



第5章 關聯表的正規化

- 5-1 正規化的基礎
- 5-2 功能相依
- 5-3 第一階到第三階正規化型式
- 5-4 多重值相依與第四階正規化型式
- 5-5 合併相依與第五階正規化型式





5-1 正規化的基礎

- 5-1-1 關聯表為什麼需要正規化
- 5-1-2 正規化的型式





5-1 正規化的基礎

- 「正規化」（**Normalization**）是一種標準處理程序來決定關聯表應該擁有哪些屬性，其目的是建立「良好結構關聯表」（**Well-structured Relation**），一種沒有重複資料的關聯表。而且在新增、刪除或更新資料時，不會造成錯誤或資料不一致的異常情況。正規化目的如下所示：
 - 去除重複性（**Eliminating Redundancy**）：建立沒有重複資料的關聯表，因為重複資料不只浪費資料庫的儲存空間，而且會產生資料維護上的問題。
 - 去除不一致的相依性（**Eliminating Inconsistent Dependency**）：資料相依是指關聯表中的屬性之間擁有關係，如果關聯表擁有不一致的資料相依，這些屬性就會在新增、刪除或更新資料時，造成異常情況。



5-1-1 關聯表為什麼需要正規化 – 非正規化型式關聯表

- 關聯式資料庫正規化的目的以一句話來說，就是避免資料重複，關聯表擁有非自然結合的屬性，就會造成資料重複的問題，稱為「非正規化型式」（Unnormalized Form）關聯表。

Student_Course_Classes

<u>sid</u>	<u>name</u>	<u>birthday</u>	<u>c_no</u>	<u>title</u>	<u>credits</u>	<u>time</u>	<u>room</u>
S001	陳會安	1967/9/3	CS101	計算機概論	4	12:00pm	180-M
S003	張三丰	1982/3/3	CS121	離散數學	4	8:00am	221-S
S001	陳會安	1967/9/3	CS203	程式語言	3	10:00am	221-S
S002	江小魚	1978/2/2	CS203	程式語言	3	14:00pm	327-S
S001	陳會安	1967/9/3	CS222	資料庫管理系統	3	13:00pm	100-M
S002	江小魚	1978/2/2	CS222	資料庫管理系統	3	13:00pm	100-M
S004	李四方	1981/4/4	CS222	資料庫管理系統	3	13:00pm	100-M
S003	張三丰	1982/3/3	CS213	物件導向程式設計	2	9:00am	622-G
S001	陳會安	1967/9/3	CS213	物件導向程式設計	2	12:00pm	500-K



5-1-1 關聯表為什麼需要正規化 – 更新異常

- 更新異常(Update Anomaly)是在Student_Course_Classes關聯表更新指定的屬性資料，因為資料重複儲存在多筆值組（記錄），所以需要同時更新多筆值組的資料。

Student_Course_Classes

<u>sid</u>	name	birthday	<u>c_no</u>	title	credits	time	room
S001	陳會安	1967/9/3	CS101	計算機概論	4	12:00pm	180-M
S001	陳會安	1967/9/3	CS203	程式語言	3	10:00am	221-S
S001	陳會安	1967/9/3	CS222	資料庫管理系統	3	13:00pm	100-M
S001	陳會安	1967/9/3	CS213	物件導向程式設計	3	13:00pm	100-M

需要更改4筆值組

Student_Course_Classes

<u>sid</u>	name	birthday	<u>c_no</u>	title	credits	time	room
S001	陳會安	1965/9/3	CS101	計算機概論	4	12:00pm	180-M
S001	陳會安	1965/9/3	CS203	程式語言	3	10:00am	221-S
S001	陳會安	1965/9/3	CS222	資料庫管理系統	3	13:00pm	100-M
S001	陳會安	1965/9/3	CS213	物件導向程式設計	2	12:00pm	500-K



5-1-1 關聯表為什麼需要正規化 – 刪除異常

- 刪除異常(Deletion Anomaly)是在Student_Course_Classes關聯表刪除值組，可能刪除合法資料。例如：CS101和CS121課程都有一筆值組，如下所示：

刪除值組連同課程資料也一併遺失

Student_Course_Classes

<u>sid</u>	name	birthday	<u>c_no</u>	title	credits	time	room
S001	陳會安	1967/9/3	CS101	計算機概論	4	12:00pm	180-M
S003	張三丰	1982/3/3	CS121	離散數學	4	8:00am	221-S
S001	陳會安	1967/9/3	CS203	程式語言	3	10:00am	221-S
S002	江小魚	1978/2/2	CS203	程式語言	3	14:00pm	327-S

.....

S003	張三丰	1982/3/3	CS213	物件導向程式設計	2	9:00am	622-G
S001	陳會安	1967/9/3	CS213	物件導向程式設計	2	12:00pm	500-K



5-1-1 關聯表為什麼需要正規化 – 新增異常

- 新增異常(Insertion Anomaly)是在Student_Course_Classes關聯表新增值組和刪除相反，可能只新增部分值組的資料。例如：新增一筆值組，課程編號是CS240，名稱為資料結構，學分是3，如下所示：

Student_Course_Classes

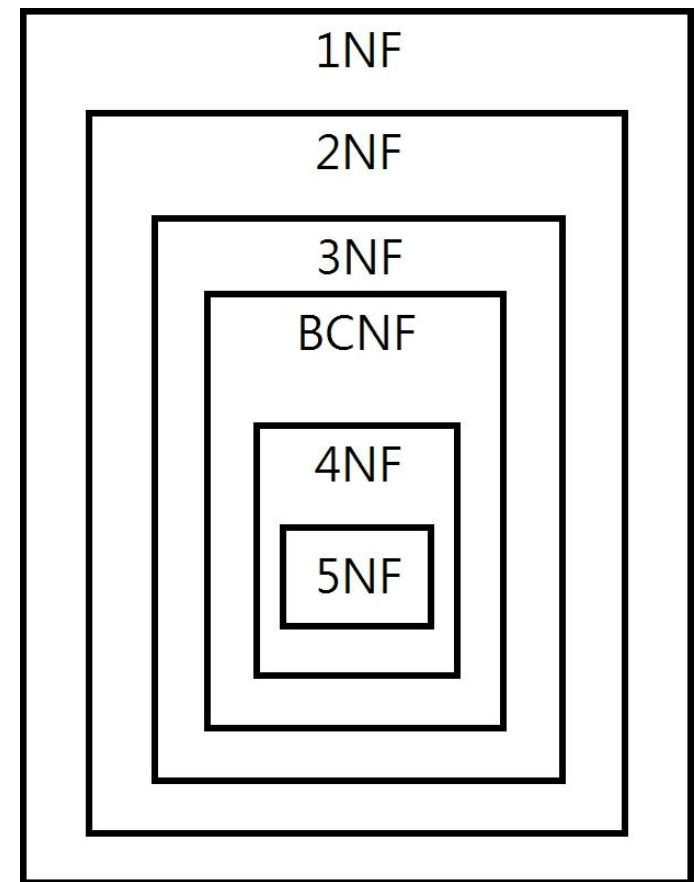
<u>sid</u>	name	birthday	<u>c_no</u>	title	credits	time	room
S001	陳會安	1967/9/3	CS101	計算機概論	4	12:00pm	180-M
S003	張三丰	1982/3/3	CS121				
S001	陳會安	1967/9/3	CS203				
S002	江小魚	1978/2/2	CS203				
.....							
S003	張三丰	1982/3/3	CS213	物件導向程	2	9:00am	622-G
S001	陳會安	1967/9/3	CS213	物件導向程 設計	2	12:00pm	500-K
--	--	--	CS240	資料結構	3	3:00pm	230-A

新增值組只有課程資料，主鍵sid是空值，違反實體完整性



5-1-2 正規化的型式 – 說明

- 正規化是從下而上（**Bottom-Up**）評估關聯表綱要是否符合正規化型式，針對的是關聯表中各屬性間的關係，正規化型式是一些組織關聯表屬性的規則。
- 正規化的處理過程分為五個階段的正規化型式，每一階正規化型式是在處理不同屬性間資料相依（**Dependency**）的問題，如右圖所示：





5-1-2 正規化的型式 – 五階正規化

- 第一階正規化型式（First Normal Form；1NF）：在關聯表刪除多重值和複合屬性，讓關聯表只擁有單元值屬性。
- 第二階正規化型式（Second Normal Form；2NF）：滿足1NF且關聯表沒有「部分相依」（Partial Dependency）。
- 第三階正規化型式（Third Normal Form；3NF）：滿足2NF，而且關聯表沒有「遞移相依」（Transitive Dependency）。
- Boyce-Codd正規化型式（Boyce-Codd Normal Form；BCNF）：廣意第三階正規化型式，關聯表如果擁有多個複合候選鍵，需要刪除候選鍵間的功能相依。
- 第四階正規化型式（Fourth Normal Form；4NF）：滿足BCNF且沒有多重值相依。
- 第五階正規化型式（Fifth Normal Form；5NF）：滿足4NF且沒有合併相依。



5-2 功能相依

- 5-2-1 功能相依的定義
- 5-2-2 功能相依的種類
- 5-2-3 找出關聯表的所有功能相依
- 5-2-4 阿姆斯特壯推論規則





5-2 功能相依

- 功能相依（**Functional Dependency**；**FD**）是描述關聯表中屬性間的相依關係，這是關聯表正規化的基礎。簡單的說，關聯表正規化的第一步，就是在關聯表找出所有屬性間的功能相依。
- 當在關聯表找出所有屬性間的功能相依後，就可以幫助我們在關聯表找出：
 - 重複資料：一些不該屬於此關聯表的屬性。
 - 候選鍵和主鍵：如果關聯表的所有屬性都功能相依於一個或一組屬性，此屬性就是候選鍵（**Candidate Key**）或主鍵（**Primary Key**）。



5-2-1 功能相依的定義 – 定義

- 功能相依是同一個關聯表屬性間的關係，其定義如下所示：

定義5.1：功能相依（Functional Dependency）以關聯表R為例，若關聯表R擁有屬性A與B，A與B可以是複合屬性，我們可以說屬性B功能相依（Functional Dependent）於A，或A功能決定（Functional Determines）B，寫成：

$$R.A \rightarrow R.B$$

或簡寫成：

$$A \rightarrow B$$

若關聯表R擁有 $A \rightarrow B$ 的功能相依，則在關聯表R的每一對t1和t2值組且 $t1(A) = t2(A)$ ，可以得到： $t1(B) = t2(B)$ 。



5-2-1 功能相依的定義 – 說明

- 在關聯表中如果有兩個值組的屬性**A**是相同的，就可以知道屬性**B**的值也一定相同。換句話說，如果知道屬性**A**的值，就可以知道屬性**B**的值。
- 功能相依是一種屬性間的關係，對比實體關聯圖中，就是實體的一對一，一對多、多對一和多對多關聯性，如下所示：

$A1 \rightarrow B1$

$A1 \rightarrow \{ B1, B2, \dots Bn \}$

$\{ A1, A2, \dots, An \} \rightarrow B1$

$\{ A1, A2, \dots, An \} \rightarrow \{ B1, B2, \dots Bn \}$



5-2-1 功能相依的定義－範例

- 例如：學生Students關聯表擁有sid、name、birthday和tel屬性，一些功能相依範例，如下所示：

sid→birthday

sid→tel

sid→{ name, birthday }

sid→{ name, birthday, tel }

{ sid, name }→{ name, birthday, tel }

Students

<u>sid</u>	name	birthday	tel
S001	陳會安	1967/9/3	02-22222222
S002	江小魚	1978/2/2	03-33333333
S003	張三丰	1982/3/3	04-44444444
S004	李四方	1981/4/4	05-55555555
S005	陳小安	1981/2/2	02-22222222
S006	江小魚	1976/9/9	06-66666666



5-2-2 功能相依的種類 – 說明

- 功能相依的種類可以分為三種：完全相依、部分相依和遞移相依。在本節使用的範例是將第3-5節的Students、Courses和Instructors關聯表硬結合成Student_Course_Instructors關聯表（新增講師辦公室office屬性），如下圖所示：

Student_Course_Instructors

sid	name	c_no	title	eid	instructor	office
S001	陳會安	CS101	計算機概論	E001	陳慶新	CS-102
S001	陳會安	CS203	程式語言	E003	李鴻章	M-100
S001	陳會安	CS222	資料庫管理系統	E002	楊金櫟	CIS-101
S001	陳會安	CS213	物件導向程式設計	E003	李鴻章	M-100
S002	江小魚	CS222	資料庫管理系統	E002	楊金櫟	CIS-101
S002	江小魚	CS203	程式語言	E003	李鴻章	M-100
S003	張三丰	CS121	離散數學	E002	楊金櫟	CIS-101
S003	張三丰	CS213	物件導向程式設計	E001	陳慶新	CS-102
S004	李四方	CS222	資料庫管理系統	E002	楊金櫟	CIS-101



5-2-2 功能相依的種類 – 完全相依

- 完全相依（Full Dependency）是指如果屬性A（複合屬性）和B是關聯表屬性，B功能相依於A，而且B沒有功能相依於任何A的子集，則稱B完全功能相依於A。
- 例如：Student_Course_Instructors關聯表的完全相依，如下所示：
 - sid→name
 - c_no→title
 - eid→instructor
 - eid→office



5-2-2 功能相依的種類 – 部分相依

- 部分相依（**Partial Dependency**）是指如果屬性A（複合屬性）和B是關聯表的屬性，B功能相依於A，如果從A中刪除部分屬性後的子集合，仍然不會影響功能相依，則稱B部分相依於A。
- 例如：**Student_Course_Instructors**關聯表的部分相依，如下所示：

{ sid, c_no } → name

{ sid, c_no } → title

{ eid, instructor } → office



5-2-2 功能相依的種類 – 遞移相依

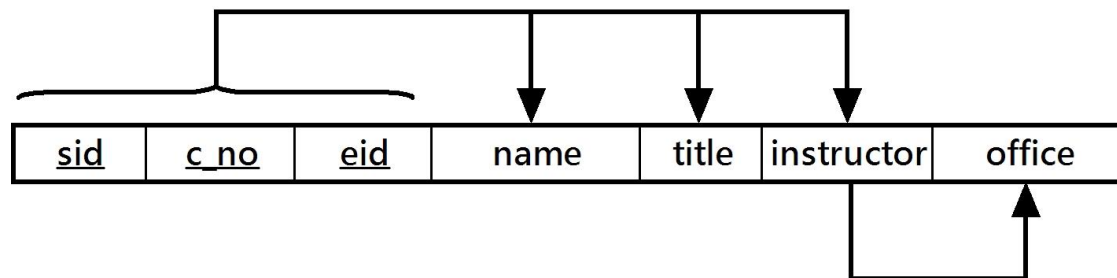
- 如果A、B和C是關聯表的屬性， $A \rightarrow B$ 且 $B \rightarrow C$ ，則稱C是遞移相依（Transitive Dependency）於A， $A \rightarrow C$ 。例如：
Student_Course_Instructors關聯表的遞移相依，如下所示：

$\{ \text{sid}, \text{c_no}, \text{eid} \} \rightarrow \text{office}$

- office屬性是遞移相依於 $\{ \text{sid}, \text{c_no}, \text{eid} \}$ ，因為：

$\{ \text{sid}, \text{c_no}, \text{eid} \} \rightarrow \text{instructor}$

$\text{instructor} \rightarrow \text{office}$





5-2-3 找出關聯表的所有功能相依

- 一般來說，我們可以使用兩種演算法找出所有的功能相依，如下所示：
 - 暴力演算法（**Brute Force Algorithm**）：一種破解密碼常用的演算法，以作苦工方式嘗試所有可能的密碼組合，也就是說，在關聯表找出各屬性間的所有可能組合，然後一一檢查是否符合功能相依，如果關聯表的屬性很多，相對的，就需花費很多時間來找出。
 - 推論基礎演算法（**Inference-based Algorithm**）：因為功能相依可以由其他功能相依來推論出。換句話說，推論基礎演算法是從關聯表已知的功能相依，使用推論規則推論出所有可能的功能相依。



5-2-4 阿姆斯壯推論規則 – 說明

- 關聯表R如果擁有一些功能相依F，「F封閉性」(Closure of F) 寫成F⁺，F⁺是F本身和由其所推論出的所有功能相依的聯集。換句話說，F⁺是功能相依F，再加上從F推論出的所有功能相依的集合，而使用的推論規則就是「阿姆斯壯推論規則」(Armstrong's Inference Rule)。
- 阿姆斯壯推論規則可以從某些關聯表已知的功能相依推論出其他隱含的功能相依，這是一種建全且完整的規則，其意義如下所示：
 - 建全 (Sound)：阿姆斯壯推論規則不會導出多餘的功能相依。
 - 完整 (Complete)：阿姆斯壯推論規則可以導出所有的功能相依。



5-2-4 阿姆斯壯推論規則 – 種類

- W. W. Armstrong在1974年提出三種推論規則，再加上依此擴充出另外兩種規則：分解和聯集規則，如下所示：
 - 反身規則。
 - 擴充規則。
 - 遞移規則。
 - 分解規則。
 - 聯集規則。



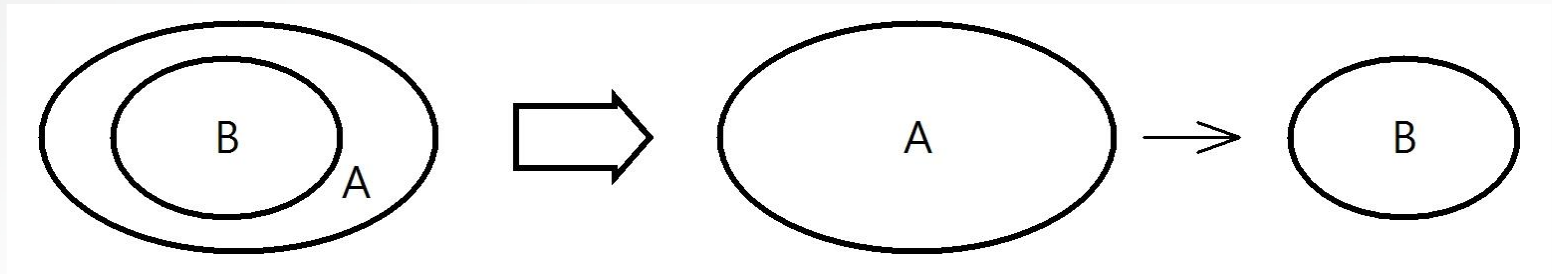
5-2-4 阿姆斯壯推論規則 – 反身規則

- 反身規則（Reflexivity Rule）是如果屬性B是屬性A的子集合，則 $A \rightarrow B$ ，其定義如下：

若： $\{ B1, B2, \dots, Bn \} \{ A1, A2, \dots, An \}$

則： $\{ A1, A2, \dots, An \} \rightarrow \{ B1, B2, \dots, Bn \}$

- 反身規則的圖例，如下圖所示：





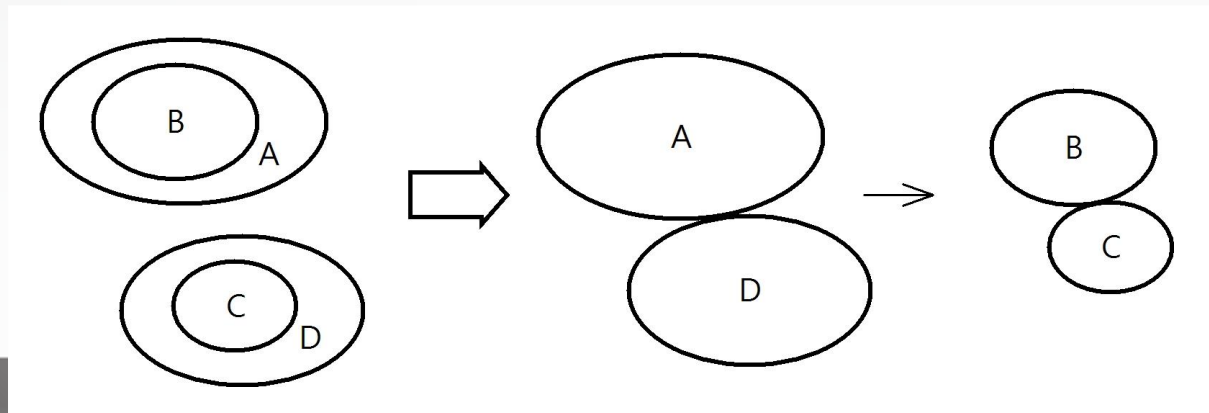
5-2-4 阿姆斯壯推論規則 – 擴充規則

- 擴充規則（Augmentation Rule）是如果 $A \rightarrow B$ 且屬性C是屬性D的子集合，則 $AD \rightarrow BC$ ，其定義如下：

若： $\{A_1, A_2, \dots, A_n\} \rightarrow \{B_1, B_2, \dots, B_n\}$ 且 $\{C_1, \dots, C_n\} \subseteq \{D_1, \dots, D_n\}$

則： $\{A_1, A_2, \dots, A_n, D_1, \dots, D_n\} \rightarrow \{B_1, B_2, \dots, B_n, C_1, \dots, C_n\}$

- 擴充規則的圖例， $A \rightarrow B$ 依反身規則就是 $B \rightarrow A$ ，如下圖所示：





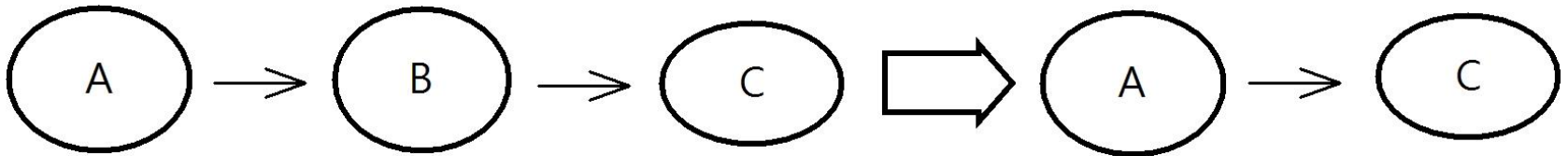
5-2-4 阿姆斯特壯推論規則 – 遞移規則

- 遞移規則（Transitivity Rule）是如果 $A \rightarrow B$ 且 $B \rightarrow C$ ，則 $A \rightarrow C$ ，其定義如下：

若： $\{ A1, \dots, An \} \rightarrow \{ B1, \dots, Bn \}$ 且 $\{ B1, \dots, Bn \} \rightarrow \{ C1, \dots, Cn \}$

則： $\{ A1, \dots, An \} \rightarrow \{ C1, \dots, Cn \}$

- 遞移規則的圖例，如下圖所示：





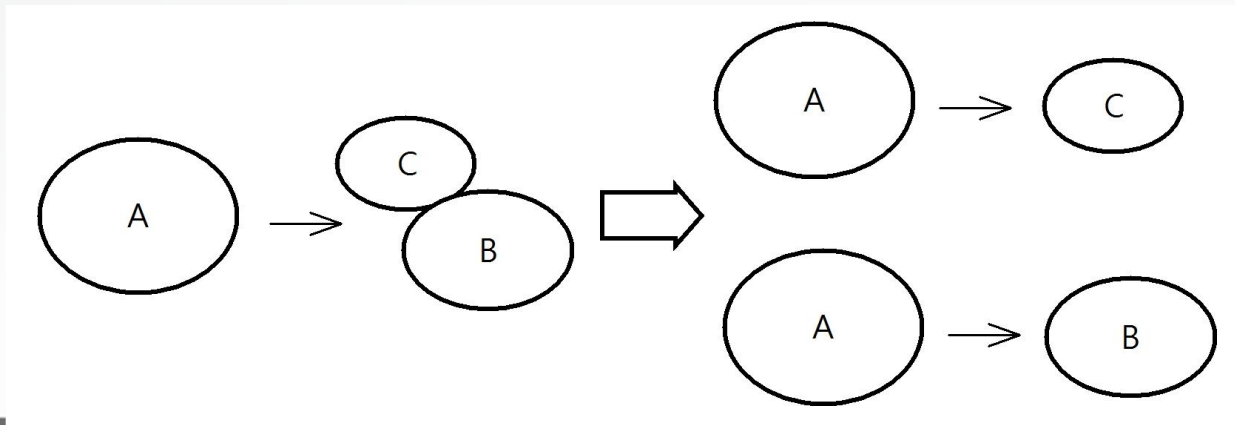
5-2-4 阿姆斯壯推論規則 – 分解規則

- 分解規則（Decomposition Rule）是如果 $A \rightarrow BC$ ，則 $A \rightarrow B$ 且 $A \rightarrow C$ ，其定義如下：

若： $\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_n, C_1, \dots, C_n\}$

則： $\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_n\}$ 且 $\{A_1, \dots, A_n\} \rightarrow \{C_1, \dots, C_n\}$

- 分解規則的圖例，如下圖所示：





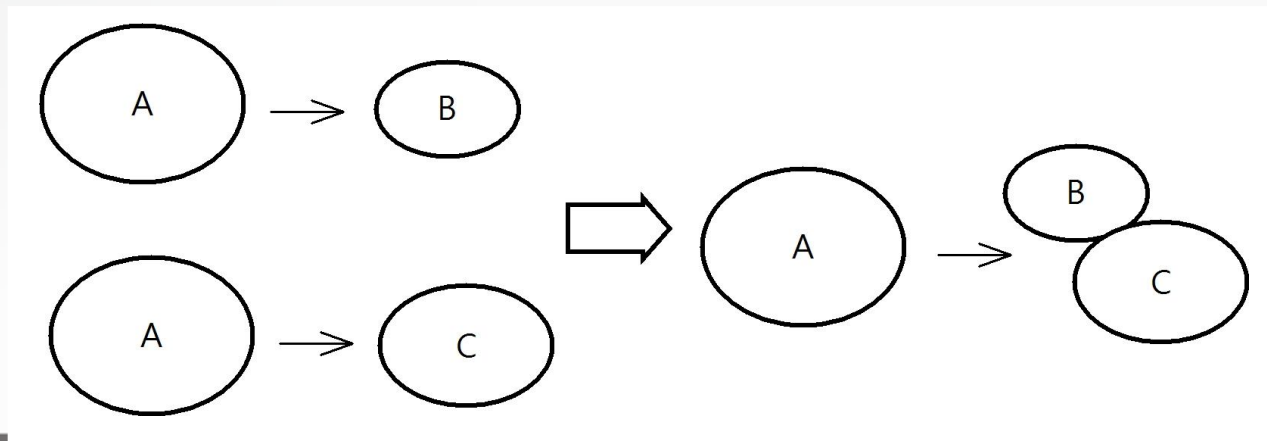
5-2-4 阿姆斯壯推論規則 – 聯集規則

- 聯集規則（Union Rule）是如果 $A \rightarrow B$ 且 $A \rightarrow C$ ，則 $A \rightarrow BC$ ，其定義如下：

若： $\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_n\}$ 且 $\{A_1, \dots, A_n\} \rightarrow \{C_1, \dots, C_n\}$

則： $\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_n, C_1, \dots, C_n\}$

- 聯集規則的圖例，如下圖所示：





5-2-4 阿姆斯壯推論規則 – 範例

- 現在我們只需使用阿姆斯壯推論規則，就可以從關聯表現有的功能相依推論出其他功能相依。例如：關聯表擁有已知的三條功能相依，如下所示：

1: $AB \rightarrow D$

2: $BD \rightarrow E$

3: $E \rightarrow C$

- 請使用上述三條功能相依推論出 $AB \rightarrow C$ 。



5-2-4 阿姆斯壯推論規則 – 推論過程

步驟	使用規則	結果
1	使用第一個FD： $AB \rightarrow D$	$AB \rightarrow D$
2	$A \subseteq B$ 使用反身規則	$AB \rightarrow AB$
3	以步驟2： $AB \rightarrow AB$ 使用分解規則	$AB \rightarrow B$
4	以步驟1： $AB \rightarrow D$ 和步驟3： $AB \rightarrow B$ 使用聯集規則	$AB \rightarrow BD$
5	以步驟4： $AB \rightarrow BD$ 和第2個： $BD \rightarrow E$ 使用遞移規則	$AB \rightarrow E$
6	以步驟5： $AB \rightarrow E$ 和第3個： $E \rightarrow C$ 使用遞移規則	$AB \rightarrow C$



5-3 第一階到第三階正規化型式

- 5-3-1 第一階正規化型式 – 1NF
- 5-3-2 第二階正規化型式 – 2NF
- 5-3-3 第三階正規化型式 – 3NF
- 5-3-4 Boyce-Codd正規化型式 – BCNF
- 5-3-5 符合Boyce-Codd正規化型式的範例





5-3-1 第一階正規化型式 – 1NF(定義)

- 第一階正規化型式是在處理關聯表本身，並沒有解決任何關聯表存在功能相依所造成的資料重複或操作異常等問題。其定義如下所示：

定義5.2： 關聯表R符合第一階正規化型式（First Normal Form；1NF）是指關聯表的每一個定義域（Domain）都是單元值（Atomic Value）且只能是單元值，也就是刪除多重值屬性型態（Multi-value Attribute Type）與複合屬性型態（Composite Attribute Type）的屬性。

- 簡單的說，上述定義是指關聯表沒有多重值和複合屬性。



5-3-1 第一階正規化型式 – 1NF(範例)

- 刪除複合屬性只需將組成的單元值屬性展開，刪除多重值屬性比較複雜，我們可以將屬性分割成關聯表、值組或屬性。例如：**Students**關聯表儲存學生的選課資料，主鍵是學號sid，如下圖所示：

Students

<u>sid</u>	name	c_no	title	eid	instructor	office	room
S001	陳會安	{ CS101, CS203, CS222, CS213 }	{ 計算機概論, 程式語言, 資料庫管理系統, 物件導向程式設計 }	{ E001, E003, E002, E003 }	{ 陳慶新, 李鴻章, 楊金櫟, 李鴻章 }	{ CS-102, M-100, CIS-101, M-100 }	{ 180-M, 221-S, 100-M, 500-K }
S002	江小魚	{ CS222, CS203 }	{ 資料庫管理系統, 程式語言 }	{ E002, E003 }	{ 楊金櫟, 李鴻章 }	{ CIS-101, M-100 }	{ 100-M, 221-S }
S003	張三丰	{ CS121, CS213 }	{ 離散數學, 物件導向程式設計 }	{ E002, E001 }	{ 楊金櫟, 陳慶新 }	{ CIS-101, CS-102 }	{ 221-S, 622-G }
S004	李四方	CS222	資料庫管理系統	E002	楊金櫟	CIS-101	100-M



5-3-1 第一階正規化型式 – 1NF (方法一：分割成不同的關聯表)

- 關聯表如果擁有多重值屬性違反1NF，第一階正規化可以將多重值屬性連同主鍵分割成新的關聯表，如下圖所示：

Students

<u>sid</u>	name
S001	陳會安
S002	江小魚
S003	張三丰
S004	李四方

Classes

<u>sid</u>	<u>c_no</u>	title	<u>eid</u>	instructor	office	room
S001	CS101	計算機概論	E001	陳慶新	CS-102	180-M
S001	CS203	程式語言	E003	李鴻章	M-100	221-S
S001	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S001	CS213	物件導向程式設計	E003	李鴻章	M-100	500-K
S002	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S003	CS203	程式語言	E003	李鴻章	M-100	221-S
S003	CS121	離散數學	E002	楊金欉	CIS-101	221-S
S003	CS213	物件導向程式設計	E001	陳慶新	CS-102	622-G
S004	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M



5-3-1 第一階正規化型式 – 1NF (方法二：分割成值組)

- 第一階正規化可以將多重值屬性改成重複值組，將屬性的每一個多重值都新增一筆值組，如下圖所示：

Students

<u>sid</u>	<u>name</u>	<u>c_no</u>	<u>title</u>	<u>eid</u>	<u>instructor</u>	<u>office</u>	<u>room</u>
S001	陳會安	CS101	計算機概論	E001	陳慶新	CS-102	180-M
S001	陳會安	CS203	程式語言	E003	李鴻章	M-100	221-S
S001	陳會安	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S001	陳會安	CS213	物件導向程式設計	E003	李鴻章	M-100	500-K
S002	江小魚	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S002	江小魚	CS203	程式語言	E003	李鴻章	M-100	221-S
S003	張三丰	CS121	離散數學	E002	楊金欉	CIS-101	221-S
S003	張三丰	CS213	物件導向程式設計	E001	陳慶新	CS-102	622-G
S004	李四方	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M



5-3-1 第一階正規化型式 – 1NF (方法三：分割成不同屬性)

- 第一階正規化還可以將多重值屬性配合空值，分割成為關聯表的多個屬性，不過，其先決條件是多重值個數是有限的。例如：一位學生規定只能修兩門課程（為了方便說明，筆者刪除講師與教室部分的屬性），如下圖所示：

Students

<u>sid</u>	name	c_no1	title1	c_no2	title2
S001	陳會安	CS101	計算機概論	CS203	程式語言
S002	江小魚	CS222	資料庫管理系統	CS203	程式語言
S003	張三丰	CS121	離散數學	CS213	物件導向程式設計
S004	李四方	CS222	資料庫管理系統	NULL	NULL



5-3-2 第二階正規化型式 – 2NF(定義)

- 第二階正規化的目的是讓每一個關聯表只能儲存同類資料，也就是單純化關聯表儲存的資料。當關聯表符合1NF後，就可以進行第二階正規化，其定義如下所示：

定義5.3： 關聯表R符合第二階正規化型式（**Second Normal Form ; 2NF**）是指關聯表符合1NF，而且所有非主鍵（**Primary Key**）的屬性都完全相依（**Fully Dependency**）於主鍵，也就是刪除所有部份相依的屬性。

- 上述定義是指關聯表中，不是主鍵的屬性需要完全相依於主鍵；反過來說，就是刪除關聯表所有部分相依（**Partial Dependency**）的屬性。



5-3-2 第二階正規化型式 – 2NF(範例)

- 當執行學生選課資料關聯表的第一階正規化後，目前關聯表已經分割成Students和Classes兩個關聯表。接著繼續執行Classes關聯表的第二階正規化，如下圖所示：

Classes

<u>sid</u>	<u>c_no</u>	title	<u>eid</u>	instructor	office	room
S001	CS101	計算機概論	E001	陳慶新	CS-102	180-M
S001	CS203	程式語言	E003	李鴻章	M-100	221-S
S001	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S001	CS213	物件導向程式設計	E003	李鴻章	M-100	500-K
S002	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M
S003	CS203	程式語言	E003	李鴻章	M-100	221-S
S003	CS121	離散數學	E002	楊金欉	CIS-101	221-S
S003	CS213	物件導向程式設計	E001	陳慶新	CS-102	622-G
S004	CS222	資料庫管理系統	E002	楊金欉	CIS-101	100-M



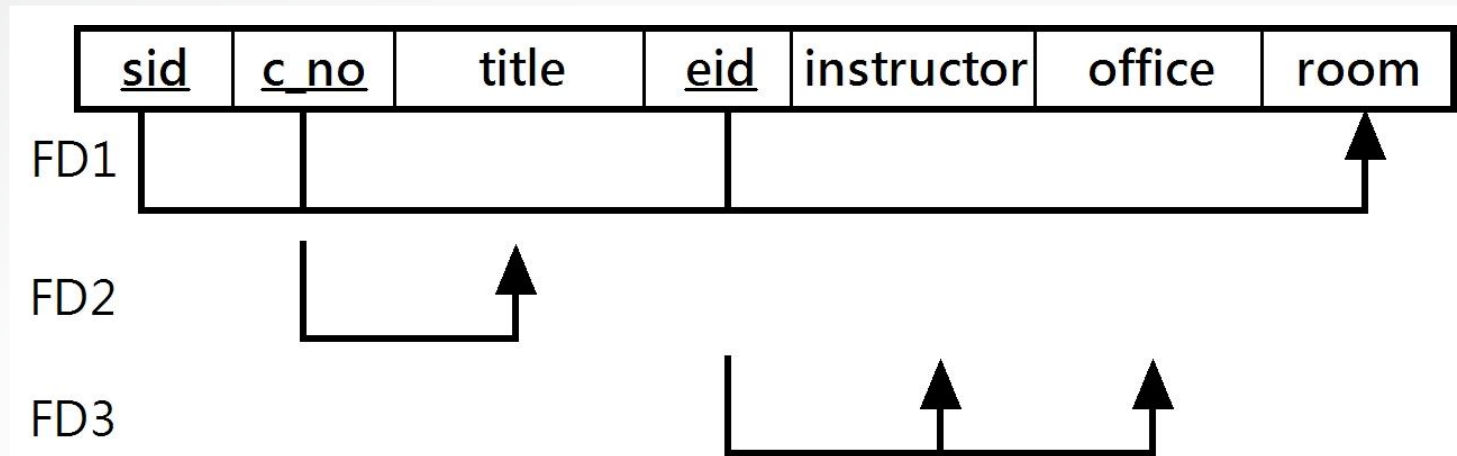
5-3-2 第二階正規化型式 – 2NF (功能相依)

- Classes 關聯表的主鍵是 (sid, c_no, eid)，關聯表已知的功能相依，如下所示：

FD1 : { sid, c_no, eid } → room

FD2 : c_no → title

FD3 : eid → { instructor, office }





5-3-2 第二階正規化型式 – 2NF (正規化結果)

- 將功能相依 $c_no \rightarrow title$ 和 $eid \rightarrow \{ instructor, office \}$ 兩邊屬性獨立成關聯表，左邊剩下的屬性就是新關聯表的主鍵，如下圖所示：

Classes

<u>sid</u>	<u>c_no</u>	<u>eid</u>	room
S001	CS101	E001	180-M
S001	CS203	E003	221-S
S001	CS222	E002	100-M
S001	CS213	E003	500-K
S002	CS222	E002	100-M
S003	CS203	E003	221-S
S003	CS121	E002	221-S
S003	CS213	E001	622-G
S004	CS222	E002	100-M

Courses

<u>c_no</u>	title
CS101	計算機概論
CS203	程式語言
CS222	資料庫管理系統
CS213	物件導向程式設計
CS121	離散數學

Instructors

<u>eid</u>	instructor	office
E001	陳慶新	CS-102
E002	楊金欉	CIS-101
E003	李鴻章	M-100



5-3-3 第三階正規化型式 – 3NF(定義)

- 第三階正規化的目的是移除哪些不是直接功能相依於主鍵的屬性，這些屬性是借由另一個屬性來功能相依於主鍵。當關聯表符合2NF後，就可以進行第三階正規化，其定義如下所示：

定義5.4： 關聯表R符合第三階正規化型式（Third Normal Form；3NF）是指關聯表符合2NF，而且所有非主鍵（Primary Key）的屬性都只能功能相依（Functional Dependency）於主鍵；沒有功能相依於其他非主鍵的屬性，即刪除遞移相依的屬性。

- 上述定義是指關聯表中不屬於主鍵的屬性都只能功能相依於主鍵，而不能同時功能相依於其他非主鍵的屬性，即刪除關聯表所有遞移相依（Transitive Dependency）。



5-3-3 第三階正規化型式 – 3NF(範例)

- 例如：請繼續第5-3-2節的關聯表，執行Instructors關聯表的第三階正規化，如下圖所示：

Instructors

<u>eid</u>	instructor	o_no	office
E001	陳慶新	CS-102	網路研究室
E002	楊金欉	CIS-101	資料庫中心
E003	李鴻章	M-100	數學系電腦中心
E004	王陽明	M-100	數學系電腦中心



5-3-3 第三階正規化型式 – 3NF (找出遞移相依)

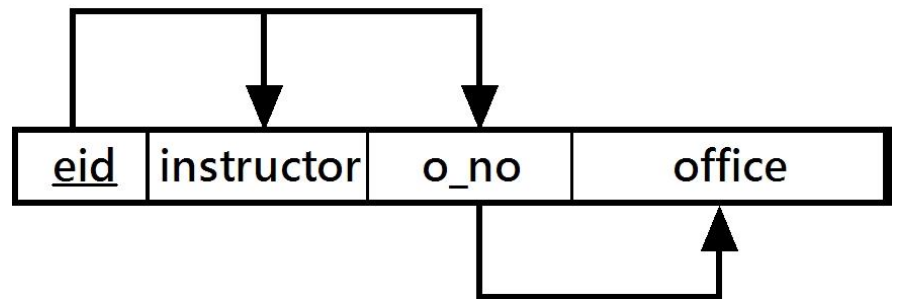
- Instructors關聯表已知的功能相依，如下所示：

$eid \rightarrow office$

$eid \rightarrow o_no$

$o_no \rightarrow office$

- 上述第office屬性功能相依於eid主鍵，它是借由倒數第o_no屬性功能相依於eid和office屬性功能相依於o_no屬性所得，所以 $eid \rightarrow office$ 是遞移相依，如下圖所示：





5-3-3 第三階正規化型式 – 3NF (正規化結果)

- 在執行第三階正規化時，就是將造成遞移相依的 $A \rightarrow B$ 功能相依兩邊的屬性獨立成關聯表，左邊的屬性就是新關聯表的主鍵，如下圖所示：

Instructors

<u>eid</u>	instructor	o_no
E001	陳慶新	CS-102
E002	楊金欉	CIS-101
E003	李鴻章	M-100
E004	王陽明	M-100

Offices

<u>o_no</u>	office
CS-102	網路研究室
CIS-101	資料庫中心
M-100	數學系電腦中心



5-3-4 Boyce-Codd正規化型式 – BCNF(說明)

- Boyce-Codd正規化型式可以視為一種更嚴格的第三階正規化型式，其目的是保證關聯表的所有屬性都功能相依於候選鍵。Boyce-Codd正規化可以讓所有屬性都完全功能相依於候選鍵，而不是候選鍵的部分屬性。
- 換句話說，Boyce-Codd正規化是在處理關聯表擁有多個候選鍵的特殊情況，在Boyce-Codd正規化處理的關聯表至少需要擁有二個或更多個候選鍵，且這兩個候選鍵是：
 - 複合候選鍵。
 - 在複合候選鍵之間擁有重疊屬性，也就是說至少擁有一個相同屬性。



5-3-4 Boyce-Codd正規化型式 – BCNF(定義)

■ BCNF正規化型式的定義，如下所示：

定義5.5： 關聯表R符合BCNF正規化型式（Boyce-Codd Normal Form；BCNF）是指關聯表所有主要的功能相依 $A \rightarrow B$ ，A一定且只可以是候選鍵（Candidate Keys），也就是刪除只功能相依候選鍵部分屬性的功能相依。

■ 簡單的說，上述定義是指關聯表中，主要功能相依 $A \rightarrow B$ 的左邊屬性A稱為「決定屬性」（Determinant），決定屬性一定是候選鍵或主鍵。



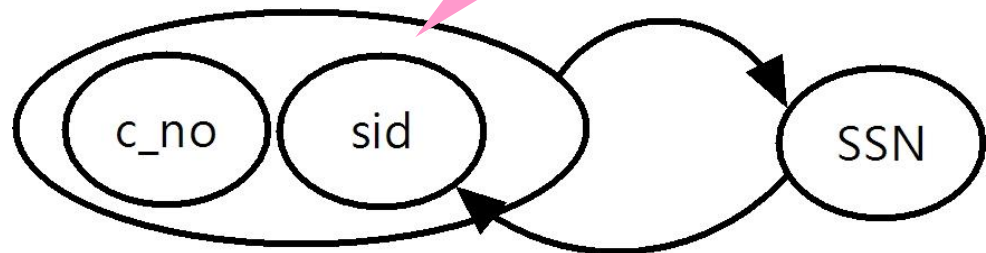
5-3-4 Boyce-Codd正規化型式 – BCNF(範例)

- 例如：學生身份證字號（SSN）與成績（grade）的 Students 關聯表，在 Students 關聯表擁有兩個候選鍵(sid, c_no)和(SSN, c_no)，如下圖所示：

Students

<u>sid</u>	SSN	<u>c_no</u>	grade
S001	H12345678	CS101	A
S001	H12345678	CS203	B
S002	J45678377	CS203	B
S003	I12345674	CS213	A-

兩個候選鍵擁有重疊
屬性c_no，在候選鍵之
間擁有功能相依
SSN→sid





5-3-4 Boyce-Codd正規化型式 – BCNF(正規化結果)

- 例如：執行前述的關聯表Students關聯表的BCNF正規化，如下圖所示：

SSN

<u>sid</u>	SSN
S001	H12345678
S002	J45678377
S003	I12345674

Transcript

<u>sid</u>	<u>c_no</u>	grade
S001	CS101	A
S001	CS203	B
S002	CS203	B
S003	CS213	A-

使用sid屬性進行分割

使用SSN屬性進行分割

SSN

<u>sid</u>	SSN
S001	H12345678
S002	J45678377
S003	I12345674

Transcript

<u>SSN</u>	<u>c_no</u>	grade
H12345678	CS101	A
H12345678	CS203	B
J45678377	CS203	B
I12345674	CS213	A-



5-3-5 符合Boyce-Codd正規化型式的範例 – 說明

- Boyce-Codd正規化型式的條件是關聯表至少擁有二個或多個複合候選鍵，不過，關聯表擁有兩個複合候選鍵，並不表示一定需要進行BCNF正規化，兩個複合候選鍵的關聯表仍然可能符合BCNF。
- 例如：學生期末考座位的Exams關聯表，因為期末考分成多天進行考試，所以學生各科的考試座位可能不同，如下圖所示：

Exams

<u>sid</u>	<u>c_no</u>	seat_no
S001	CS101	C101-02
S001	CS203	C121-03
S002	CS203	C121-11
S003	CS213	C121-03



5-3-5 符合Boyce-Codd正規化型式的範例 – 範例說明

- 在Exams關聯表有兩個候選鍵，如下所示：
 (sid, c_no)
 (seat_no, c_no)
- 兩個候選鍵擁有重疊屬性c_no，在候選鍵之間並沒有功能相依seat_no→sid，例如：seat_no屬性值C121-03，同時有S001和S003兩位學生在不同科目進行考試。在Exams關聯表擁有的功能相依清單，如下所示：
 { sid, c_no }→seat_no
 { seat_no, c_no }→sid
- 因為功能相依的決定屬性都是候選鍵，符合BCNF定義。Exams關聯表雖然擁有兩組候選鍵，但是關聯表不但符合3NF，而且符合BCNF。



5-4 多重值相依與第四階正規化型式

- 5-4-1 多重值相依（Multi-valued Dependency）
- 5-4-2 第四階正規化型式 – 4NF





5-4-1 多重值相依 – 說明

- 第四階正規化型式的基礎是「多重值相依」（**Multi-valued Dependency ; MVD**），這是1977年R. Fagin所提出，屬於功能相依的推廣定義，或是說，功能相依是多重值相依的一種特例。
- 多重值相依（**Multi-valued Dependency**）的先決條件是關聯表擁有3個以上屬性，因為多重值相依的定義有些複雜，我們準備使用一個關聯表範例來說明多重值相依。



5-4-1 多重值相依 – 範例

- 例如：一門課可能由多位講師授課，每一門課可以使用多本教課書，儲存這些資料的 **Course_Instructor_Textbook** 關聯表，簡稱為 **CIT**，如下圖所示：

CIT

<u>course</u>	<u>instructor</u>	<u>textbook</u>
物件導向程式設計	陳慶新	Java程式設計範例教本
物件導向程式設計	陳慶新	物件導向程式設計
物件導向程式設計	李鴻章	Java程式設計範例教本
物件導向程式設計	李鴻章	物件導向程式設計
程式語言	李鴻章	C/C++程式設計範例教本
程式語言	李鴻章	Java程式設計範例教本
程式語言	李鴻章	Visual Basic程式設計範例教本



5-4-1 多重值相依 – 定義

定義5.6：多重值相依（Multi-valued Dependency）以關聯表R為例，若關聯表R是由屬性A、B與C組成R(A, B, C)，A、B與C可以是複合屬性，則屬性B多重值相依（Multi-valued Dependent）於A，或A多重決定（Multi-determines）於B，寫成：

$$R.A \twoheadrightarrow R.B$$

或簡寫成：

$$A \twoheadrightarrow B$$

若關聯表R擁有 $A \twoheadrightarrow B$ 多重值相依，則表示存在t1和t2值組且 $t1(A) = t2(A)$ ，且另外存在二個值組t3和t4滿足下列條件：

$$t1(A) = t2(A) = t3(A) = t4(A)$$

$$t3(B) = t1(B)$$

$$t3(C) = t2(C)$$

$$t4(B) = t2(B)$$

$$t4(C) = t1(C)$$



5-4-1 多重值相依 – 定義說明

- 多重值相依的定義有些複雜，以CIT關聯表為例，因為關聯表CIT擁有 $\text{course} \rightarrow \rightarrow \text{textbook}$ ($A \rightarrow \rightarrow B$) 多重值相依，所以在關聯表可以找到交換 instructor 屬性的2對值組t1、t2和t3、t4，如下圖所示：

	A	C	B
CIT	<u>course</u>	<u>instructor</u>	<u>textbook</u>
t1	物件導向程式設計	陳慶新	物件導向程式設計
t2	物件導向程式設計	李鴻章	Java程式設計範例教本
t3	物件導向程式設計	李鴻章	物件導向程式設計
t4	物件導向程式設計	陳慶新	Java程式設計範例教本



5-4-1 多重值相依 – 異常狀況

- 在CIT關聯表擁有很多重複資料，這些重複資料會導致新增或更新異常，例如：新增講師陳慶新教授程式語言這門課，雖然只有一門課，但是需要同時新增3筆值組，每一個值組是一本教課書，否則就會產生資料不一致的問題。

CIT

<u>course</u>	<u>instructor</u>	<u>textbook</u>
物件導向程式設計	陳慶新	Java程式設計範例教本
物件導向程式設計	陳慶新	物件導向程式設計
物件導向程式設計	李鴻章	Java程式設計範例教本
物件導向程式設計	李鴻章	物件導向程式設計
程式語言	李鴻章	C/C++程式設計範例教本
程式語言	李鴻章	Java程式設計範例教本
程式語言	李鴻章	Visual Basic程式設計範例教本
程式語言	陳慶新	C/C++程式設計範例教本
程式語言	陳慶新	Java程式設計範例教本
程式語言	陳慶新	Visual Basic程式設計範例教本



5-4-1 多重值相依 – 推論規則1

- 多重值相依的推論規則是1977年C. Beeri、R. Fagin和J. H. Howard擴充自阿姆斯壯推論規則，其中前3個推論規是源自功能相依，如下所示：
 - FD反身規則（FD Reflexivity）：如果屬性B是屬性A的子集合，則 $A \rightarrow B$ 。
 - FD擴充規則（FD Augmentation）：如果 $A \rightarrow B$ 且屬性C是屬性D的子集合，則 $AD \rightarrow BC$ 。
 - FD遞移規則（FD Transitivity）：如果 $A \rightarrow B$ 且 $B \rightarrow C$ ，則 $A \rightarrow C$ 。
 - MVD擴充規則（MVD Augmentation）：如果 $A \twoheadrightarrow B$ 且屬性C是屬性D的子集合，則 $AD \twoheadrightarrow BC$ 。
 - MVD遞移規則（MVD Transitivity）：如果 $A \twoheadrightarrow B$ 且 $B \twoheadrightarrow C$ ，則 $A \twoheadrightarrow C - B$ 。



5-4-1 多重值相依 – 推論規則2

- MVD互補規則（MVD Complementation）：如果 $A \twoheadrightarrow B$ ，則 $A \twoheadrightarrow (R - A - B)$ 。
- MVD聯集規則（MVD Union）：如果 $A \twoheadrightarrow B$ 且 $A \twoheadrightarrow C$ ，則 $A \twoheadrightarrow BC$ 。
- MVD分解規則（MVD Decomposition）：如果 $A \twoheadrightarrow BC$ ，則 $A \twoheadrightarrow B$ ， $A \twoheadrightarrow C$ 且 $A \twoheadrightarrow B - C$ 。
- 複製規則（Replication或稱為升級Promotion）：如果 $A \rightarrow B$ ，則 $A \twoheadrightarrow B$ 。
- 合併規則（Coalescence）：如果 $A \twoheadrightarrow B$ 且 $C \rightarrow D$ ， D 是 B 的子集合， C 與 B 的交集是空集合，則 $A \rightarrow D$ 。
- 偽裝遞移規則（Pseudotransitivity）：如果 $A \twoheadrightarrow B$ 且 $CB \rightarrow D$ ，則 $CA \rightarrow D$ 。
- 混合偽裝遞移規則（Mixed Pseudotransitivity）：如果 $A \twoheadrightarrow B$ 且 $AB \rightarrow C$ ，則 $A \twoheadrightarrow C - B$ 。



5-4-2 第四階正規化型式 – 4NF(定義)

- 第四階正規化的目的是刪除多重值相依，當關聯表符合BCNF後，如果關聯表擁有多重值相依，就需要進行第四階正規化，其定義如下所示：

定義5.7： 關聯表R符合第四階正規化型式（Fourth Normal Form；4NF）是指關聯表符合BCNF，而且所有多重值相依（Multi-valued Dependency）都是功能相依（Functional Dependency），也就是刪除多重值相依。

- 簡單的說，上述定義是指關聯表沒有存在的多重值相依，只有功能相依。



5-4-2 第四階正規化型式 – 4NF(正規化)

- CIT關聯表擁有2個多重值相依，如下所示：

course $\rightarrow\rightarrow$ textbook

course $\rightarrow\rightarrow$ instructor

- 執行第四階正規化，只需將多重值相依A $\rightarrow\rightarrow$ B兩邊的屬性獨立成關聯表即可，如下圖所示：

CI

<u>course</u>	<u>instructor</u>
物件導向程式設計	陳慶新
物件導向程式設計	李鴻章
程式語言	李鴻章

CT

<u>course</u>	<u>textbook</u>
物件導向程式設計	Java程式設計範例教本
物件導向程式設計	物件導向程式設計
程式語言	C/C++程式設計範例教本
程式語言	Java程式設計範例教本
程式語言	Visual Basic程式設計範例教本



5-5 合併相依與第五階正規化型式

- 5-5-1 合併相依 (Join Dependency)
- 5-5-2 第五階正規化型式 – 5NF





5-5 合併相依與第五階正規化型式

- 第五階正規化型式的基礎則是擴充多重值相依的「合併相依」（**Join Dependency**）。
- 前四階正規化主要是將關聯表每一種相依，分割成「兩」個關聯表來滿足各階的正規化型式，但是，一些非常特殊的關聯表，如果只分割成兩個並不能解決資料重複和異常操作問題。此時，我們需要使用第五階正規化型式，將關聯表分割成三個或以上的關聯表。



5-5-1 合併相依 – 說明與範例

- 合併相依（Join Dependency）是將三個循環擁有關聯性的屬性置於同一個關聯表。例如：每一個科系（**department**）開多門課；課程（**course**）可以給多位學生修；學生（**student**）可以修不同科系的課。三個屬性循環擁有關聯性，而且儲存在同一個關聯表**Department_Course_Student**，簡稱**DCS**，如下圖所示：

DCS

<u>department</u>	<u>course</u>	<u>student</u>
資訊系	101	陳會安
資訊系	202	江小魚
資管系	202	江小魚
資管系	101	張三丰
資管系	222	江小魚



5-5-1 合併相依－投影分割

- 將DCS關聯表使用投影運算（投影運算是取出部分關聯表的屬性，即關聯表屬性集合的子集合，詳見第16-2-2節的說明）分割成三個關聯表：DC (department, course)、CS (course, student)、SD (student, department)，如下圖所示：

DC

<u>department</u>	<u>course</u>
資訊系	101
資訊系	202
資管系	202
資管系	101
資管系	222

CS

<u>course</u>	<u>student</u>
101	陳會安
202	江小魚
101	張三丰
222	江小魚

SD

<u>student</u>	<u>department</u>
陳會安	資訊系
江小魚	資訊系
江小魚	資管系
張三丰	資管系



5-5-1 合併相依 – 自然合併

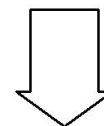
- 先執行DC和CS的自然合併，接著再與SD進行自然合併，就恢復成原始關聯表DCS，如右圖所示：

DCS_Temp

<u>department</u>	<u>course</u>	<u>student</u>
資訊系	101	陳會安
資訊系	101	張三丰
資訊系	202	江小魚
資管系	202	江小魚
資管系	101	陳會安
資管系	101	張三丰
資管系	222	江小魚

SD

<u>student</u>	<u>department</u>
陳會安	資訊系
江小魚	資訊系
江小魚	資管系
張三丰	資管系



DCS

<u>department</u>	<u>course</u>	<u>student</u>
資訊系	101	陳會安
資訊系	202	江小魚
資管系	202	江小魚
資管系	101	張三丰
資管系	222	江小魚



5-5-1 合併相依 – 定義

- 合併相依是多重值相依的推廣定義，其定義如下所示：

定義5.8：合併相依（Join Dependency）以關聯表R為例，若關聯表R等於自然合併運算 $R1 \bowtie R2 \bowtie \dots \bowtie Rn$ ， Ri 是關聯表R屬性的子集合，則，關聯表R滿足合併相依 $(R1, R2, \dots, Rn)$ 。

- 以上述定義來檢視關聯表DCS，關聯表DCS可以使用投影運算分割成DC、CS和SD三個關聯表，執行 $DC \bowtie CS \bowtie SD$ 自然合併運算就可以還原成原始DCS關聯表，所以關聯表DCS擁有合併相依。

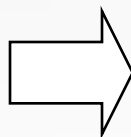


5-5-1 合併相依 – 異常狀況

- 關聯表如果擁有合併相依，在新增和刪除時就會產生異常情況，例如：**DCS**關聯表擁有合併相依，我們準備以新增異常為例來說明合併相依產生的異常情況。在**DCS**關聯表新增一筆值組{資管系, 101, 江小魚}，如下圖所示：

DCS

<u>department</u>	<u>course</u>	<u>student</u>
資訊系	101	陳會安
資訊系	202	江小魚
資管系	101	江小魚



DCS

<u>department</u>	<u>course</u>	<u>student</u>
資訊系	101	陳會安
資訊系	101	江小魚
資訊系	202	江小魚
資管系	101	江小魚

將上述關聯表以投影運算分割成DC、CS和SD三個關聯表，再將它使用合併運算結合起來，可以發現多了一筆值組



5-5-2 第五階正規化型式 – 5NF(定義)

- 第五階正規化是在處理合併相依的問題，當關聯表符合4NF後，如果關聯表擁有合併相依，就需要進行第五階正規化，其定義如下所示：

定義5.9： 關聯表R符合第五階正規化型式（Fifth Normal Form；5NF），也稱為投影-合併正規化型式（Project-Join Normal Form；PJNF），這是指關聯表符合4NF，且所有合併相依($R_1, R_2, \dots, R_n$)的關聯表屬性子集 R_i ，都是關聯表R的超鍵（Superkey），也就是刪除合併相依。

- 上述定義是指關聯表沒有存在合併相依。以上一節的DCS關聯表為例，DCS關聯表擁有合併相依，所以不符合5NF。



5-5-2 第五階正規化型式 – 5NF(正規化)

- DCS關聯表擁有合併相依，可以使用投影運算進行再次分割，並不符合5NF，所以需要執行第五階正規化，使用投影運算分割成三個關聯表，如下圖所示：

DC

<u>department</u>	<u>course</u>
資訊系	101
資訊系	202
資管系	202
資管系	101
資管系	222

CS

<u>course</u>	<u>student</u>
101	陳會安
202	江小魚
101	張三丰
222	江小魚

SD

<u>student</u>	<u>department</u>
陳會安	資訊系
江小魚	資訊系
江小魚	資管系
張三丰	資管系



End
