

資料庫 理論與實務

Access 2007



第 12 章

資料表正規化



本章提要

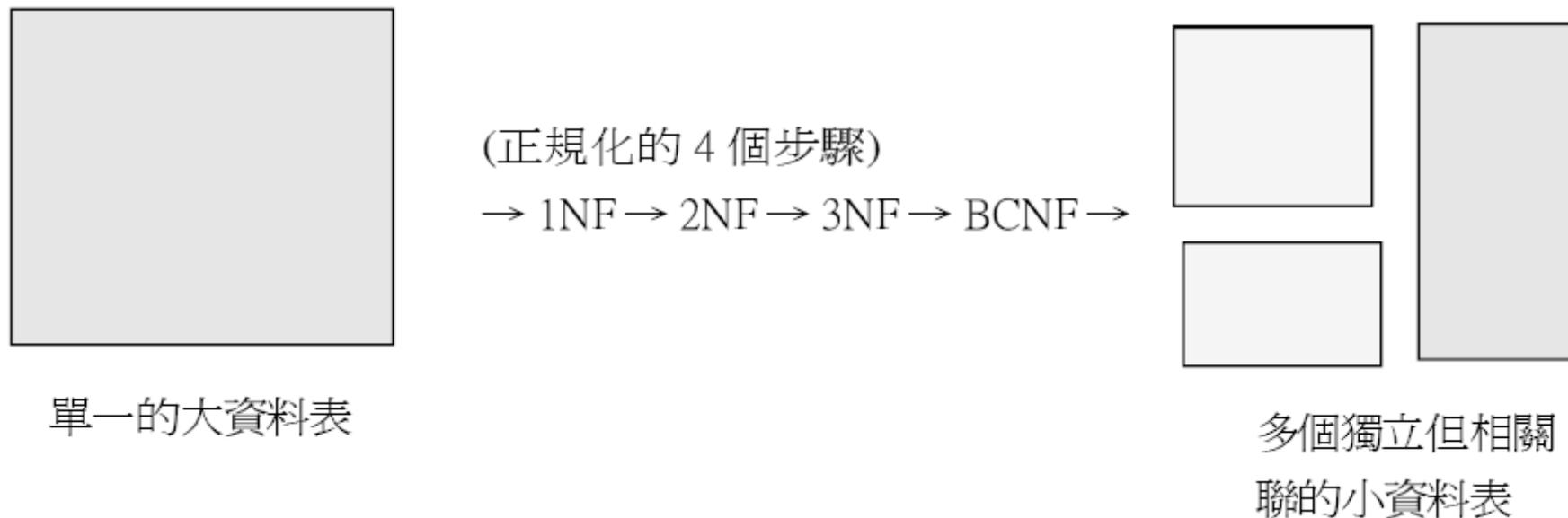
- 12-1 實體 - 關係圖與正規化的關係
- 12-2 欄位相依
- 12-3 第一階正規化 (1NF)
- 12-4 第二階正規化 (2NF)
- 12-5 第三階正規化 (3NF)
- 12-6 Boyce-Codd 正規化 (BCNF)
- 12-7 正規化的另類思考



12-1 實體 - 關係圖與正規化的關係

- 資料表的『**正規化理論**』(Normalization Theory) 其實是一套『**資料表分割**』的法則。
- 在資料庫設計之初, 我們可將所有的資料欄位合併成一個大資料表, 然後依循正規化理論所提出的各個步驟, 逐步分割, 進而形成許多獨立, 但彼此關聯的小資料表。
- 正規化的目的, 則是要**避免資料重複或相互矛盾**的情形, 並使資料庫在使用時能**更有效率、更容易維護**。

實體 - 關係圖與正規化的關係



- 其實利用**實體 - 關係圖**所建立的資料表, 也可避免資料重複或相互矛盾的情形發生。

實體 - 關係圖與正規化的關係

- 不過，對於初學者來說，利用實體 - 關係圖發展資料庫的過程中，思緒未必周延，可能仍會出現資料重複或相互矛盾的情形，此時便可再利用正規化來檢驗這些資料表。
- 資料庫的正規化共可分為 **1NF (Normal Form)**、**2NF**、**3NF**、**BCNF (Boyce-Codd Normal Form)**、**4NF**、**5NF** 等多個階段，不過對於一般資料庫設計來說，通常只要執行到 **BCNF** 即可，其他更高階的正規化只有在特殊的情況下才用得到。
- 因此本章也只介紹到 **BCNF** 的正規化。

實體 - 關係圖與正規化的關係

- 資料表正規化的過程 (1NF 到 BCNF) 中, 每個階段都是以欄位的相依性, 做為分割資料表的依據之一。
- 所以下一節中, 將先為各位介紹何謂欄位相依, 待觀念建立後, 再來介紹正規化。

12-2 欄位相依

- 在一個資料表中，若乙欄位的值必須搭配甲欄位才有意義，則我們說『乙欄位相依於甲欄位』。
- 例如：

* 課程編號	* 學 號	成 績
M102	963123	89
A205	963145	75

欄位相依

欄位相依

- **成績**欄本身如果單獨存在是沒有意義的，因為不曉得是哪一門課、哪個學生的成績。
- 不過，當**成績**欄相依於**課程編號**及**學號**二欄時，就可以了解某個學生修某堂課的成績，這樣的成績資料才有意義。
- 我們再看看下面的例子，一個**員工**資料表如下：

* 員工編號	姓名	地址
101	陳鐵雄	...
102	王大明	...



欄位相依

- 員工編號欄為主鍵，做為唯一辨識該筆記錄的欄位。
- 對此資料表來說，姓名欄必須要相依於員工編號欄，姓名欄才有意義，否則同名同姓的陳鐵雄都可對應到該筆記錄。
- 同理，地址欄亦必須相依於員工編號欄，才有意義。



12-3 第一階正規化 (1NF)

- 正規化過程是循序漸進的，資料表必須在滿足第一階正規化的條件之下，才能進行第二階正規化。
- 也就是說，第二階正規化必須建立在符合第一階正規化的資料表上，依此類推。而第一階正規化正是所有正規化的基礎。

第一階正規化的規則

- 如果資料表符合以下條件，我們就說這個資料表符合『第一階正規化的形式 (First Normal Form, 簡稱 1NF)』：
 - 1. 資料表中有**主鍵**，而其他所有的欄位都相依於主鍵。
 - 2. 每個欄位中都只儲存**單一值**，例如同一筆記錄的**姓名**欄位中不能存放 2 個人的姓名。
 - 3. 資料表中沒有**意義相同**的多個欄位，例如**姓名 1**、**姓名 2**...等重複的欄位。

第一階正規化的規則

- 反之, 若資料表的欄位不符合以上條件, 則稱為『非正規化』的資料表。

未符合 1NF 資料表的缺點

- 首先我們來看一個非正規化的資料表：

選課資料表

課程編號	課程名稱	教師編號	教師姓名	學號	學生姓名	成績
CS101	資料庫系統	1001	曾聰明	96010	陳鐵雄	70
				96013	王大明	90
				96020	李珍珍	88
CS103	影像處理	1015	郝天才	96033	趙阿丁	86
				96045	鄭阿龍	93

- 這個資料庫乍看之下似乎很精簡，而且一目瞭然，但卻有以下兩個缺點：

未符合 1NF 資料表的缺點

- 1. 『學號』、 『學生姓名』 及 『成績』 欄的長度無法確定：
 - 由於修課學生的人數可多可少, 所以必須預留很大的空間給這兩個欄位, 如此反而造成儲存空間的浪費。
- 2. 降低存取資料的效率：
 - 例如要找出 『陳鐵雄』 的成績, 必須先在學生姓名欄中找出 『陳鐵雄』 所在的位置, 然後才能從成績欄中擷取出對應的成績資料,
 - 這不僅減緩了資料處理的速度, 而且也增加了程式出錯的機會。



未符合 1NF 資料表的缺點

- 很明顯的, 此資料表違反了第一階正規化的第 2 個條件。
- 另外, 可能有兩個老師同時都開了『資料庫系統』的課程, 也可能有兩個學生都叫做『王大明』, 因此這個資料表缺少具有唯一性的主鍵, 也違反了第一階正規化的第 1 個條件。
- 接下來看一個違反第一階正規化第 3 個條件的例子：

未符合 1NF 資料表的缺點

修課資料表

課程名稱	學生 1	學生 2	學生 3	成績 1	成績 2	成績 3
資料庫系統	陳鐵雄	王大明	李珍珍	70	90	88
影像處理	趙阿丁	鄭阿龍		86	93	

- 像學生 1、學生 2、學生 3 這樣一群意義相同的欄位，其問題同樣是無法確定要有多少個重複的欄位，而且存取效率低落，
- 例如要找『陳鐵雄』的成績，必須在學生群組的每一個欄位中搜尋，找到後還得要找到成績群組中的相同位置欄位中讀取，相當麻煩。

建構 1NF 資料表的方法

- 對於不具第一階正規化形式的資料表，我們可將重複的資料項分別儲存到不同的記錄中，並加上適當的主鍵 (加 "*" 符號者為主鍵)：

選課資料表

* 課程編號	* 學號	課程名稱	教師編號	教師姓名	學生姓名	成績
CS101	96010	資料庫系統	1001	曾聰明	陳鐵雄	70
CS101	96013	資料庫系統	1001	曾聰明	王大明	90
CS101	96020	資料庫系統	1001	曾聰明	李珍珍	88
CS103	96033	影像處理	1015	郝天才	趙阿丁	86
CS103	96045	影像處理	1015	郝天才	龔阿龍	93



建構 1NF 資料表的方法

- 如此一來，雖然增加了許多記錄，但每一個欄位的長度及數目都可以固定，而且我們可用課程編號欄加上學號欄做為主鍵，那麼在查詢某學生修某堂課的成績時，就非常方便而快速了。

12-4 第二階正規化 (2NF)

- 在執行符合一階正規化的資料表時，應該會發覺：我們輸入了許多重複的資料。
- 如此，不但浪費儲存的空間，更容易造成新增、刪除或更新資料時的異常狀況。所以，我們必須進行第二階正規化，來消除這些問題。

第二階正規化的規則

- 如果資料表符合以下的條件，我們說這個資料表符合『第二階正規化的形式 (Second Normal Form, 簡稱 2NF)』：
 - 1. 符合 1NF 的格式。
 - 2. 各欄位與主鍵間沒有『部分相依』的關係。
- 『部分相依』只有在主鍵是由多個欄位組成時才會發生，它是指某些欄位只與主鍵中的部分欄位有相依性，而與另一部分的欄位沒有相依性。

第二階正規化的規則

- 以選課資料表來說，其主鍵為課程編號 + 學號欄位，但課程名稱欄只和課程編號欄有相依性，而學生姓名欄只和學號欄有相依性：

第二階正規化的規則



只和學號有相依性

選課資料表

* 課程編號	* 學號	課程名稱	教師編號	教師姓名	學生姓名
CS101	96010	資料庫系統	1001	曾聰明	陳鐵雄
CS101	96013	資料庫系統	1001	曾聰明	王大明
CS101	96020	資料庫系統	1001	曾聰明	李珍珍
CS103	96033	影像處理	1015	郝天才	趙阿丁
CS103	96045	影像處理	1015	郝天才	鄭阿龍

只和課程編號有相依性

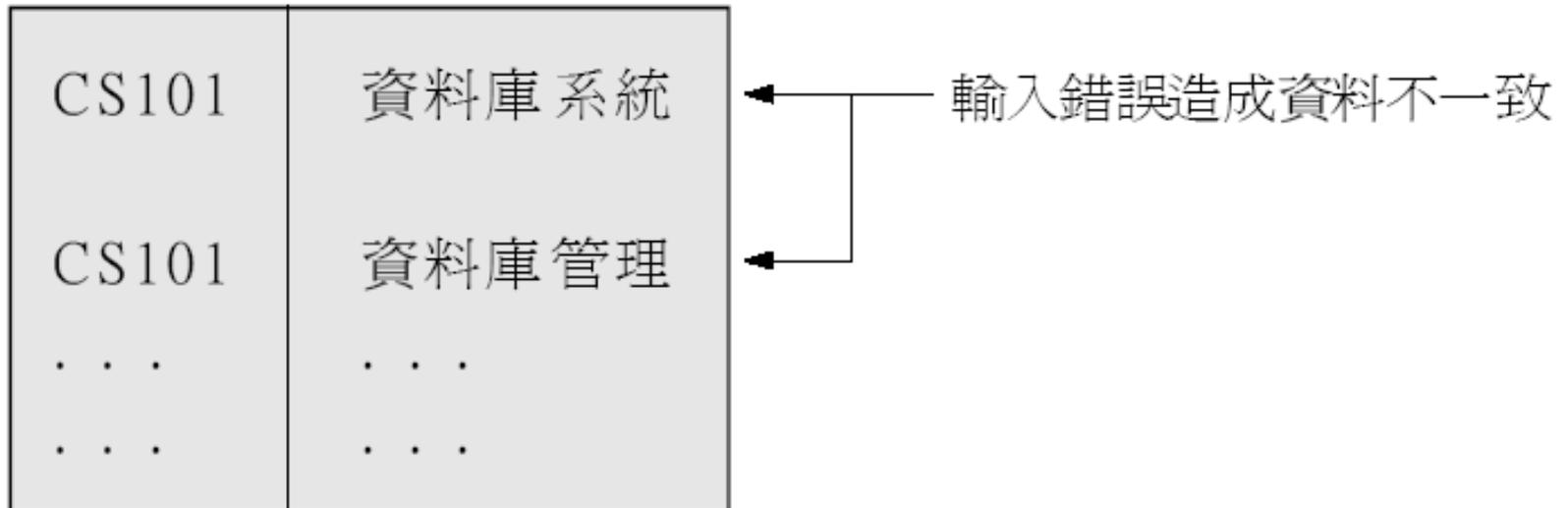


第二階正規化的規則

- 部分相依會造成下列問題：
 - 新增資料時：若有一個新來的轉學生『吳技安』，但還沒有選修任何課程，那麼它的資料將無法輸入 (因為主鍵中的欄位值是不允許有空白的，但此時根本沒有課程編號可輸入)。
 - 更改資料時：當我們想要將課程名稱『資料庫系統』更改為『資料庫管理』時，必須搜尋整個資料庫並一一更改，非常沒有效率。
 - 刪除資料時：由於陳鐵雄只修了『CS101』一門課，如果將該筆記錄刪除，那麼陳鐵雄的資料也就跟著消失了。

第二階正規化的規則

- 另外，部分相依也會造成資料重複出現的問題，例如『CS101，資料庫系統』這組資料每次都必須同時輸入，不但浪費時間及儲存空間，而且也容易因疏忽而造成資料不一致的錯誤。
- 例如：



建構 2NF 資料表的方法

- 要除去資料表中的部分相依性, 只需將部份相依的欄位分割成另外的資料表即可。
- 例如我們將選課資料表分割成 3 個較小的資料表 (加 "*" 號的欄位為主鍵) :

成績資料表

* 課程編號	* 學號	成績
CS101	96010	70
CS101	96013	90
CS101	96020	88
CS121	96020	86
CS121	96033	93

建構 2NF 資料表的方法

課程資料表

* 課程編號	課程名稱	教師編號	教師姓名
CS101	資料庫系統	1001	曾聰明
CS121	影像處理	1015	郝天才

學生資料表

* 學號	學生姓名
96010	陳鐵雄
96013	王大明
96020	李珍珍
96033	趙阿丁

12-5 第三階正規化 (3NF)

- 經過了第二階正規化後的資料表，其實還存在一些問題：
 - 在課程資料表中，如果新來了一位教師『甄莠』，在尚未安排他教授的課程之前，我們無法輸入該教師的資料。
 - 若要刪除影像處理的課程，勢必會將該門課程的授課教師一併刪除。

第三階正規化 (3NF)

- 若要更改影像處理課程的名稱, 則必須同時更改多筆記錄 (同一門課程, 會有多位教師開課), 造成不便。
- 基於上述理由, 我們必須再執行第三階正規化。

第三階正規化的規則

- 如果資料表符合以下條件，我們就說這個資料表符合『第三階正規化的形式 (Third Normal Form, 簡稱 3NF)』：
 - 符合 2NF 的格式。
 - 各欄位與主鍵間沒有『間接相依』的關係。
- 『間接相依』是指在二個欄位間並非直接相依，而是借助第三個欄位來達成資料相依的關係，例如 A 相依於 B；而 B 又相依於 C，如此 A 與 C 之間就是間接相依的關係。

第三階正規化的規則

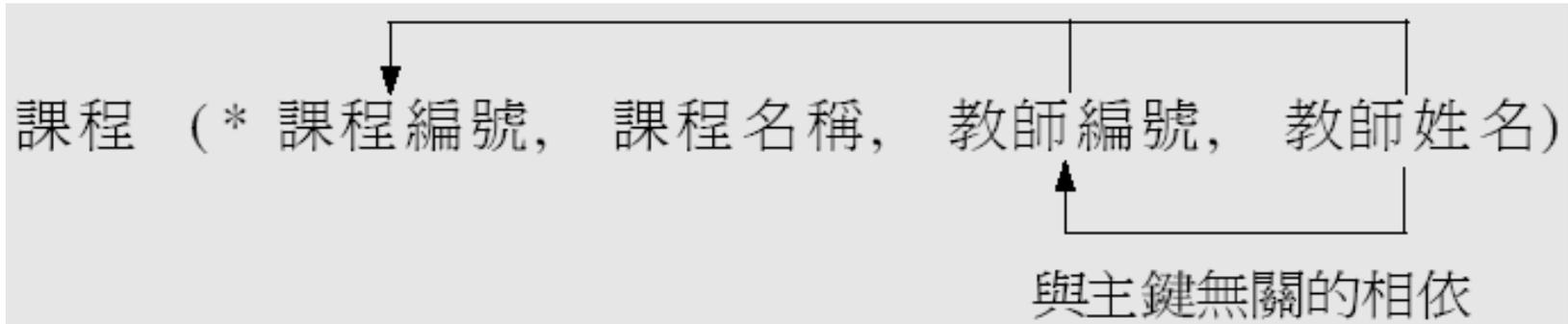
- 要找出各欄位與主鍵間の間接相依性，最簡單的方式就是看看資料表中有沒有『與主鍵無關的相依性』存在。
- 例如在課程資料表中：

課程資料表

* 課程編號	課程名稱	教師編號	教師姓名
CS101	資料庫系統	1001	曾聰明
CS121	影像處理	1015	郝天才

第三階正規化的規則

- 由於每一門課程都會有授課的教師，所以教師編號欄和教師姓名欄都相依於課程編號欄。
- 但教師姓名又同時相依於教師編號欄，而這個相依性是與主鍵完全無關的：



第三階正規化的規則

- 事實上，它們之間的相依關係為：教師姓名→教師編號→課程編號，因此有『間接相依』的關係存在，也就是有『與主鍵無關的相依性』。

建構 3NF 資料表的方法

- 要除去資料表中的『間接相依性』，其方法和除去『部分相依性』完全相同。例如課程資料表可再分割成兩個資料表：

課程資料表

*課程編號	課程名稱	教師編號
CS101	資料庫系統	1001
CS103	影像處理	1015

教師資料表

*教師編號	教師姓名
1001	曾聰明
1015	郝天才

與直覺式的分割技巧做比較

- 當您設計資料庫一段時間，累積了經驗及技術後，您便可依照己身的經驗，以直覺的方式對資料表執行最佳化，底下是兩種方法在功能上的對照：

正規化	功能相同的直覺式分割法
1NF：有主鍵 欄位中只有一個單一值 沒有意義相同的重複欄位	無
2NF：除去『部分相依性』	分割『欄位值一再重複』的欄位
3NF：除去『間接相依性』	分割『與主鍵無關』的欄位

12-6 Boyce-Codd 正規化 (BCNF)

- 對於大部分資料庫來說，通常只需要執行到第三階段的正規化即足夠了。
- 如果資料表的**主鍵是由多個欄位組成的**，則必須再執行 Boyce-Codd 正規化來檢驗。

Boyce-Codd 正規化的規則

- 如果資料表的主鍵只由單一欄位組成，則符合第三階正規化的資料表，亦符合 Boyce-Codd 正規化。
- 但若資料表的主鍵由多個欄位組成，則資料表只要符合以下條件，我們就說這個資料表符合『Boyce-Codd 正規化的形式』(Boyce-Codd Normal Form, 簡稱 BCNF)：
 - 符合 3NF 的格式。
 - 主鍵中的各欄位不可以相依於其他非主鍵的欄位。

檢驗『成績』資料表是否滿足 BCNF 規範



- 我們利用 Boyce-Codd 正規化的條件，來檢驗主鍵由多個欄位組成的成績資料表：

成績資料表

* 課程編號	* 學號	成績
CS101	96010	70
CS101	96013	90
CS101	96020	88
CS121	96020	86
CS121	96033	93

所有欄位都相依於主鍵，
主鍵中的欄位亦無相依於其他非主鍵欄位



檢驗『成績』資料表是否滿足 BCNF 規範



- 成績欄相依於課程編號及學號欄，對課程編號欄而言，並無相依於成績欄；對學號欄而言，也無相依於成績欄。
- 所以成績資料表是符合『Boyce-Codd 正規化的形式』的資料表。
- 第 12-5 頁的選課資料表經過了 1NF 到 BCNF 正規化的過程後，產生了下列互相關聯的資料表。



檢驗『成績』資料表是否滿足 BCNF 規範



- 到此，正規化的工作就算大功告成了，現在我們就將它們列出來，提供參考：

課程資料表

* 課程編號	課程名稱	教師編號
CS101	資料庫系統	1001
CS103	影像處理	1015

教師資料表

* 教師編號	教師姓名
1001	曾聰明
1015	郝天才



檢驗『成績』資料表是否滿足 BCNF 規範



成績資料表

*課程編號	*學號	成績
CS101	96010	70
CS101	96013	90
CS101	96020	88
CS121	96020	86
CS121	96033	93

學生資料表

*學號	學生姓名
96010	陳鐵雄
96013	王大明
96020	李珍珍
96033	趙阿丁



12-7 正規化的另類思考

- 正規化固然是設計資料庫的好方法，但它只是一個基本的原則而已，在原則之外，我們還要考慮系統的需求自行作一些變化，
- 例如下面兩種情況：
 - 不必要的分割：
 - 正規化的工作有時不必做的非常徹底，因為過多的資料表可能會降低系統執行的效能。
 - 我們以學生資料表來看：

學生
*學號
姓名
年齡
電話
鄉鎮市(區)
街牌號碼
*郵遞區號

正規化的另類思考

- 按照正規化的原理, 郵遞區號其實是相依於『鄉鎮市(區)』, 因此這個資料表可再進行分割, 才能符合正規化的要求。
- 如下圖所示：

學生
*學號
姓名
年齡
電話
鄉鎮市(區)編號
街牌號碼

郵遞區號對照
*鄉鎮市(區)編號
鄉鎮市(區)
郵遞區號

正規化的另類思考

- 但實際作業上，我們一定是使用整筆的地址資料，如果將它們分割了，那麼每次查詢地址時，都要透過資料表的關聯，從郵遞區號對照資料表中取得郵遞區號欄位。
- 當資料量增加到一定程度時，就會明顯感受到系統效能降低的事實，實非明智之舉。

正規化的另類思考

- 人工的分割：
 - 有時為了增加資料處理的效率，我們可將已經符合 **BCNF** 的資料表再做分割。
 - 例如一個資料表擁有非常多的欄位，而其中又有許多欄位根本很少用到，那麼我們就可將這些很少用到的欄位分離出來，存放另外一個資料表中。