003	100
S0004	李安	D003	S0001	C003	100
S0001	張三	D001	S0002	C004	89
S0002	李四	D001	S0002	C004	89
S0003	王五	D002	S0002	C004	89
S0004	李安	D003	S0002	C004	89
S0001	張三	D001	S0003	C002	90
S0002	李四	D001	S0003	C002	90
S0003	王五	D002	S0003	C002	90
S0004	李安	D003	S0003	C002	90

## 步驟3：進行「合併（JOIN）」

- 本題以「內部合併」為例，亦即在Where中加入「相關表格」的關聯性。

```
use ch13_db;  
SELECT *  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號
```

學號	姓名	系碼	學號	課號	成績
S0001	張三	D001	S0001	C001	67
S0001	張三	D001	S0001	C002	85
S0001	張三	D001	S0001	C003	100
S0002	李四	D001	S0002	C004	89
S0003	王五	D002	S0003	C002	90

## 步驟4：加入限制條件(成績大於或等於70分者)。

```
use ch13_db;  
SELECT *  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號  
And B.成績>=70
```

學號	姓名	系碼	學號	課號	成績
S0001	張三	D001	S0001	C002	85
S0001	張三	D001	S0001	C003	100
S0002	李四	D001	S0002	C004	89
S0003	王五	D002	S0003	C002	90

## 步驟5：投影使用者欲「輸出的欄位名稱」。

```
use ch13_db;  
SELECT A.學號, 姓名, 課號, 成績  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號  
And B.成績>=70
```

結果 #1 (4×4)			
学號	姓名	課號	成績
S0001	張三	C002	85
S0001	張三	C003	100
S0002	李四	C004	89
S0003	王五	C002	90

## 步驟6：使用群組化及聚合函數。

```
use ch13_db;  
  
SELECT A.學號, 姓名, AVG(成績) AS 平均成績  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號  
And B.成績>=70  
GROUP BY A.學號, 姓名
```

執行結果▶

群組化(前)				群組化(後)		
				學生資料表 (3x3)	平均成績	
學號	姓名	課號	成績	學號	姓名	
50001	張三	C002	85	學生資料表 (3x3)		
50001	張三	C003	100	50001	張三	92.5000
50002	李四	C004	89	50002	李四	89.0000
50003	王五	C002	90	50003	王五	90.0000

$$\rightarrow (85+100)/2 = 92.5$$

## 步驟7：使用「聚合函數」之後，再進行篩選條件(各人平均成績大於或等於90分者)。

```
use ch13_db;  
SELECT A.學號, 姓名, AVG(成績) AS 平均成績  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號  
And B.成績>=70  
GROUP BY A.學號, 姓名  
HAVING AVG(成績)>=90
```

執行結果▶

學生資料表 (3×2)		
學號	姓名	平均成績
S0001	張三	92.5000
S0003	王五	90.0000

## 步驟8：依照某一欄位或「聚合函數」結果，來進行「排序」(由低分到高分)。

```
use ch13_db;  
  
SELECT A.學號, 姓名, AVG(成績) AS 平均成績  
FROM 學生資料表 AS A, 選課資料表 AS B  
WHERE A.學號=B.學號  
And B.成績>=70  
GROUP BY A.學號, 姓名  
HAVING AVG(成績)>=90  
ORDER BY AVG(成績) ASC
```

執行結果▶

學生資料表 (3×2)		
學號	姓名	平均成績
S0003	王五	90.0000
S0001	張三	92.5000

## 結論

- 「學生表」與「課程表」在經過「卡氏積」之後，會展開成各種組合，並產生龐大紀錄，但大部分都是不太合理的配對組合。
- 所以，我們就必須要再透過「內部合併（Inner Join）」來取出符合「限制條件」的紀錄。
- 因此，我們從上面的結果，可以清楚得知「內部合併」的結果就是「卡氏積」的子集合。

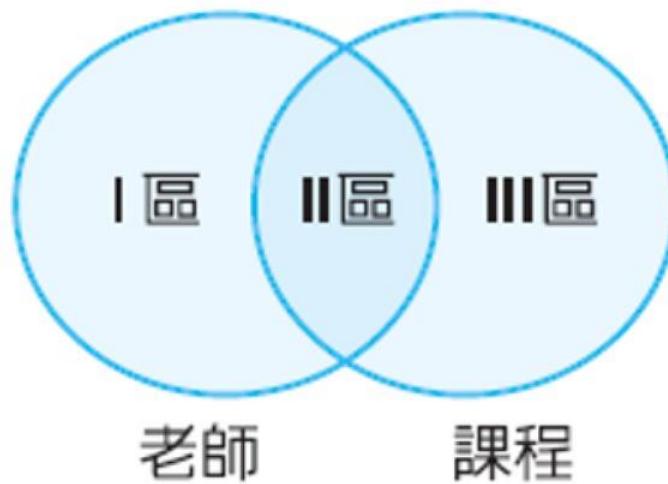
內部合併

卡氏積

## 外部合併 ( OUTER JOIN )

- 定義：當在進行合併 ( Join ) 時，不管紀錄是否符合條件，都會被列出其中一個資料表的所有紀錄時，則稱為「外部合併」。此時不符合條件的紀錄就會被預設為NULL值。即左右兩邊的關聯表，不一定要有對應值組。

- 用途：是應用在異質性分散式資料庫上的整合運算，其好處是不會讓資訊遺漏。
- 分類：可分為兩種：



## (一) 左外部合併 (Left Outer Join，以 表示)。

範例 » 如果要查詢尚未開課的老師，則會使用到「左外部合併」。

如圖13-24中的 I 區。

## (二) 右外部合併 (Right Outer Join，以 表示)。

範例 » 如果查詢有哪些課程尚未被老師開課，則會使用到「右外部合併」。

如圖13-24中的 III 區。

## 格式 »

```
SELECT *
FROM 表格A [RIGHT | LEFT ] [OUTER ][JOIN] 表格B
ON 表格A.PK=表格B.FK
```

# 範例 1 左外部合併

- 假設有兩個資料表，分別是「老師資料表」與「課程資料表」，現在欲查詢每一位老師開課資料，其中包括尚未開課的老師也要列出。

老師資料表(A)		課程資料表(B)		
	老師編號	老師姓名	課程代碼	課程名稱
#1	T0001	張三	C001	資料庫
#2	T0002	李四	C002	資料結構
#3	T0003	王五	C003	程式設計
#4	T0004	李安	C004	系統分析

## 解答

- 1. 分析：當兩個關聯做合併運算時，會保留第一個關聯（左邊）中的所有值組（Tuples）。找不到相匹配的值組時，必須填入NULL（空值）。
- 2. 撰寫SQL程式碼

SQL指令

```
use ch13_db;  
SELECT *  
FROM 老師資料表 AS A LEFT OUTER JOIN 課程資料表 AS B  
ON A.老師編號=B.老師編號
```

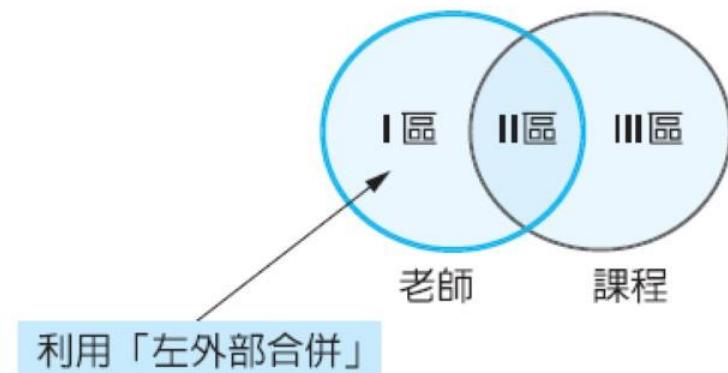
### ○ 3. 執行結果

老師編號	老師姓名	課程代碼	課程名稱	老師編號
T0001	張三	C001	資料庫	T0001
T0002	李四	C002	資料結構	T0002
T0003	王五	(NULL)	(NULL)	(NULL)
T0004	李安	(NULL)	(NULL)	(NULL)

## 範例 2 左外部合併

- 承上一題，請撰寫出尚未開課的老師的SQL指令。

解答▶ 1. 利用圖解說明



◆ 圖13-27

2. 撰寫SQL程式碼

SQL指令
use ch9_DB; SELECT A.老師編號,A.老師姓名 FROM 老師資料表 AS A LEFT OUTER JOIN 課程資料表 AS B ON A.老師編號=B.老師編號 WHERE B.老師編號 IS NULL



### ○ 3. 執行結果

老師資料表 (2×2)	
老師編號	老師姓名
T0003	王五
T0004	李安

## 範例 3 右外部合併

- 假設有兩個資料表，分別是「老師資料表」與「課程資料表」，現在欲查詢每一門課程資料，其中包括尚未被老師開課的課程也要列出。

老師資料表(A)		課程資料表(B)		
#	老師編號	老師姓名	課程代碼	課程名稱
#1	T0001	張三	C001	資料庫
#2	T0002	李四	C002	資料結構
#3	T0003	王五	C003	程式設計
#4	T0004	李安	C004	系統分析

## 解析

- 1. 分析
- 當兩個關聯做合併運算時，會保留第二個關聯(右邊)中的所有值組(Tuples)。找不到相匹配的值組時，必須填入NULL(空值)。
- 2. 撰寫SQL程式碼

SQL 指令
<pre>use ch13_db; SELECT * FROM 老師資料表 AS A RIGHT JOIN 課程資料表 AS B ON A.老師編號=B.老師編號 ORDER BY B.課程代碼</pre>



### ○ 3. 執行結果

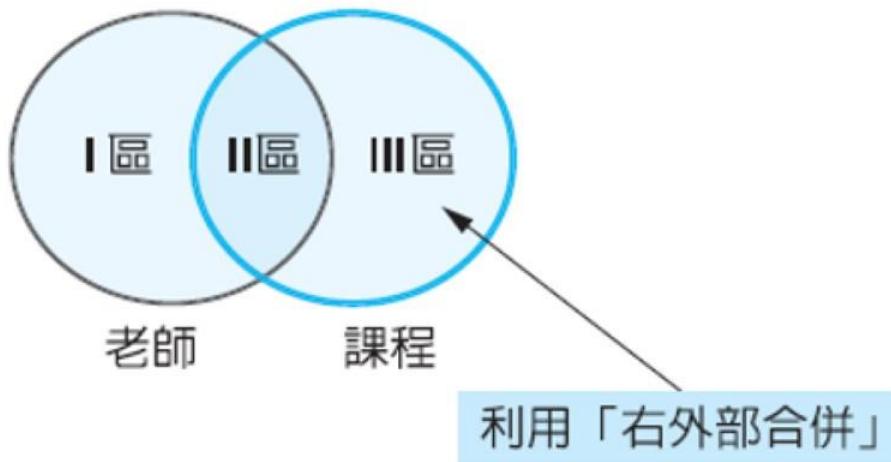
老師編號	老師姓名	課程代碼	課程名稱	老師編號
T0001	張三	C001	資料庫	T0001
T0002	李四	C002	資料結構	T0002
(NULL)	(NULL)	C003	程式設計	
(NULL)	(NULL)	C004	系統分析	

## 範例 4 右外部合併

- 承上一題，請撰寫出那些課程尚未被老師開課的SQL指令。

## 解析

- 1. 利用圖解說明



## ○ 2. 撰寫SQL程式碼

```
use ch9_DB;  
  
SELECT B.課程代碼, B.課程名稱  
FROM 老師資料表 AS A RIGHT OUTER JOIN 課程資料表 AS B  
ON A.老師編號=B.老師編號  
WHERE A.老師編號 IS NULL
```

## ○ 3. 執行結果

課程資料表 (2×2)	
課程代碼	課程名稱
C003	程式設計
C004	系統分析

# 除法(Division)

## 【定義】

此種運算如同數學上的除法一般，有二個運算元：第一個關聯表R

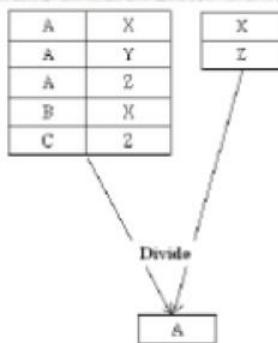
當作「被除表格」，第二個關聯表S當作「除表格」。

其中「被除表格」的屬性必須比「除表格」中的任何屬性中的值域都要與「被除表格」中的某屬性之值域相符合。

## 【關聯式代數】 $R \div S$

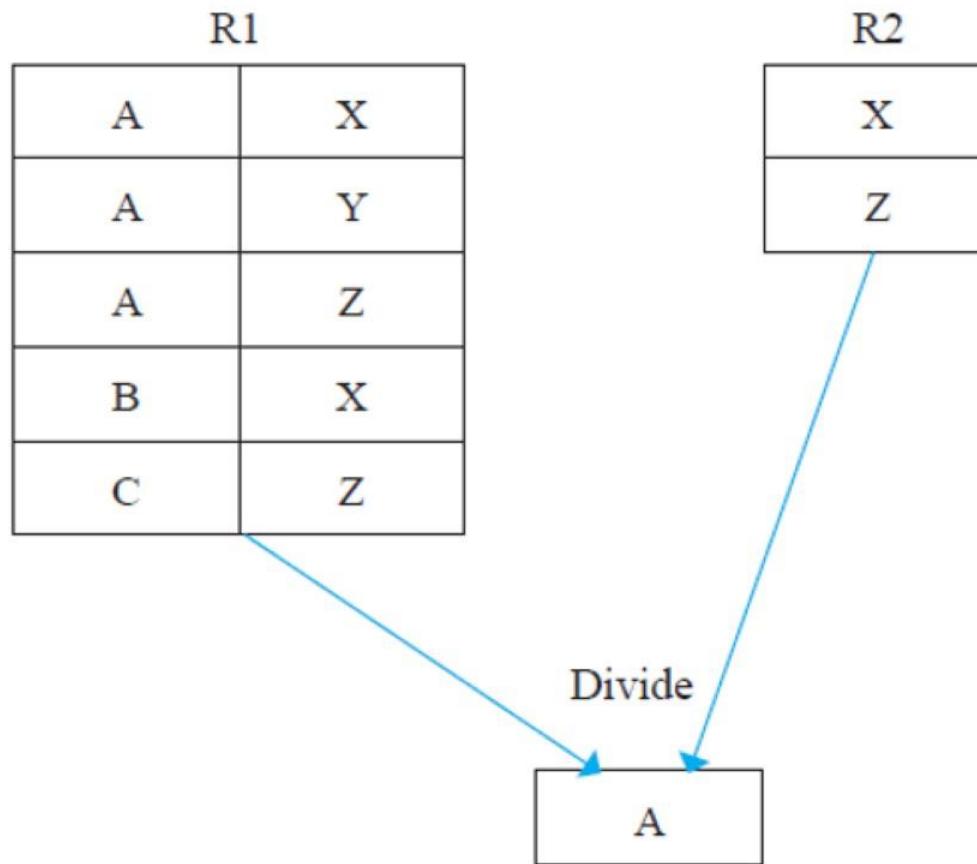
## 【概念圖】

關聯 R 與關聯 S作「除法」運算時，只作用在兩個關聯中相同的部份。



代表符號▶▶  $R1 \div R2$

概念圖▶▶



## 【簡易作法】

關聯R「除以」關聯S時，則分兩個步驟來處理

步驟1：

檢查關聯R中的每一列資料，若有包含關聯S中的某一列時，則將該列資料取出

步驟2：

將步驟1取出資料列，再刪掉關聯S之資料行

## 【實例分析】

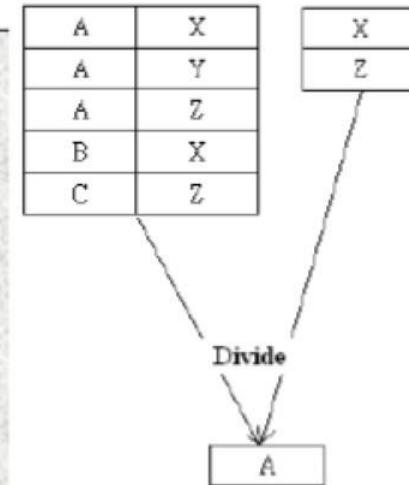
假設現在有關聯R與關聯S，如下圖所示：

關聯R

A	X
A	Y
A	Z
B	X
C	Z

關聯S

X
Z

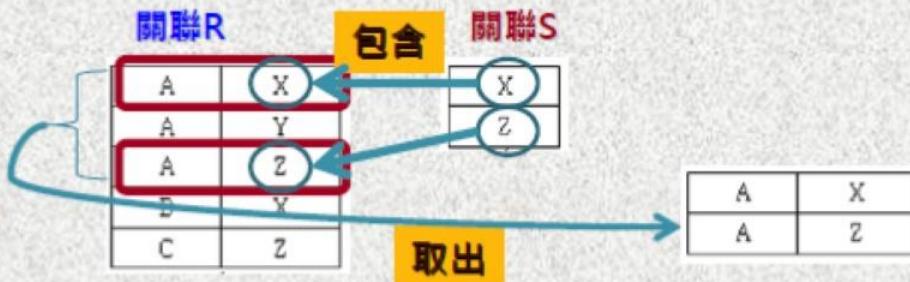


現在欲作  $R \div S$  時，則其執行過程為何？

【解答】

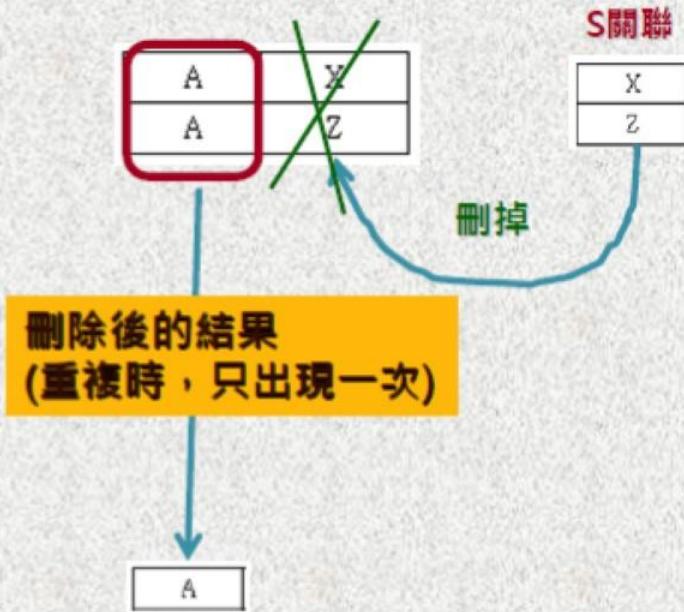
步驟1：

檢查關聯R中的每一列資料，若有包含關聯S中的每一列時，則將該列資料取出



步驟2：

將步驟1取出資料列，再刪掉S關聯之資料行



## 基本型格式▶

```
SELECT 目標屬性  
FROM 目標表格  
WHERE NOT EXISTS  
(SELECT *  
FROM 除式表格  
WHERE NOT EXISTS  
(SELECT *  
FROM 被除式表格  
WHERE 目標表格.合併屬性1=被除式表格.合併屬性1  
AND 除式表格.合併屬性2=被除式表格.合併屬性2))
```

## 範例 1

- 假設有三個檔案，分別為：學生檔、選課檔及課程檔。請列出所有學生均選修的課程？



## 學生檔

學號	姓名	性別	電話	地址
S001	一心	男	11111111	前金區
S002	二聖	男	22222222	小港區
S003	三多	女	33333333	新興區
S004	四維	男	44444444	三民區
S005	五福	女	55555555	小港區

## 選課檔

學號	課號	成績
S001	C001	56
S001	C005	73
S002	C002	92
S002	C005	63
S003	C004	92
S003	C005	70
S004	C003	75
S004	C004	88
S004	C005	68
S005	C005	95

## 課程檔

課程表 (3×8)		
課號	課名	學分數
C001	程式設計	4
C002	計算機概論	3
C003	資料結構	3
C004	系統分析	4
C005	資料庫系統	4
C006	數位學習	3
C007	手機程式	3
C008	機器人程式	3

**分析►►** 1. 分析的方法：辨識「目標屬性」及「相關表格」。



2. 分析題型：屬於「間接合併」，指合併時必須要借助中間表格。

## 解答▶

### SQL指令

```
use ch13_db;  
SELECT 課名  
FROM 課程檔 As C  
WHERE NOT EXISTS  
(  
    SELECT *  
    FROM 學生檔 As A  
    WHERE NOT EXISTS  
(  
        SELECT *  
        FROM 選課檔 As B  
        WHERE C.課號=B.課號 AND A.學號=B.學號  
    )  
)
```

課程檔 (1×1)	
課名	
資料庫系統	

## 巢狀結構查詢

- 定義：是指在Where敘述中再嵌入另一個查詢敘述，此查詢敘述稱為「子查詢」。換言之，您可以將「子查詢」的結果拿來作為另一個查詢的條件。
- 注意：「子查詢」可以獨立地被執行，其執行結果稱為「獨立子查詢」。
- 分類：
  - 1. 傳回單一值 ( = )
  - 2. 傳回多值 ( IN )
  - 3. 測試子查詢是否存在 ( 利用EXIST )

## 第一種作法

- 範例：利用子查詢來找出選修「資料庫系統」的學生學號及姓名。



**解答▶** 請參考前一單元中的三個資料表。

SQL指令
use ch13_db;
SELECT A.學號, 姓名 _____
FROM 學生檔 AS A, 選課檔 AS B
WHERE A.學號=B.學號 AND B.課號= _____
(SELECT C.課號 FROM 課程檔 AS C _____
WHERE 課名= '資料庫系統') _____

**執行結果▶**

學生檔 (2×5)	
學號	姓名
5001	一心
5002	二聖
5003	三多
5004	四維
5005	五福

## 第二種作法

- 範例：利用子查詢來找出選修「課號為C005」的學生學號及姓名。

## 解答▶▶

SQL指令
use ch13_db; SELECT A.學號, 姓名 FROM 學生檔 AS A WHERE A.學號= _____  (SELECT 學號 FROM 選課檔 AS B WHERE A.學號=B.學號 AND B.課號='C005') _____

主查詢

子查詢

## 執行結果▶▶

學生檔 (2×5)	
學號	姓名
5001	一心
5002	二聖
5003	三多
5004	四維
5005	五福



## 練習

學生成績表 (5×10)

學號	姓名	資料庫	資料結構	程式設計
50001	一心	100	85	80
50002	二聖	70	75	90
50003	三多	85	75	80
50004	四維	95	100	100
50005	五福	80	65	70
50006	六合	60	55	80
50007	七賢	45	45	70
50008	八德	55	30	50
50009	九如	70	65	70
50010	十全	60	55	80

1

請撰寫一段「子查詢」的SQL指令來查詢哪些同學的「資料庫」成績高於平均成績。

2

請撰寫一段「子查詢」的SQL指令來查詢哪些同學的「資料庫」成績是最高分。



### SQL指令

```
USE ch13_db;  
SELECT *  
FROM 學生成績表  
WHERE 資料庫 > (SELECT AVG(資料庫)  
FROM 學生成績表);
```

執行結果：

學號	姓名	資料庫	資料結構	程式設計
S0001	一心	100	85	80
S0003	三多	85	75	80
S0004	四維	95	100	100
S0005	五福	80	65	70

### SQL指令

```
USE ch13_db;  
SELECT *  
FROM 學生成績表  
WHERE 資料庫 = (SELECT MAX(資料庫)  
                  FROM 學生成績表);
```

執行結果：

學生成績表 (5×1)				
學號	姓名	資料庫	資料結構	程式設計
50001	一心	100	85	80