

## 第九章 資料庫邏輯設計

### 9-1 資料庫的設計流程

如何建立一套合乎需求的資料庫系統，這是一件非常抽象的工作，並非讀了很多書就可以達成，老師所教的方法也大多派不上用場。它是一件非常接近實務的工作，需要長時間經驗累積才可以達成，因此我建議初學者不要研讀過於複雜的方法，由最簡的步驟設計完成一個資料庫來玩玩，熟悉這方面的遊戲規則之後，再學習較嚴謹的方法，這樣比較容易達成。

設計一套資料庫最基礎的方法，是思考如何將真實環境利用圖形將它的運作流程繪出來，再將它表格化。這裡面有一些方法，我們會詳加說明。首先，我們來看一般教科書所說的建立步驟如圖 9-1 所示，以下說明各個階段的工作。

另外需特別注意的是，目前大多採用關聯式資料庫系統，所有資料庫的邏輯設計方法，都是針對它而設計。以下所介紹的方法也是如此。

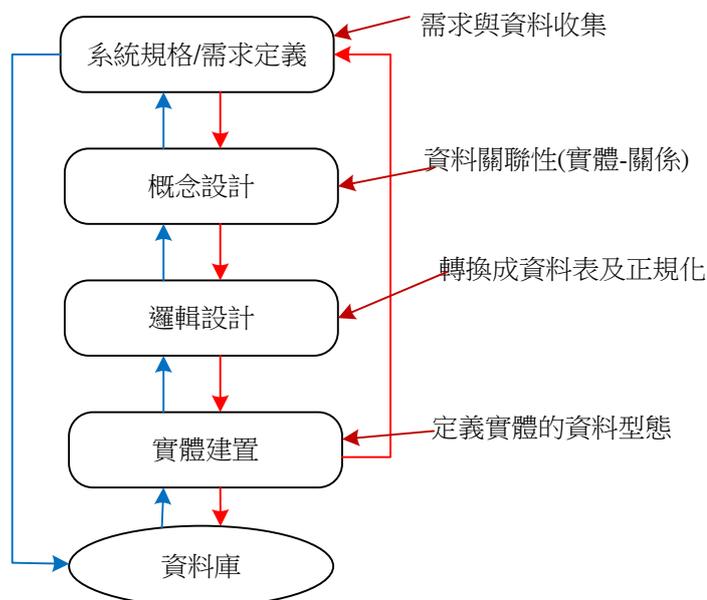


圖 9-1 資料庫邏輯設計步驟

#### 9-1-1 步驟一：定義系統規格與需求

在此階段必須到現場收集資料並瞭解運作情形，主要包含下列兩項：

1. 針對客戶需求，確定設計範圍：瞭解客戶的工作流程、各部門職掌範圍、資料處理方式，以確定資料庫設計的範圍及應具備的功能。
2. 收集與分析資料：除了要明確而具體的找出客戶的需求外，還要收集他們平時使用的各類表單、報表、、、等等，這些都是規劃資料庫的重要參考資料。

### 9-1-2 步驟二：概念設計

在此階段不需考慮資料的儲存及處理等問題，主要是分析及整理收集到的資料，產生一個能符合使用者需求的資料庫模型。一般都用 E-R 模式的圖形來製作。通常我們將此階段再細分為兩個階段：第一個階段是建立分區的概念設計；其次再將各分區設計整合成全區的概念設計，當然這些設計過程都使用 E-R 圖形來規劃，如圖 9-2。

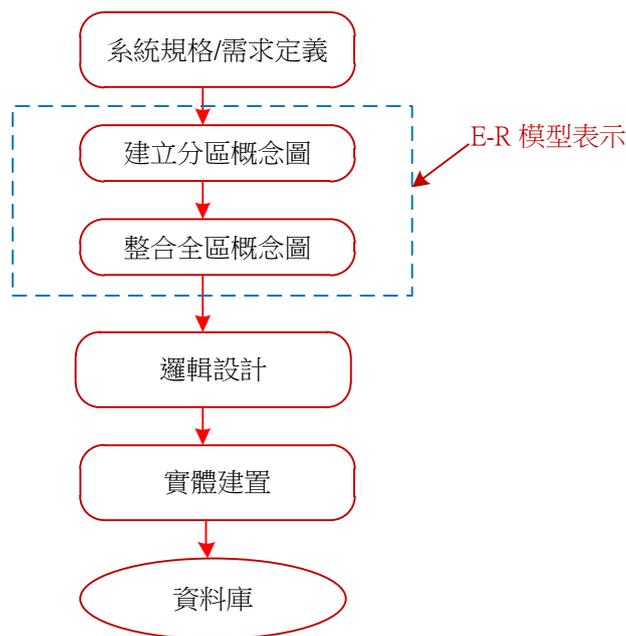


圖 9-2 概念設計

### 9-1-3 步驟三：邏輯設計

此階段是將 E-R 圖形的設計概念，轉換為實際的資料表，並符合資料表的正規化(確保資料的完整性)(後面會說明到)，如圖 9-3 所示。

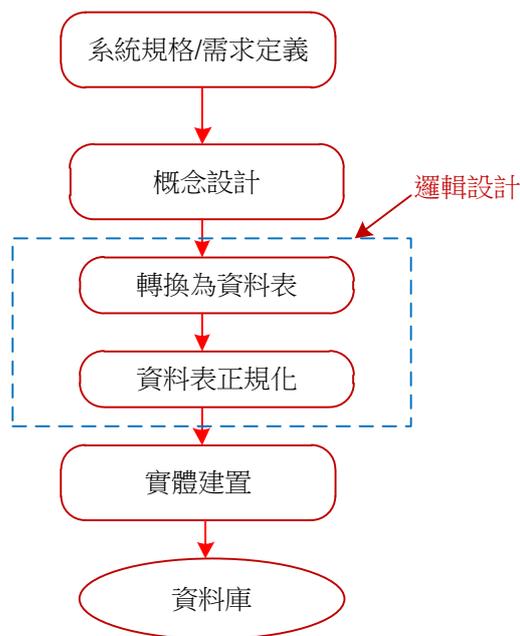


圖 9-3 邏輯設計

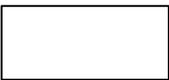
### 9-1-4 步驟四：實體建置

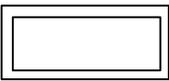
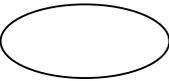
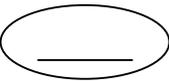
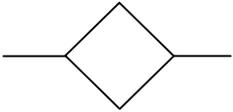
選定所要使用的資料庫，譬如：Access、SQL Server、MySQL、等等，再將上述所設計的資料表建立起來即完成。(下一章實例說明)

## 9-2 E-R Model 實體關係圖

### 9-2-1 何謂 E-R 模型

『實體關係模型』(Entity-Relation (E-R) Model)是將現場實境以圖形化顯現出來的工具。資料庫設計者到客戶現場將訪談或自己所認識的情境以 E-R 圖繪製出來，再利用它和客戶之間溝通。所繪製出來的圖形會隨時改變，一直到雙方都認可才可以繼續往下設計資料庫，也就是說，這個階段會耗費最多的時間。從另一方面來講，我們將現場情境用圖形繪出來，此圖形需要大家都看得懂，因此，需要一個標準化圖示來表示，這就是標準化的 E-R Mode。首先，我們來看標準化的 E-R 圖的符號，如下：

E-R 的元素	表示符號	說明
實體(Entity)		表示真實環境的個體(無形或有形)，譬如：學生、老師、課程、。

弱實體(Weak Entity)		表示該實體是依附別的實體而存在，譬如：監護人是依附學生個體而存在。
屬性(Attribute)		用來描述實體的現象，譬如學生實體的學號、姓名、等等。
衍生屬性 (Derived Attribute)		用來描述屬性再細分其它小屬性，譬如：姓名屬性再區分為姓與名。
鍵值(Key)		用來辨識在實體集合中唯一性的屬性。譬如：學號是學生實體中唯一性的辨識。
關係(Relationship)		用來表示實體之間連繫的管道(關係)，譬如：學生與班別之間的聯繫是班別代碼。

### 9-2-2 實體個體的表示

在真實環境裡，無論有形或無形的個體，都稱為實體。每一個實體是由若干個屬性來描述它的特質。圖 9-4 是表示學生的個體，它是由學號、姓名、性別、地址、電話與電子郵件等屬性來描述。為了達成每一筆記錄代表一個學生，學生之間不可重複，我們發現僅有『學號』滿足條件(1. 不可重複性、2. 最小性)，因此我們選擇它為主鍵。

同學開始設計時，不要太在乎需要哪些屬性來描述，只要能連結起來就好，到底要哪些屬性需視系統需求而定。

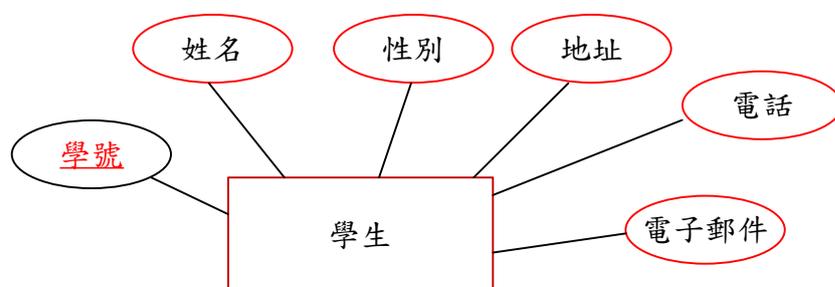


圖 9-4 實體表示

### 9-2-3 含有衍生屬性的個體

如果某些屬性可再細分其它小屬性的話，稱之衍生屬性，如圖 9-5 所示，我們再將姓名區分為『姓』與『名』兩屬性。

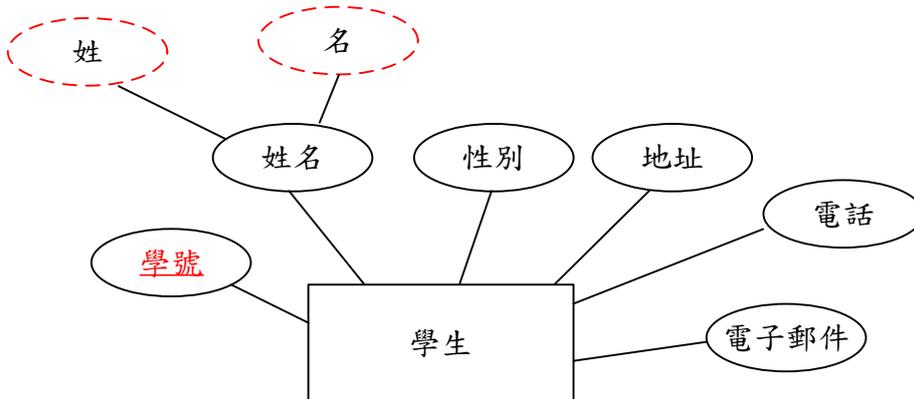


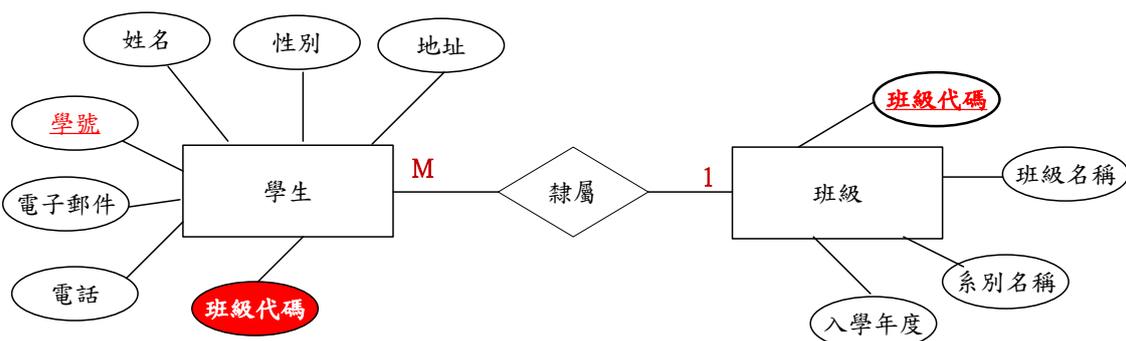
圖 9-5 衍生屬性

## 9-2-4 個體間關係表示

個體之間需建立關係才可以由一個個體找到另一個個體，如果有一個個體它無法和其它個體建立關係，則他就沒有存在的價值。實體之間關係有『一對多，1:M』、『多對一，M:1』、『一對一，1:1』與『多對多，M:N』等四種，前面兩種幾乎一樣，只不過將它倒過來顯示。

### (A) 多對一關係

圖 9-6 表示學生個體與班級個體之間的關係，表示學生是屬於哪一個班級的，因此，它們之間是屬於『隸屬』關係。首先，我們看班級個體，它是由班級名稱、系別名稱、入學年度等屬性描述，其中班級名稱好像擁有唯一性的特質，但我們考慮到他每年都會修改，譬如：今年是『進四資二丙』，明年可能是『進四資三丙』，因此它也不適合當主鍵。我們只好再增加『班級代碼』屬性，確定它不會變更，選定為『主鍵』。



### 圖 9-6 多對一關係

既然，學生要與班級建立隸屬關係，只好在學生個體中增加『班級代碼』屬性，用來表示他是哪一個班級的學生，也稱之為『外來鍵』。一個班級可能擁有若干個學生，因此，班級與學生之間的關係是『多對一，M：1』，我們在圖示表示哪一個是多(M)、哪一個是一(1)。

### (B) 多對多關係

圖 9-7 是多對多的範例，系統設計學生同時可以修讀多門課，每一門課也可以讓多位學生選讀。另外，關係也可以附加屬性，譬如圖(七)中，選課關係增加『分數』的屬性，其用來記錄學生修讀的分數。

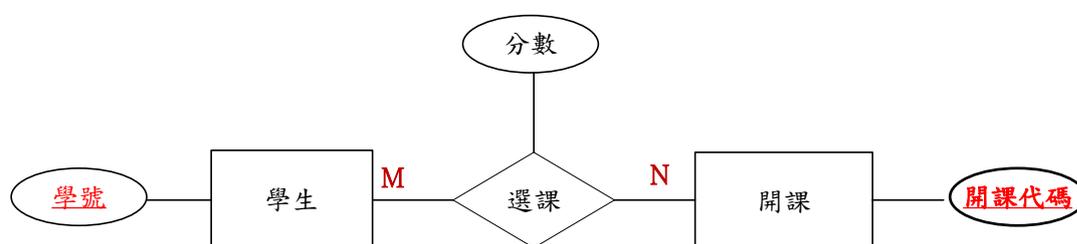


圖 9-7 多對多關係

### (C) 一對一關係

我們用學生與監護人之間關係來說明一對一關係，假設每一位學生只有一位監護人，每一位監護人也僅隸屬一位學生，如圖 9-8 所示。

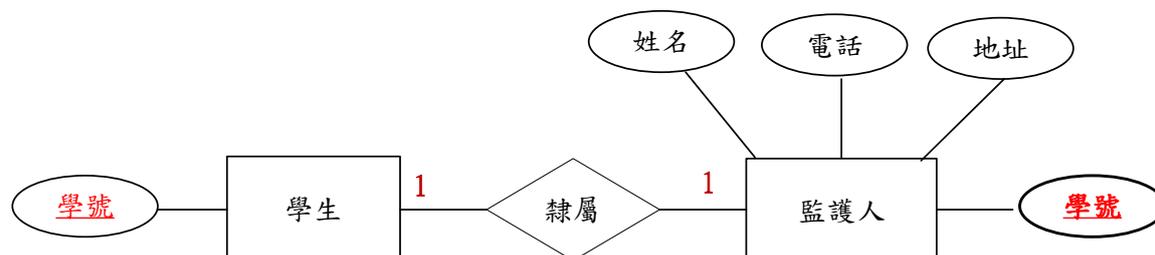


圖 9-8 一對一關係

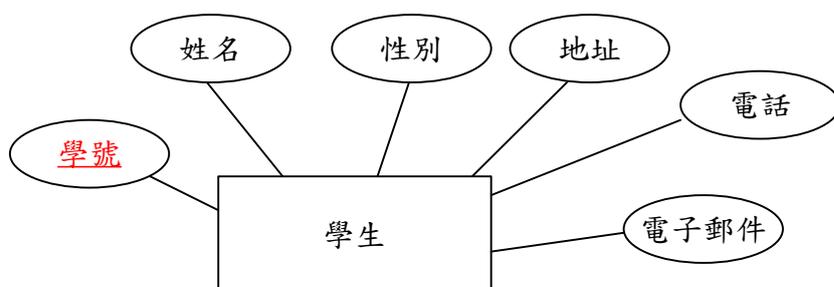
## 9-3 E-R 圖轉換為資料表

利用 E-R Mode 圖形描述真實情境之後，接下來必須將 E-R 圖轉換為資料表，其實轉換資料表並沒有一定常規可循，只要能描述每一個實體，並且每一表格都與其它表格建立關係

即可。

### 9-3-1 個體轉換

我們用學生實體來說明如何將實體轉換為資料表，如圖 9-9。將此實體取名為『學生資料表』= {學號、姓名、班級代碼、性別、地址、電話、電子郵件}，其中有底線者表示為主鍵。

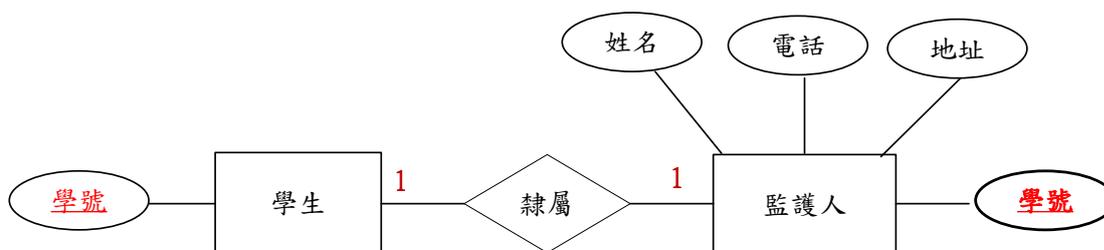


學生資料表 - {學號、姓名、性別、地址、電話、電子郵件}

圖 9-9 學生實體

### 9-3-2 一對一關係轉換

兩個實體之間是一對一的關係，大多兩實體都使用相同屬性為主鍵，譬如學生與監護人之間是一對一的關係，兩者都採用『學號』當主鍵，其關係就不需要另外建立資料表。



監護人資料表 - {學號、姓名、電話、地址}

圖 9-10 一對一關係轉換

### 9-3-3 一對多關係轉換

如果關係沒有增加屬性的話，它大多不需另外建立資料表，譬如學生與班級之間的隸屬關係，只要分別建立學生資料表與班級資料表即可，如 9-11。

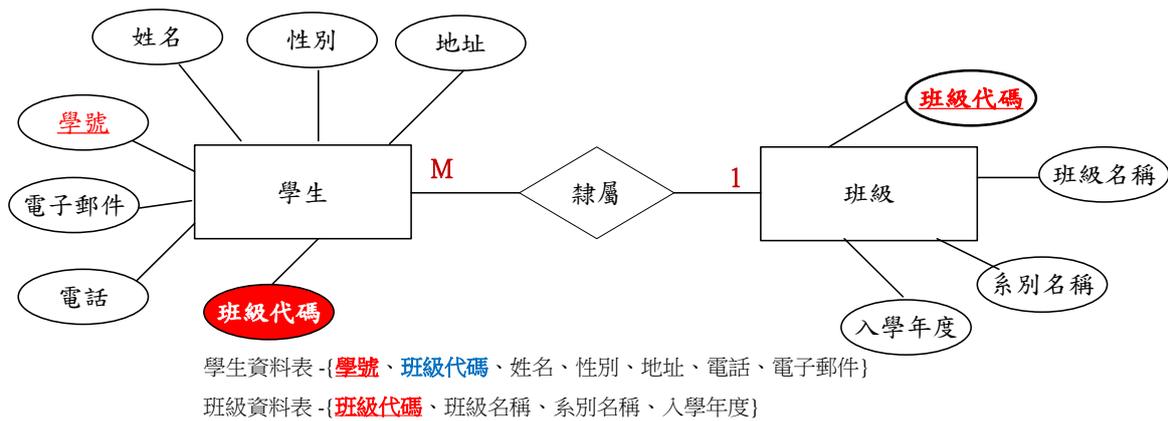


圖 9-11 一對多關係轉換

### 9-3-4 多對多關係轉換

系統設定每位學生可以選讀多門課，每一門課可讓學生修讀，但同一位學生、同一門課僅能選讀一次，不可點選兩次以上。因此，我們定選課的資料表為『選課總表』，其屬性為 {學號、開課代碼、分數}，其中學號與開課代碼為複合主鍵，如圖 9-12。

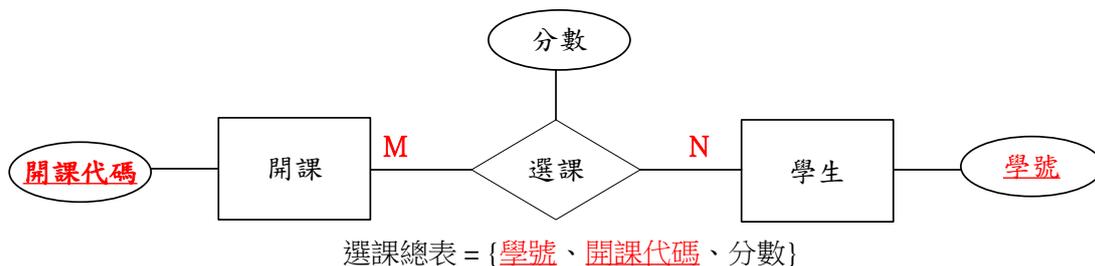


圖 9-12 多對多轉換

### 9-3-5 複雜的多對多關係

開課關係是複雜且多對多的關係，表示同一學期教師可以開多門課、每班也可開多門課、每一課程也可以讓不同老師在不同班級開課，如圖 9-13。我們將此關係定名為『開課總表』，他的屬性可以 {課程代碼、班級代碼、教師代碼} 等三個屬性，並組成主鍵。如果這樣的話，當它當別資料表的『外來鍵』時，該資料表也要建立這三個欄位，就會很浪費空間，因此，我們再增加一個『開課代碼』之欄位，並選定它為主鍵。

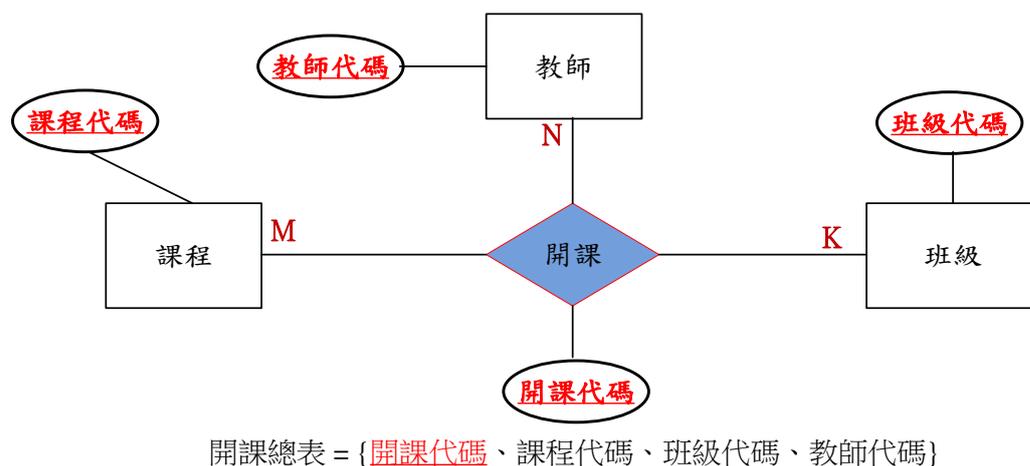


圖 9-13 複雜多對多轉換

## 9-4 資料表的正規化

前一節介紹了如何將實體或關係轉換成資料表，但轉換後的資料表是否恰當，必須要有一套方法來評估它，這就是『正規化』( Normalization )。我們可以由兩個觀點來說明為何要正規化：

### 9-4-1 正規化的目的

資料表為何需要正規化，有下列兩個原因：

#### (A) 降低資料重複性

同一筆資料盡可能僅存放一個位置，雖然可以解決資料不一致性的問題，但有會造成許多額外的負擔。譬如，需要學生資料的地方有教務處、學務處、以及系所辦公室，如果三個地方都儲存，當我們需要修改、增加或刪除時，三個地方都要去執行，否則就會發生資料不一致性問題。但如果僅存放一個位置，其它地方必須能搜尋得到，則必須建立『關係』來聯繫，如果會增加了許多表格，查詢時也許要多利用合併或其它多重表格的方法，如此也會造成系統執行速度降慢。如何降低資料重複性問題，也是滿困難的事，因此，我們需利用正規化來評估它。

#### (B) 避免資料更新的異常

設計資料表的簡單構想，就是儘量將特定資料（如學生資料）儲存在某一個地方，以避

免重複的問題，可能會將一個資料表分割成若干個資料表，之間再建立聯繫以維持原來的特性。但當某一個資料被更新時，其關係是否還是存在，否則就會造成資料更新的異常。

## 9-4-2 正規化型式的階層架構

正規化之目的是要檢測資料表是否符合減少重複性與避免資料更新的異常，但這兩者常常相違背，如果資料要越安全，資料表可能需要分割越精細，這樣一來，會增加查詢資料的困難。如果我們用另外的角度來看，資料表設計也許有漏洞，但可以永遠不會去觸動它，此種漏洞也許可以容忍它存在，如此設計資料表就可以簡單許多。也就是說，正規化並不需要要求非常嚴謹，只要能符合環境需求就可以了。在下圖中顯示正規化的階層架構有 1NF、2NF、3NF、BCNF、4NF、以及 PJ/NF 等六個層次，越高層次所要求條件越高，資料表可能被分割得越細，一般系統應用只到 3NF 或 BCNF 就不錯了，我們也討論到這個層次就好。

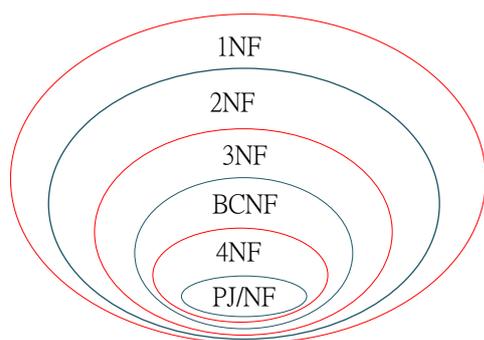


圖 9-14 正規化的階層

## 9-4-3 正規化的關鍵 – 功能相依性

在一個資料表裡有若干個欄位，或是說一個資料表示由若干個欄位所構成。如果學術一點用詞是『一個關聯是由數個屬性所描述』。接下來，就是如何選擇這些欄位，或欄位之間限制，這牽涉到欄位之間的功能相依性。可區分為：

- (1) **直接相依**：如果 A 欄位因 B 欄位而產生，則稱  $A \rightarrow B$  直接相依。
- (2) **傳遞相依**：如果 A 因 B 而產生，B 因 C 而產生，則  $A \rightarrow C$  為傳遞相依。

功能相依性關係到資料更新的異常，假設學生資料表為 {姓名、學號、地址、監護人、關係}，其內容可能如下：

學 號	姓 名	地 址	監 護 人	監 護 人 電 話
40011223	張有明	高雄市	張先生	07-3221456
40011224	劉志明	彰化市	劉媽媽	04-7891234
40011225	陳曉東	台南市	陳大姊	06-3445671

我們將此資料表內各欄位之間相依性關係劃出如圖 9-15 所示。姓名、地址與監護人欄位都直接相依於學號(主鍵)，監護人電話相依於監護人、監護人再相依於學號。監護人電話與學號之間就是傳遞相依。

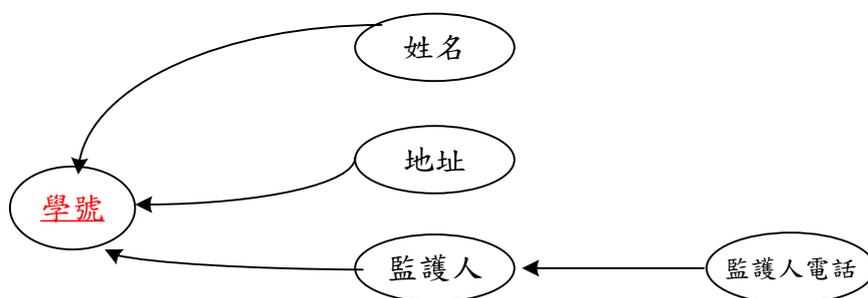


圖 9-15 傳遞功能性相依

## 9-5 正規化步驟

雖然我們並不需要達到第六階層的正規化，但最起碼還是需要到 3NF 的正規化，因此，我需要由 1NF、2NF、到 3NF 之間按部就班達成。

### 9-5-1 1NF-第一正規化

第一正規化定義如下：『在一個資料表內，所屬欄位的內容都是單元值(Atomic Value)，則稱該資料表滿足 1NF。』簡單的說，每個欄位的內容都是最小單元，不可以再分割的。如需要再分割，則需另外再設計資料表的欄位。譬如：學生資料表內某一筆資料的地址是『高雄市鳥松區澄清路 840 號』，它就是一個單元值，不可以再區分其它欄位，譬如縣市、區域等等。

學 號	姓 名	地 址	性 別	電 話
40011223	張有明	高雄市鳥松區澄清路 840 號	男	07112234
40011224	劉志明	彰化市		

真的需要將地址再區分縣市、區域的話，則資料表必須重新設計，將地址劃分為縣市、區域、道路，如下：(但它每個欄位也是單元值)

學 號	姓 名	縣 市	區 域	道 路	性 別	電 話
40011223	張有明	高雄市	烏松區	澄清路 840 號	男	07112234
40011224	劉志明	彰化市				

如果，我們如下面設計，姓名與地址同一個欄位存放，那它就不能滿足第一正規化。

學 號	監 護 人	關 係	電 話
40011223	張有明(高雄市烏松區澄清路 840 號)	男	07112234
40011224	劉志明(彰化市 ....)		

## 9-5-2 2NF-第二正規化

第二正規化定義如下：『一個資料表，除了符合第一正規化特性外，所有主鍵以外的欄位都必須與主鍵功能相依，則稱該資料表滿足 2NF。』簡單的說，所有欄位都必須與主鍵之間功能性相依。假設學生選課資料表，記錄著學生選讀那些課程，每一位學生可修讀多門課，主鍵為 {學號、課程名稱}，設計如下：

<u>學 號</u>	姓 名	地 址	<u>課程名稱</u>	學分數	成 績	必 選 修
40011223	張有明	高雄市###	資料庫管理系統	4	80	必
40011224	劉志明	彰化市###	資料庫管理系統	3	90	必
40011234	陳美麗	台南市###	計算機概論	2	85	選
49921234	林秀氣	屏東縣###	null	null	null	null

從上表可以看出它包含學生個人資料與成績，由資料表看得出來，我們必須選擇{學號、課程名稱}當複合主鍵。是否主鍵以外的欄位都與主鍵功能相依，探討如下：

- (1) 姓名與地址僅與學號相依，並不與課程名稱相依。
- (2) 學分數、成績、必選修等欄位與課程名稱相依，並不與學號相依。

如此看起來它不符合第二正規化，可能出現問題如下：

- (1) 新增可能發生錯誤：如果僅增加一位學生『林秀氣』，當還未修讀任何課程，結果『課

程名稱』欄位空白，違背主鍵不可 Null 之規定。

- (2) 更新可能發生錯誤：系上規定『資料庫管理系統』的學分數改為 4，如果僅更新『學號=40011223』的資料，其它同學的學分數就沒有變更到。
- (3) 刪除可能發生錯誤：如果刪除了學號=40011234 這筆資料，以後可能找不到『計算機概論』的資料了。

解決的方法很簡單，只要將表格拆成：學生資料表與選課總表等兩個表格即可，當然不僅這樣，往後還會慢慢談到相關技巧。

### 9-5-3 3NF-第三正規化

第三正規化定義如下：『一個資料表，除了符合第二正規化特性外，所有主鍵以外的欄位都僅能與主鍵功能相依，且沒有與其它欄位相依，則稱該資料表滿足 3NF。』簡單的說，所有欄位都必須、且僅能與主鍵之間功能性相依，其它欄位之間不可以有相依的特性存在。假設我們設計學生資料表如下圖，欄位包含有：{學號、姓名、地址、郵遞區號、性別、電話}，很容易可以看出必須選擇『學號』為主鍵，是否滿足 3NF 分析如下：

1. 姓名、地址、郵遞區號、性別與電話都與學號之間功能性相依，符合 2NF 特性。
2. 郵遞區號與地址之間功能相依，不符合 3NF 特性。

學 號	姓 名	地 址	郵遞區號	性 別	電 話
40011223	張有明	高雄市鳥松區澄清路 840 號	850	男	07112234
40011224	劉志明	彰化市			
40011234	陳美麗	台南市安南區中華路 2 號	644	女	06234567
49921234	林秀氣	高雄市鳥松區澄清路 450 號	833	女	07231134

由此可以看出此學生資料表並不能符合 3NF 特性，期可能出現問題如下：

1. 更新可能出現錯誤：如果高雄市鳥松區的郵遞區號改為 850，如僅修改 40011223 這筆資料，則出現與 49921234 資料內郵遞區號不一致的現象。
2. 刪除可能出現錯誤：如刪除掉 40011234 這筆資料，可能無法再找到台南市安南區

的郵遞區號資料了。

解決的方法是：

1. 將郵遞區號另外再開啟一個資料表來存放，雖然可以滿足 3NF 規範，但也增加了一個資料表，是否需要這樣呢？這要看您系統的使用環境，並非一昧的追求最完善就是最好。
2. 或者，如果郵遞區號不需要另外查詢，則將它加入地址欄位，也可以滿足 3NF。

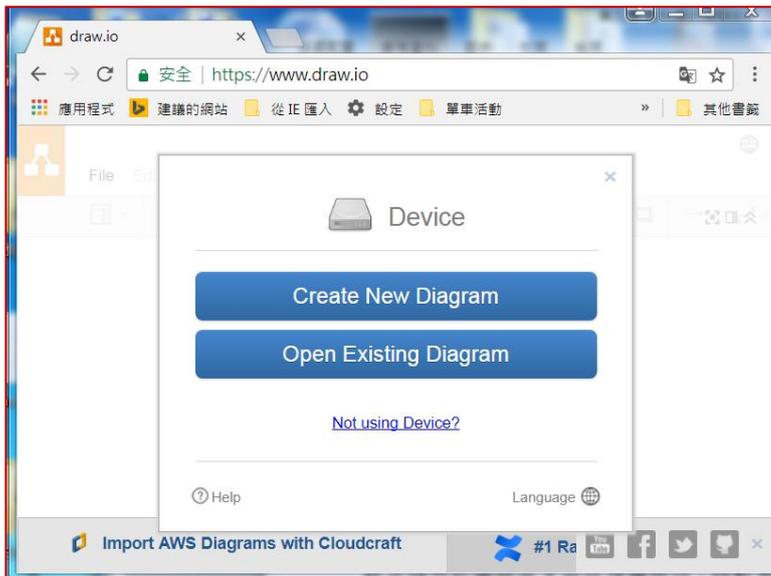
由上述正規化的操作可以發現，為了滿足更嚴謹的正規化，就是需要再細分資料表，如此可能再延伸查詢、更新與其它操作的困難。本人建議初學者正規化到這個階段就可以了，目前市面上 70% 以上資料庫系統，也幾乎只到規劃到這樣，如果熟悉資料庫運作之後，有興趣或需要再學習更進一步的正規化階段。

## 9-6 E-R Model 繪製工具 – draw.io

我們需要一套方便工具，才能將 E-R Model 關聯圖繪製美觀。有 Microsoft Visio 當然最方便但需要購買，許多同學妹有此軟體。在網路上有很多免費繪圖軟體，經過本人試用結果，感覺上 draw.io 套件也非常方便。它是線上工具不需要安裝，並且以 XML 格式儲存於雲端或本地磁碟機皆可。缺點是下載圖形並須整張圖下載，無法選擇區域下載。

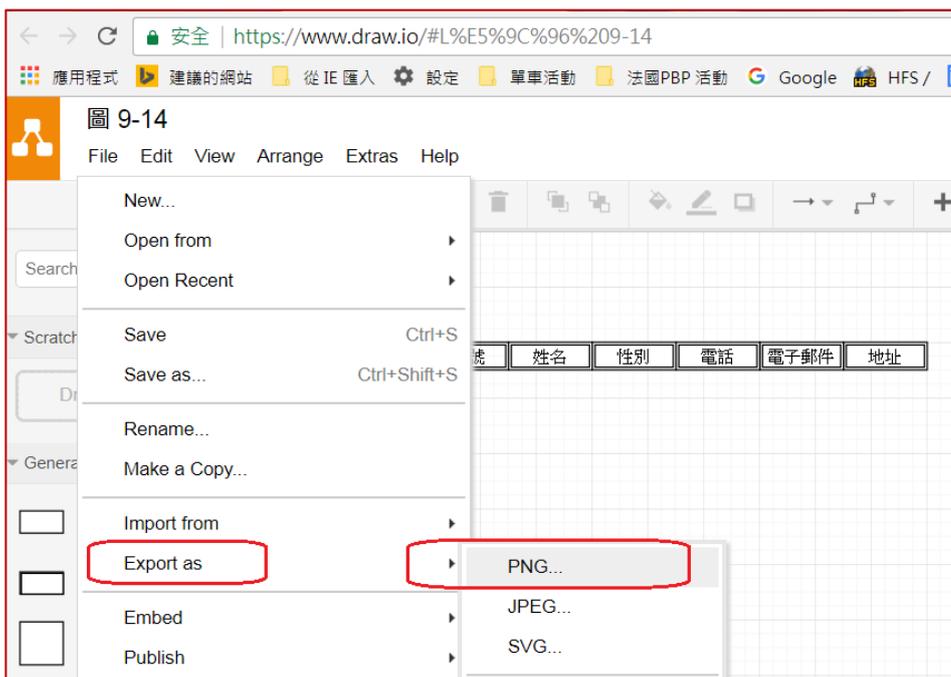
### (A) 連結上網 – [www.draw.io](http://www.draw.io)

利用瀏覽器連結如下：



### (B) 匯出圖形

開啟舊檔或新建立檔案，完成繪圖後，匯出圖形如下：



由電腦的『下載』可觀察到該檔案，如下：



只要將 PNG 檔插入 Word 檔案內即可，XML 檔可以重複編輯，如果發現設計不良時，可再開啟重新修正。

## 9-7 邏輯設計演練

### 9-7-1 範例研討 – 商品管理系統

#### (A) 情境與資料彙集

『廣招進出口貿易公司』對於辦理進出口商品種類非常繁複，期望建立一套系統來管理，目前對商品資料如下：

1. 商品名稱：
2. 單價：美金
3. 單位：個、批、套、
4. 產地：中國、台灣、馬來西亞、越南、、、、。
5. 分類：電器、廚具、寢物、工具、、、、。
6. 製造商
7. 製造商地址
8. 製造商電話
9. 製造商負責人

#### (B) 資料分類

我們將此資料庫定名為：Trader\_db。由上面資料可以看出，包含產品、製造商等兩個實體，我們先將兩實體分類如下：(請利用 draw.io 繪製)



圖 9-16 Trade\_db 資料庫資料分類

### (C) 實體間建立關聯 – E-R 關係圖

商品實體內缺少唯一識別值，我們增加了一個『商品編碼』屬性，每一個產品都有一個獨立編碼。同樣的，也許會有相同名稱的製造商，因此，在製造商實體中，也增加了一個『製造商編碼』的屬性。如此兩實體之間並沒有關聯性，吾人在商品實體內再增加一個『製造商編碼』，並將其設定為『外來鍵』參考到製造商實體的『製造商編碼』的屬性上，如下圖所示：

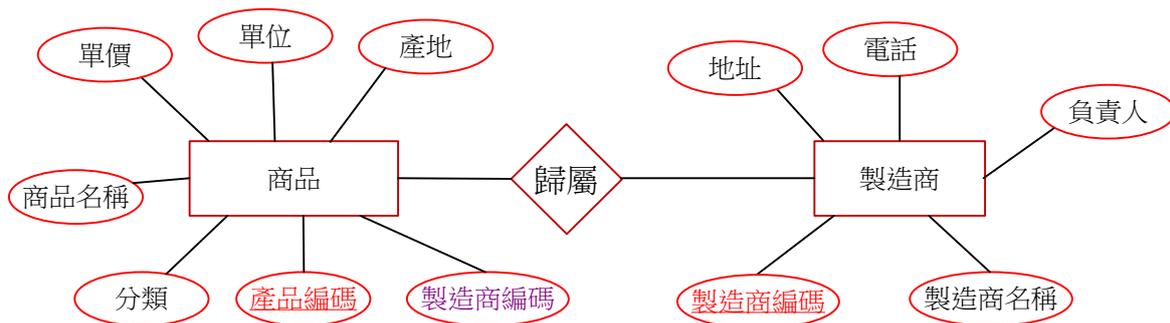
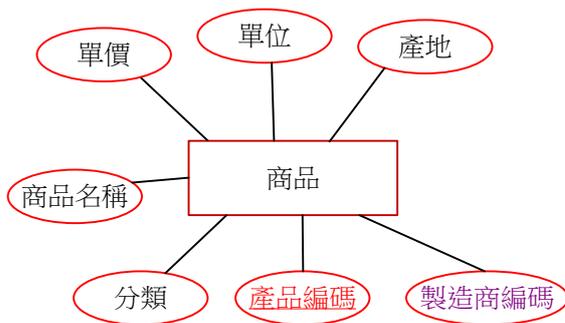


圖 9-17 貿易商 E-R 實體關聯圖

### (C) 商品實體轉換資料表

依照圖 9-17 所設計的 E-R 實體關聯圖，商品實體定名為 Product 表。



轉換成 Product 資料表如下：(主鍵：商品編碼(Product\_ID)、外來鍵：Manu\_ID)

<u>商品編號</u>	製造商編碼	單價	單位	產地	分類
<u>Product_ID</u>	Manu_ID	price	unit	origin	classify
CHAR(20)	CHAR(20)	int	CHAR(20)	CHAR(20)	CHAR(20)

#### 正規化分析：

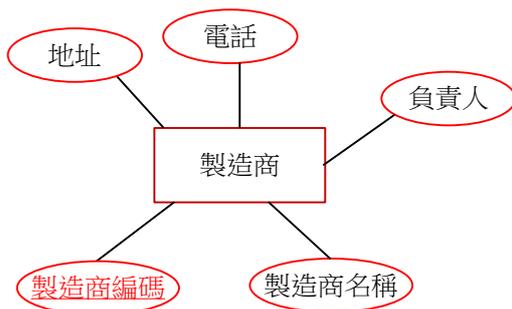
- 1NF：所有欄位都是最小單元(Atomic value)：符合。

2. 2NF：除了主鍵外，所有欄位都與主鍵相依：符合。
3. 3NF：除了主鍵外，所有欄位之間都沒有功能性相依：**不符合**。

原因：產地欄位與製造商編碼相依。產地與製造商地址有功能性相依。

#### (D) 製造商實體轉換資料表

依照圖 9-17 所設計的 E-R Model 圖，商品實體定名為 Manufacturer 表。



轉換成 Manufacturer 資料表如下：主鍵是：商品編碼(Product\_ID)

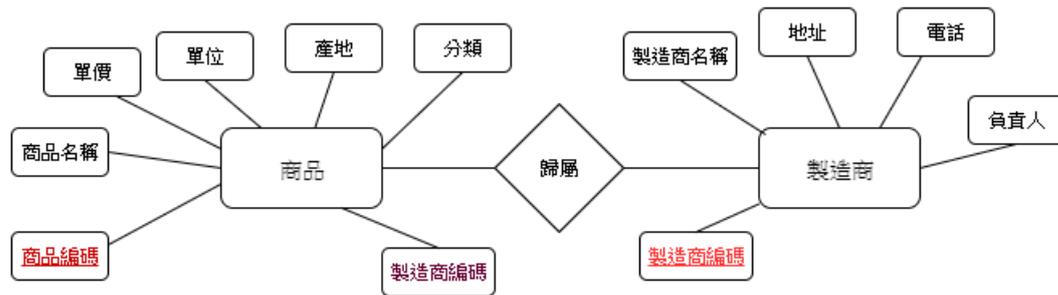
<u>製造商編號</u>	製造商名稱	負責人	電話	地址
<u>Manu_ID</u>	Manu_name	contact	tel	address
CHAR(20)	CHAR(20)	CHAR(20)	CHAR(20)	CHAR(50)

#### 正規化分析：

1. 1NF：所有欄位都是最小單元(Atomic value)：符合。
2. 2NF：除了主鍵外，所有欄位都與主鍵相依：符合。
3. 3NF：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

### 9-7-2 自我挑戰 – 改善 Trader\_db 系統

『廣招進出口公司』 Trader\_db 資料庫的 E-R 實體關聯圖如下：



經過分析後，發現商品資料表(Product)並不滿足第三正規化，請您重新設計 E-R 實體關聯圖，使其所有資料表都能滿足第三正規化，請依下列順序完成：

- (A) 資料分類
- (B) 建立 E-R 關聯圖
- (C) 轉換資料表與正規化分析

### 9-7-3 自我挑戰 – 人事管理系統

#### (A) 情境與資料彙集

『鴻達精密公司』希望建立一套人事資料管理系統，張系統分析師到該公司去收集相關資料，如下：

1. 員工姓名：
2. 員工性別：
3. 員工地址：
4. 員工聯絡電話：
5. 服務單位名稱：
6. 服務單位負責人：
7. 薪資等級：
8. 薪資金額：
9. 加級等級：
10. 加級金額：

請依照下列次序完成：

- (1) 資料分類。
- (2) 建立 E-R 關聯圖。
- (3) 轉換成資料表，並正規化分析。

**(B) 提示：**

E-R 圖方案之一：四個資料表達成。