

第4章 關聯式資料庫模型

- 4-1 關聯式資料庫模型的基礎
- 4-2 關聯式資料庫模型的資料結構
- 4-3 關聯式資料庫模型的完整性限制條件
- 4-4 關聯式資料庫模型的數學理論
- 4-5 關聯式資料庫



4-1 關聯式資料庫模型的基礎

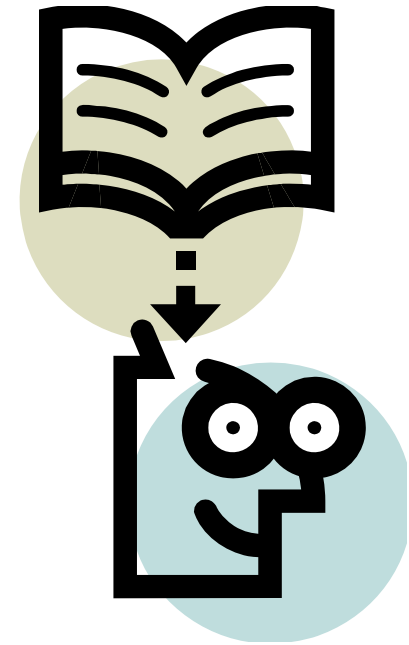
- 關聯式資料庫模型（Relational Database Model）是1969年E. F. Codd博士在IBM公司的研究成果。
- 發表著名的關聯式資料庫12條規則。
 - **Rule 0:** The system must qualify as *relational*, as a *database*, and as a *management system*.
 - **Rule 1:** The *information rule*:
 - All information in the database is to be represented in one and only one way, namely by values in column positions within rows of tables.
 - **Rule 2:** The *guaranteed access rule* (primary keys)
 - **Rule 3:** *Systematic treatment of null values*
 - ...

4-1 關聯式資料庫模型的基礎

- 關聯式資料模型的組成元素，如下所示：
 - 資料結構（**Data Structures**）：資料的組成方式；以關聯式資料模型來說，就是欄和列組成表格的關聯表（**Relations**）。
 - 資料操作或運算（**Data Manipulation或Operations**）：資料相關操作的關聯式代數和計算。
 - 完整性限制條件（**Integrity Constraints**）：維護資料完整性條件，其目的是確保儲存資料是合法資料。

4-2 關聯式資料庫模型的資料結構

- 4-2-1 關聯表綱要
- 4-2-2 關聯表實例
- 4-2-3 關聯表的特性
- 4-2-4 關聯表的種類



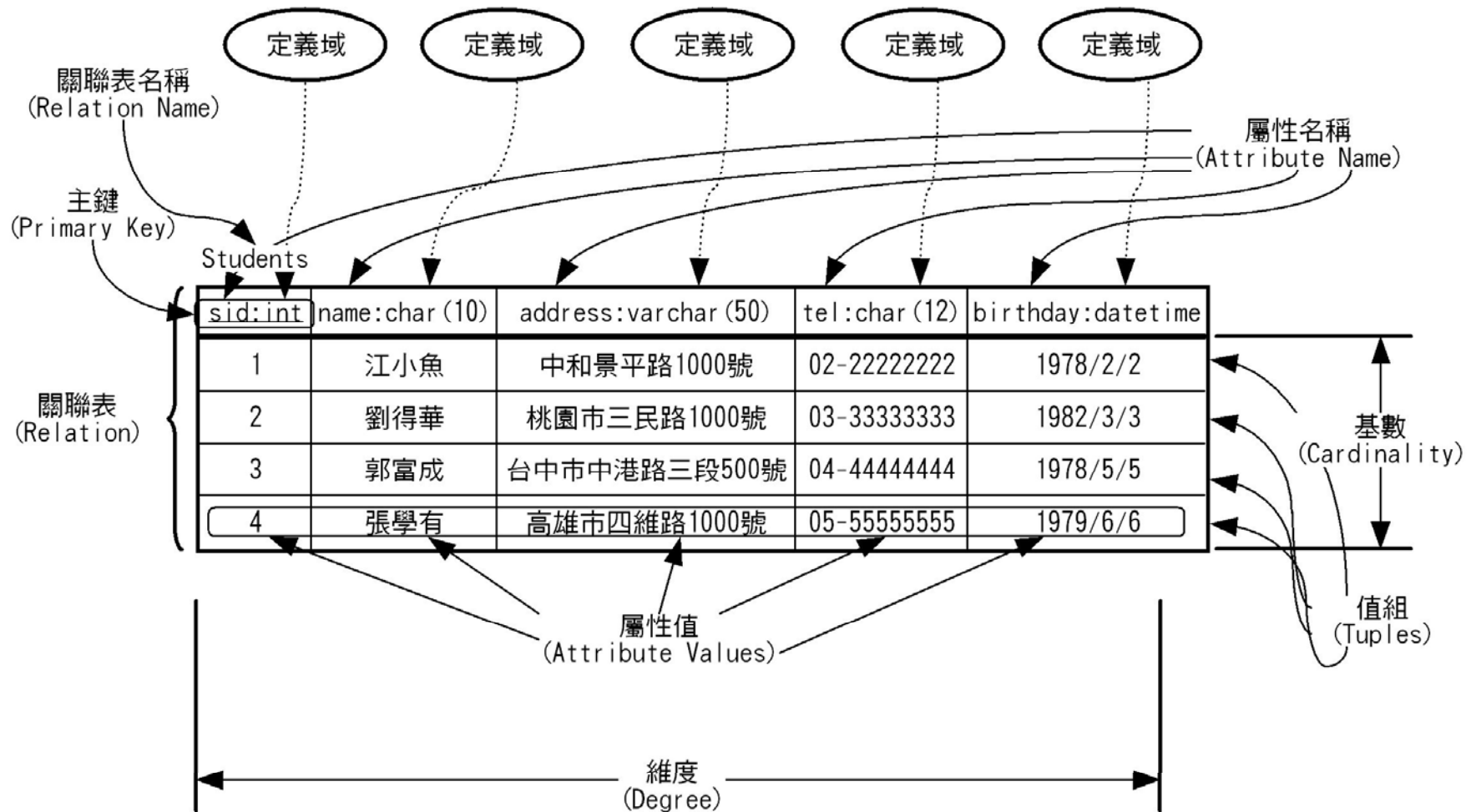
4-2 關聯式資料庫模型的資料結構

- 關聯式資料庫是一組關聯表（**Relations**）的集合；關聯表是關聯式資料庫模型的資料結構（**Data Structure**），使用二維表格組織資料，如下所示：
 - － 關聯表綱要（**Relation Schema**）：包含關聯表名稱、屬性名稱和其定義域。
 - － 關聯表實例（**Relation Instance**）：指某個時間點儲存在關聯表的資料（因為儲存的資料可能隨時改變），可以視為是一個二維表格，其儲存的每一筆記錄是一個「值組」（**Tuples**）。

4-2 關聯式資料庫模型的資料結構- 圖例

關聯表綱要 (Relation Schema)	Students				
	sid:Integer	name:string	address:string	tel:string	birthday:date
關聯表實例 (Relation Instance)	S001	江小魚	中和景平路1000號	02-22222222	1978/2/2
	S002	劉得華	桃園市三民路1000號	03-33333333	1982/3/3
	S003	郭富成	台中市中港路三段500號	04-44444444	1978/5/5
	S004	張學有	高雄市四維路1000號	05-55555555	1979/6/6

4-2-1 關聯表綱要-相關術語1



4-2-1 關聯表綱要-相關術語2

- 關聯表（**Relations**）：相當於一個二維表格，每一個關聯表需要指定關聯表名稱。
 - 不同於表格的是，關聯表並不用考慮各列和各欄資料的順序
- 屬性（**Attributes**）：屬性有其名稱與定義域(domain)。
 - 在關聯表的所有屬性是一個「屬性集合」（Attribute Set），關聯表的屬性並不能重複。
- 定義域（**Domains**）：是一組可接受的屬性值的集合，通常使用資料型態來代表值集合的範圍。

4-2-1 關聯表綱要-相關術語3

- 值組（**Tuples**）：關聯表的一列，也就是一筆記錄，是一組目前屬性值的集合。
- 維度（**Degree**）：關聯表的維度指關聯表的屬性數目。
- 基數（**Cardinality**）：關聯表的基數指關聯表的值組數目。
- 主鍵（**Primary Key**）：可用來識別值組的一個或一組屬性。
 - －故在關聯表中，主鍵的值不會重複出現。

4-2-1 關聯表綱要-相關術語對照

- 關聯式資料庫、資料庫與檔案系統術語對照表

關聯式資料庫模型	資料庫	檔案系統
關聯表 (Relation)	資料表 (Table)	檔案 (File)
值組 (Tuple)	資料列 (Row)	記錄 (Record)
屬性 (Attribute)	資料欄 (Column)	欄位 (Field)

4-2-1 關聯表綱要-表示法(語法)

- 關聯表綱要表示法的語法為：
relation_name (attribute1, attribute2,
attribute3, ... , attributeN)
- 說明如下：
 - relation_name：關聯表名稱。
 - attribute1, attribute2, attribute3, , attributeN：
括號中是屬性清單，通常省略屬性的定義域。
 - 在屬性加上底線表示它是主鍵，外來鍵可以使用虛線底線或其他表示方法。

4-2-1 關聯表綱要-表示法(範例)

- 以關聯表Students為例，其關聯表綱要如下所示：
`Students (sid, name, address, tel, birthday)`
- 上述關聯表Students的主鍵是sid。
- 如果在關聯表綱要需要標示定義域的int、char和datetime，可以使用括號標示在屬性之後：
`Students (sid(int), name(char), address(varchar), tel(char), birthday(datetime))`

4-2-2 關聯表實例-說明

- 「關聯表實例」（Relation Instance）是一個有限個數的集合，集合內容是關聯表的值組（Tuples）。
- 更正確的說，因為關聯表資料可能隨時更改，所以關聯表實例是指某一時間點的值組集。

1	江小魚	中和景平路1000號	02-22222222	1978/2/2
2	劉得華	桃園市三民路1000號	03-33333333	1982/3/3
3	郭富成	台中市中港路三段500號	04-44444444	1978/5/5
4	張學有	高雄市四維路1000號	05-55555555	1979/6/6

4-2-2 關聯表實例-定義域(說明)

定義域 (Domains)

- 定義域是一組可接受值的集合，這些值是
不可分割的單元值 (Atomic)，不可以是另
一個集合。
- 對比程式語言來說，定義域相當於變數的
資料型態；值組的屬性值相當於是變數值，
滿足資料型態的定義域範圍。

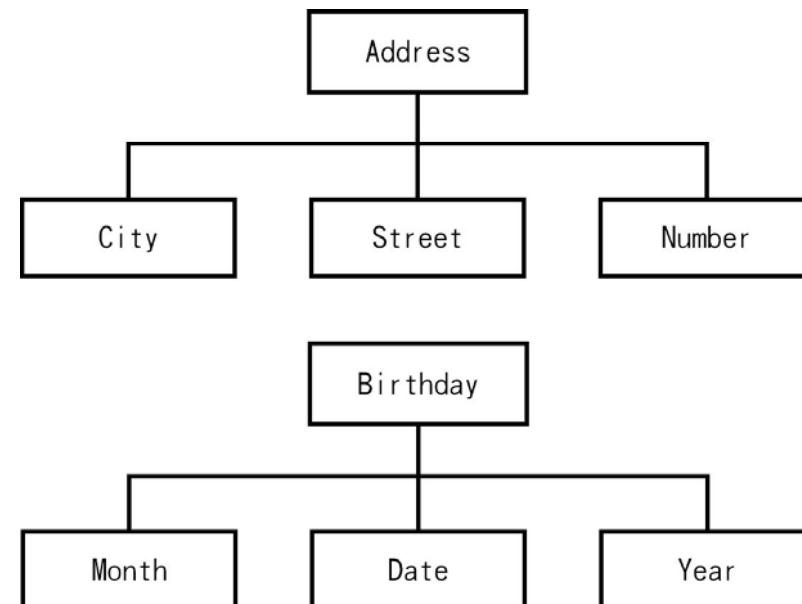
4-2-2 關聯表實例-定義域(種類)

- 屬性主要可以分為兩種：
 - 簡單屬性（**Simple Attributes**）：簡單屬性是一種不可再分割的屬性，其定義域是相同型態的單元值（**Atomic**）集合。
 - 複合屬性（**Composite Attributes**）：複合屬性是由簡單屬性所組成的屬性，它可以形成一個階層架構。例如：地址屬性和生日屬性可以由數個簡單屬性所組成，如下所示：
Address = City+Street+Number
Birthday = Month+Day+Year

4-2-2 關聯表實例-定義域(圖例)

- Address屬性是由City、Street和Number所組成，屬性值的定義域也是由City、Street和Number屬性的定義域組成，生日Birthday屬性的定義域則是月份Month、Day和Year屬性的整數定義域所組成。

目前大多數關聯式
資料庫系統並不支
援自訂定義域



4-2-2 關聯表實例-屬性值(說明1)

屬性值 (Attribute Values)

- 屬性值是關聯表實際儲存資料的最小單位，在關聯表屬性集合的每一個屬性都擁有一組可接受的值，即屬性的定義域。例如：學生**Students**關聯表，如下圖所示：

Students

<u>sid</u>	name	city	age	GPA
1	江小魚	中和市	25	3
2	劉得華	桃園市	26	3.5
3	郭富成	台中市	30	4
4	張學有	高雄市	21	1

4-2-2 關聯表實例-屬性值(說明2)

- 關聯表實例的屬性值集合（**Attribute Value Set**）是指目前關聯表實例各屬性所包含的值，如下所示：
city屬性值 = { '中和市', '桃園市', '台中市', '高雄市' }
age屬性值 = 21~30
GPA屬性值 = 1~4.0
- 屬性值集合可以作為定義定義域依據，不過仍然需要參考實際情況，才能定義出可接受值的定義域，例如：**GPA**實際的範圍是0.0~4.0。

4-2-2 關聯表實例-屬性值(特點)

1. **單元值 (Atomic)**：屬性值是不可分割的單元值。
2. **需要指定定義域**：屬性值一定需要指定定義域，而且只有一個定義域，雖然屬性值屬於指定的定義域，但並不表示所有的定義域值都會出現，屬性值集合可能只是定義域的部分集合。
3. **可能為空值**：屬性值可以是空值(見下頁)。

4-2-2 關聯表實例-空值

空值（Null Values）

- 在關聯表的屬性值可能是一個未知或無值的空值，這個值是一個特殊符號，它不是0，也不是空字串。
- 所有定義域都會包含空值。
- 空值並沒有意義，所以不能作為真偽的比較運算。
- 空值有兩類：未知值與不適性

4-2-2 關聯表實例-空值(意義)

1. 未知值（Unknown）

- 1-1. 找不到（Missing）：屬性值存在但是找不到，例如：不知道學生【陳大安】的地址，因為address屬性值一定存在只是找不到，所以這是一個找不到的空值，如下圖所示：

Students

<u>sid</u>	name	address	age	GPA
1	江小魚	中和市景平路1000號	25	3
2	劉得華	桃園市三民路10號	26	3.5
3	郭富成	台中市中港路5號	30	4
4	陳大安	NULL	21	1

4-2-2 關聯表實例-空值(意義) (續)

- 1-2. 完全未知 (Total Unknown) : 不知道屬性值是否存在。例如：不知道張先生是否有配偶，所以配偶欄spouse是完全未知的空值，如下圖所示：

Label

<u>id</u>	name	spouse	age
1	張先生	NULL	25
2	劉先生	江小姐	30

4-2-2 關聯表實例-空值(意義) (續)

2. 不適性 (Not applicable)

- 不適性空值是指屬性沒有適合的屬性值。
例如：公司員工劉先生沒有手機，所以 **cellphone** 屬性值的手機號碼是一個不適性的空值，如下圖所示：

Employees

<u>id</u>	name	cell phone	age
1	張先生	0938-000123	25
2	劉先生	NULL	30

4-2-3 關聯表的特性

- 關聯表的名稱是唯一的，在資料庫不能有兩個關聯表擁有相同名稱。同一個關聯表的屬性名稱也是唯一，不過不同關聯表之間允許擁有相同名稱的屬性。
- 在關聯表中除了名稱唯一性的特性外，還有四個特性：
 1. 沒有重複的值組
 2. 值組是沒有順序
 3. 屬性也沒有順序
 4. 所有的屬性值都是單元值

第一正規化
(ch7)

4-2-3 關聯表的特性(續)

1. 沒有重複的值組

- 關聯表是數學集合，在集合中不可有重複元素，所以關聯表沒有重複的值組
- 換個角度來說，其隱含的意義是指關聯表的主鍵。因為主鍵可以是值組的識別，沒有2個值組是完全相同的，故沒有重複的值組

4-2-3 關聯表的特性(續)

2. 值組是沒有順序

- 在關聯表的值組因為是集合，所以並沒有順序分別
- 也就是說，如果重新排列關聯表的值組，也不會產生新的關聯表。

4-2-3 關聯表的特性(續)

3. 屬性也沒有順序

- 在關聯表的屬性也沒有順序差別，如果重新排列關聯表的屬性，也不會產生新的關聯表。
- 但事實上，大部分資料庫管理系統並不支援此特性，資料庫管理系統提供的資料庫存取函式庫，不但可以取得屬性的原始順序，而且允許使用順序存取屬性值。

4-2-3 關聯表的特性(續)

4. 所有屬性值都是單元值

- 關聯表的屬性值都是「單元值」(Atomic)，這是指二維表格中的每一個儲存格的值都是單一值，而不是一組值的集合。

4-2-4 關聯表的種類

- 具名關聯表（**Named Relations**）：在資料庫管理系統使用CREATE TABLE/VIEW和SNAPSHOT指令建立的關聯表。
- 基底關聯表（**Base Relations**）：一種具名關聯表，也稱為「真實關聯表」（**Real Relations**）。
- 導出關聯表（**Derived Relations**）：這是由其他具名關聯表，經過運算而得的具名關聯表。
- 視界（**Views**）：定義在其他基底關聯表之上的一種「虛擬關聯表」（**Virtual Relations**）。

4-2-4 關聯表的種類(續)

- 快照關聯表（**Snapshots**）：這是具名的導出關聯表，在一個時間點的關聯表內容，不過它有真正的儲存資料，屬於一種唯讀關聯表。
- 查詢結果（**Query Results**）：一種沒有具名的導出關聯表。
- 中間結果（**Intermediate Results**）：一種沒有具名的導出關聯表，這是JOIN合併查詢指令中子查詢的查詢結果。
- 暫存關聯表（**Temporary Relations**）：這是一種具名關聯表，儲存的是資料庫管理系統暫時所需的資料。
- 儲存型關聯表（**Stored Relations**）：實際儲存在儲存裝置的關聯表，這是真正可儲存資料的關聯表。

4-2-4 關聯表的種類-重新整理

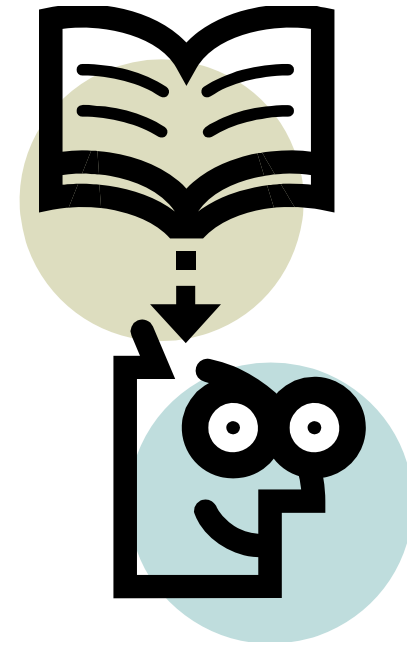
- 基底關聯表（Base Relations）：一種具名關聯表，也稱為「真實關聯表」（Real Relations）(table)，真正存放資料的關聯表。
- 導出關聯表（Derived Relations）：由其它具名關聯表(基底或其它種類的)，經過運算而得的具名關聯表。
 1. 視界（Views）：定義在其它基底關聯表之上的「虛擬關聯表」（Virtual Relations），本身並不包含任何資料，皆為關聯表運算所得(邏輯上的組合)，是具名的導出關聯表。
 2. 快照關聯表（Snapshots）：在某個時間點的關聯表內容，它有真正的儲存資料，屬於一種具名的唯讀關聯表。

4-2-4 關聯表的種類-重新整理(續)

- 具名關聯表（Named Relations）：在資料庫管理系統使用CREATE TABLE/VIEW和SNAPSHOT指令建立的關聯表。
- 不具名關聯表（Unnamed Relations）：
 - 查詢結果（Query Results）：一種沒有具名的導出關聯表。而且只顯示在螢幕上後便捨去，不存起來。
 - 中間結果（Intermediate Results）：一種沒有具名的導出關聯表，這是JOIN合併查詢指令中子查詢的查詢結果。求取查詢結果時所產生的中間結果，它沒有名稱，而且執行後立刻捨去，不存起來。
- 暫存關聯表（Temporary Relations）：這是一種具名關聯表，儲存的是資料庫管理系統暫時所需的資料。
- 儲存型關聯表（Stored Relations）：實際儲存在儲存裝置的關聯表，這是真正可儲存資料的關聯表。

4-3 關聯式資料庫模型的完整性限制條件

- 4-3-1 鍵限制條件
- 4-3-2 定義域限制條件
- 4-3-3 實體完整性
- 4-3-4 參考完整性
- 4-3-5 其他完整性限制條件



4-3 關聯式資料模型的 完整性限制條件

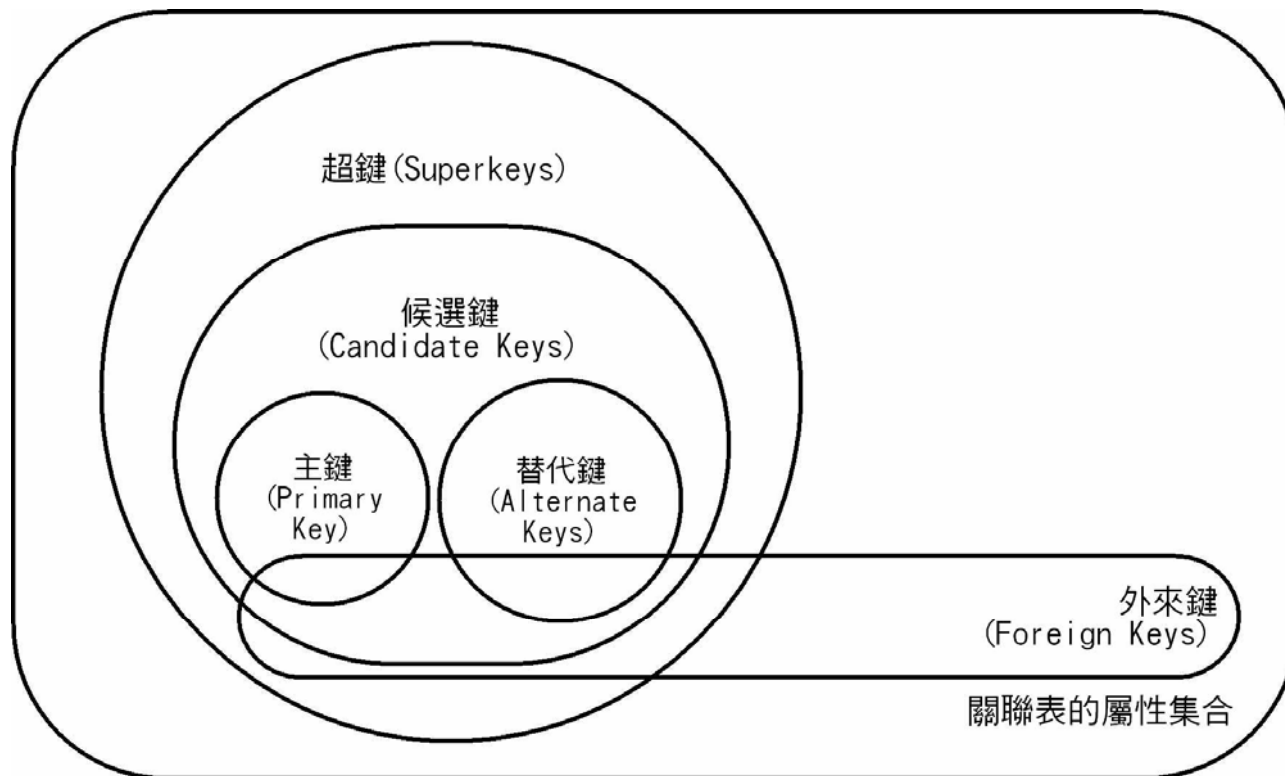
- 關聯式資料庫模型的完整性限制條件（**Integrity Constraints**）是資料庫設計的一部分，其目的是檢查資料庫儲存的資料和保障資料的**正確性**，不但可以防止授權使用者將不合法的資料存入資料庫，還能夠避免關聯表之間的資料**不一致**。

4-3 關聯式資料模型的 完整性限制條件-種類

1. **鍵限制條件 (Key Constraints)**：關聯表一定擁有一個唯一和最小的主鍵 (Primary Key)。
2. **定義域限制條件 (Domain Constraints)**：關聯表的屬性值一定是屬於定義域的單元值。
3. **實體完整性 (Entity Integrity)**：關聯表的主鍵不可以是空值。
4. **參考完整性 (Referential Integrity)**：關聯表的所有外來鍵值，都必須能參考到另一關聯表的主鍵值。

4-3-1 鍵限制條件

- 關聯式資料庫模型的鍵是一個重要觀念，關聯表的「鍵」（**Keys**）是指關聯表綱要中單一屬性或一組屬性的集合。



4-3-1 鍵限制條件-超鍵

超鍵 (Superkeys)

- 超鍵是關聯表綱要的單一屬性或一組屬性的集合，超鍵需要滿足唯一性：
 - 唯一性 (**Uniqueness**)：在關聯表絕不會有2個值組擁有相同的超鍵的值。
- 換句話說，只需透過超鍵的識別，就可以在關聯表存取指定的值組。

4-3-1 鍵限制條件-候選鍵

候選鍵 (Candidate Keys)

- 在每一個關聯表至少擁有一個候選鍵，候選鍵是一個超鍵，不只滿足超鍵的唯一性，還需要滿足最小性，如下所示：
 - 最小性 (Minimality)：最小屬性數的超鍵，在超鍵中沒有一個屬性可以刪除，否則將違反唯一性。
- 關聯表的候選鍵需要同時滿足唯一性和最小性，簡單的說，候選鍵是最小屬性數的超鍵
 - 單一屬性的超鍵一定是候選鍵

4-3-1 鍵限制條件-主鍵

主鍵（Primary Key）

- 主鍵（Primary Key, PK）是關聯表各候選鍵中的其中之一，而且只有一個。

4-3-1 鍵限制條件-挑選主鍵

- 在眾多候選鍵中如何挑選主鍵，選擇原則如下：
 - 絕對不是空值（**Not Null**）：候選鍵的屬性值不能是空值，如果是複合鍵，所有屬性都保證不會是空值。
 - 永遠不會改變（**Never Change**）：候選鍵的屬性值永遠不會改變。
 - 本身不是識別值（**Non-identifying Value**）：候選鍵的屬性值本身沒有其他意義。
 - 簡短且簡單的值（**Brevity and Simplicity**）：儘可能選擇單一屬性的候選鍵。

4-3-1 鍵限制條件-替代鍵

替代鍵（ Alternate Keys ）

- 在候選鍵中不是主鍵的其他候選鍵稱為替代鍵，因為這些是可以用來替代主鍵的候選鍵。

練習 (candidate keys, superkeys)

A	B	C	D
1	b	w	3
2	b	x	2
3	a	y	4
4	a	z	4
5	c	x	2

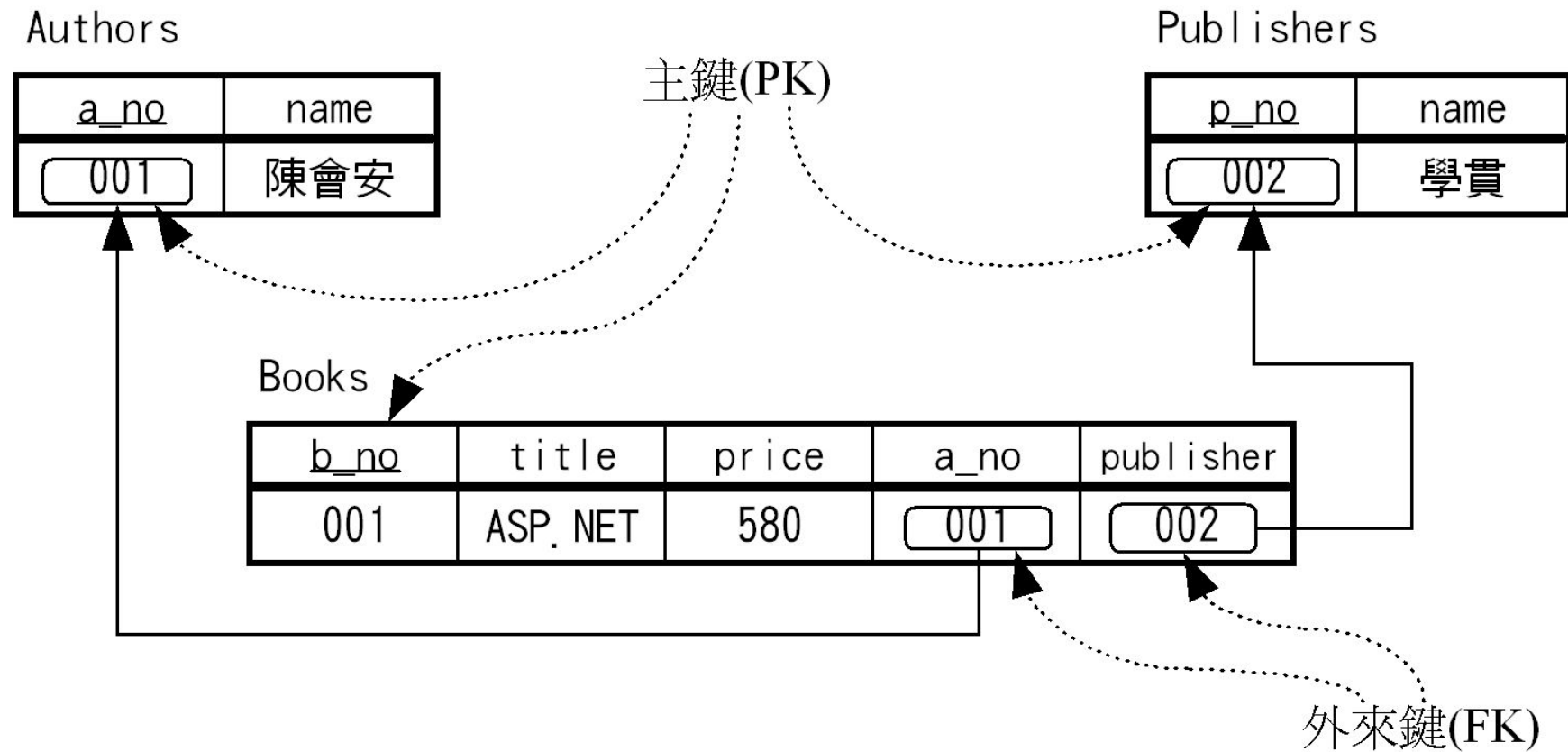
- candidate keys:
 - A, BC
- superkeys:
 - A, AB, AC, AD, ABC, ABD, ACD, ABCD
 - BC, BCD

4-3-1 鍵限制條件-外來鍵

外來鍵（ Foreign Keys ）

- 外來鍵（ Foreign Keys, FK ）是關聯表的單一屬性或一組屬性的集合，它的值是參考其他關聯表的主鍵（或候選鍵），當然也可能參考同一個關聯表的主鍵。
- 外來鍵和其他關聯表的主鍵是對應的，在關聯式資料庫是扮演連結關聯表的膠水功能（連結）。

4-3-1 鍵限制條件-外來鍵(圖例)



4-3-1 鍵限制條件-外來鍵(特性)

1. 外來鍵一定參考其他關聯表的主鍵（或候選鍵），這是兩個關聯表間的連結。
2. 外來鍵在關聯表內不一定是主鍵。
3. 外來鍵和參考的主鍵屬於相同定義域，不過屬性名稱可以不同。
 - 外來鍵和參考主鍵中的主鍵如果是單一屬性，外來鍵就是單一屬性；主鍵是屬性集合，外來鍵一樣也是屬性集合。
4. 外來鍵可以是空值NULL。
5. 外來鍵可以參考同一個關聯表的主鍵。

4-3-2 定義域限制條件

- 定義域限制條件（**Domain Constraints**）是指在關聯表的屬性值一定是**定義域的單元值**（**Atomic**）。
- 例如：屬性**age**的定義域是**int**，故屬性值可以為**5**，但不可以為**4.5**。
- 對比程式語言，就是一種「強調型態」（**Strongly Typed**）程式語言。

4-3-3 實體完整性

實體完整性（Entity Integrity）

- 實體完整性是指在基底關聯表主鍵的任何部分都不可為空值，其規則如下所示：
 1. 如果主鍵是多個屬性的集合，則任何一個屬性不可為空值，例如：**(ename, cname)**是主鍵，那**ename**屬性不可為空值，**cname**屬性也不可為空值。
 2. 在關聯表只有主鍵不可為空值，替代鍵並不適用。
 3. 實體完整性是針對基底關聯表，從其導出的關聯表並不用遵守。

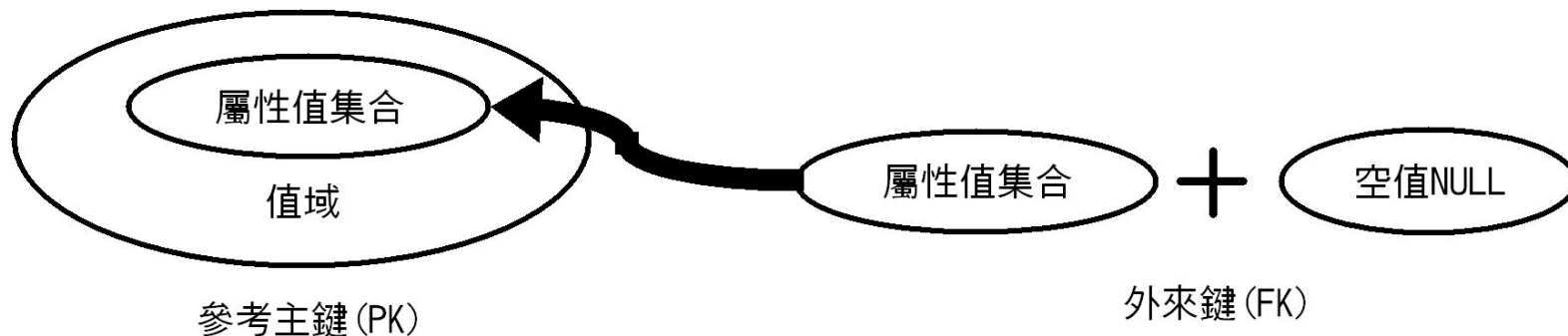
4-3-3 實體完整性-主鍵的使用規則

- 關聯式資料庫管理系統都支援實體完整性，我們可以定義主鍵的更新規則（**Update Rule**）：
 - 在基底關聯表的一個值組更新主鍵或新增值組時，如果主鍵是空值就違反實體完整性，資料庫管理系統必須拒絕這項操作。

4-3-4 參考完整性

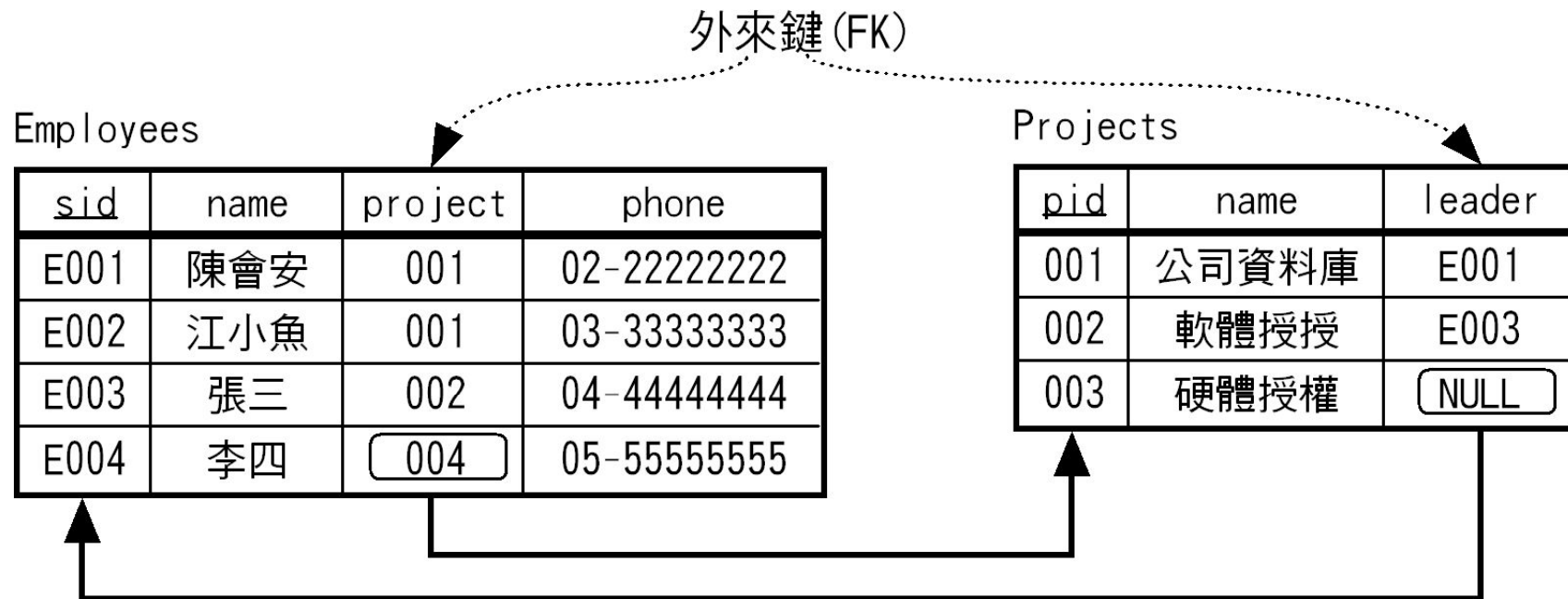
參考完整性（Referential Integrity）

- 參考完整性（Referential Integrity）是當關聯表存在外來鍵時，外來鍵的值一定是來自參考主鍵，或為空值；也就是說，外來鍵的屬性值集合即是參考主鍵的屬性值集合，如下圖所示：



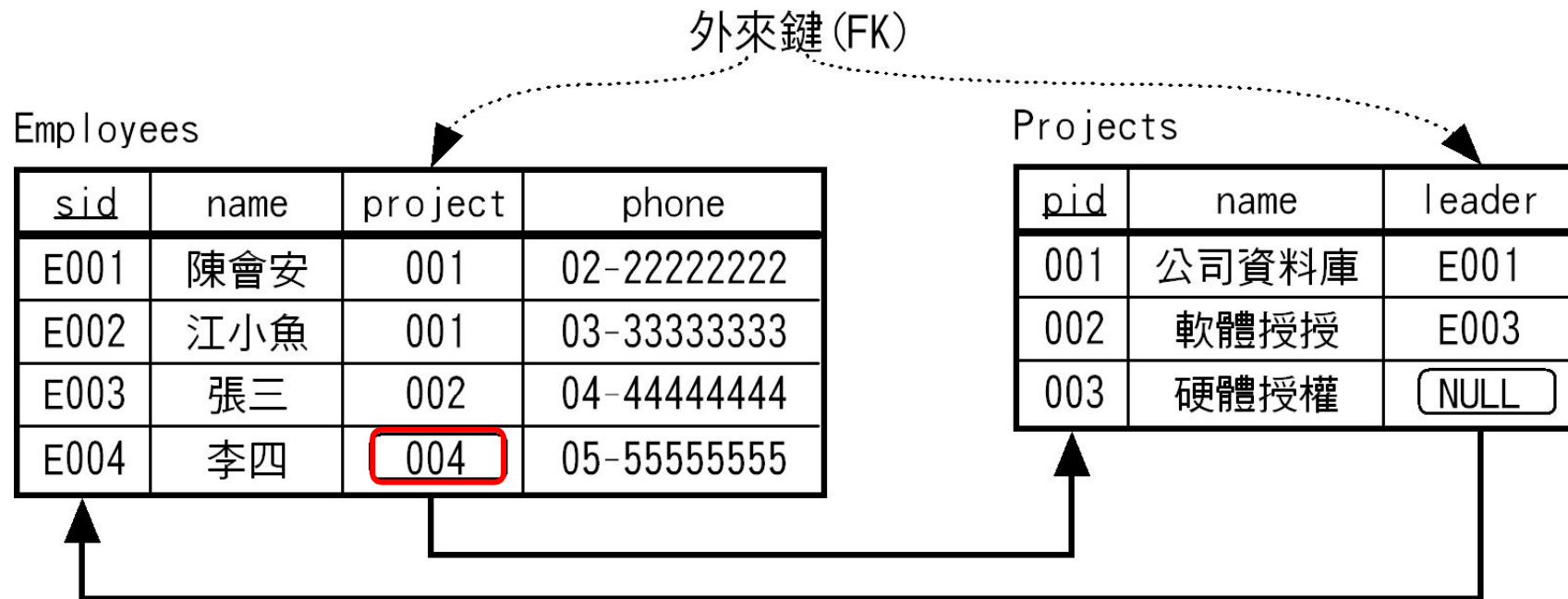
4-3-4 參考完整性-範例

- 例如：公司員工關聯表Employees都會參與公司的專案關聯表Projects，如下圖所示：



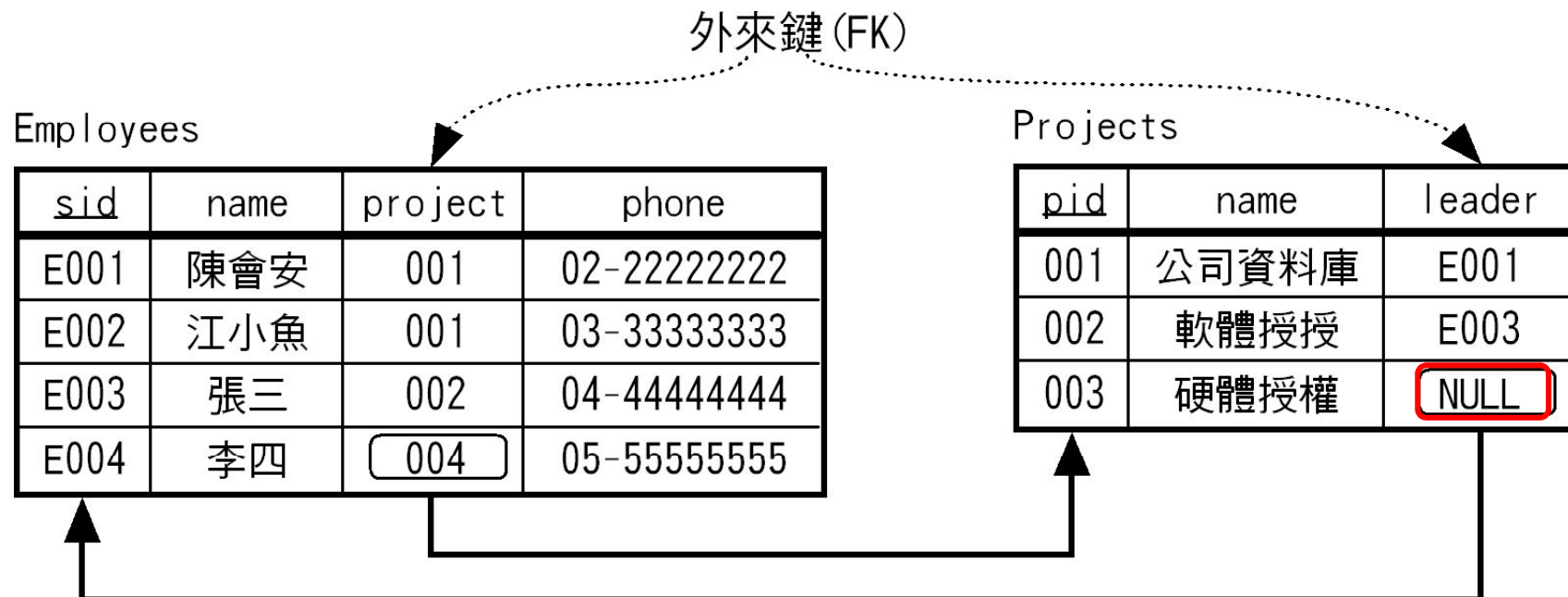
4-3-4 參考完整性-規則

- 在關聯表不可包含無法參考的外來鍵，例如：**並不存在**員工李四的**project**外來鍵值(004) **之**參考主鍵**pid**，表示該值組違反參考完整性



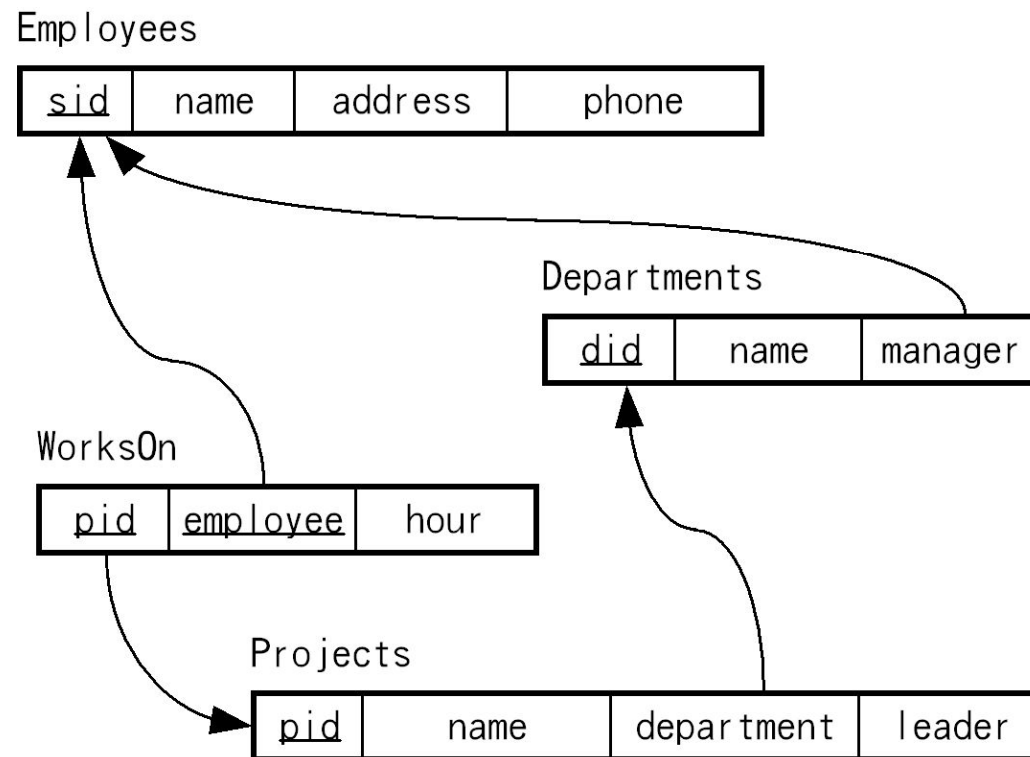
4-3-4 參考完整性-規則

- 如果外來鍵不是關聯表的主鍵，其屬性值可以為空值，例如：**Projects**關聯表pid=003的值組的**leader**外來鍵是空值
 - 因為可能尚未指定專案領導者，並沒有違反參考完整性。



4-3-4 參考完整性-外來鍵參考圖

- 在建立資料庫綱要時，通常我們會使用圖形標示關聯表間的外來鍵關係，稱為「外來鍵參考圖」(Referential Diagram)，如下圖所示：



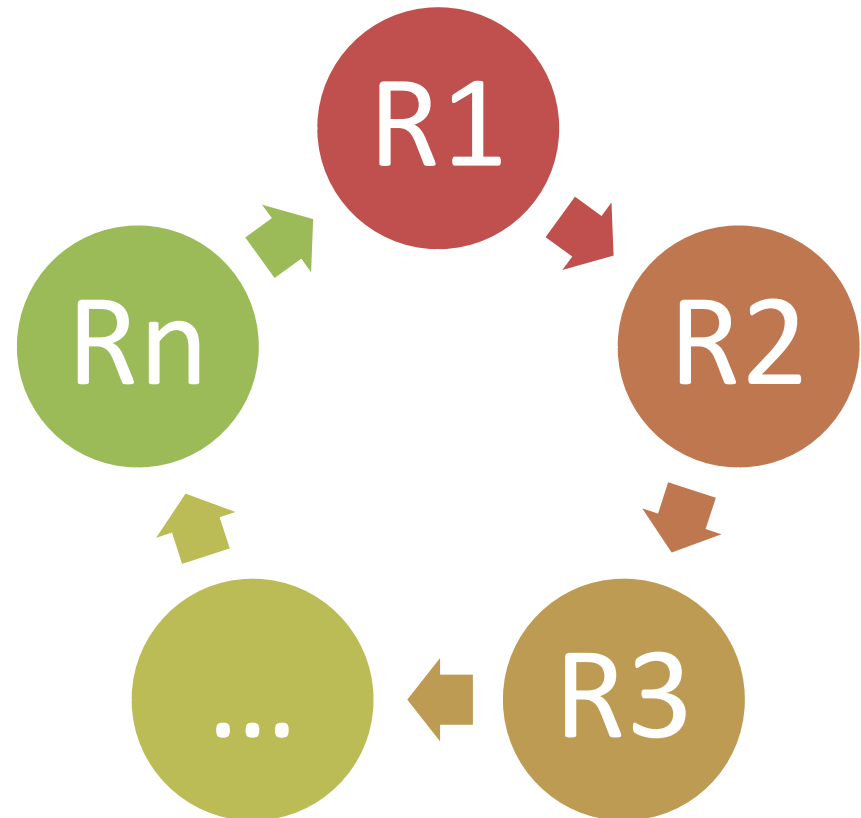
4-3-4 參考完整性-外來鍵參考鏈

- 關聯表的外來鍵是參考其他關聯表的主鍵；被參考的關聯表也可能擁有其他的外來鍵，參考其他關聯表。
- 如果我們將這些外來鍵的參考關係依序繪出，就可以建立「外來鍵參考鏈」
（Referential Chain），如下所示：

Projects → Departments → Employees

4-3-4 參考完整性-外來鍵參考環

- 如果外來鍵參考最後回到原關聯表，稱為「外來鍵參考環」(Referential Cycle)。
- 例如：關聯表R1的參考鏈最後又回到R1。其外來鍵參考環，如右所示：



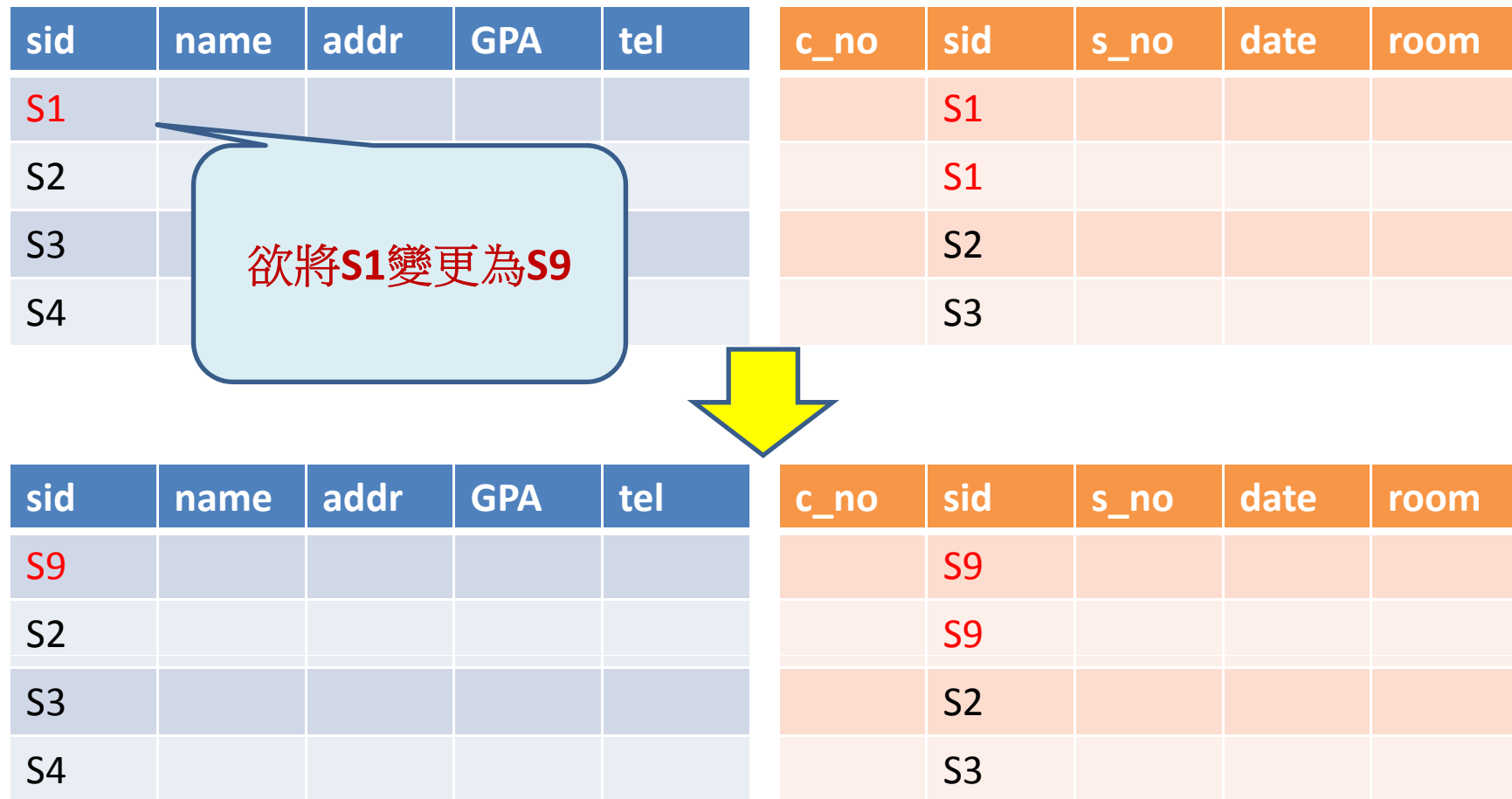
4-3-4 參考完整性-外來鍵使用規則

- 外來鍵的更新規則（**Update Rule**）：如果一個值組擁有外來鍵，當合法使用者試圖在更新或新增值組時，更改外來鍵的值，資料庫管理系統會有如何處理？
- 外來鍵的刪除規則（**Delete Rule**）：如果一個值組擁有外來鍵，當合法使用者試圖刪除參考的主鍵時，資料庫管理系統會怎麼處理？
- 一般有三種方式...下頁投影片

4-3-4 參考完整性-處理方式

1. 限制性處理方式（**Restricted**）：拒絕刪除或更新操作。
2. 連鎖性處理方式（**Cascades**）：連鎖性處理方式是當更新或刪除時，需要作用在所有影響的外來鍵，否則拒絕此操作。
3. 空值化處理方式（**Nullifies**）：將所有可能的外來鍵都設為空值，否則拒絕此操作。

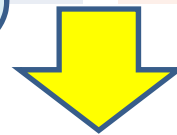
連鎖性處理方式（Cascades）



空值化處理方式（Nullifies）

sid	name	addr	GPA	tel
S1				
S2				
S3				
S4				

欲將S1該筆紀錄刪除



sid	name	addr	GPA	tel
S2				
S3				
S4				

c_no	sid	s_no	date	room
	S1			
	S1			
	S2			
	S3			

c_no	sid	s_no	date	room
	NULL			
	NULL			
	S2			
	S3			

4-3-5 其他完整性限制條件

- 資料庫管理師除了前述的一般完整性限制條件外，還可以依照實際需求，在基底關聯表的屬性新增額外的完整性限制條件。
 - － 通常所有導出關聯表也會繼承在基底關聯表設定的完整性條件。這些額外條件是在關聯表新增、刪除和更新資料時，觸發的一些額外檢查條件。
- 「語意完整性」（Semantic Integrity）屬於特定資料庫管理系統支援的完整性條件，這是屬性內容的一些限制條件，可以檢查關聯表值組的屬性是否為合法資料。

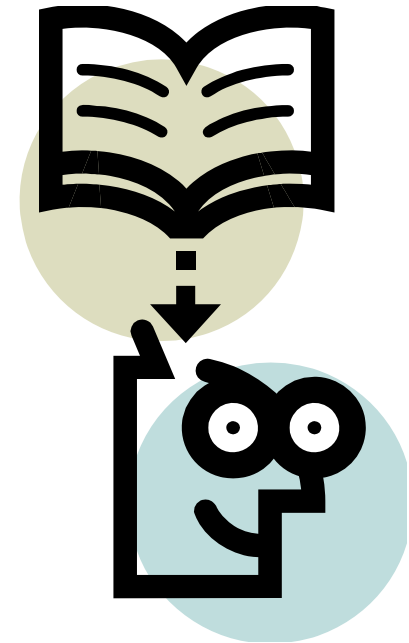
4-3-5 其他完整性限制條件-

語意完整性

1. 空值限制條件（**Null Constraint**）：限制屬性值不可為空值，也就是說，這個屬性一定要輸入資料。
2. 預設值（**Default Value**）：如果沒有輸入指定的屬性值，值組的屬性預設會填入指定的資料，其主要的目的是避免屬性為空值。
3. 檢查限制條件（**Check Constraint**）：這是布林值的邏輯運算式，輸入的屬性值一定需要滿足運算式，即邏輯運算式為真（**true**）。

4-4 關聯式資料模型的數學理論

- 4-4-1 集合的基礎
- 4-4-2 集合運算
- 4-4-3 關聯表的定義
- 4-4-4 鍵的定義



4-4-1 集合的基礎-說明

- 關聯式資料庫的數學基礎就是集合論(set theory)
- 集合（**Set**）是任何東西組合成的群體，集合本身並不考量集合中元素是否有任何關聯，或是集合有何特殊用途，因為集合是由包含的元素來完全決定之。
- 例如：沒有任何元素的集合，稱為空集合（**Empty Set**）；元素都是奇數，就是奇數的集合。

4-4-1 集合的基礎-集合表示法

- 集合的表示方法有很多種，本書主要是使用逐一列舉法，使用大括號括起集合的元素，如下所示：

$A = \{ 1, 2, 3, 4, 5, 6, 7 \}$

$B = \{ a, b, c, d, e, f, g \}$

$C = \{ 001, \text{'陳會安'}, 1966/09/05, 20 \}$

- 如果集合沒有任何元素稱為空集合（Empty set），我們使用 \emptyset 符號來表示。

4-4-1 集合的基礎-集合關係

- 屬於「 \in 」：如果A是集合，x是集合中的元素，我們稱x是集合A的元素，屬於集合A，寫成： $x \in A$ 。
- 不屬於「 \notin 」：如果b不是集合A的元素，表示b不屬於A，寫成： $b \notin A$ 。
- 包含於「 \subseteq 」或「 \supseteq 」：如果A和B是兩個集合，集合A的每一個元素都是集合B的元素，即元素 $x \in A \rightarrow x \in B$ ，我們可以說A是B的「子集」（subset），寫成： $A \subseteq B$ 或 $B \supseteq A$ 。

4-4-1 集合的基礎-集合關係(續)

- 包含於且不相等「 \subset 」或「 \supset 」：如果A和B是兩個集合，集合A的每一個元素都是集合B的元素，且 $A \neq B$ ，稱A是B的**真子集**。
- 等於「 $=$ 」：如果A和B兩個集合的元素完全相同，即A等於B，寫成 $A=B$ ，也就是證明： $A \subseteq B$ 且 $B \subseteq A$ 。

4-4-2 集合運算-聯集（Union）

- 若A和B是兩個集合，將A的所有元素和B的所有元素合起來的集合，稱為A與B的聯集，寫成：

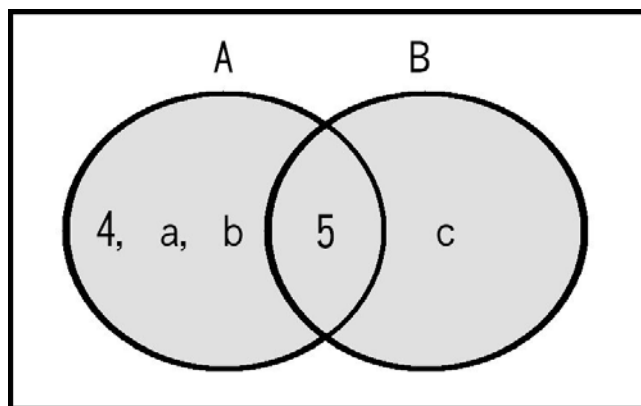
$A \cup B$ 。例如：

$$A = \{4, 5, a, b\}$$

$$B = \{c, 5\}$$

$$A \cup B = \{4, 5, a, b\} \cup \{c, 5\} = \{4, 5, a, b, c\}$$

- 文氏圖（Venn Diagram）表示的 $A \cup B$ ，如下示：



4-4-2 集合運算-聯集(續)

- 交換律： $A \cup B = B \cup A$
- 結合律： $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$
- 包含性： $C \subseteq C \cup D$ ，亦即C一定是 $C \cup D$ 聯集的子集
- 吸收律：若 $C \subseteq D \rightarrow C \cup D = D$ ，因為C是D的子集，則C與D的聯集就是D

4-4-2 集合運算-交集（Intersection）

- 若A和B是兩個集合，將A和B的共同元素合起來的集合，稱為A與B的交集，寫成：

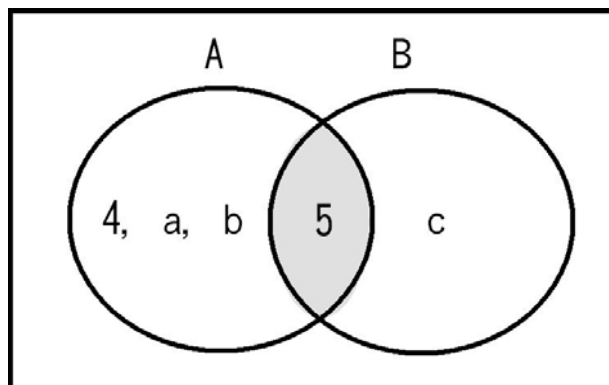
$A \cap B = \{ x \mid x \in A \text{ 且 } x \in B \}$ 。例如：

$$A = \{ 4, 5, a, b \}$$

$$B = \{ c, 5 \}$$

$$A \cap B = \{ 4, 5, a, b \} \cap \{ c, 5 \} = \{ 5 \}$$

- 文氏圖（Venn Diagram）表示的 $A \cap B$ ，如下：



4-4-2 集合運算-交集 (續)

- 交換律： $A \cap B = B \cap A$ 。
- 結合律： $(A \cap B) \cap C = A \cap (B \cap C) = A \cap B \cap C$ 。
- 包含性： $C \cap D \subseteq C$ ，亦即C與D的交集一定是C的子集。
- 吸收律：若 $C \subseteq D \rightarrow C \cap D = C$ ，因為C是D的子集，則C與D的交集就是子集C。

4-4-2 集合運算-差集（Difference）

- 若A和B是兩個集合，所有屬於A但不屬於B的元素所成的集合，稱為A與B的差集，寫成：

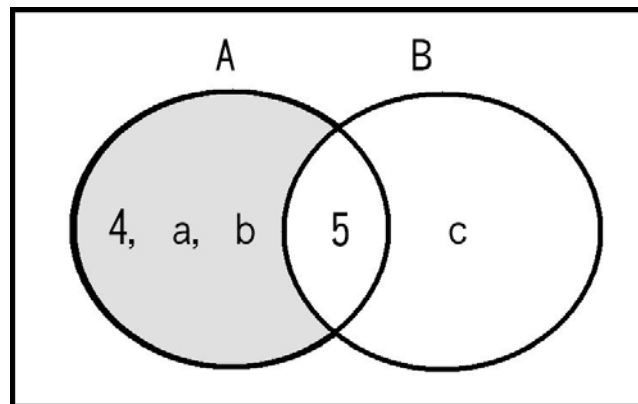
$A - B = \{x \mid x \in A \text{ 且 } x \notin B\}$ 。例如：

$$A = \{4, 5, a, b\}$$

$$B = \{c, 5\}$$

$$A - B = \{4, 5, a, b\} - \{c, 5\} = \{4, a, b\}$$

- 文氏圖（Venn Diagram）表示的 $A - B$ ，如下：



4-4-2 集合運算-卡笛生乘積 (Cartesian Product)

- 若集合 $A_1, A_2, A_3, \dots, A_n$ ，卡笛生乘積是所有可能集合元素的集合， $C = A_1 \times A_2 \times A_3 \times \dots \times A_n$ 。例如：

$A = \{a, b, c\}$

$B = \{1, 2\}$

則

$C = A \times B = \{a, b, c\} \times \{1, 2\}$

$= \{(a,1), (a,2), (b,1), (b,2), (c,1), (c,2)\}$

A		B		C
a	x	1	=	a 1
b		2		a 2
c				b 1
				b 2
				c 1
				c 2

4-4-3 關聯表的定義-定義域

定義4.1：定義域（**Domain**）是一些指定型態值的集合，在關聯表屬性**A**的定義域，寫成：**dom(A)**。

- 一些定義域的範例，如下所示：

正整數 = { 1, 2, 3, 4, 5, ... }

英文小寫字母 = { a, b, c, d, ..., z }

4-4-3 關聯表的定義

定義4.2：關聯表（**Relations**）是各屬性定義域卡笛生乘積的子集。若關聯表**R**擁有 $A_1, A_2, A_3, \dots, A_n$ 共 n 個屬性，屬性的定義域分別為：

$\text{dom}(A_1), \text{dom}(A_2), \text{dom}(A_3), \dots, \text{dom}(A_n)$

定義域的卡笛生乘積為：

$\text{dom}(A_1) \times \text{dom}(A_2) \times \text{dom}(A_3) \times \dots \times \text{dom}(A_n)$

則，關聯表**R**是定義域卡笛生乘積的子集：

$R \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \text{dom}(A_3) \times \dots \times \text{dom}(A_n)$

表示關聯表**R**是「 n 值組」（**n-tuples**）的集合 = $\langle v_1, v_2, v_3, \dots, v_n \rangle$ ， $v_i \in \text{dom}(A_i)$ or $v_i = \text{NULL}$ ， $1 \leq i \leq n$ ， n 是關聯表的屬性數，即維度（**Degree**）。

4-4-3 關聯表的定義-範例

- 關聯表Emp1和Emp2擁有2個屬性name和salary，其定義域分別為dom(name)和dom(salary)如下：

$\text{dom}(\text{name}) = \{ \text{'陳會安'}, \text{'江小魚'}, \text{'張三丰'} \}$

$\text{dom}(\text{salary}) = \{ 35000, 40000 \}$

$\text{dom}(\text{name}) \times \text{dom}(\text{salary}) = \{ \langle \text{'陳會安'}, 35000 \rangle, \langle \text{'陳會安'}, 40000 \rangle, \langle \text{'江小魚'}, 35000 \rangle, \langle \text{'江小魚'}, 40000 \rangle, \langle \text{'張三丰'}, 35000 \rangle, \langle \text{'張三丰'}, 40000 \rangle \}$

- 關聯表Emp1和Emp2如下：

$\text{Emp1} \subseteq \text{dom}(\text{name}) \times \text{dom}(\text{salary})$

$\text{Emp1} = \{ \langle \text{'陳會安'}, 40000 \rangle, \langle \text{'江小魚'}, 35000 \rangle, \langle \text{'張三丰'}, 35000 \rangle \}$

$\text{Emp2} \subseteq \text{dom}(\text{name}) \times \text{dom}(\text{salary})$

$\text{Emp2} = \{ \langle \text{'陳會安'}, 40000 \rangle, \langle \text{'江小魚'}, 40000 \rangle \}$

4-4-3 關聯表的定義-關聯表綱要

定義4.3：關聯表綱要（**Relation Schema**）是屬性的有限元素集合。若關聯表名稱為**R**，包含屬性**A₁, A₂, ..., A_n**與對應的定義域**D₁, D₂, ..., D_n**，則關聯表綱要通常寫成：

R (A₁(D₁), A₂(D₂), ..., A_n(D_n))

或簡寫成：

R (A₁, A₂, ..., A_n)

- 例如：前述**Emp1**關聯表綱要，如下所示：

Emp1 (name, salary)

4-4-3 關聯表的定義-關聯表實例

定義4.4：關聯表實例（**Relation Instance**）是「**n**值組」的集合。若關聯表綱要**RS**包含屬性 **A_1, A_2, \dots, A_n** ，每一筆**n**值組對應的屬性值為 **V_1, V_2, \dots, V_n** ，通常寫成：

$$t = \{ \langle A_1, V_1 \rangle, \langle A_2, V_2 \rangle, \dots, \langle A_n, V_n \rangle \}$$

或簡寫成：

$$t = \langle V_1, V_2, \dots, V_n \rangle$$

則，關聯表綱要**RS**的關聯表實例是 **$r(RS)$** ，寫成：

$$r(RS) = \{ t_1, t_2, \dots, t_m \}$$

m是基數（**Cardinality**），就是實例包含的值組數。

練習

- $r1(R1) = \{a,1,b,2,c,3\}$
- $r2(R2) = \{(a,1),(b,2),(c,3)\}$
- $r3(R3) = \{(a,1,b),(2,c,3)\}$

- 維度(n)=?
- 基數(m)=?

4-4-4 鍵的定義-超鍵

定義4.5：鍵K是關聯表R的超鍵（**Superkey**），則滿足：

(1) $K \subseteq R$ ：K是R的子集。

(2) $\forall t_1, t_2 \in r(R) \ t_1 \neq t_2 \rightarrow t_1[K] \neq t_2[K]$ ：存在 t_1 和 t_2 兩個不同值組，超鍵值 $t_1[K]$ 不等於 $t_2[K]$ ，即滿足**唯一性**。

4-4-4 鍵的定義-候選鍵

定義4.6：鍵**CK**是關聯表**R**的候選鍵（**Candidate Key**），則滿足：

(1) **CK**是超鍵。

(2) **CK**是最小超鍵，沒有其他超鍵**K**是**CK**的真子集：
 $K \subset CK$ ，即滿足最小性。

4-4-4 鍵的定義-外來鍵

定義4.7：鍵FK是關聯表R1的外來鍵（**Foreign Key**），則FK是R1屬性的子集，如下所示：

$FK = \{A_1, A_2, \dots, A_n\}$ FK是R1的子集： $FK \subseteq R1$

且滿足：

- (1) FK與關聯表R2的主鍵擁有相同定義域，即FK參考關聯表R2。
- (2) FK的值是參考關聯表R2主鍵的值，或是空值。

4-5 關聯式資料庫

- 關聯式資料庫（Relational Database）是一種使用關聯式資料模型的資料庫，它是由多個已正規化的關聯表所組成。
- 在關聯表間使用外來鍵與參考主鍵的資料值來建立連結，以便實作一對一、一對多和多對多的關聯性。
- 如果關聯式資料庫只擁有一個關聯表，也是一種合法的資料庫。
- 關聯式資料庫綱要（Relational Database Schema）是一組關聯表綱要和各綱要附屬的完整性限制條件。

4-5 關聯式資料庫-關聯表綱要

- 學校選課與排課系統資料庫是一個擁有4個關聯表的資料庫，各關聯表綱要如下所示：

Student (sid, name, birthday, GPA, tel)

Instructors (eid, name, rank, department)

Courses (c_no, title, credits)

Classes (eid, sid, c_no, time, room)

4-5 關聯式資料庫-關聯表圖例

Students

<u>sid</u>	name	birthday	GPA	tel
S001	陳會安	1967/9/3	3.7	02-22222222
S002	江小魚	1978/2/2	3.0	03-33333333
S003	張三丰	1982/3/3	3.2	04-44444444
S004	李四方	1981/4/4	2.9	05-55555555

Instructors

<u>eid</u>	name	rank	department
E001	陳慶新	教授	CS
E002	楊金欉	副教授	CIS
E003	李鴻章	講師	MATH

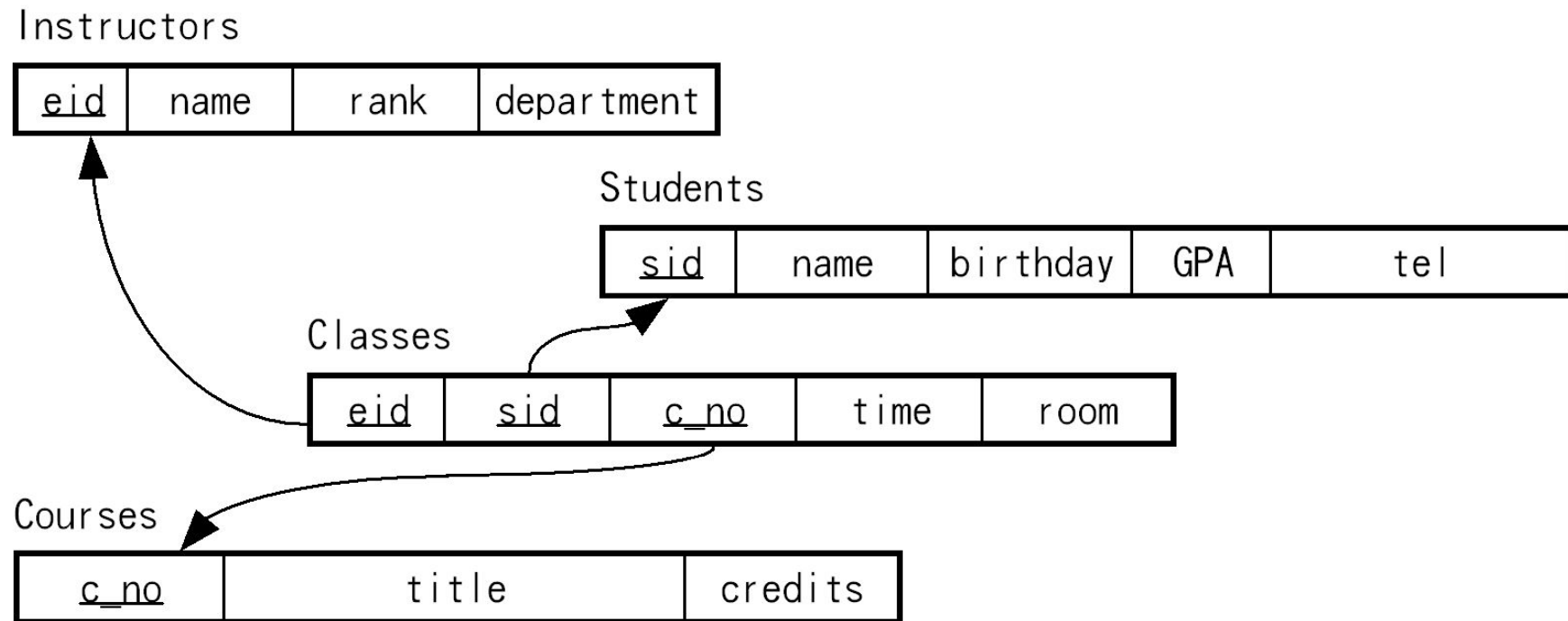
Courses

<u>c_no</u>	title	credits
CS101	計算機概論	4
CS121	離散數學	4
CS203	程式語言	3
CS222	資料庫管理系統	3
CS213	物件導向程式設計	2

Classes

<u>eid</u>	<u>sid</u>	<u>c_no</u>	time	room
E001	S001	CS101	12:00pm	180-M
E002	S003	CS121	8:00am	221-S
E003	S001	CS203	10:00am	221-S
E003	S002	CS203	14:00pm	327-S
E002	S001	CS222	13:00pm	100-M
E002	S002	CS222	13:00pm	100-M
E002	S004	CS222	13:00pm	100-M
E001	S003	CS213	9:00am	622-G
E003	S001	CS213	12:00pm	500-K

4-5 關聯式資料庫-外來鍵參考圖



練習

- 請繪出外來鍵參考圖：
- Professor (id, name, address, tel)
- Teach (c_no, id, day)
- Course (c_no, title, credit)

練習

Professor

<u>id</u>	name	address	tel
123	Mary	NY	000000
345	John	NJ	111111
456	Tim	GS	222222

Course

<u>c_no</u>	title	credit
DS	資料庫	3
PG	程式語言	3
OO	物件導向	3

哪些違反鍵限制條件、實體完整性、參考完整性？

Teach

<u>c_no</u>	<u>id</u>	day
DS	345	一
PG	345	二
DS	456	NULL
PG	NULL	三
OO	345	NULL
DS	456	六
PHY	456	五
OO	789	五

解答

- 違反鍵限制條件: Teach
 - 值組 $\langle \text{DS}, 456, \text{null} \rangle$ 與 $\langle \text{DS}, 456, \text{六} \rangle$ 之主鍵值重覆
- 違反實體完整性: Teach
 - 值組 $\langle \text{PG}, \text{null}, \text{三} \rangle$ 之主鍵(部份)值包含有null值
- 違反參考完整性: Teach
 - 值組 $\langle \text{PHY}, 456, \text{五} \rangle$ 的PHY
 - 值組 $\langle \text{OO}, 789, \text{五} \rangle$ 的789