

第 A 章 善用 View 視界查詢

備註：可在網站上下載：[createView.sql](#) 則包含 `simple_teachers`、`simple_students` 與 `public_course` 三只 View，但建議自行練習產生。

A-1 視界(View)簡介

A-1-1 視界與資料表

在資料庫的邏輯設計時，為了考慮正規化原則，我們將資料表分割再分割。原本只需一、兩張資料表就可以達成，說不定將它分割成近十張資料表。滿足越多階層的正規化原則，資料表分割得越細，也造成編寫查詢程式越困難。因此，一般資訊系統只要能滿足第三階層正規化就可以了，不然真的自討苦吃。但僅滿足第三正規化原則，資料表還是會被分割得非常凌亂，有補救方法嗎？有的，就看您如何靈活運用『視界』(View)，這就是本章介紹的重點。

資料表在資料庫內佔有一定的儲存空間，是真實存在的表格。然而，視界僅描述表格空間，但沒有真正儲存內容，它是一個或多個資料表的投影組合，也就是說，視界的表格內容還是分散在真實資料表內，它依賴原始資料表而存在。到底如何投影組合，基本上有下列方式：

- (A) 水平投影、
- (B) 垂直投影、
- (C) 水平與垂直投影、
- (D) 表格合併投影。

以下分別介紹之。

A-1-2 開啟視界命令

視界是資料表的投影組合，它的產生是由一個或一個以上的資料表所構成，其基本命令格式如下：

基本語法	範 例
------	-----

<pre> Create view view_name as Select 欄位_1, 欄位_2, 欄位_3, .. From 資料表_1, ... Where 條件敘述 </pre>	<pre> Create view simple_students Select student_ID, name, sex From students Where class_ID = "902113"; </pre>
--	--


A-2 視界的產生

我們介紹幾種視界產生方法與運用。

A-2-1 水平投影的視界 – 資管系教師

我們僅取資料表某些記錄(水平排列)，則稱為水平投影。如下圖所示，我們由 teachers 資料表內，僅取 dep_ID = “11” 資訊管理系的老師。

Dep_ID = “11”



teacher_ID	name	Dep_ID	sex	title	tel	mailbox	address
0107	張永誠	11	男	系主任	07-7310606	tech1@csu.edu.tw	高雄市三民區澄清路 100 號
0108	粘添壽	11	男	教師	07-7310607	tech2@csu.edu.tw	高雄市鳥松區澄清路 101 號
0109	林正忠	10	男	教師	07-7310608	tech3@csu.edu.tw	高雄市三民區澄清路 102 號
0110	林純純	10	女	教師	07-7310609	tech4@csu.edu.tw	台南市大同區澄清路 103 號
0111	林自立	09	男	教師	07-7310610	tech5@csu.edu.tw	高雄市岡山區澄清路 104 號
0112	吳錦秀	08	女	教師	07-7310611	tech6@csu.edu.tw	高雄市三民區澄清路 105 號
0113	張群偉	11	男	教師	07-7310612	tech7@csu.edu.tw	台北市大安區澄清路 106 號
0114	劉維銘	11	男	教師	07-7310613	tech8@csu.edu.tw	高雄市左營區澄清路 107 號
0115	王玉珠	10	女	教師	07-7310614	tech9@csu.edu.tw	高雄市新興區澄清路 108 號

我們用 SQL 命令建立該 view，並取名為 11_teachers，如下：

```

create view 11_teachers as
select *
from teachers
where dep_ID = "11"

```

新增 view 表格

資料表	執行	記錄	型態	校對	大小
11_teachers	     	0 ¹	檢視	---	unknown
all_course	     	14	MyISAM	utf8_unicode_ci	3.6 KB
classes	     	11	MyISAM	utf8_unicode_ci	3.4 KB
department	     	11	MyISAM	utf8_unicode_ci	2.3 KB
open_course	     	20	MyISAM	utf8_unicode_ci	4.5 KB
public_course	     	0 ¹	檢視	---	unknown
select_course	     	106	MyISAM	utf8_unicode_ci	5.3 KB
simple_students	     	0 ¹	檢視	---	unknown
simple_teachers	     	0 ¹	檢視	---	unknown

執行成功後，可以發現在 course_db 上增加了 11_teachers 表，它的型態是“檢視”(view)，記憶體空間未知(unknown)。我們利用 SQL 命令觀察 11_teachers 表的內容如何，如下：





























































```
Select *
From 11_teachers;
```

teacher_ID	name	Dep_ID	sex	title	tel	mailbox	address
0107	張永誠	11	男	系主任	07-7310606	tech1@csu.edu.tw	高雄市三民區澄清路 100 號
0108	粘添壽	11	男	教師	07-7310607	tech2@csu.edu.tw	高雄市鳥松區澄清路 101 號
0113	張群偉	11	男	教師	07-7310612	tech7@csu.edu.tw	台北市大安區澄清路 106 號
0114	劉維銘	11	男	教師	07-7310613	tech8@csu.edu.tw	高雄市左營區澄清路 107 號
0116	李詠勇	11	男	教師	07-7310615	tech10@csu.edu.tw	高雄市三民區澄清路 109 號

如果使用完，不再需要它時，可將它刪除 SQL 命令(刪除 11_teachers 表)，如下：

```
drop view 11_teachers;
```

11_teachers View 表已不見

all_course	     	14	MyISAM	utf8_unicode_ci	3.6 KB
classes	     	11	MyISAM	utf8_unicode_ci	3.4 KB
department	     	11	MyISAM	utf8_unicode_ci	2.3 KB
open_course	     	20	MyISAM	utf8_unicode_ci	4.5 KB
public_course	     	0 ¹	檢視	---	unknown
select_course	     	106	MyISAM	utf8_unicode_ci	5.3 KB
simple_students	     	0 ¹	檢視	---	unknown
simple_teachers	     	0 ¹	檢視	---	unknown
students	     	58	MyISAM	utf8_unicode_ci	8.6 KB
teachers	     	20	MyISAM	utf8_unicode_ci	5.0 KB

A-2-2 垂直投影的視界 – 減化學生資料

我們擷取資料表某些欄位，如同垂直投影一般。如下圖，一般學生選課管理很少用到電話、電子郵件、地址、電話等資料。因此，我們只要投影有用到的地方即可，譬如姓名、學號、性別。

僅投影部分欄位 smp_students

student_ID	name	class_ID	sex	tel	mailbox	address
90211233	陳邦峰	902113	男	073453211	stu3@csu.edu.tw	高雄市三民區澄清路 102 號
90209218	劉奕佑	902113	男	077310606	stu2@csu.edu.tw	高雄市鳥松區澄清路 101 號
90211302	蔡明真	902113	男	077310607	stu4@csu.edu.tw	台南市三民區澄清路 103 號
90211303	李雯玲	902113	女	077310609	stu5@csu.edu.tw	高雄市三民區澄清路 104 號
90211304	劉雅玲	902113	女	077310609	stu6@csu.edu.tw	高雄市三民區澄清路 105 號
90211305	吳真儀	902113	女	077310608	stu7@csu.edu.tw	高雄市三民區澄清路 106 號
90211306	劉政維	902113	男	077315643	stu8@csu.edu.tw	高雄市三民區澄清路 107 號
90211307	張祥宇	902113	男	077310709	stu9@csu.edu.tw	高雄市三民區澄清路 108 號

我們用 SQL 命令建立該 view，並取名為 smp_students，如下：

```

Create view smp_students as
Select student_ID, name, class_ID, sex
From students;
    
```

新增 view 表格

資料表	執行	記錄	型態	校對	大小
all_course		14	MyISAM	utf8_unicode_ci	3.6 KB
classes		11	MyISAM	utf8_unicode_ci	3.4 KB
department		11	MyISAM	utf8_unicode_ci	2.3 KB
open_course		20	MyISAM	utf8_unicode_ci	4.5 KB
public_course		0 ¹	檢視	---	unknown
select_course		106	MyISAM	utf8_unicode_ci	5.3 KB
simple_students		0 ¹	檢視	---	unknown
simple_teachers		0 ¹	檢視	---	unknown
smp_students		0 ¹	檢視	---	unknown
students		58	MyISAM	utf8_unicode_ci	8.6 KB
teachers		20	MyISAM	utf8_unicode_ci	5.0 KB

執行成功後，可以發現在 course_db 上增加了 smp_students 表，它的型態是“檢視”(view)。我們利用 SQL 命令觀察 smp_students 表的內容如何，如下：

```

Select *
From smp_students;
    
```

student_ID	name	class_ID	sex
90211233	陳邦峰	902113	男
90209218	劉奕佑	902113	男
90211302	蔡明真	902113	男
90211303	李雯玲	902113	女
90211304	劉雅玲	902113	女

如果使用完，不再需要它時，可將它刪除 SQL 命令(刪除 11_teachers 表)，如下：

```
drop view smp_students;
```

A-2-3 多表格投影的視界 – 簡化教師資料

我們可以經由巢狀、合併或混合等技巧，將查詢的結果建立一個視界，這種應用可能更加廣泛。譬如，教師開課大多是系所辦理，很少牽涉到其它系老師，處理老師開課時大多不需要電子郵件、地址等資料，因此，我們可以針對每一個系所，建立一個 view 來專門處理該系所開課使用的教師表格，也許可以減少不需要的資料負荷。譬如開啟一個 11_course_teachers 視界{teacher_ID, name, sex}，專門給資管系使用。我們用 SQL 命令建立該 view，並取名為 11_course_teachers，如下：

```
create view 11_course_teachers as
select teacher_ID, name, sex
from teachers
where dep_ID = (select dep_ID
                from department
                where dep_name="資訊管理系")
```

新增 view 表格

資料表	執行	記錄!	型態	校對	大小
11_course_teachers		0 ¹	檢視	---	unknown
all_course		14	MyISAM	utf8_unicode_ci	3.6 KB
classes		11	MyISAM	utf8_unicode_ci	3.4 KB
department		11	MyISAM	utf8_unicode_ci	2.3 KB
open_course		20	MyISAM	utf8_unicode_ci	4.5 KB

執行成功後，可以發現在 course_db 上增加了 11_course_teachers 表，它的型態是“檢視”(view)。

我們利用 SQL 命令觀察其內容如何，如下：

```
Select *
From 11_course_teachers;
```

teacher_ID	name	sex
0107	張永誠	男
0108	粘添壽	男
0113	張群偉	男
0114	劉維銘	男

如果使用完，不再需要它時，可將它刪除 SQL 命令，如下：

```
drop view 11_course_teachers;
```

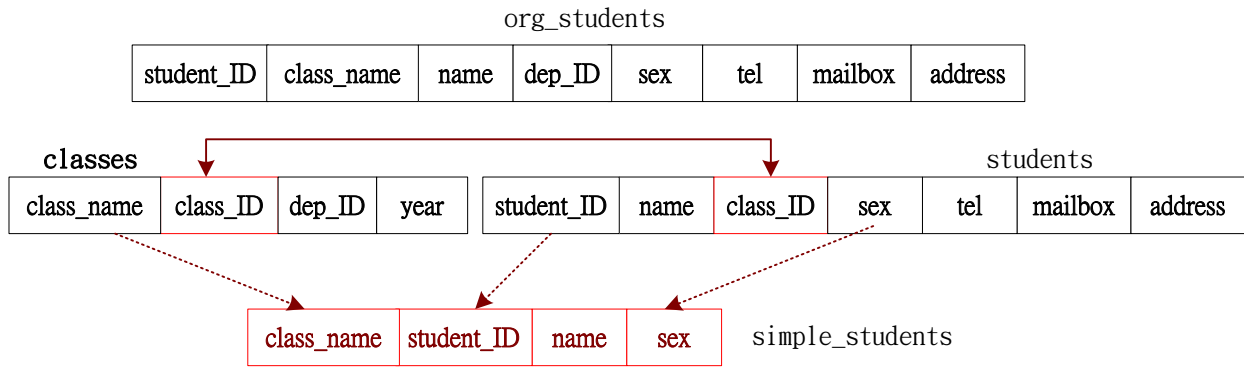
A-3 視界的運用

瞭解視界的產生方法之後，接下來我們用幾個範例來說明視界的運用。

A-3-1 視界 - 還原資料表的運用

(A) 視界 simple_students

我們用 students 與 classes 兩張資料表來說明資料庫邏輯設計的問題。如下圖所示，原來記錄學生資料只要如同 org_students 表，裡面登錄了該學生在哪一班(class_name)，在哪一系(dep_ID)。但班級名稱每年要修改，譬如今年四資三甲，明年就變成四資四甲，所以每年必須變更每一位學生的資料，這樣設計好像不理想。另外，學生可能轉系或轉班，又必須修改 class_name 與 dep_ID，如果僅修改一個地方也是發生錯誤。因此，為了此正規化原則，我們將它拆解成 students 與 classes 兩張資料表，每年只要修改 classes 內的 class_name，則全校學生資料隨之修改。此設計好像不錯，但延伸了新的問題，我們查詢學生資料時，大多以班級大多以代碼表示，但如要顯示班級名稱時，又要結合 classes 表格才能達成。再思考下一個問題，一般處理學生選課事務時，很少用到地址、電子郵件與電話，何不建立一個 view，來回復原來 org_students 的需求就好，如圖中的 simple_students。



在本書 course_db 已建立了此資料表，其結構 {class_ID, student_ID, name, sex}，SQL 命令如下：(輸入之前，必須先把原有的刪除)

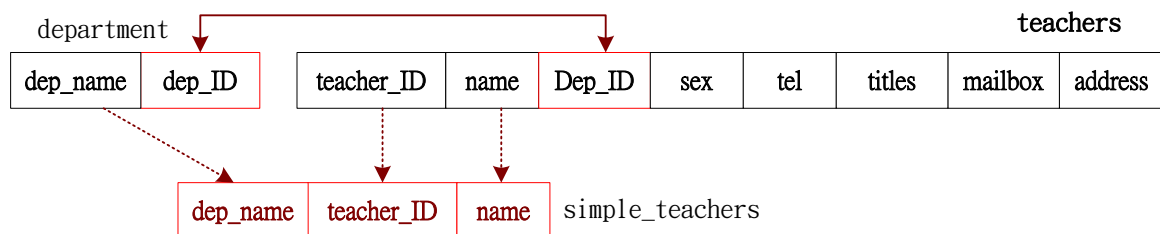
```

Create view simple_students as
Select B.class_name, A.student_ID, A.name, A.sex
From students as A, classes as B
Where A.class_ID = B.class_ID;
    
```

欄位	型態	校對	屬性	Null	預設值	附加	檢視
class_name	varchar(40)	utf8_unicode_ci		否			
student_ID	varchar(40)	utf8_unicode_ci		否			
name	varchar(40)	utf8_unicode_ci		否			
sex	varchar(40)	utf8_unicode_ci		否			

(B) 視界 simple_teachers

同樣的問題也發生在 teachers 與 department 資料表。如下圖所示，我們將它們回復成 simple_teachers 視界，不但能方便運用，也不會破壞原來的設計原則。



在本書 course_db 已建立了此資料表，其結構 {dep_name, teacher_ID, name}，SQL 命令如下：(輸入之前，必須先把原有的刪除)

```

Create view simple_teachers as
Select B.dep_name, A.teacher_ID, A.name
    
```

From teachers as A, department as B

Where A.dep_ID = B.dep_ID

欄位	型態	校對	屬性	Null	預設值	附加	檢視
dep_name	varchar(40)	utf8_unicode_ci		否			
teacher_ID	varchar(40)	utf8_unicode_ci		否			
name	varchar(40)	utf8_unicode_ci		否			

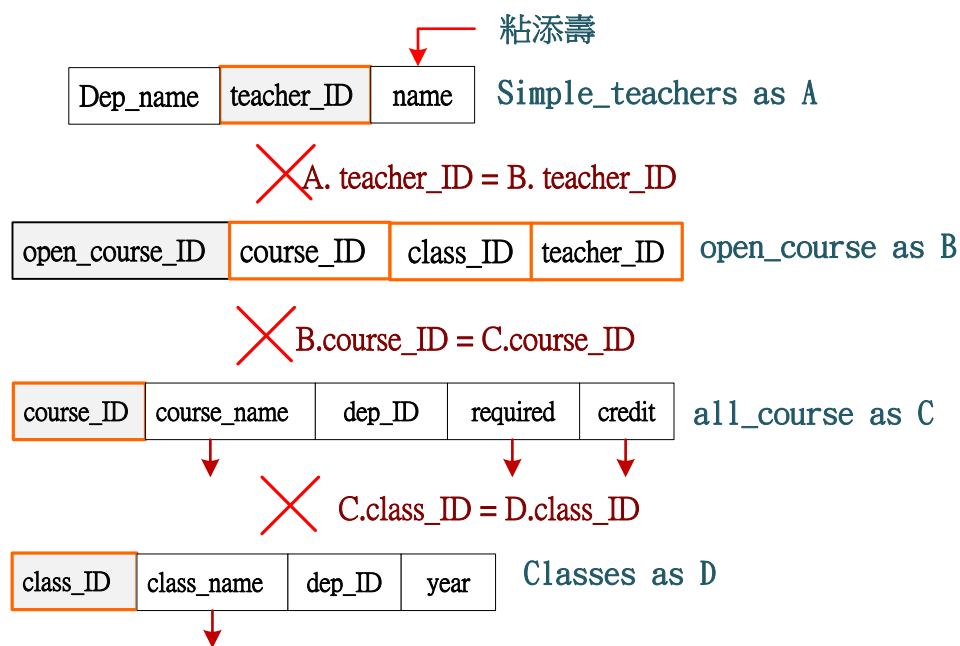
A-3-2 範例研討 - 查詢教師開課課程

(A) 系統功能：ExA_1

請查詢『粘添壽』老師這學期開了那些課程，請依照班級名稱、課程名稱、必選修、學分數支順序印出。

(B) 系統分析

此題目，我們在 6-4-1 節討論過，我們改用 simple_teachers 視界來處理，如下：



(C) SQL 命令：

將上圖的關連，以 SQL 命令如下：


```

Select D.class_name, B.course_ID, C.course_name, C.required, C.credit
From simple_teachers as A, open_course as B, all_course as C, classes as D
where A.name = "粘添壽"

And A.teacher_ID = B.teacher_ID
And B.course_ID = C.course_ID
And B.class_ID = D.class_ID;

```

class_name	course_ID	course_name	required	credit
日四資管四乙	40BQ31	Java程式設計(一)	必	3
日四資工三甲	40BQ41	網路管理技術	選	3
進四資管二丙	90B252	資料庫管理系統	必	4

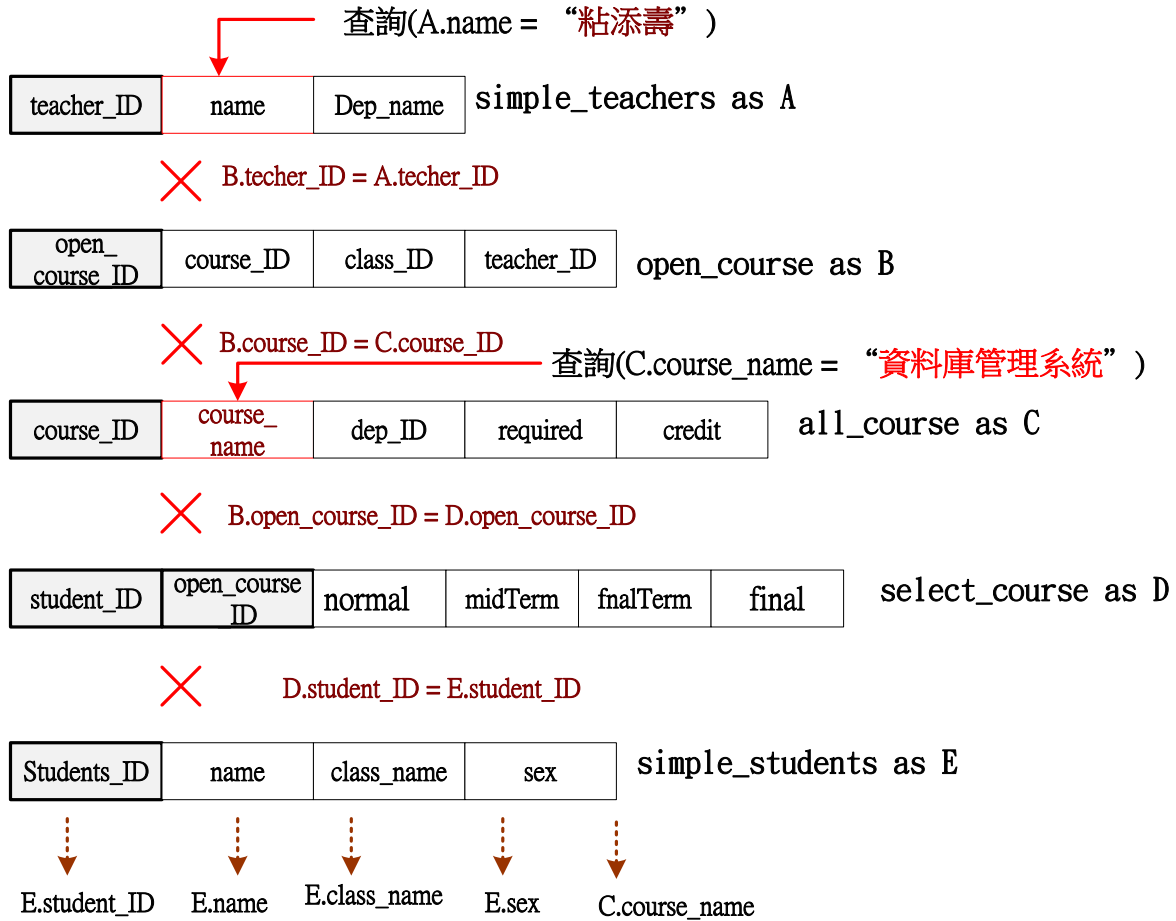
A-3-3 範例研討 - 查詢教師授課學生名冊

(A) 系統功能：ExA_2

請查詢這學期 "粘添壽" 老師所開 "資料庫管理系統" 課程的學生名冊。

(B) 系統分析

此題目，我們用原資料表做過，現使用 simple_teachers 與 simple_students 兩個 view 表格，來處理看看，下圖是資料表關連圖，假設：**查詢選修粘添壽老師所開 "資料庫管理系統" 的學生名冊。**



(C) SQL 命令：(查詢選修粘添壽老師所開 "資料庫管理系統" 的學生名冊)

將上圖的關連，以 SQL 命令如下：

```

select C.course_name, E.class_name, E.student_ID, E.name, E.sex
from simple_teachers as A,
      open_course as B,
      all_course as C,
      select_course as D,
      simple_students as E
where A.name = "粘添壽"
and A.teacher_ID = B.teacher_ID
and C.course_name = "資料庫管理系統"
and B.course_ID = C.course_ID
and B.open_course_ID = D.open_course_ID
and D.student_ID = E.student_ID;
    
```

course_name	class_name	student_ID	name	sex
資料庫管理系統	日四資管三甲	40011111	王口毓	男
資料庫管理系統	進四資管二丙	90209218	劉奕佑	男
資料庫管理系統	進四資管二丙	90211233	陳邦峰	男
資料庫管理系統	進四資管二丙	90211302	蔡明真	男
資料庫管理系統	進四資管二丙	90211303	李雯玲	女
資料庫管理系統	進四資管二丙	90211304	劉雅玲	女
資料庫管理系統	進四資管二丙	90211305	吳真儀	女
資料庫管理系統	進四資管二丙	90211306	劉政維	男
資料庫管理系統	進四資管二丙	90211307	張祥宇	男
資料庫管理系統	進四資管二丙	90211308	張敦揮	男
資料庫管理系統	進四資管二丙	90211309	劉郁萱	女
資料庫管理系統	進四資管二丙	90211310	陳姿美	女

A-4 多表格結合的 view

A-4-1 複雜的視界 – 通用課程表

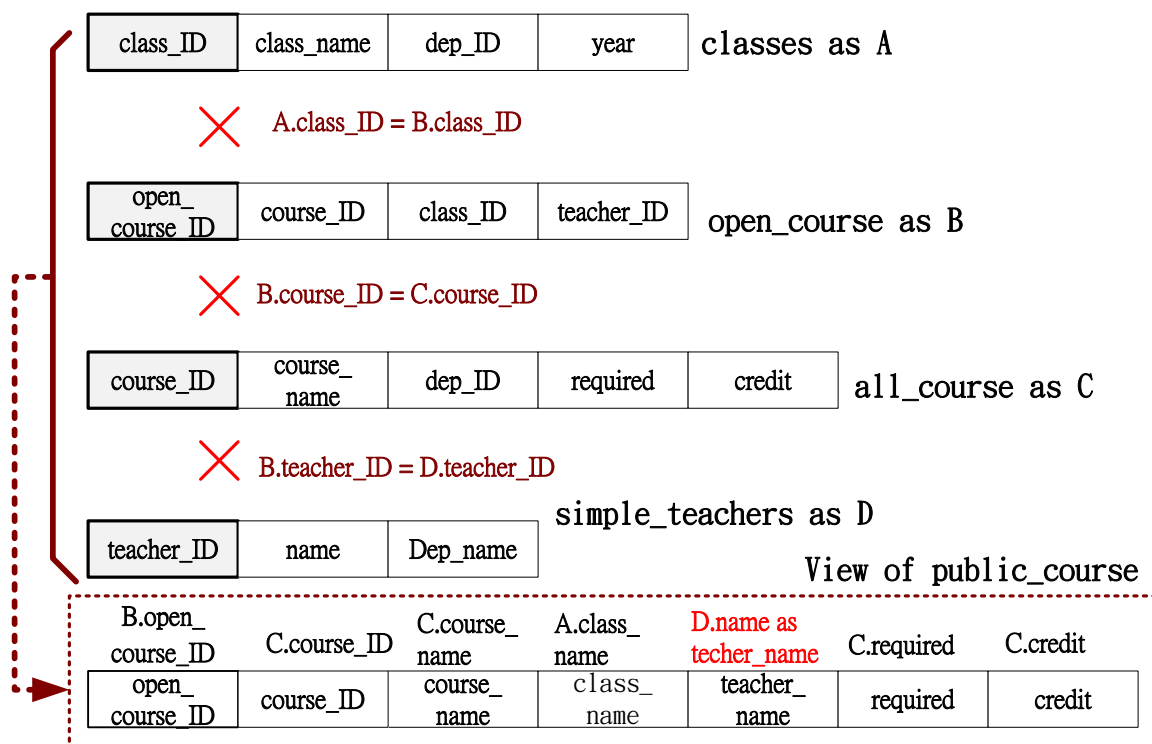
(A) 系統功能：public_course View

到目前我們認識可以利用 View 來回復原來資料表的抽象格式，但回復功能不僅 simple_students 與 simple_teachers 兩視界範圍而已。我們以 course_db 為例，此資料庫最主要提供系所規畫課程與每學期教師開課、學生選課、登錄成績功能，但有許多複雜關係，歸類如下：

- (1) 系所全學年規畫了哪些課程、
- (2) 老師在哪一個班級、開哪一門課、
- (3) 同一門課，不同老師可能在不同班級開課、
- (4) 學生可以隨意選課，不一定在自己班級、
- (5) 開課以班級為單位，不列入系所、
- (6) 學生同一門課不可同時修 2 次以上、、、等條件限制。

從資料庫邏輯設計(或正規化原則)的觀點，我們為了滿足上述條件，我們將課程相關資料分割為若干個資料表來存放，並在資料表之間建立關聯，這是關聯式資料庫系統的最基本原理。我們用巢狀、合併與混合等技巧就是為了要克服這些雜亂的資料表中萃取資訊之目的。

如果，我們可以建立一個 View 來回復原來需求的資料，摒除掉一些為了滿足正規化所增加的欄位，也許在資料查詢就可以方便許多，然而，它又是一個抽象不存在的表格，也不會破壞原來正規化原則。如下圖所示，我們將 classes、open_course、all_course 與 simple_teachers 等表格合併，找出『這學期』與教師開課全部相關的資料，建立一個 View，如果要查詢系所開課、教師開課、班級開課、課程資料、學生選課、等等，只要與課程相關的資料都由這個 View 就可以找到。所建立 public_course 視界的結構是 {開課代碼、課程名稱、班級名稱、教師名稱、必選修、學分數}。



(B) 利用 SQL 命令產生 public_course View，語法如下：

```

Create view public_course as
Select B.open_course_ID, C.course_ID, C.course_name, A.class_name,
       D.name as Teacher_name, C.required, C.credit
From classes as A, open_course as B, all_course as C, simple_teachers as D
Where A.class_ID = B.class_ID
And B.course_ID = C.course_ID
And B.teacher_ID = D.teacher_ID;
    
```

欄位	型態	校對	屬性	Null	預設值	附加	檢視
open_course_ID	int(40)			否	0		
course_ID	varchar(40)	utf8_unicode_ci		否			
course_name	varchar(40)	utf8_unicode_ci		否			
class_name	varchar(40)	utf8_unicode_ci		否			
Teacher_name	varchar(40)	utf8_unicode_ci		否			
required	varchar(40)	utf8_unicode_ci		否			
credit	smallint(6)			否			

(C) 直接用 SQL 驗證該視窗是否滿足需求：

```
select *
from public_course;
```

open_course_ID	course_ID	course_name	class_name	Teacher_name	required	credit
2	90B252	資料庫管理系統	進四資管二甲	林正忠	必	4
3	90B252	資料庫管理系統	進四資管二乙	李詠勇	必	4
1	90B252	資料庫管理系統	進四資管二丙	粘添壽	必	4
4	90B253	資料結構	進四資管二甲	林正忠	必	4
5	90A902	電腦動畫設計	進四資管二乙	劉維銘	選	2
6	90A900	數位內容概論	進四資管二丙	張永誠	選	2
7	909257	互動式網頁設計	進四資管二甲	林自立	必	3
8	90B908	資料庫程式設計	進四資管二乙	王玉珠	必	3

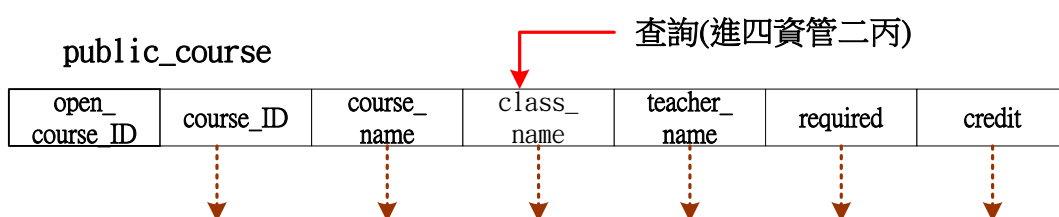
A-4-2 範例研討 – 查詢班級所開課程

(A) 系統功能：ExA_2

請查詢 "進四資管二丙" 班級這學期開了那些課程，請列出開教師、課程名稱、學分數，以及必選修。

(B) 系統分析

這個題目用原來的系統結構，可需要合併好幾個資料表，我們嘗試使用 public_course 視界來做看看。如下圖所示，我們只要一個查詢語句就可以完成。



(C) SQL 語法：(查詢進四資管二丙所開的課程)

```
Select course_ID, course_name, class_name, teacher_name, required, credit
From public_course
Where class_name = “進四資管二丙”;
```

course_ID	course_name	class_name	Teacher_name	required	credit
90A900	數位內容概論	進四資管二丙	張永明	選	2
90B252	資料庫管理系統	進四資管二丙	粘添壽	必	4

A-4-3 自我挑戰 – 查詢課程開在哪些班級

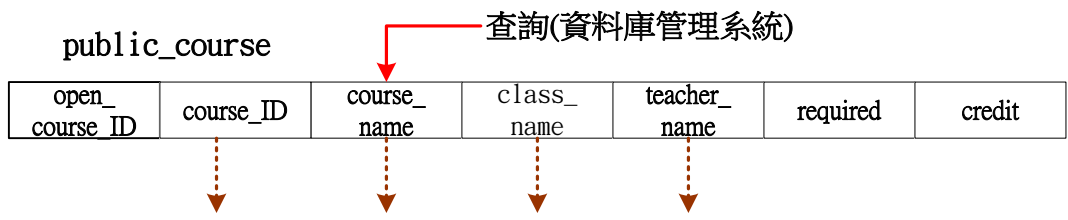
(A) 程式功能：PMA_1

請查詢全校 "資料庫管理系統" 課程開在那些班級。請列出：課程代碼、課程名稱、班級名稱、開課教師名稱。如下：

course_ID	course_name	class_name	Teacher_name
90B252	資料庫管理系統	進四資管二乙	李詠勇
90B252	資料庫管理系統	進四資管二丙	粘添壽

(B) 提示：

只要利用課程名稱，在 public_course 視界上就可以查到相關資訊，如下圖所示。



A-4-4 自我挑戰 – 查詢授課班級成績單

(A) 程式功能：PMA_2

請查詢：粘添壽老師在進四資二丙班級，所開資料庫管理系統課程的學生成績單。結果如下：

student_ID	class_name	name	final
40011111	日四資管三甲	王〇堯	76
90209218	進四資管二丙	劉奕佑	90
90211233	進四資管二丙	陳邦峰	90
90211302	進四資管二丙	蔡明真	86
90211303	進四資管二丙	李雯玲	90
90211304	進四資管二丙	劉雅玲	100
90211305	進四資管二丙	吳真儀	78
90211306	進四資管二丙	劉政維	70
90211307	進四資管二丙	張祥宇	80
90211308	進四資管二丙	張敦揮	81
90211309	進四資管二丙	劉郁萱	82
90211310	進四資管二丙	陳姿美	83

(B) 提示：

學生選課與登錄成績 (select_course)並沒有在 public_course 視界內。如下圖所示，執行網頁收到所欲查詢的 3 個資料(course_name, class_name, teacher_name)之後，由 public_course 視界查詢出該課程的開課代碼(open_course_ID)，再到 select_course 查詢出修課學生的 student_ID 與成績(final)，最後與 simple_students 視界合併查出學生的班級名稱與姓名。

