

第二章 資料庫簡介與 course_db

2-1 資料庫簡介

我們利用下面幾個章節來簡述資料庫的基本概念。

2-1-1 資料庫功能

顧名思義，資料庫是儲存資料的地方，就好像材料倉庫是存放材料的地方，但如果材料沒有分類排列放置的話，當材料數量過多時，我們將很難在一個大倉庫裡找尋所要的東西。另一方面，也很難清點、計算、統計材料的數量。當然啦，資料數量、種類不夠多的話，根本不要考慮用到資料庫儲存，但目前任何一家公司，不論客戶資料、工作項目、員工人數、貨品、等等，數量與種類都非常的大，如不用資料庫來儲存的話，在增加、修改、擷取、刪除、等等管理上會非常困難。

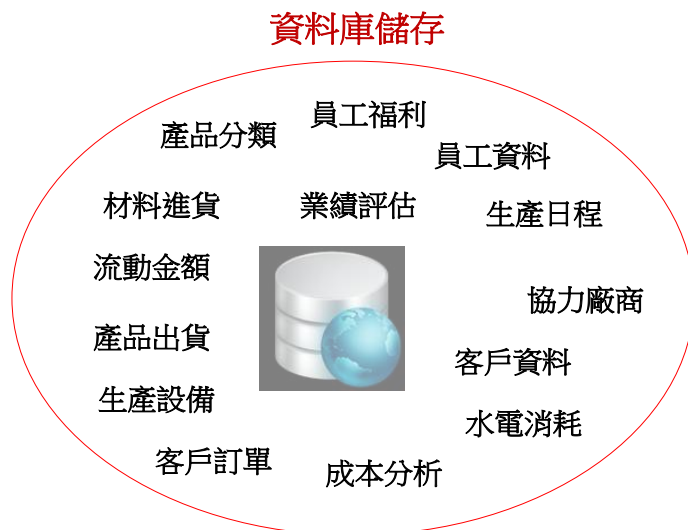


圖 2-1 資料庫內容

圖 2-1 是一般公司行號的資料庫環境。目前任何公司行號內的作業，大多仰賴資訊系統來整合管理。由電腦來管理公司內人事、生產、行銷、客戶等等運作事項。亦是，有一大堆的資料需要擲入電腦內，讓電腦來分類管理。彙整管理所有資料的裝置就是『資料庫』。可以說，資料庫是公

司所有運作的核心，它的重要性是無可倫比的。因此，學習資料庫是現代資訊人員不可或缺的基本能力。

2-1-2 資料庫結構

從簡單的構思，大量資料一定要將它分類儲存，才會比較容易搜尋、更新或刪除等等管理。資料庫也不例外，它也是將資料分類再儲存。因此，在一個資料庫內必須建立許多『資料表』(table)，再將資料分類儲存在各個資料表內。圖 2-2 是一般資料庫的結構。

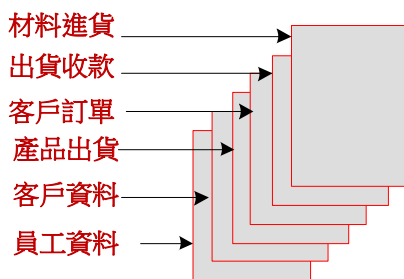


圖 2-2 一個資料庫內有若干個資料表

2-1-3 資料表的關聯

當資料表夠多時，我們希望資料表之間建立關聯，才比較容易由一個資料表找到另一個資料。這樣的設計，可以減低資料表的欄位與數量，也可降低不一致性問題(這裡不討論如何降低)。我們以圖 2-3 說明資料表的關聯性，我們為了簡化資料表的數量，將與客戶有關的資料分為四個資料表存放。客戶資料表專門存放與公司往來的客戶資料，客戶訂單表存放客戶的訂單，只要這兩個資料表建立關聯，在訂單資料表內就不需放置客戶的詳細資料，譬如客戶地址、電話、負責人、甚至信用度等等。另一方面，在客戶資料表內就不需放置有關客戶訂購的詳細清單。另外，再加上產品出貨與出貨收款兩個資料表，就可以將所有客戶的資料、訂貨、出貨、收款等資料分類收集。只要四個資料表之間建立關聯，我們就很容易可以查詢出：某位客戶訂購哪些東西、要寄到甚麼地方、已經出貨了沒有、出貨後是否已經收款、、、等等。

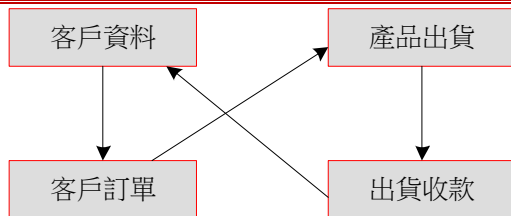


圖 2-3 資料表之間的關聯

資料庫內所有資料表都建立完善的關聯，我們就可以利用這些關聯製作出一套完美的資訊管理系統。圖 2-4 是資料庫內資料表之間的關聯。

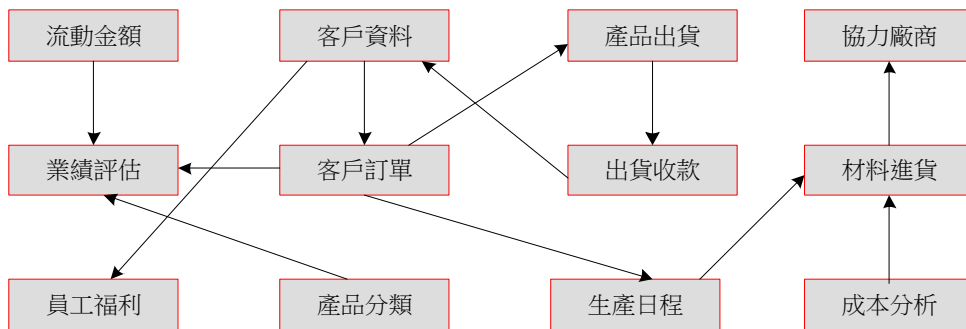


圖 2-4 資料庫內資料表之間的關聯

2-1-4 資料表的屬性

基本上，一個資料表描述一個獨立的『實體』 (Entity) 或『個體』 (Body)，它可能是有形、抽象、或無形的事件或實體東西。譬如：客戶資料是描述有形的實體，銷售業績是無形的個體。另外，到底要如何描述一個實體呢？這是一個很實際的問題，可能要依照系統的需求而定。譬如：描述員工資料可能需要：姓名、員工代號、地址、電話號碼、等等現象，這些現象就稱為資料表的『屬性』 (Attribute)。以員工資料為例，如在銷售管理、生產管理、人事管理或會計管理，它們所要的資料也許不同，建構員工資料的屬性也不會相同。

簡單的說，一個資料表是由若干個屬性所構成，選擇哪些屬性才滿足系統需求，這是另外一門學問不在此探討。另外，每一個屬性是由一個資料來表示，這表示此資料需要一個有規範的資料型態，譬如：整數、浮點數、字串、時間、日期、等等，每一種資料型態也會規範其大小，譬如字

串有可能是 20 字元(VARCHAR(20))、40 字元(VARCHAR(40))等等；甚至，規定某一屬性是否允許『空白』(NULL)。如果違反屬性的規範，資料庫系統將不允許存入，否則即破壞規則。

『屬性』這個名詞並不親民，習慣上都稱為資料表的『欄位』(Field)。圖 2-5 是 classes 資料表的結構，它由四個欄位所構成，分別是班級代碼(class_ID)、班級名稱(class_name)、所屬系別的代碼(Dep_ID)與入學年度(year)，也分別限定它們的資料型態與範圍。

欄位	型態	校對	屬性	Null	預設值	附加
class_ID	varchar(40)	utf8_unicode_ci		否		
class_name	varchar(40)	utf8_unicode_ci		否		
Dep_ID	varchar(40)	utf8_unicode_ci		否		
year	varchar(40)	utf8_unicode_ci		否		

圖 2-5 資料表的屬性(欄位)

2-1-5 資料表的主鍵

在一個資料表內大多會放置若干筆資料，這些資料都是由相同的欄位描述著。但不同資料內的欄位內容可能相同，甚至兩筆以上資料的所有欄位都相同也有可能。因此，我們有需要在眾多欄位中至少設定一個欄位的內容，不允許與其他資料相同，甚至可以代表該筆資料，這些欄位就稱為『主鍵』(Primary Key)。以班級資料表為例，class_ID(班級代碼)就是主鍵，它不可以重複，也不允許空值，如圖 2-6 所示。

主鍵

class_ID	class_name	Dep_ID	year
400101	日四企管四甲	10	100
400102	日四企管四乙	10	100
400111	日資管四甲	11	102
400112	日四資管四乙	11	100
401081	日四電子三甲	08	101
401091	日四資工三甲	09	101

圖 2-6 資料表的主鍵

2-1-6 資料表的外來鍵

大致上，我們知道一個資料庫是由多個資料表所構成，一個資料表是由若干個欄位(或稱屬性)所結合而成。然而，又希望這些資料表之間能夠建立關聯，我們才能整合多個資料表，經過處理後才能得到珍貴的訊息(利用 SQL 工具)，正因如此，才能開發出完善的資訊管理系統。但到底要如何讓資料表之間產生關聯呢？很簡單，就是建立『外來鍵』(Foreign Key)。

外來鍵是在資料表內選擇一個欄位參考其他資料表的某一欄位。以圖 2-7 為例，譬如 students 的 class_ID 欄位參考到 classes 的 class_ID 欄位，是表明學生是屬於哪一個班級，如此一來，在 students 資料表內就不需要列出班級名稱(class_name)、哪一系(dep_ID)等資料。另外，teachers 的 Dep_ID 欄位參考到 department 的 dep_ID 欄位，是說明某位老師是屬於哪一系的。

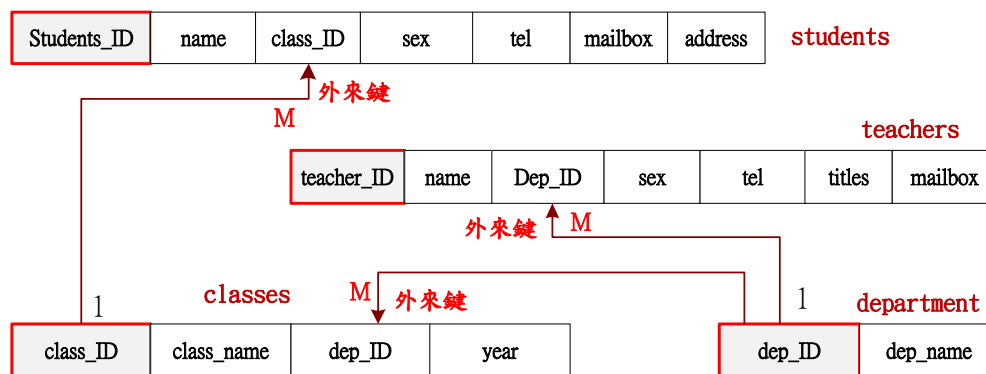


圖 2-7 資料表的外來鍵

2-2 資料庫程式語言

2-2-1 SQL 語言

『結構化查詢語言』(Structure Query Language, SQL)是管理資料庫的主要工具，可用來新增、刪除、更新、查詢資料表內容，也可用來產生、移除資料表，所有對資料庫的操作都要透過它。因此，學習 SQL 語言是管理資料庫的最基本入門技術。如果 SQL 語言運用得宜，就可以由資料庫內粹取各式各樣的資訊或知識。基本上，資料庫的內容是『死的』，如何利用 SQL 來活用它，才是現代資訊人員最需學習的技巧。

SQL 是直譯式語言，進入資料管理介面後，直接輸入 SQL 命令即可操作資料庫。如圖 2-8 所示，由開始進入 MySQL Command Line Client，則出現右邊命令視窗。首先輸入資料庫密碼，再選擇資料庫 (Use course_system)，就可以開始操作。

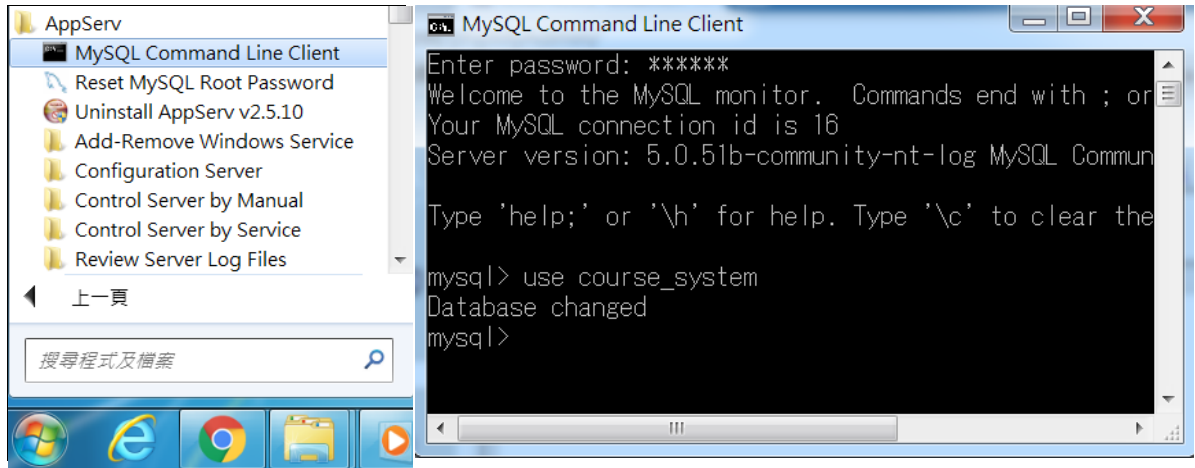


圖 2-8 MySQL 命令行

但我們透過 AppServ 更容易操作 SQL 命令如圖 2-9，(1) 選擇資料庫，接著(2) 按 SQL 選項，(3) 則可輸入 SQL 命令，(4) 點選執行後，將會出現查詢資料的結果。



圖 2-9 AppServ SQL 操作視窗

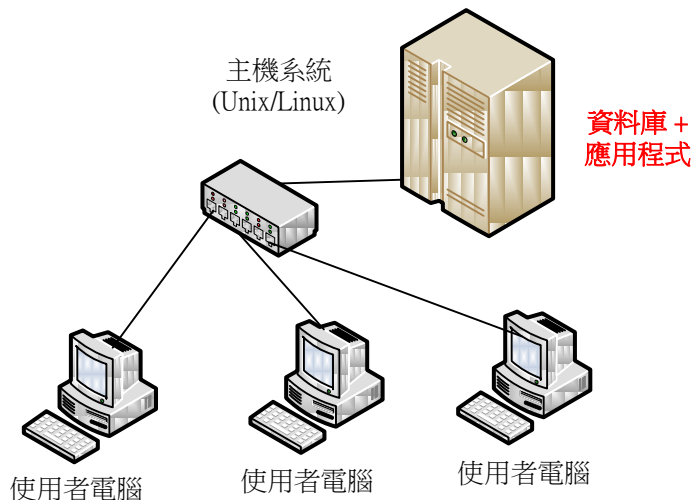
2-2-2 SQL 嵌入式程式

基本上，SQL 僅能以交談式處理資料表而已，並不具有發展應用系統的功能，因此需配合其它發展系統工具才能行。甚麼是合適的發展工具呢？這要看應用系統的操作環境，基本上有兩種架構以下說明之。

(A) 主機型架構 – Basic + SQL

表示『資料庫』與『應用系統』安裝在同一主機上，此主機的作業系統大多允許多人同時使用，譬如 Unix/Linux、PS2 等系統。多位使用者可同時透過網路連線登入系統操作。此系統是最傳統架構，但也是最安全的，任何安全措施只要在主機系統上實施即可。目前許多需要安全性較高的系統，或者小系統還是採用此架構。一般傳統程式語言如 Basic、Fortran、Pascal、C、Java、... 如具有 SQL 函數功能即可發展此類應用系統。

在主機型架構中，使用者電腦僅是終端機功能，系統處理工作大多落在主機上。一般為了簡化主機負荷以及美化操作介面，增強端機功能而設計出『智慧型終端機』，譬如 IBM 3270/5250。



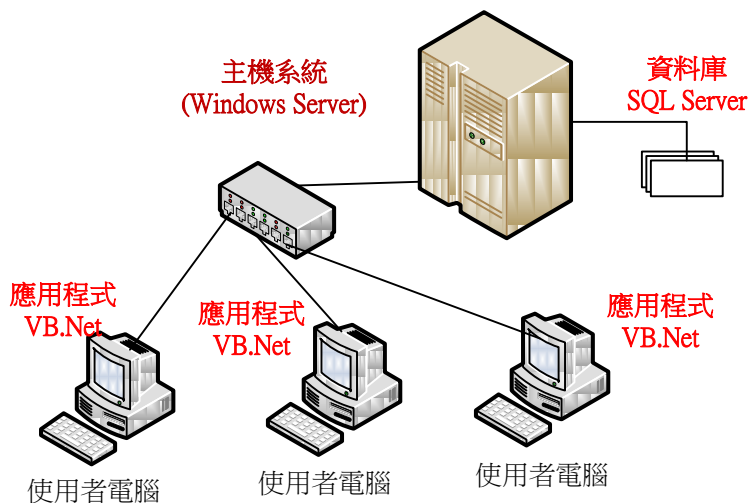
(B) 主從式架構 – Visual 系列

此架構的特點是應用程式與資料庫分開，主機上存放資料庫系統，客戶端電腦上安裝應用程式。運作情況如下：客戶端程式向資料庫伺服器發出要求，伺服器再回應給客戶端，應用程式需要運算與處理都在客戶端電腦上完成，伺服器僅提供資料儲存與處理而已。如此即可減少主機電腦的工作負荷，再說目前客戶端電腦的處理能力大多非常強，分散主機運算處理，增加系統處理能力。由此

可見，主從式架構大多運用於企業內的資訊管理系統，譬如，銷售管理系統、庫存管理系統、等等，主要特質是公司內有安裝相關應用程式的電腦才可以使用。

目前客戶端大多在 Windows 環境下操作，則具有發展視窗介面能力，且將 SQL 編譯程式嵌入，使他們具有『資料庫程式設計』的能力，可採用 Visual Basic、Visual C++、Java WindowBuilder、等等『整合發展工具』(Interaction Development Key, IDK)。

此系統環境安全性雖沒有主機型系統高，但處理能力已提高許多，另外許多安全性問題目前大多可以克服，許多大型公司企業也都採用此架構。但它最大的缺點是當『應用系統更新』時，每一只客戶端電腦都需要更新，如有電腦沒有更新可能產生錯誤。還好目前系統大多有自動下載更新的功能，此問題也大多可以克服。



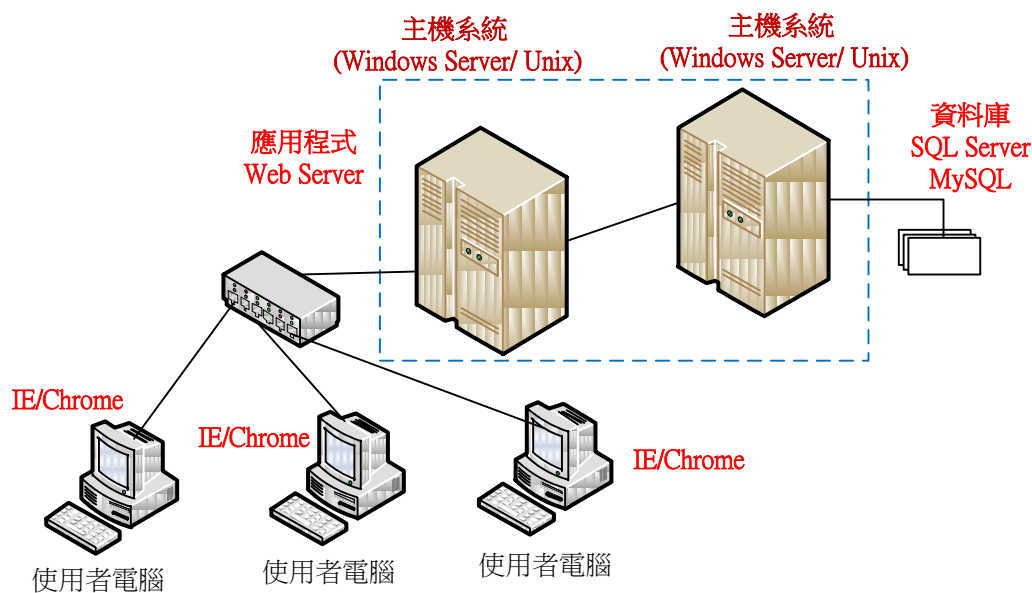
(B) 態網頁架構 – PHP+MySQL、ASP.Net 或 JSP

隨著網際網路的發展，許多資訊系統需拓展到公司外任何電腦都可以操作，不要侷限於特定電腦。但外部電腦型態有許多類型，每部電腦的操作環境也不同，如何達成呢？還好有一個全球通用的介面就是『瀏覽器』。只要客戶端所安裝的瀏覽器符合國際標準，他就可以瀏覽 HTML 所建立的文件。然而提供 HTML 的網頁伺服器(Web Server)如具有 PHP 的編譯能力，我們再將 SQL 嵌入 PHP 語言中，就成為『動態網頁架構』。在此架構下，網頁就不僅顯示文件的功能而已，他可

要求使用者輸入需求，透過 PHP 編譯器向資料庫伺服器索取資料，網頁不再僅是靜態顯示，已成為使用者與資訊系統之間動態的運作模式。

此架構目前有 PHP+MySQL、ASP.Net 或 JSP 等發展工具，它們就是資料庫程式設計的發展工具，亦可發展網頁環境下的應用資訊系統。吾人在網際網路上所使用的系統大多在此環境下發展出來的。

此系統架構是『主機架構』與『主從式架構』的整合體，它將資料庫與應用程式都放在『伺服器端』。『客戶端』向伺服器索取程式(HTML 與 PHP 程式)，並在客戶端的瀏覽器上執行與顯示，當它需要資料時再向伺服器要求，伺服器回應再所需的資料。如此程式雖放在伺服器端但它並不負責執行，當應用系統更新時，僅更新伺服器端程式 (HTML 與 PHP 程式) 即可。因此架構需建構在網頁系統上，又稱為『網頁資料庫系統』。



2-3 網頁資料庫系統架構

欲瞭解網頁資料庫之前，必須由網頁伺服器、PHP 解譯器、資料庫伺服器等三方面，按部就班來說明。

2-3-1 超文字標示語言

『超文字標示語言』(Hyper Text Markup Language, HTML)是一種描述文件格式的語言。利用 HTML 可以建構一份多媒體的文件，文件內不但可以顯示各種字體的字型、大小、顏色，甚至可包含影像、動畫、影片。簡單的說，同樣的文件內容，利用 Word 可以建立多采多姿的版面的樣式，但如利用 WordPad 則能僅能描述文件的內容而已。當我們利用 Word 建立文件時，隨時修改文字的大小、顏色或插入圖形時，Word 會自動插入特殊標識記號來辨識，所用的標識記號就是它的標示語言。但 Word 標示語言所建立的文件僅類似 Microsoft Word 套件才可以觀賞，其它文書工具就無法顯示。HTML 則不然，它希望所建立的文件任何系統都可以顯示，只要系統上安裝有『瀏覽器』(Browser)。並且 HTML 是一種國際標準語言，『瀏覽器』只要含有 HTML 的解譯器，就可以顯示任何人所建立的文件。其運作情況如圖 2-10 所示。

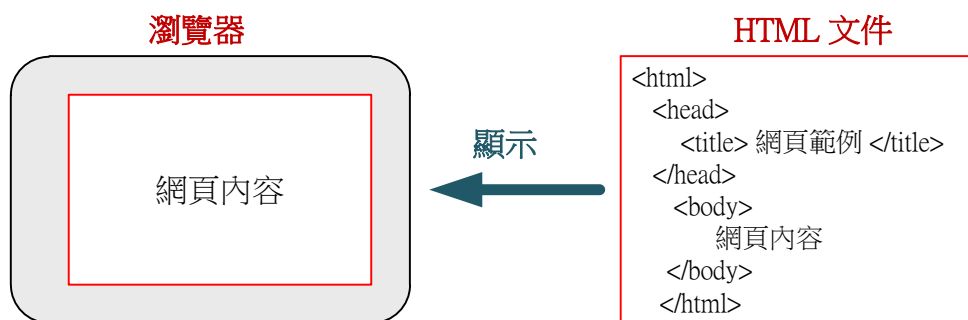
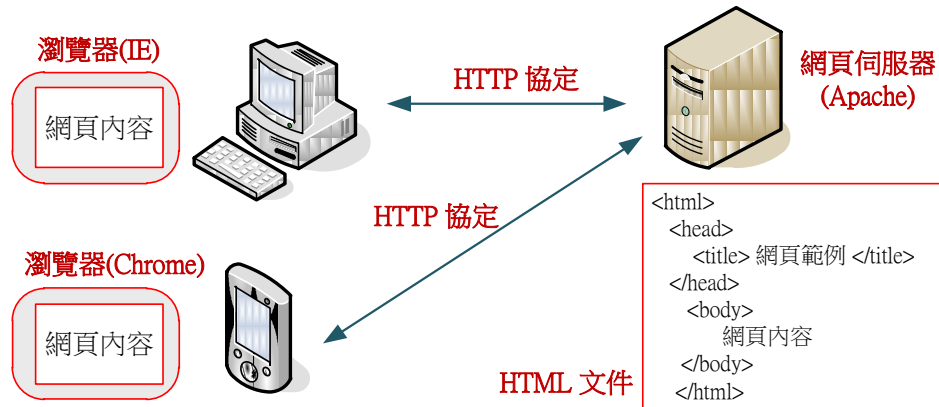


圖 2-10 HTML 與瀏覽器

2-3-2 網頁伺服器

我們所建立的網頁希望達到共通閱讀的目的，它應該放在哪裡才可以呢？因此，我們在網路上建立一個專門存放 HTML 文件的地方，這就是『網頁伺服器』(Web Server)，只要電腦上安裝有瀏覽器就可以閱讀它的文件。HTML 描述文件的格式類似傳統書本上一頁接一頁的顯示，因此將 HTML 顯示的文件稱之為『網頁』(Web Page)。翻動網頁不需按要一頁接下一頁順序，可直接跳頁，跳頁的方法稱為『超連結』(Hyper link)。我們對於每一個網頁伺服器都要有一個專屬名稱，使用者才可以找得到所需的網頁在那裡，此名稱不應該受限於地理位置，應與所屬網頁的特質有關，因此，定義一種稱之為『一致性資源位址』(Universal Resource Locator, URL)。知道文件所在位址，

應告訴對方要用何種協定去存取，HTML 文件存取協定是『超文件傳輸協定』(HyperText Transfer Protocol)。因此，對於網頁伺服器的存取位址格式為『http://www.tsnien.idv』，其中 http 是指定 HTML 傳輸協定、www.tsnien.idv 為網頁的 URL 位址。其運作情況如圖 2-11 所示。



2-3-3 PHP 直譯器

『超文字預處理器』(HyperText Preprocessor, PHP)是被植入 HTML 文件內，預先處理某些特殊程序，將結果表現於 HTML 文件內。基本上，HTML 僅描述文件顯示的方法，它能將一份文件顯示成為美輪美奐的網頁，包含多彩文字、動畫、影像與視訊等等，但僅這些功能並不能滿足目前多元化的社會。譬如，文件內要顯示十句『我愛您』，僅利用 HTML 製作可能要輸入十句同樣的話，對我們學資訊的人一定會感到很不耐，為何不叫它連續顯示十次、一百次就好，這可要仰賴 PHP 的功能了。

PHP 並無法獨立運作，它是被嵌入在 HTML 文件內。當客戶端(如 IE)索取某網頁伺服器(http://www.tsnien.idv)的網頁時，網頁伺服器(如 Apache)會先瀏覽該網頁，發現有 PHP 程式時，它會先執行該程式並將結果張貼於 HTML 文件內，再將網頁傳給客戶端。以圖 2-12 為例，客戶端瀏覽網頁時，發現內網頁內有 PHP 程式(<?php?>)，則呼叫 PHP 直譯程式執行它，再結果張貼將於網頁上(顯示十次我愛您)，瀏覽器就觀察到 PHP 執行後的結果。

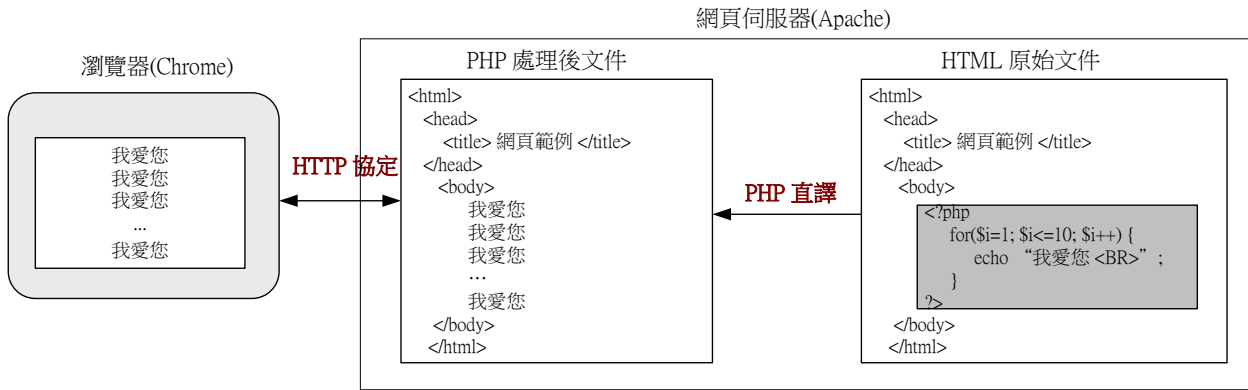


圖 2-12 PHP 程式與 HTML 文件

2-3-4 網頁資料庫架構

網頁伺服器加入了 PHP 處理器之後，增加了許多系統處理的能力。但要利用它來設計資訊處理系統(譬如，醫院管理系統)，好像少了甚麼東西？就是資料儲存核心：『資料庫系統』。試想看看，如果網頁系統再加入了資料庫系統，不但可以製作出各種資訊管理系統，並且還包含多媒體的顯示能力，如此一來，它的應用將更為廣泛，因此，就有所謂三層式應用系統架構的誕生，我們稱它為『網頁資料庫系統架構』，如圖 2-13 所示。

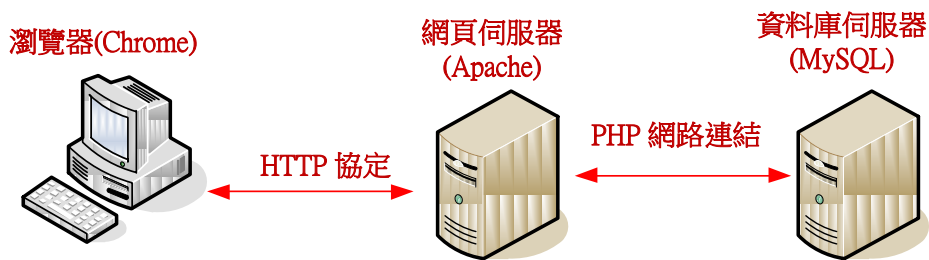


圖 2-13 三層式應用系統架構

既然 PHP 具有如同 C 或 Java 程式設計的功能，也應該如同一般語言具有網路存取的介面程式。我們在 HTML 文件上嵌入一段 PHP 程式，而該程式可連結資料庫、存取資料庫內容，將結果張貼於 HTML 文件上，再傳給瀏覽器顯示，如此即可建立一套美輪美奐的資訊處理系統。

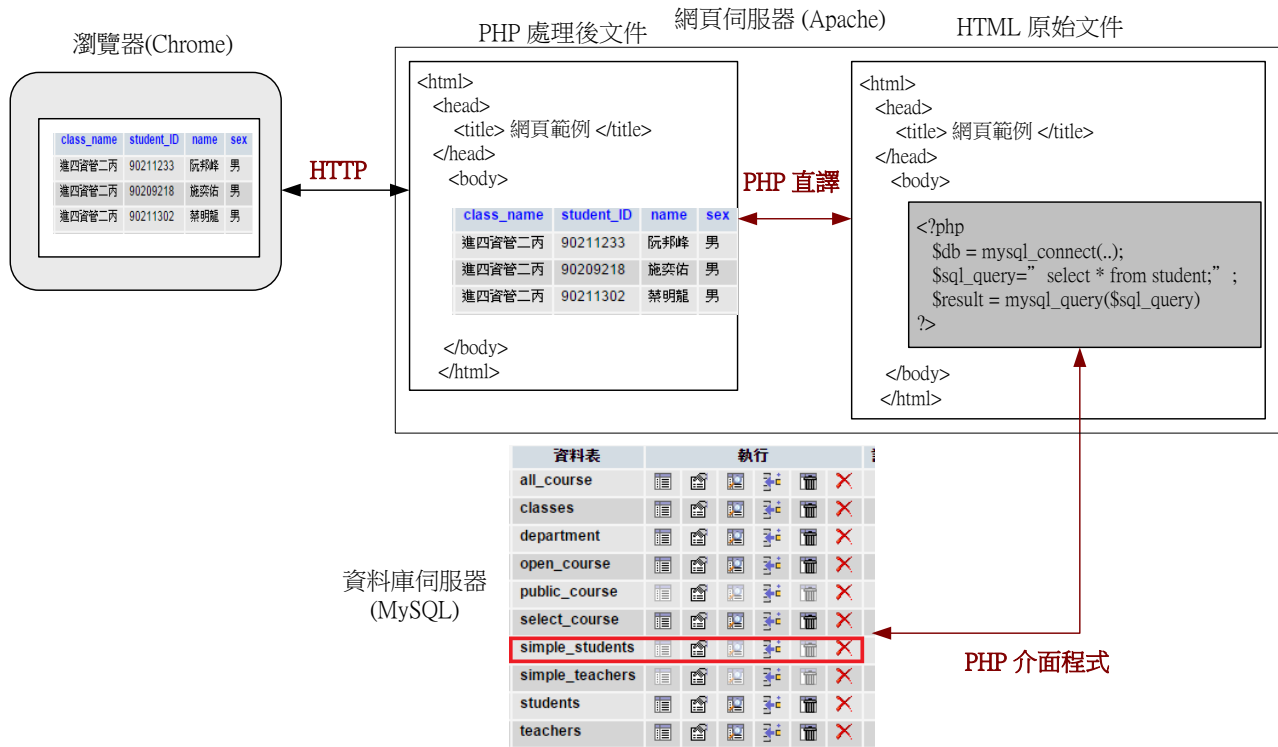


圖 2-14 網頁資料庫的運作程序

圖 2-14 是網頁資料庫系統的運作程序。首先在網頁上發現有 PHP 程式，則呼叫 PHP 直譯器來執行，動作是連結資料庫與查詢資料庫內容。PHP 再將查詢結果寫入 HTML 網頁內(容後說明其技巧)，客戶端就可觀察到查詢資料庫的結果了。

2-4 資料庫範例 – course_db

(有關 course_db 資料與操作方法請參考 『資料庫系統概論』 翻轉電子書)

2-4-1 course_db 資料庫功能

本書在 AppServ 系統上建構一套網頁資料庫，以供本書編寫範例使用，資料庫名為 course_System。它是用來開發學校的『課程管理系統』使用，如圖 2-15，我們期望能製作出下列應用程式：

- (1) 登錄/管理學生資料。
- (2) 登錄/管理教師資料。

- (3) 處理學生轉系、轉班之功能。
- (4) 系所變更名稱、主任。
- (5) 教師開課。
- (6) 學生選課。
- (7) 教師登錄成績。

....

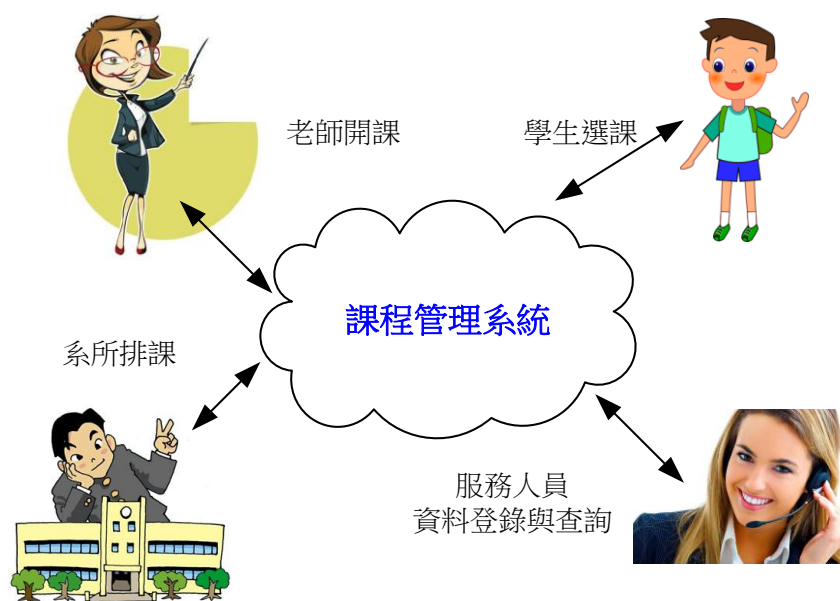
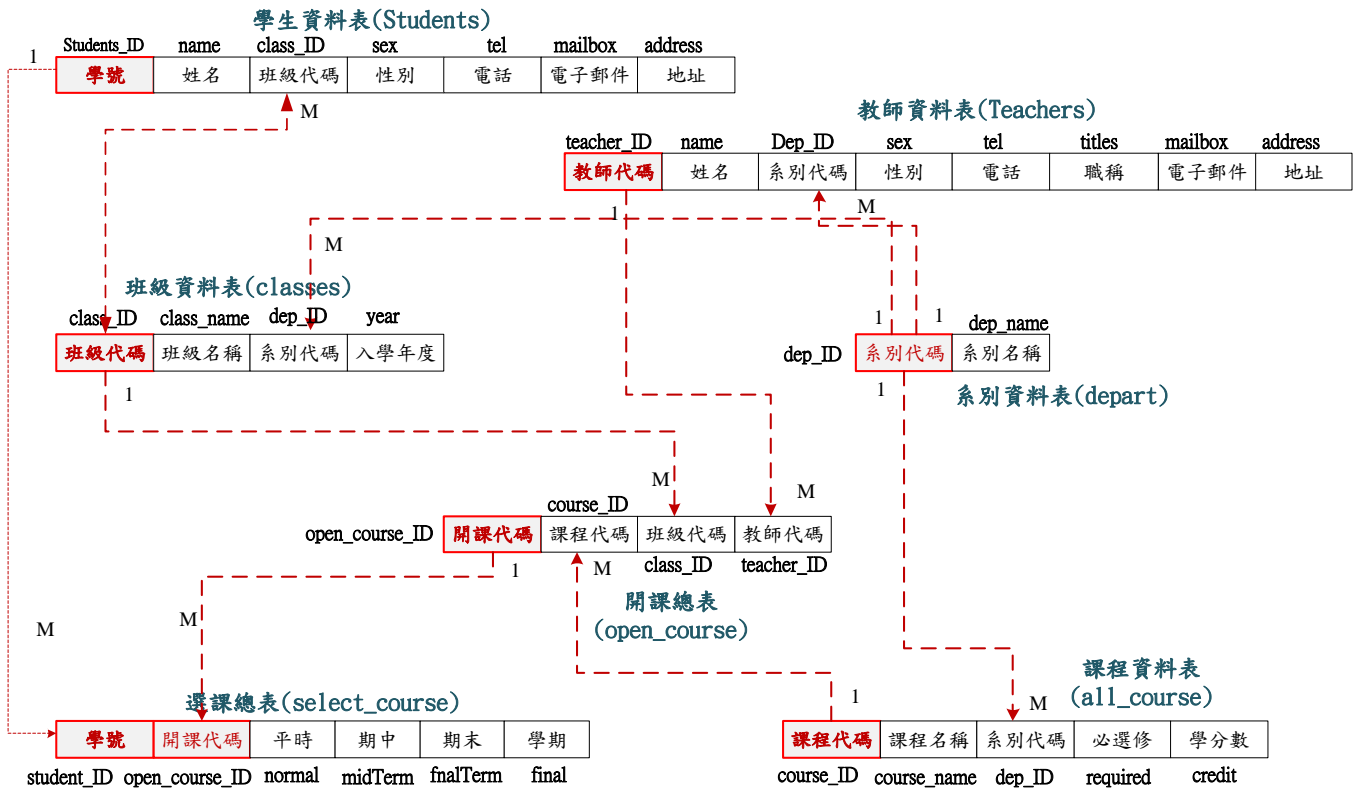


圖 2-15 課程管理系統的功能

2-4-2 course_db 資料表關聯圖

course_db 資料表關聯圖



2-4-3 course_db 資料庫結構

如圖 2-16，在 course_db 資料庫內，我們建立了 7 個資料表，以及 3 個視界，其功能如下：

(個資料表內容隨後說明)

- (1) Students 資料表：供登錄學生資料使用。
- (2) Teachers 資料表：供登錄教師資料使用。
- (3) Classes 資料表：供登錄班級資料使用。
- (4) Department 資料表：供登錄系所資料使用。
- (5) All_course 資料表：全校各系全年級所規劃的課程皆登錄在資料表內。
- (6) Open_course 資料表：供本學期開課使用。每學期各系由 All_course 表內選擇所欲開的課程，並登錄那一位老師、在哪一個班級開課。
- (7) Select_course 資料表：供登錄全校學生選課使用。每學期開始學生由 Open_course 表內選課，其中並登錄該生修課成績。
- (8) Simple_students 視界：簡化學生資料。
- (9) Simple_teachers 視界：簡化教師資料。

(10) Public_course 視界：通用型記錄教師、系所開課之狀況。

資料表	執行	記錄	型態	校對	大小	多餘
<input type="checkbox"/> all_course		14	MyISAM	utf8_unicode_ci	3.6 KB	-
<input type="checkbox"/> classes		11	MyISAM	utf8_unicode_ci	3.4 KB	-
<input type="checkbox"/> department		11	MyISAM	utf8_unicode_ci	2.3 KB	-
<input type="checkbox"/> open_course		20	MyISAM	utf8_unicode_ci	4.5 KB	-
<input type="checkbox"/> public_course		01	檢視	---	unknown	-
<input type="checkbox"/> select_course		106	MyISAM	utf8_unicode_ci	5.5 KB	28 Bytes
<input type="checkbox"/> simple_students		01	檢視	---	unknown	-
<input type="checkbox"/> simple_teachers		01	檢視	---	unknown	-
<input type="checkbox"/> students		58	MyISAM	utf8_unicode_ci	8.6 KB	-
<input type="checkbox"/> teachers		20	MyISAM	utf8_unicode_ci	5.0 KB	-

圖 2-16 course_db 內的資料表

(1) students 資料表

如圖 2-17，資料表 students 包含有 7 個欄位：{student_ID, name, class_ID, sex, tel, mailbox, address}，並以 student_ID 為主鍵(Primary key)、class_ID 為外來鍵，連結到 classes 資料表的主鍵 class_ID，作為登錄該學生所屬班級。

欄位	型態	校對	屬性	Null	預設值	附加
<input type="checkbox"/> <u>student_ID</u>	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/> name	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/> class_ID	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/> sex	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/> tel	varchar(40)	utf8_unicode_ci		是	NULL	
<input type="checkbox"/> mailbox	varchar(40)	utf8_unicode_ci		是	NULL	
<input type="checkbox"/> address	varchar(60)	utf8_unicode_ci		是	NULL	

圖 2-17 students 資料表的屬性

其中，比較特殊的是學號(student_ID)，我們用 8 個數字表示，由左至右排列是：

- (1) 學制(一個數字)：日四技(4)、日二技(2)、日五專(5)、日二專(3)、進四技(9)、進二技(7)...
- (2) 入學年度(兩個數字)：100(00)、101(01)、102(02)、103(03)、104(04)、...
- (3) 系別(兩個數字)：管系(11)、企管系(10)、資工系(09)、電子系(08)、...
- (4) 班級(一個數字)：甲班(1)、乙班(2)、丙班(3)、...
- (5) 座號(兩個數字)：01 ~99。

譬如，學號 = 40311101，表示該生是日間四技、103 年度入學、資管系、甲班、座號為 1。

(2) teachers 資料表

如圖 2-18，資料表 teachers 係登錄教師資料，包含 8 個欄位，以 teacher_ID(員工代號)作為主鍵，另外 Dep_ID 表示該教師目前在哪一系工作，它參考到 department(系別資料表)的主鍵 Dep_ID。欄位 title 表示該教師的職務，譬如教師或系主任。

	欄位	型態	校對	屬性	Null	預設值	附加
<input type="checkbox"/>	<u>teacher_ID</u>	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/>	name	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/>	Dep_ID	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/>	sex	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/>	title	varchar(40)	utf8_unicode_ci		否		
<input type="checkbox"/>	tel	varchar(40)	utf8_unicode_ci		是	NULL	
<input type="checkbox"/>	mailbox	varchar(40)	utf8_unicode_ci		是	NULL	
<input type="checkbox"/>	address	varchar(60)	utf8_unicode_ci		是	NULL	

圖 2-18 teachers 資料表的屬性

(3) classes 資料表

如圖 2-19，資料表 classes 係登錄各班級的資料，包含有 4 個欄位，其中 class_ID 為主鍵，Dep_ID 表示該班是屬於哪一個系所、year 試入學年度。另外，class_ID 包含 6 個數字，表示方法如同 student_ID(學號)的前 6 個數字一樣。

	欄位	型態	校對	屬性	Null	預設值	附加
	<u>class_ID</u>	varchar(40)	utf8_unicode_ci		否		
	class_name	varchar(40)	utf8_unicode_ci		否		
	Dep_ID	varchar(40)	utf8_unicode_ci		否		
	year	varchar(40)	utf8_unicode_ci		否		

圖 2-19 classes 資料表的屬性

(4) department 資料表

如圖 2-20，資料表 department 是用來記錄各系的資料，包含有兩個欄位，以 dep_ID 為主

鍵。

欄位	型態	校對	屬性	Null	預設值	附加
<u>Dep_ID</u>	varchar(40)	utf8_unicode_ci		否		
Dep_name	varchar(40)	utf8_unicode_ci		否		

圖 2-20 department 資料表的屬性

(5) all_course 資料表

如圖 2-21，資料表 all_course 是登錄全校各系所規劃的課程，包含有 5 個欄位。每一個課程都有一個專屬編號(course_ID)，並永遠保存著。各欄位說明如下：

欄位	型態	校對	屬性	Null	預設值	附加
<u>course_ID</u>	varchar(40)	utf8_unicode_ci		否		
course_name	varchar(40)	utf8_unicode_ci		否		
dep_ID	varchar(40)	utf8_unicode_ci		否		
required	varchar(40)	utf8_unicode_ci		否		
credit	smallint(6)			否		

圖 2-21 all_course 資料表的屬性

(1) Course_ID：課程代碼。由 7 個字母編制而成，由左至右說明如下：

- (a) 學制(一個字母)：日四技(4)、日二技(2)、日五專(5)、日二專(3)、進四技(9)、...
- (b) 系別(兩個字母)：資管系(0B)、企管系(0A)、資工系(09)、電子系(05)、...
- (c) 課程屬性(一個字母)：共同科(A)、通識(B)、專業必修(C)、專業選修(D)、...
- (d) 課程編號(三個字母)：000 ~ 999

(2) Course_name：課程名稱。各系可能開相同名稱的課程。

(3) Dep_ID：表示此課程是哪一系所開的。

(4) Required：必修或選修。

(5) Credit：學分數。

(6) open_course 資料表

如圖 2-22，資料表 open_course 記錄著本學期全校所開的課程，這就比較複雜，同一課程可

能由多位老師開在不同班級上，每位老師可能開若干個課程。它包含 4 個欄位，說明如下：

欄位	型態	校對	屬性	Null	預設值	附加
<u>open_course_ID</u>	int(40)			否		auto_increment
course_ID	varchar(40)	utf8_unicode_ci		否		
class_ID	varchar(40)	utf8_unicode_ci		否		
teacher_ID	varchar(40)	utf8_unicode_ci		否		

圖 2-22 open_course 資料表的屬性

- (1) Open_course_ID：開課代碼。該表格的主鍵，我們採用數字，並請系統自動產生一個遞增不可重複的數字(自動編號)。
- (2) Course_ID：課程代碼。是參考到 all_course 的主鍵 course_ID，表示開哪一個課程。
- (3) Class_ID：班級代碼。表示該課程是開在哪一班級。
- (4) Teacher_ID：教師代碼。表示此課程是哪一位老師授課。

(7)select_course 資料表

如圖 2-23，資料表 select_course 是學生的選課記錄，並可登錄成績。它包含 6 個欄位，並且是雙主鍵(student_ID, open_course_ID)，表示哪一位學生、修讀哪一門課。其它是登錄平時成績(normal)、期中考成績(midTerm)、期末考成績(finalTerm)、與學期成績(final)。

欄位	型態	校對	屬性	Null	預設值	附加
<u>student_ID</u>	varchar(40)	utf8_unicode_ci		否		
<u>open_course_ID</u>	varchar(40)	utf8_unicode_ci		否		
normal	smallint(6)			是	NULL	
midTerm	smallint(6)			是	NULL	
finalTerm	smallint(6)			是	NULL	
final	smallint(6)			是	NULL	

圖 2-23 select course 資料表屬性

(8)simple_students 視界

如圖 2-24，視界(view) simple_students 是簡化 students 資料表的欄位，一般選課作業並不需要那麼詳細的個人資料。它包含 4 個欄位，如下：

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			

圖 2-24 simple_students 視界的屬性

(9) simple_teachers 視界

圖 2-25，同樣，簡化 teachers 資料表欄位，如下：

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			

圖 2-25 simple_teachers 視界的屬性

(10) public_course 視界

我們為了資料表的正規化問題(請參考拙著『資料庫管理系統』講義)，將許多資料拆到各個資料表內，如此會造成查詢資料的困難，因此，我們建立 public_course 視界來操作會比較容易。至於，製作技巧容後再說明。它包含 7 個欄位，如圖 2-26 所示。


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			

圖 2-26 public_course 視界的屬性