合併理論

限制(Restrict)

【定義】是指在關聯表中選取符合某些條件的值組(記錄)，然後另成一個新的關聯。

【代表符號】σ(唸成sigma)

【假設】P為選取的條件，則以$\sigma_p(R)$代表此運算。其結果為原關聯表R記錄的「水平」子集合。

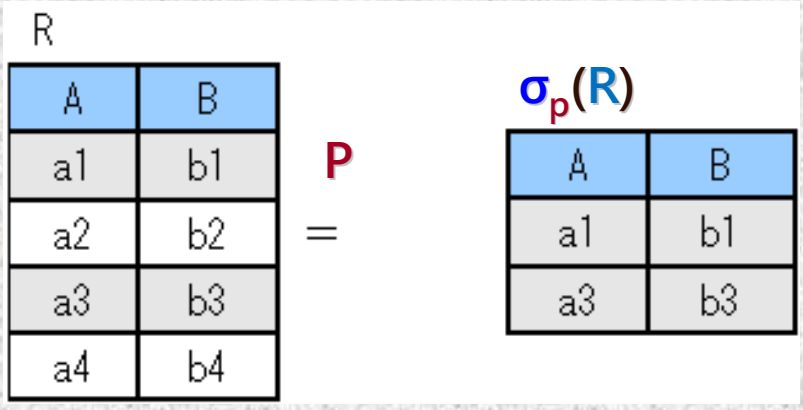
【關聯式代數】$\sigma_{條件}(關聯)$

【SQL語法】關聯 Where 條件

其中條件可用邏輯運算子(AND, OR , NOT)來組成。

【概念圖】

從關聯表 R中選取符合條件 (Predicate) P的值組。其結果為原關聯表R記錄的「水平」子集合。如下圖所示：



【對應SQL語法】

```
SELECT    屬性集合
FROM      關聯表 R
WHERE     選取符合條件 P     //水平篩選
```

【實例】請在下列的學生選課表中，請找出課程學分數為3的記錄？

	學號	姓名	課號	課程名稱	學分數
#1	S0001	張三	C001	MIS	3
#2	S0002	李四	C005	XML	2
#3	S0003	王五	C002	DB	3
#4	S0004	林六	C004	SA	3

【解答】以SQL達成關聯式代數的運算功能

關聯式代數	SQL
$\sigma_{學分數=3}$(學生選課表) 相當於	SELECT * FROM 學生選課表 WHERE 學分數='3'

【執行結果】

	學號	姓名	課號	課程名稱	學分數
#1	S0001	張三	C001	MIS	3
#2	S0003	王五	C002	DB	3
#3	S0004	林六	C004	SA	3

投影(Project)

【定義】是指在「關聯表R」中選取想要的「欄位」，然後另成一個新的關聯表。

【代表符號】π(唸成pai)

【假設】關聯表R中選取想要的欄位為A1，A2，A3，...An，則以 $\pi_{A1,A2,A3...An}(R)$表示此投影運算。其結果為原關聯表R的「垂直」子集合。

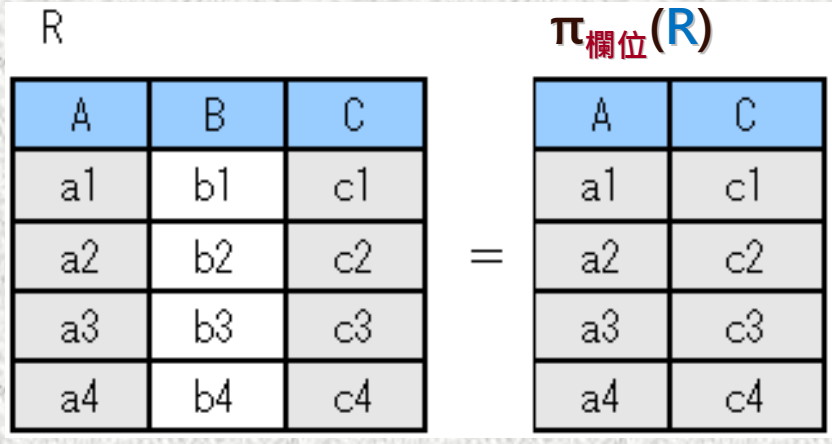
【關聯式代數】$\pi_{欄位}(關聯表)$

【SQL語法】Select 欄位 From 關聯表

其中欄位可以由數個欄位所組成。

【概念圖】

從關聯表 R中選取想要的欄位。其結果為原關聯表R記錄的「垂直」子集合。如下圖所示：



【對應SQL語法】

```
SELECT  屬性集合              //垂直篩選
FROM  資料表名稱
```

【實例】請在下列的學生選課表中，請找出學生「姓名」與「課程名稱」？

	學號	姓名	課號	課程名稱	學分數
#1	S0001	張三	C001	MIS	3
#2	S0002	李四	C005	XML	2
#3	S0003	王五	C002	DB	3
#4	S0004	林六	C004	SA	3

【解答】以SQL達成關聯式代數的運算功能

關聯式代數	SQL
相當於	SELECT 姓名,課程名稱 FROM 學生選課表
$\pi_{姓名，課程名稱}$(學生選課表)	

【執行結果】

	姓名	課程名稱
#1	張三	MIS
#2	李四	XML
#3	王五	DB
#4	林六	SA

<資料庫檔案名稱： ch8-2-2.accdb>

卡氏積(Cartesian Product)

【定義】是指將兩關聯表R_1與R_2的記錄利用集合運算中的乘積運算形成新的關聯表R_3。卡氏積(Cartesian Product)；也稱交叉乘積(Cross Product)；或稱交叉合併(Cross Join)。

【代表符號】 ×

【假設】R_1有r_1個屬性，m筆記錄，R_2有r_2個屬性，n筆記錄，R3會有（$r_1 + r_2$）個屬性，（$m \times n$）筆記錄。

【關聯式代數】 $R_3 = R_1 X R_2$

【 SQL語法】SELECT *

FROM A表格,B表格

【概念圖】

R_1有r_1個屬性，m筆記錄，R_2有r_2個屬性，n筆記錄，R3會有（$r_1 + r_2$）個屬性，（$m \times n$）筆記錄。如下圖所示：

R1

A	B
a1	b1
a2	b2
a3	b3

X

R2

X	Y
x1	y1
x2	y2

=

R3=R1XR2

A	B	X	Y
a1	b1	x1	y1
a2	b2	x1	y1
a3	b3	x1	y1
a1	b1	x2	y2
a2	b2	x2	y2
a3	b3	x2	y2

【格式1】

```
SELECT *
FROM A表格,B表格
```

【格式2】

```
SELECT *
FROM A表格 CROSS JOIN B表格
```

【實例】請在下列的「學生表」與「課程表」中，請找出學生表與課程表的所有可能配對的集合？

學生表

	學號	姓名	課號
#1	S0001	張三	C001
#2	S0002	李四	C002

課程表

課號	課名	學分數
C001	MIS	3
C002	DB	3
C003	VB	2

【解答】<資料庫檔案名稱： ch8-2-3.accdb>

1.分析：

已知：學生表R_1 (學號，姓名，課號)

課程表R_2 (課號，課名，學分數)

兩個資料表的「卡氏積」，可以表示為：

學生表R_1 (學號，姓名，課號) × 課程表R_2 (課號，課名，學分數) = 新資料表R_3

R_1有($r_1=3$)個屬性，($m=2$)筆記錄，R_2有($r_2=3$)個屬性，($n=3$)筆記錄，

R_3會有（r_1+r_2）個屬性＝6個屬性

新資料表R_3(學號，姓名，學生表.課號，課程表.課號，課名，學分數)

R_3會有（$m \times n$）筆記錄＝6筆記錄

在資料記錄方面，每一位學生(2位)均會對應到每一門課程資料(3門)，亦即二位學生資料，產生(2×3)=6筆記錄。如下圖所示：

學生表

	學號	姓名	課號
#1	S0001	張三	C001
#2	S0002	李四	C002

課程表

課號	課名	學分數
C001	MIS	3
C002	DB	3
C003	VB	2

因此，「學生表」與「課程表」在經過「卡氏積」之後，共會產生6筆記錄，如下圖所示：

「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	MIS	3
#2	S0001	張三	C001	C002	DB	3
#3	S0001	張三	C001	C003	VB	2
#4	S0002	李四	C002	C001	MIS	3
#5	S0002	李四	C002	C002	DB	3
#6	S0002	李四	C002	C003	VB	2

每一位學生對應三門課程

以上面所產生的六筆記錄中，不知您是否有發現，有一些不太合理的記錄。

例如：「張三」只選修課號為C001，但是卻多出了兩筆不相關的紀錄(C002,C003)。因此，如何從「卡氏積」所展開的全部組合中，挑選出合理的記錄，就必須要再透過下一章節所要介紹的「內部合併(Inner Join)」。

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。 如下圖所示：

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)

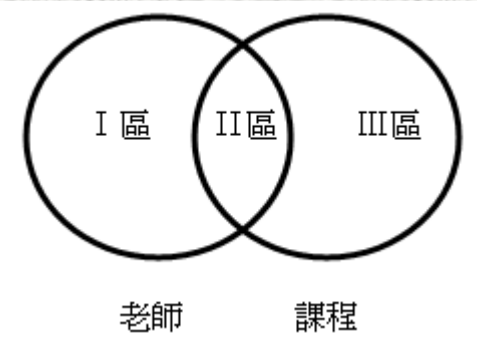
R1⋈ R2= R3

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

【合併的分類】

廣義而言，合併可分為「來源合併」與「結果合併」兩種。

一、來源合併：(需要FK→PK)



(一)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」。如上圖中的Ⅱ區。

(二)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」。如上圖中的Ⅰ區。

2.如果查詢有那些課程尚未被老師開課，則會使用到「右外部合併」。

如上圖中的Ⅲ區。

二、結果合併：(不需要FK→PK)

(一)Cross Join(卡氏積)

(二)Union(聯集)

(三)Intersect(交集)

(四)Except(差集)

內部合併(Inner Join)

【定義】

內部合併(Inner Join)又稱為「條件式合併(Condition Join)」，也就是說，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」。如下表所示：

運算子	條件式說明
= (等於)	學生表.課號=課程表.課號
<> (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

【兩種作法】

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

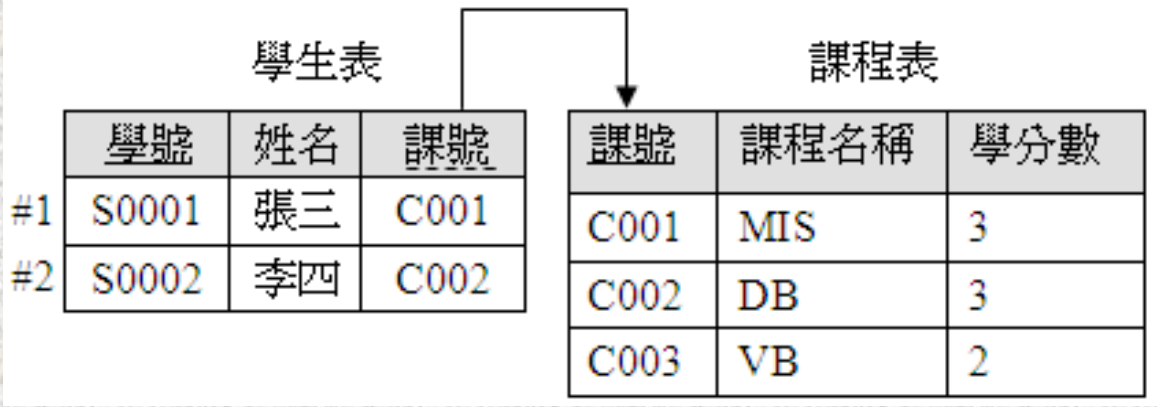
```
From A ,B
Where (A.c=B.c)
```

2.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
```

【實例】

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」對應到「課程表」的「課號」，如下圖所示：



學生表

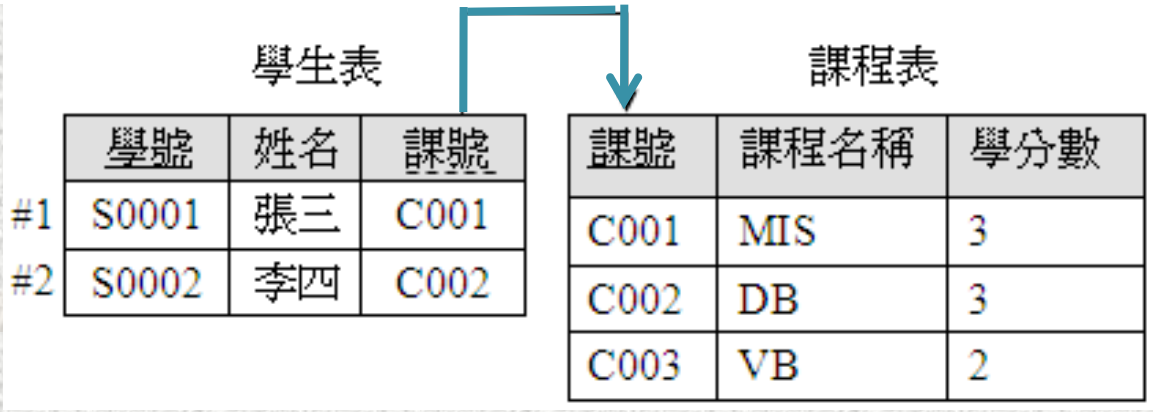
	學號	姓名	課號
#1	S0001	張三	C001
#2	S0002	李四	C002

課程表

課號	課程名稱	學分數
C001	MIS	3
C002	DB	3
C003	VB	2

【解析】

從下一頁開始...



	學號	姓名	課號
#1	S0001	張三	C001
#2	S0002	李四	C002

課號	課程名稱	學分數
C001	MIS	3
C002	DB	3
C003	VB	2

1.分析

從上圖中，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，從展開後的記錄中，找尋那幾筆記錄具有符合「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	MIS	3
#2	S0001	張三	C001	C002	DB	3
#3	S0001	張三	C001	C003	VB	2
#4	S0002	李四	C002	C001	MIS	3
#5	S0002	李四	C002	C002	DB	3
#6	S0002	李四	C002	C003	VB	2

2.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】

	學號	姓名	課號	課名	學分數
#1	S0001	張三	C001	MIS	3
#2	S0002	李四	C002	DB	3

3.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，則必須要進行以下的分析： ch8-2.4.1A.accdb

步驟一：辨識「目標屬性」及「相關表格」

學生資料表(學號，姓名，系碼)

? ?

選課資料表(學號，課號，成績)

?

❶目標屬性：學號, 姓名, 平均成績

❷相關表格：學生資料表, 選課資料表

步驟二：將相關表格進行「卡氏積」

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

步驟三：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

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

5筆記錄

A.學號	姓名	系碼	B.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

步驟四：加入限制條件

```
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
```

4筆記錄

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

步驟五：投影使用者欲「輸出的欄位名稱」

```
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
```

輸出的欄位名稱

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

步驟六：使用群組化及聚合函數

```
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
```

(85+100)/2=92.5

依照學號與姓名來分群

學號	姓名	平均成績
S0001	張三	92.5
S0002	李四	89
S0003	王五	90

步驟七：使用「聚合函數」之後，再進行篩選條件

```
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)>=70
```

70(含)以上

學號	姓名	平均成績
S0001	張三	92.5
S0002	李四	89
S0003	王五	90

步驟八：依照某一欄位或「聚合函數」結果，來進行「排序」。

```
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>=60
GROUP BY A.學號, 姓名
HAVING AVG(成績)>=70
ORDER BY AVG(成績) DESC;
```

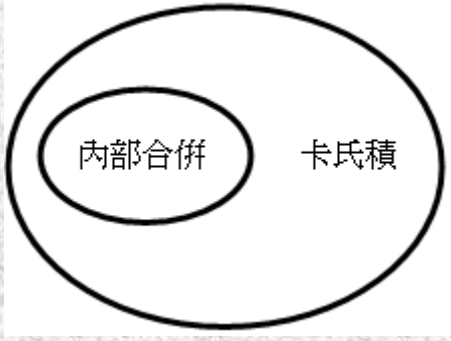
由高到低分排序

學號	姓名	平均成績
S0001	張三	92.5
S0003	王五	90
S0002	李四	89

4.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，並產生龐大記錄，但大部份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」的記錄。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。如下圖所示：



外部合併(Outer Join)

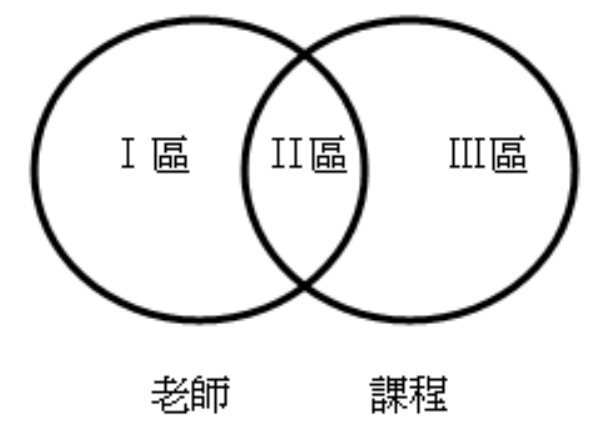
【定義】

當在進行合併(Join)時不管記錄是否符合條件，都會被列出其中一個資料表的所有記錄時，則稱為「外部合併」。此時不符合條件的記錄就會被預設為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】



1.左外部合併(Left Outer Join，以 ⟕ 表示)。

舉例：如果要查詢尚未開課的老師，則會使用到「左外部合併」。

如上圖中的 I 區。

2.右外部合併(Right Outer Join，以 ⟖ 表示)。

舉例：如果查詢有那些課程尚未被老師開課，則會使用到「右外部合併」。

如上圖中的III區。

3.完全外部合併(Full Outer Join，以 ⟗ 表示)。

舉例：如果要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，

並且也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

【格式】

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

【舉例1】左外部合併

資料庫檔名：ch8-2-4-2.accdb

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。如下圖所示：

老師資料表(A)

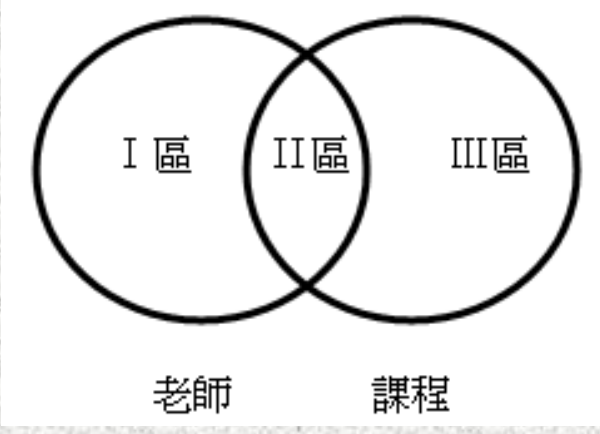
	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

【解析】

從下一頁開始...



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)。

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

左外部合併

利用SQL Server 2008執行結果如下：

	A.老師編號	老師姓名	課程代碼	課程名稱	B.老師編號
#1	T0001	張三	C001	資料庫	T0001
#2	T0001	張三	C002	資料結構	T0001
#3	T0002	李四	NULL	NULL	NULL
#4	T0003	王五	NULL	NULL	NULL
#5	T0004	李安	NULL	NULL	NULL

利用Access2007執行結果如下：

1_外部合併(左外部合併查詢_舉例1)

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

2.撰寫SQL程式碼

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

【實例2】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」，請撰寫出尚未開課的老師的SQL指令。

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

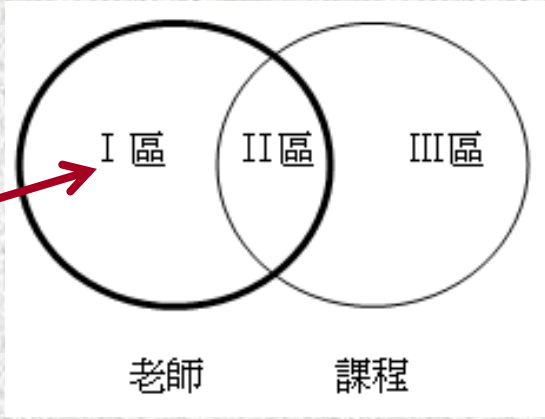
課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

【解析】

1.利用圖解說明

利用「左外部合併」



2.撰寫SQL程式碼

```
--1.1查詢尚未開課的老師(利用左外部合併)
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
```

1_外部合併(左外部合併查詢_舉例1)

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

3.執行結果

老師編號	老師姓名
T0002	李四
T0003	王五
T0004	李安

【舉例1】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。如下圖所示：

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

【解析】

從下一頁開始...

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)。

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

右外部合併

利用SQL Server 2008執行結果如下：

A.老師編號	老師姓名	課程代碼	課程名稱	B.老師編號
T0001	張三	C001	資料庫	T0001
T0001	張三	C002	資料結構	T0001
NULL	NULL	C003	程式設計	NULL
NULL	NULL	C004	系統分析	NULL

利用Access2007執行結果如下：

3_外部合併(右外部合併查詢_舉例2)

A.老師編號	老師姓名	課程代碼	課程名稱	B.老師編號
T0001	張三	C001	資料庫	T0001
T0001	張三	C002	資料結構	T0001
		C003	程式設計	
		C004	系統分析	

2.撰寫SQL程式碼

```
--2.右外部合併
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
```

【舉例1】全外部合併

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，並且也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。如下圖所示：

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

【解析】

從下一頁開始...

1.分析

當兩個關聯做合併運算時，會保留左右兩邊關聯中的所有值組(Tuples)。找不到相匹配的值組時，必須填入NULL(空值)。

老師資料表(A)

	老師編號	老師姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

課程資料表(B)

課程代碼	課程名稱	老師編號
C001	資料庫	T0001
C002	資料結構	T0001
C003	程式設計	NULL
C004	系統分析	NULL

全外部合併

結果 訊息

	老師編號	老師姓名	課程代碼	課程名稱	老師編號
1	T0001	張三	C001	資料庫	T0001
2	T0001	張三	C002	資料結構	T0001
3	T0002	李四	NULL	NULL	NULL
4	T0003	王五	NULL	NULL	NULL
5	T0004	李安	NULL	NULL	NULL
6	NULL	NULL	C003	程式設計	NULL
7	NULL	NULL	C004	系統分析	NULL

2.撰寫SQL程式碼

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

註：FULL OUTER JOIN在SQL Server中才能執行此指令。

集合運算子

基本上，集合運算子有三種：

1.交集(Intersection)

2.聯集(Union)

3.差集(Difference)

交集(Intersection)

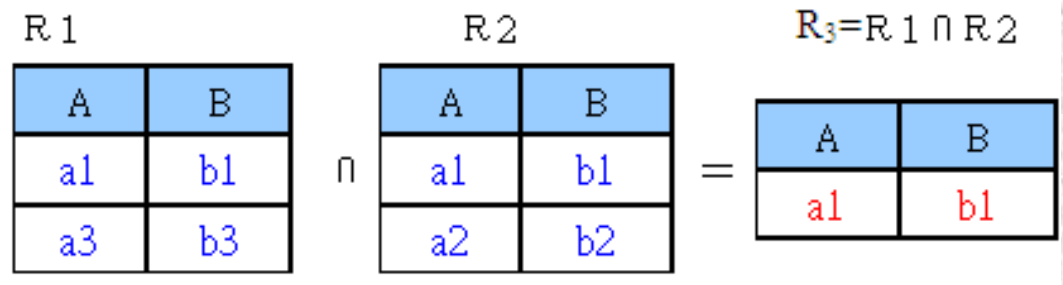
【定義】

是指關聯表R_1與關聯表R_2作「交集」時，則將原來在兩個關聯式中都有出現的值組(記錄)組合在一起成為新的關聯式R_3。

【代表符號】$R_1 \cap R_2$

【SQL語法】From 關聯表R_1 Intersect 關聯表R_2

【概念圖】



【格式】

關聯式代數		SQL
A∩B	相當於	Select * From A Intersect B

【舉例】

列出97與98學年度「都有」在網路開課老師名單

97學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

98學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0004	李安
#3	T0005	小雄
#4	T0006	碩安

【解答】 **<此功能只能在SQL Server上執行>**

SQL指令

```
SELECT * FROM [97學年度網路開課老師表]
INTERSECT
SELECT * FROM [98學年度網路開課老師表]
```

【執行結果】選取兩資料表「都有」的資料列。

	教師編號	姓名
#1	T0001	張三
#2	T0004	李安

聯集(Union)

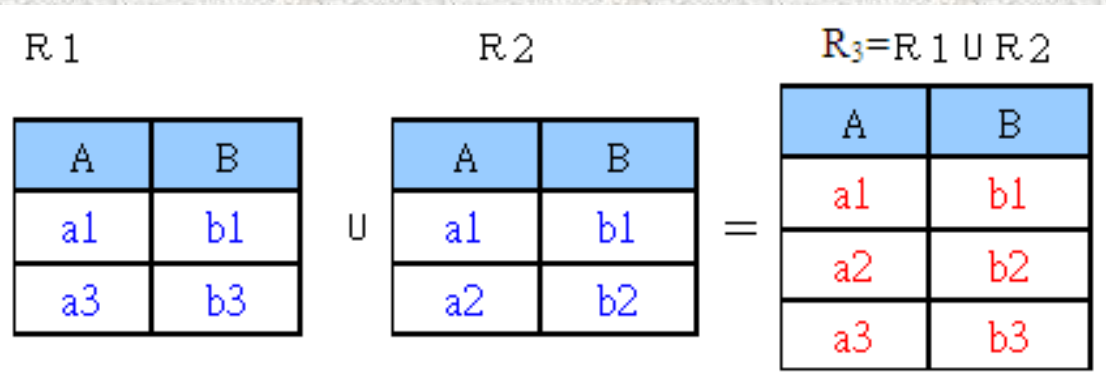
【定義】

是指關聯表R_1與關聯表R_2作「聯集」時，則會重新組合成一個新的關聯表R_3，而新的關聯表R_3中的記錄為原來兩關聯表的所有記錄，若有重複的記錄，則只會出現一次。

【表示符號】$R_1 \cup R_2$

【SQL語法】From 關聯表R_1 Union 關聯表R_2

【概念圖】



【對應的SQL語法】

關聯式代數		SQL
A∪B	相當於	Select * From A ***Union*** B

【舉例】

列出97與98學年度有在網路開課老師名單

97學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

98學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0004	李安
#3	T0005	小雄
#4	T0006	碩安

【解答】

SQL指令
SELECT * FROM [97學年度網路開課老師表]
UNION
SELECT * FROM [98學年度網路開課老師表]

【執行結果】聯集運算乃在選取兩資料表「所有」的資料列，但重複的資料列只取一次

	教師編號	姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安
#5	T0005	小雄
#6	T0006	碩安

差集(Difference)

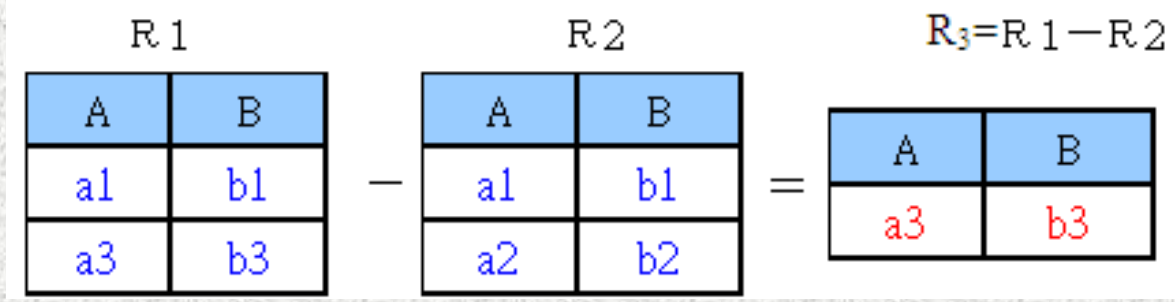
【定義】

是指將一個關聯表R_1中的記錄減去另一個關聯表R_2的記錄，形成新的關聯表R_3的記錄。亦即關聯表R_1差集關聯表R_2之後的結果，則為關聯表R_1減掉R_1R_2兩關聯共同的值組。

【代表符號】 $R_1 - R_2$

【 SQL語法】From 關聯表R_1 Except 關聯表R_2

【概念圖】



【格式】

關聯式代數		SQL
A−B	相當於	Select * From A ***Except*** B

事實上差集的運算相當於將關聯表R1中的記錄減去R1與R2共有的記錄，也就是R1-R2 = R1-(R1∩R2)。

【舉例】

列出有在97學年度網路開課，但沒有在98學年度網路開課的老師名單

97學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0002	李四
#3	T0003	王五
#4	T0004	李安

98學年度網路開課老師表

	教師編號	姓名
#1	T0001	張三
#2	T0004	李安
#3	T0005	小雄
#4	T0006	碩安

【解答】 **<此功能只能在SQL Server上執行>**

SQL指令
SELECT * FROM [97學年度網路開課老師表] ***Except*** SELECT * FROM [98學年度網路開課老師表]

【執行結果】

	教師編號	姓名
#1	T0002	李四
#2	T0003	王五

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，你可以將「子查詢」的結果拿來做為另一個查詢的條件。

【注意】

「子查詢」是可以獨立地被執行，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)

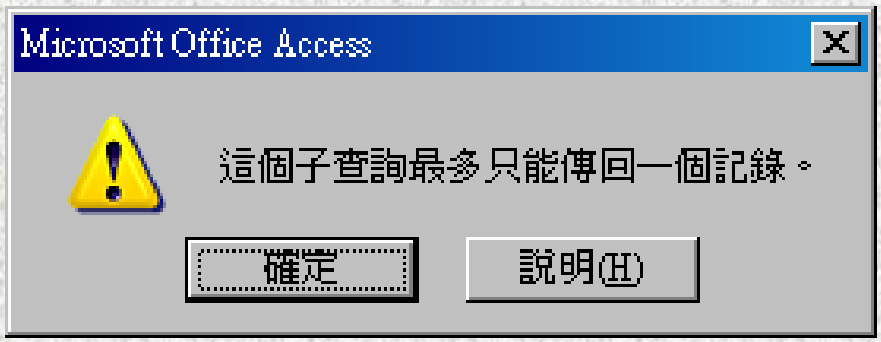
3.測試子查詢是否存在(利用EXIST)

比較運算子「=」

【定義】

由於主查詢的條件中使用了比較運算子「=」，所以子查詢就只能傳回一個結果。一旦子查詢傳回了一個以上的結果，那麼主查詢的Where子句中的條件就無法成立了。

【使用時機】子查詢就只能傳回一個結果，否則會出現如下的畫面：



【實例1】

利用子查詢來找出選修「資料庫系統」的學生學號及姓名

一、學生資料表

	學號	姓名	系碼
#1	S0001	張三	D001
#2	S0002	李四	D002
#3	S0003	王五	D003
#4	S0004	陳明	D001
#5	S0005	李安	D004

【解答】《資料表見8-4　第一個案例》

SQL指令

```
SELECT A.學號, 姓名
FROM 學生資料表 AS A, 選課資料表 AS B
WHERE A.學號=B.學號 AND B.課號=
(SELECT C.課號 FROM 課程資料表 AS C
  WHERE 課名= '資料庫系統');
```

主查詢

子查詢

課號
C005

子查詢就只能傳回一個結果

【查詢結果】

學號	姓名
S0001	張三
S0002	李四
S0003	王五
S0004	陳明
S0005	李安

三、選課資料表

	學號	課號	成績
#1	S0001	C001	56
#2	S0001	C005	73
#3	S0002	C002	92
#4	S0002	C005	63
#5	S0003	C004	92
#6	S0003	C005	70
#7	S0004	C003	75
#8	S0004	C004	88
#9	S0004	C005	68
#10	S0005	C005	NULL

四、課程資料表

	課號	課名	學分數	老師編號
#1	C001	資料結構	4	T0001
#2	C002	資訊管理	4	T0001
#3	C003	系統分析	3	T0001
#4	C004	統計學	4	T0002
#5	C005	資料庫系統	3	T0002
#6	C006	數位學習	3	T0003
#7	C007	知識管理	3	T0004

IN集合條件

【定義】

如果我們想讓子查詢可以傳回一個以上的值，我們可以在主查詢條件之中使用IN運算子來接收子查詢傳回的結果，因為IN可以處理多個值，也就是說，當某列的學號等於IN之內的任何一個學號，此列就會被傳回。

【使用時機】子查詢可以傳回一個以上的結果

【實例】

若授課老師想了解有修「資料」開頭的課程之同學(利用子查詢來找出，使用IN)

【解答】《資料表見8-4 第一個案例》

SQL指令

```
SELECT A.學號, 姓名
FROM 學生資料表 AS A, 選課資料表 AS B
WHERE A.學號=B.學號 AND B.課號 IN
(SELECT C.課號 FROM 課程資料表 AS C
  WHERE 課名 LIKE '資料*');
```

主查詢

子查詢

子查詢可以傳回兩個或兩個以上的結果

課號
C001
C005

【查詢結果】

學號	姓名
S0001	張三
S0002	李四
S0001	張三
S0003	王五
S0004	陳明
S0005	李安

一、學生資料表

	學號	姓名	系碼
#1	S0001	張三	D001
#2	S0002	李四	D002
#3	S0003	王五	D003
#4	S0004	陳明	D001
#5	S0005	李安	D004

四、課程資料表

	課號	課名	學分數	老師編號
#1	C001	資料結構	4	T0001
#2	C002	資訊管理	4	T0001
#3	C003	系統分析	3	T0001
#4	C004	統計學	4	T0002
#5	C005	資料庫系統	3	T0002
#6	C006	數位學習	3	T0003
#7	C007	知識管理	3	T0004

EXIST測試子查詢是否存在

【定義】

是指用來判斷子查詢結果是否存在於合併後的結果中。如果存在，則會傳回TRUE，若不存在的話則會傳回FALSE。若是TRUE的話，則會執行主查詢，若是FALSE的話則不會被執行。

【實例】

請利用EXISTS來找出選修「資料庫系統」的學生學號及姓名。

【解答】《資料表見8-4　第一個案例》

SQL指令

```
SELECT DISTINCT A.學號, 姓名
FROM 學生資料表 AS A, 選課資料表 AS B
WHERE A.學號=B.學號 AND EXISTS
(SELECT C.課號 FROM 課程資料表 AS C
WHERE 課名='資料庫系統');
```

主查詢

子查詢

子查詢傳回TRUE

【查詢結果】

學號	姓名
S0001	張三
S0002	李四
S0003	王五
S0004	陳明
S0005	李安

ALL與ANY集合條件

【定義】

假設有兩個資料表(主關聯與子關聯)，如果想比較主關聯與子關聯資料時，就可以利用ALL與ANY集合條件來篩選資料。

【 ALL的例子1】 A >ALL B

A集合：　　取出

2025　60　80　100

B集合：

最大值

30　50

A >ALL B➔是指取出A集合中比B集合「最大值」還要大的資料。

所以，在上面的例子中，顯示的結果為：{60,80,100}

【實例1】

請列出「甲班成績單」中有那些同學的「成績」比「乙班成績單」中所有同學的「成績」高

【解答】

SQL指令

```
SELECT 學號, 姓名, 成績
FROM 甲班成績單
WHERE 成績>ALL (SELECT 成績 FROM 乙班成績單);
```

主查詢

子查詢

甲班成績單

學號	姓名	成績
S0001	一心	20
S0002	二聖	25
S0003	三多	60
S0004	四維	80
S0005	五福	100

乙班成績單

學號	姓名	成績
S0011	張三	30
S0012	李四	50

【查詢結果】是指取出甲班成績單中比乙班成績單「最高分」還要高的成

學號	績°姓名	成績
S0003	三多	60
S0004	四維	80
S0005	五福	100

View視界

View有人稱為「視界」、「檢視表」或「虛擬資料表」，事實上，不管稱為「視界」或「檢視表」，這些都是由View來翻譯過來的名詞，因此，View這個英文字還是最能傳達關聯式資料庫「過濾」的觀念。

【 View與ANSI/SPARC架構的關係】

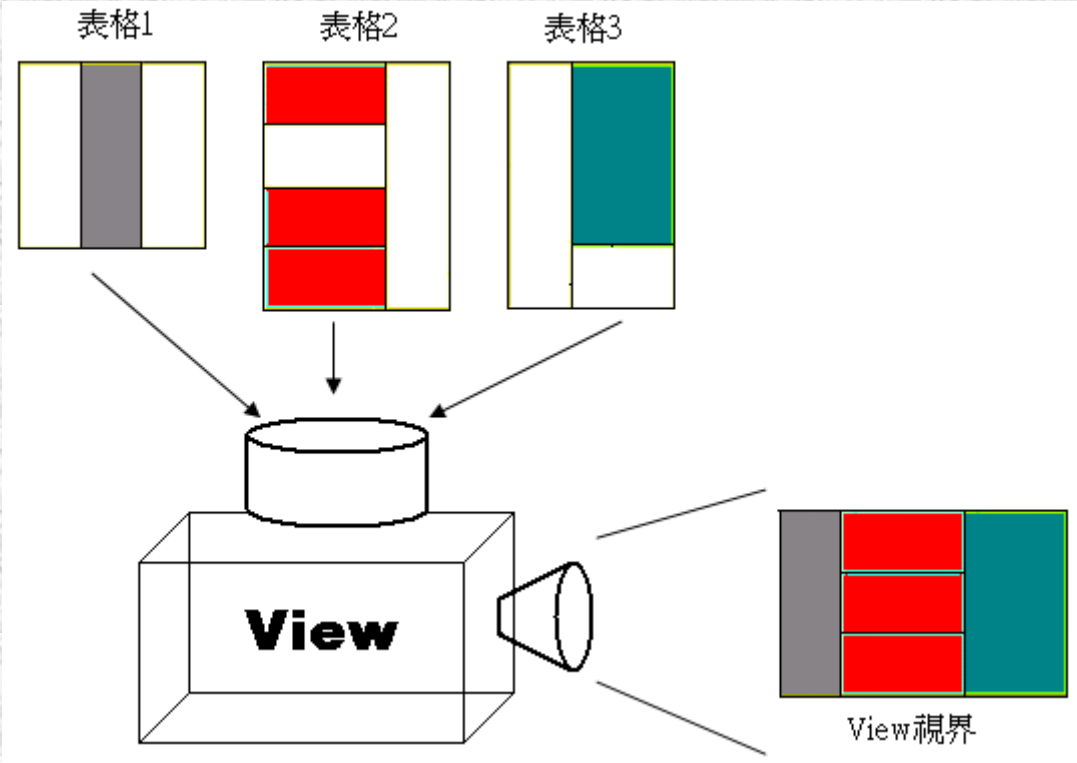
視界表格(View Table)在關聯式系統中的地位相當於ANSI/SPARC資料庫的三層綱目架構上的外部層（External Level）。因為它只是在實際資料表之外的一個虛擬資料表，實際上並沒有儲存資料。

什麼是VIEW？

【定義】

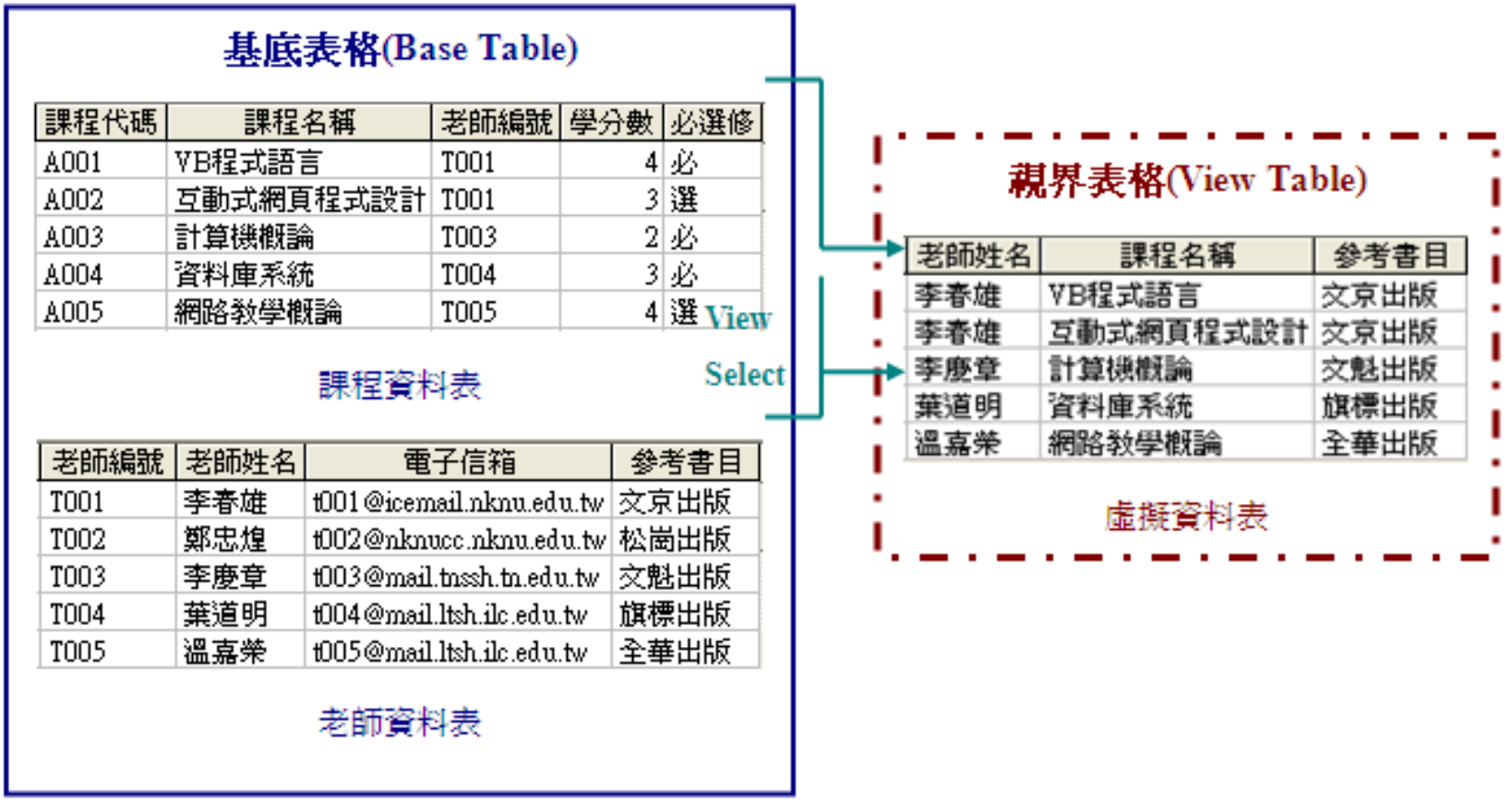
視界(View)其實只是基底表格(Base Table)的一個「小窗口」而已，因為視界表格(View Table)往往只是基底表格的一部份而非全部，我們可以利用SQL結構化查詢語言，將我們需要的資料從各個資料表中挑選出來，整合成一張新的資料表。

【概念圖】



➢「基底表格」與「虛擬資料表」的關係

假設現在有兩個基底表格(Base Table)，分別為「課程資料表」及「老師資料表」，在透過SQL查詢之後，合併成一個使用者需求的資料表，即稱為虛擬資料表。如下圖所示：



【說明】

視界表格(View Table)的資料來源在於數個基底表格(Base Table)，也就是說，透過View Select 語法的查詢，來建立一個新的虛擬資料表，使用者可以依不同的需求，來撰寫不同的SQL指令，進而查詢出使用者所需要的結果。

View的用途

View視界的主要用途就是可以提供不同的使用者不同的查詢資訊。因此，我們可以歸納為下列幾項用途：

1.讓不同使用者對於資料有不同的觀點與使用範圍。

例如：教務處是以學生的「學業成績」為主要觀點。

學務處是以學生的「操行成績」為主要觀點。

2.定義不同的視界，讓使用者看到的資料過濾後的資訊。

例如：一般使用者所看到的資訊只是管理者的部份子集合。

3.有保密與資料隱藏的作用。

例如：個人可以看到個人全部資訊，但是，無法觀看他人的資料(如：薪資、紅利、年終獎金等)。

4.絕大部份的視界僅能做查詢，不能做更新。