



資訊管理學系 陳士杰老師

# 資料庫系統管理

## Database System Management

### ER及EER Model轉換為關聯表格

ER and EER Model Mapping to Related Tables



國立聯合大學  
NATIONAL UNITED UNIVERSITY

# [ ■ Outlines ]

- 如何利用E-R Model來從事正規化
- 將E-R Model對映到關聯
- 轉換細則
- 將EER Model對映到關聯

【講義：Ch. 2, Section 3】

【原文：Ch. 7】

【補充講義 2】

# [ ■ 如何利用E-R Model來從事正規化 ]

- E-R Model  $\Rightarrow$  Normalization
- E-R Model  $\Rightarrow$  Relation  $\Rightarrow$  Normalization (Ch. 4)

# [ ■ 將E-R Model轉換到關聯 ]

Step 1: Mapping of **Regular Entity Types**

Step 2: Mapping of **Weak Entity Types**

Step 3: Mapping of **Binary 1:1 Relationship Types**

Step 4: Mapping of **Binary 1:N Relationship Types**

Step 5: Mapping of **Binary M:N Relationship Types**

Step 6: Mapping of **Multivalued Attributes**

Step 7: Mapping of **N-ary Relationship Types**

## Step 1: Mapping of Regular Entity Types

- 針對一般的個體，為其建立新的關聯表格
- 該關聯表格的Key即為個體的主鍵(Primary Key)
- 該個體的所有一般Attributes均需納入表格中
- 此時不考慮Foreign Key; 關係上的屬性; 衍生屬性; 多值屬性

Step 2: Mapping of Weak Entity Types

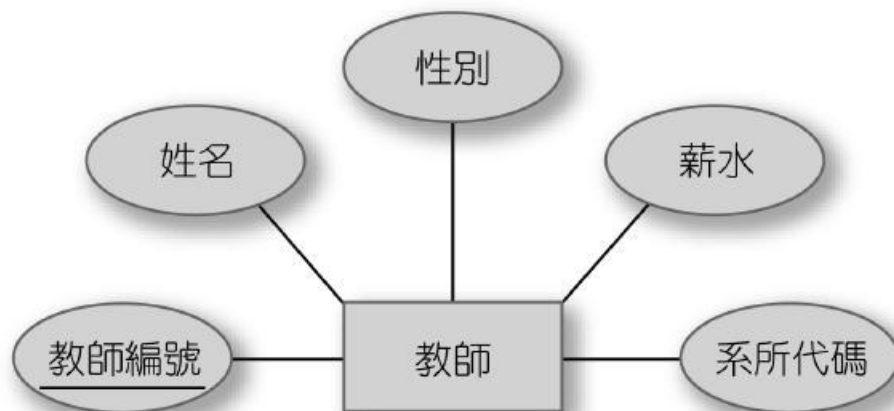
Step 3: Mapping of Binary 1:1 Relationship Types

Step 4: Mapping of Binary 1:N Relationship Types

Step 5: Mapping of Binary M:N Relationship Types

Step 6: Mapping of Multivalued Attributes

Step 7: Mapping of N-ary Relationship Types



教師

<u>教師編號</u>	姓名	性別	薪水	系所代碼
-------------	----	----	----	------

[

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

- 針對弱個體，建立一個新的關聯表格
- Key (所有者) + Key (Weak entity) → 做為此新表格的複合主鍵
- 以Key(所有者)做為外來鍵，以關連到原本所屬的一般個體
- 該弱個體上所有的Attributes均需納入

Step 3: Mapping of Binary 1:1 Relationship Types

Step 4: Mapping of Binary 1:N Relationship Types

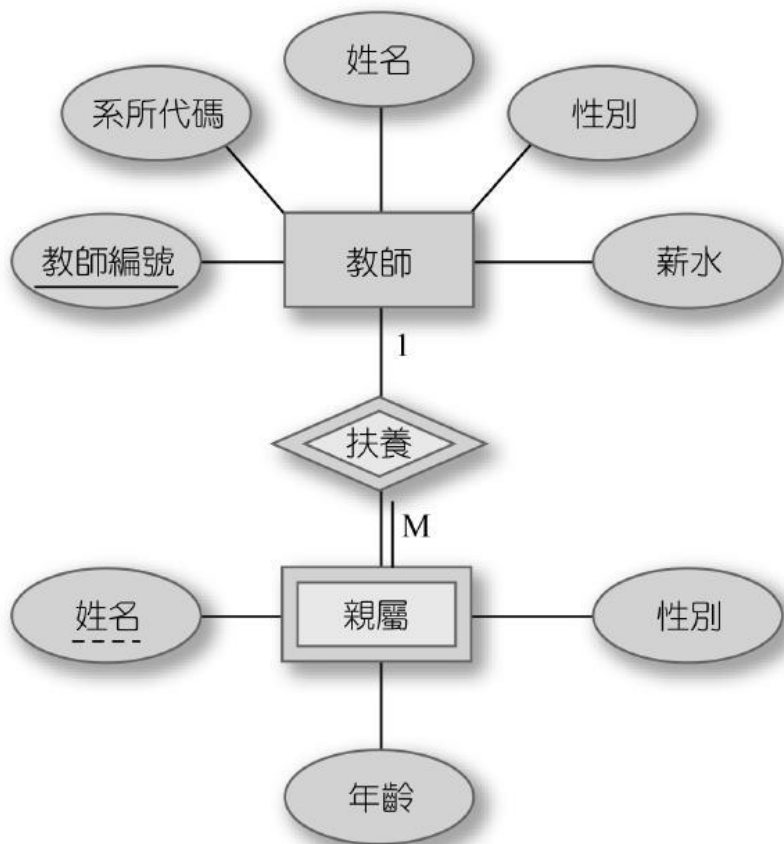
Step 5: Mapping of Binary M:N Relationship Types

Step 6: Mapping of Multivalued Attributes

Step 7: Mapping of N-ary Relationship Types

[

]





Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relationship Types

- 若兩個個體中，只有一個個體為 **Total participation (全部參與)**：
  - 則 **1:1關係上的屬性** 加入到 全部參與的個體所構成的關聯表格 中。
  - 將 **部份參與個體上之主鍵**，加入到全部參與的個體所構成之關聯表格中，並將之設定為 **Foreign Key**。
- 若為其它的狀況：
  - 將 **1:1關係上的屬性** **向左或向右** 轉移到任一個參與的個體型態中。
  - 此個體並加入另一個體的主鍵以做為外來鍵。

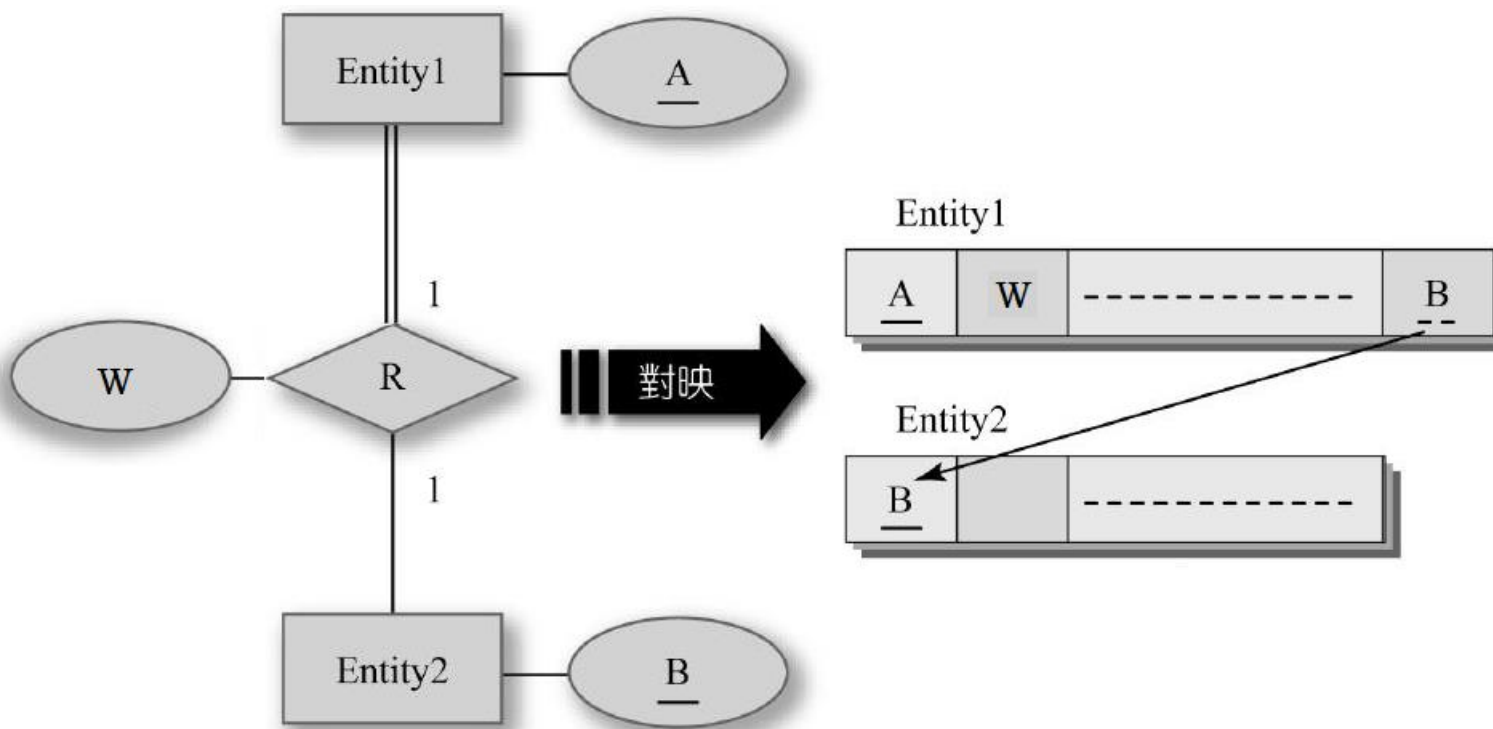
Step 4: Mapping of Binary 1:N Relationship Types

Step 5: Mapping of Binary M:N Relationship Types

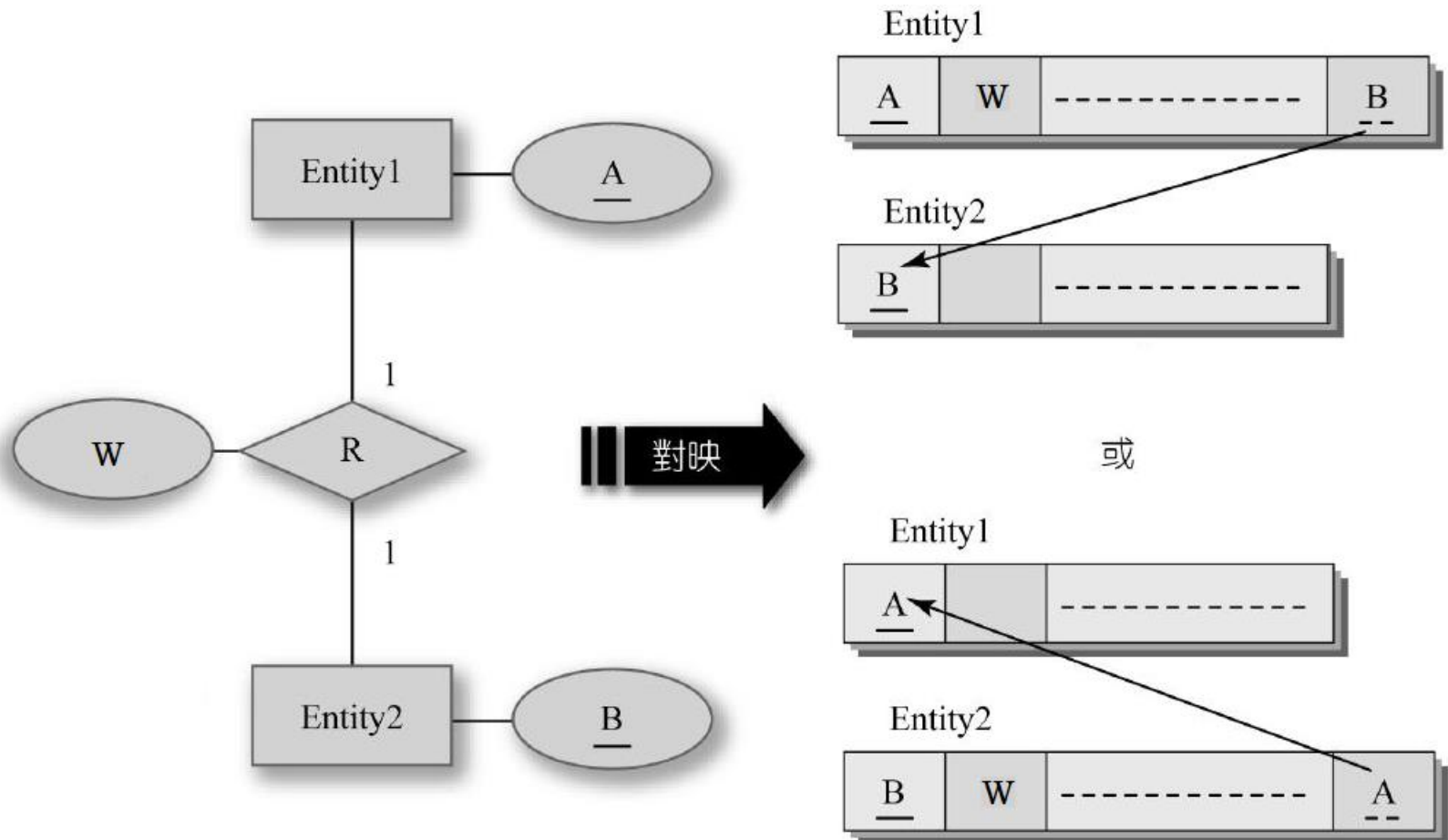
Step 6: Mapping of Multivalued Attributes

Step 7: Mapping of N-ary Relationship Types

- 只有一個個體為 **Total participation (全部參與)** :



## ■ 其它狀況：



[

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relationship Types

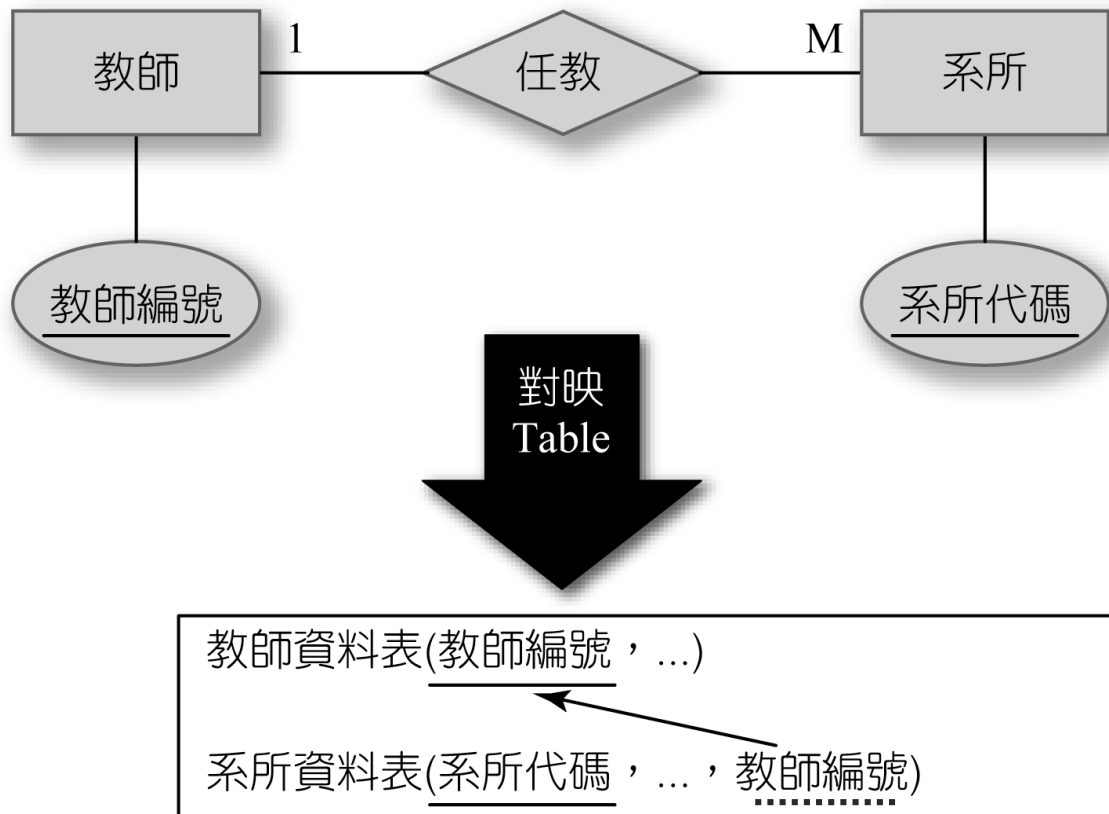
Step 4: Mapping of Binary 1:N Relationship Types

- 針對**Non-Weak 1對多**的關係
- **N端(多端)的個體所產生的表格**中，加入**1端的個體之主鍵**以成為其**Foreign Key**
- 若有關係上的屬性，則將該屬性**加入N端的個體所產生之關聯表格中**

Step 5: Mapping of Binary M:N Relationship Types

Step 6: Mapping of Multivalued Attributes

Step 7: Mapping of N-ary Relationship Types



[

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relationship Types

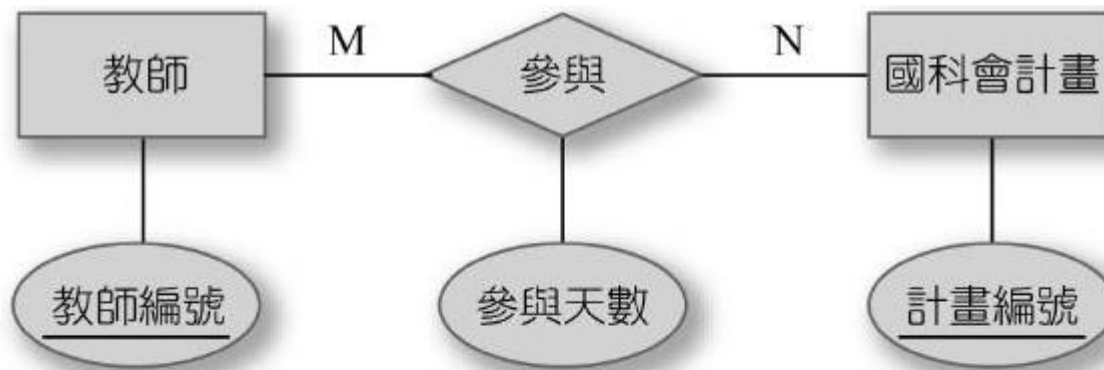
Step 4: Mapping of Binary 1:N Relationship Types

**Step 5: Mapping of Binary M:N Relationship Types**

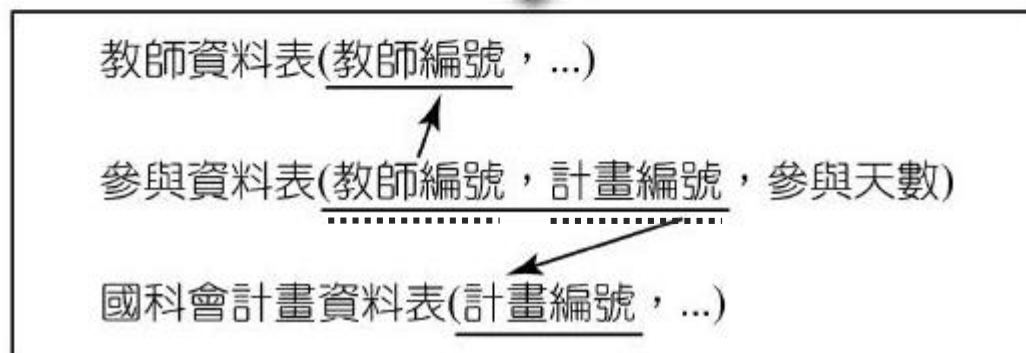
- 針對此多對多的關係，建立一個**新的關聯表格**
- $\text{Key}(\text{M端}) + \text{Key}(\text{N端}) \rightarrow$ 做為此新關聯表格的**複合主鍵**
- $\text{Key}(\text{M端})$  與  $\text{Key}(\text{N端})$ 在此新的表格中，也分別表示為**外來鍵**，以關連到原本所屬的一般個體。

Step 6: Mapping of Multivalued Attributes

Step 7: Mapping of N-ary Relationship Types



對映  
Table



[

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relationship Types

Step 4: Mapping of Binary 1:N Relationship Types

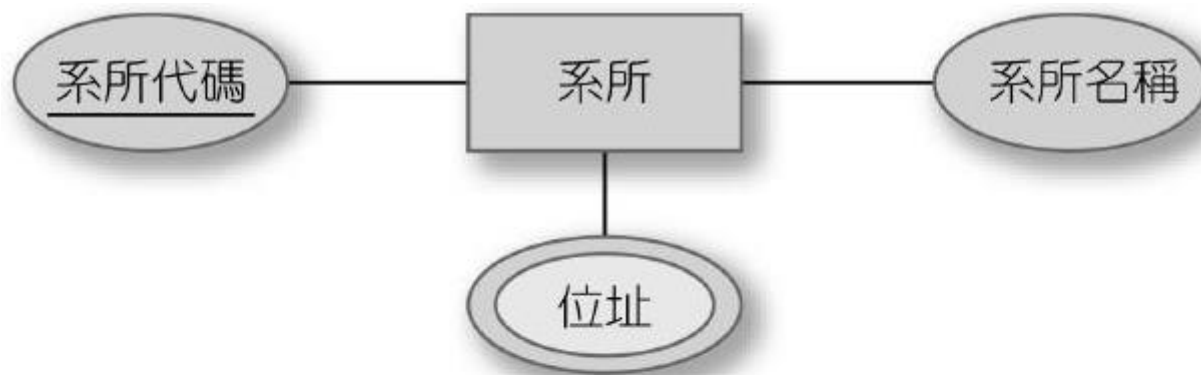
Step 5: Mapping of Binary M:N Relationship Types

**Step 6: Mapping of Multivalued Attributes**

- 針對此多值屬性，建立一個**新的關聯表格**
- **Key(原本所在之個體)+Multivalued Attribute**→做為此新關聯表格的**複合主鍵**
- 以該**Key(原本所在之個體)**為**外來鍵**，以關連到原本所屬的一般個體

Step 7: Mapping of N-ary Relationship Types







Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relationship Types

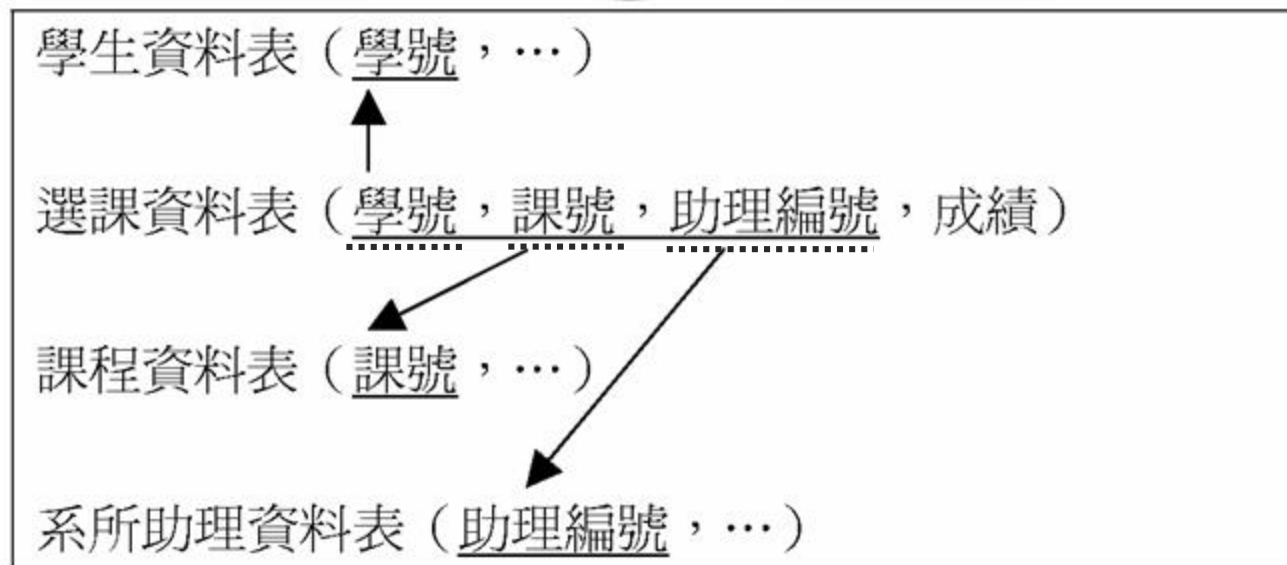
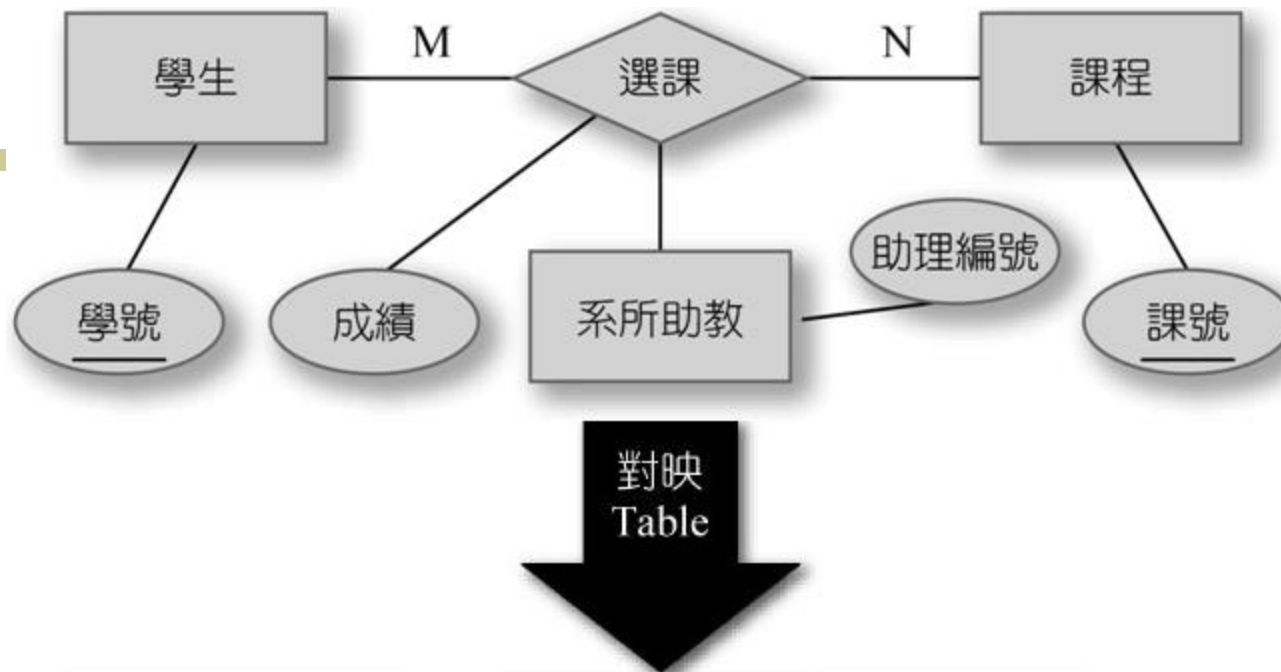
Step 4: Mapping of Binary 1:N Relationship Types

Step 5: Mapping of Binary M:N Relationship Types

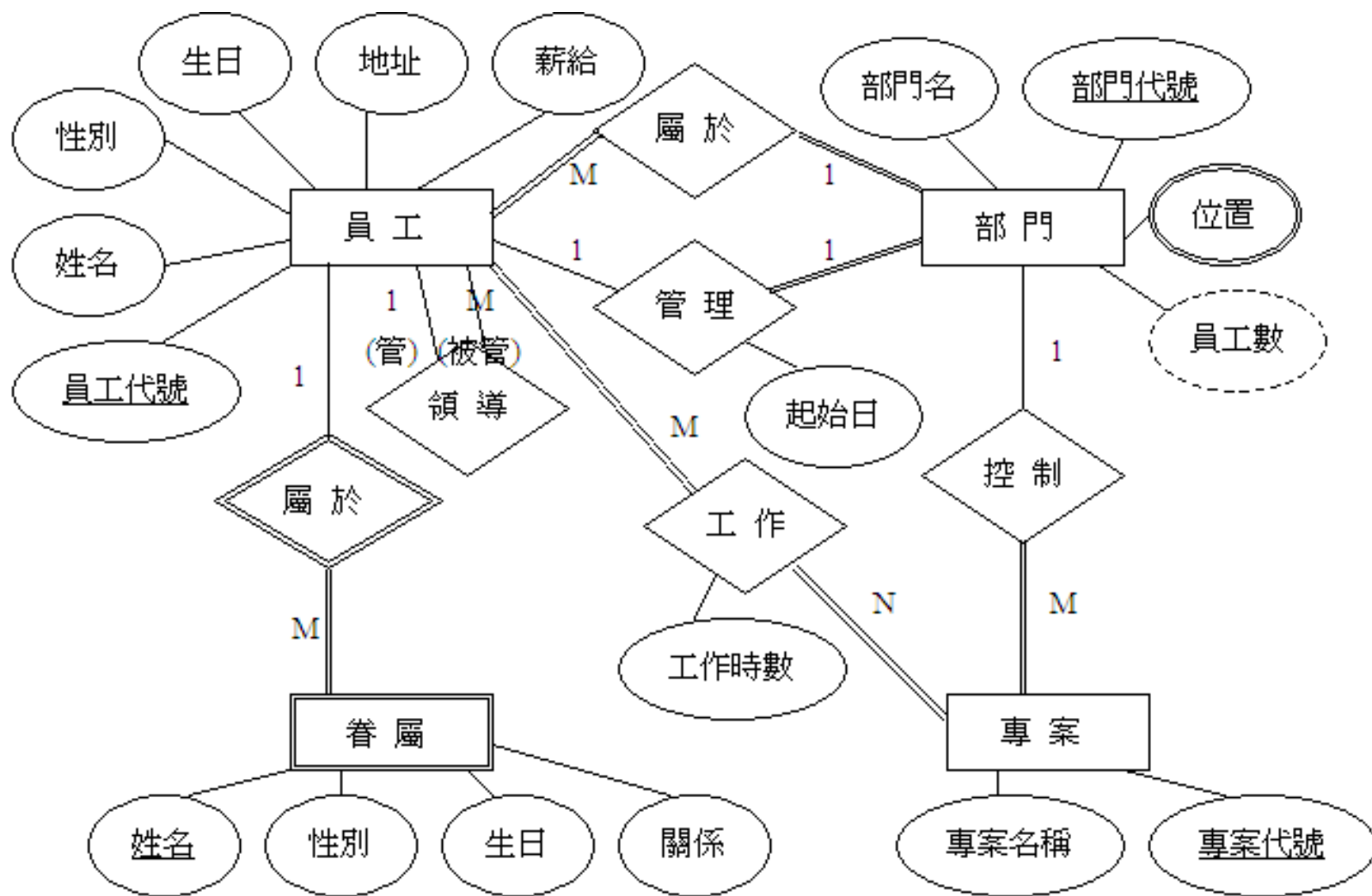
Step 6: Mapping of Multivalued Attributes

**Step 7: Mapping of N-ary Relationship Types**

- 針對此多元關係，建立一個**新的關聯表格**
- 將所有關連到此關係的個體之主鍵，做為此新關聯表格的**複合主鍵**。而通常這些主鍵同時也是**外來鍵**，分別與其所屬的一般個體有所關連。



# 範例 1



## Step 1: Mapping of Regular Entity Types

- 針對一般的個體，為其建立**新的關聯表格**
- 該關聯表格的**Key**即為個體的**主鍵(Primary Key)**
- 該個體的所有一般Attributes均需納入表格中
- 此時不考慮Foreign Key; 關係上的屬性; 衍生屬性; 多值屬性

### Step 1：員工表

<u>員工代號</u>	姓名	出生日	性別	地址	薪給
-------------	----	-----	----	----	----

### 部門表

<u>部門代號</u>	部門名
-------------	-----

### 專案表

<u>專案代號</u>	專案名稱
-------------	------

## Step 2: Mapping of Weak Entity Types

- 針對弱個體，建立一個新的關聯表格
- Key (所有者) + Key (Weak entity) → 做為此新表格的複合主鍵
- 以Key(所有者)做為外來鍵，以關連到原本所屬的一般個體
- 該弱個體上所有的Attributes均需納入

Step2 : 眷屬表

<u>員工代號</u>	眷屬姓名	性別	出生日	關係
-------------	------	----	-----	----

### Step 3: Mapping of Binary 1:1 Relationship Types

- 若兩個個體中，只有一個個體為**Total participation (全部參與)**：
  - 則**1:1關係上的屬性**加入到**全部參與的個體所構成的關聯表格**中。
  - 將**部份參與個體上之主鍵**，加入到全部參與的個體所構成之關聯表格中，並將之設定為**Foreign Key**。
- 若為其它的狀況：
  - 將**1:1關係上的屬性****向左或向右**轉移到任一個參與的個體型態中。
  - 此個體並加入另一個體的主鍵以做為外來鍵。

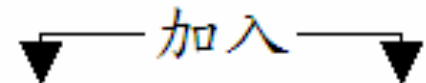
Step3：部門表

部門代號

部門名

員工編號

起始日



## Step 4: Mapping of Binary 1:N Relationship Types

- 針對**Non-Weak 1對多**的關係
- **N端(多端)的個體所產生的表格中**，加入**1端的個體之主鍵**以成為其**Foreign Key**
- 若有關係上的屬性，則將該屬性加入**N端的個體所產生之關聯表格中**

Step4：員工表

員工代號 姓名 出生日 性別 地址 薪給 領導員工代號 部門代號

專案表

專案代號 專案名稱 部門代號

來自員工表

來自部門表

來自部門表



## Step 5: Mapping of Binary M:N Relationship Types

- 針對此多對多的關係，建立一個**新的關聯表格**
- Key(M端) + Key (N端) → 做為此新關聯表格的**複合主鍵**
- Key(M端) 與 Key (N端)在此新的表格中，也分別表示為**外來鍵**，以關連到原本所屬的一般個體。

Step5：專案員工表

員工代號

專案代號

工作時數

## Step 6: Mapping of Multivalued Attributes

- ◆ 針對此多值屬性，建立一個**新的關聯表格**
- ◆ **Key(原本所在之個體)+Multivalued Attribute**→做為此新關聯表格的**複合主鍵**
- ◆ 以該**Key(原本所在之個體)**為**外來鍵**，以關連到原本所屬的一般個體。

Step6：部門位置表

部門代號

部門位置

## ■ 本範例的結果為：

**員工表**(員工代號，姓名，出生日，性別，地址，薪給，部門代號，領導員工代號)

**部門表**(部門代號，部門名，員工代號，起始日)

**眷屬表**(員工代號，眷屬姓名，性別，出生日，關係)

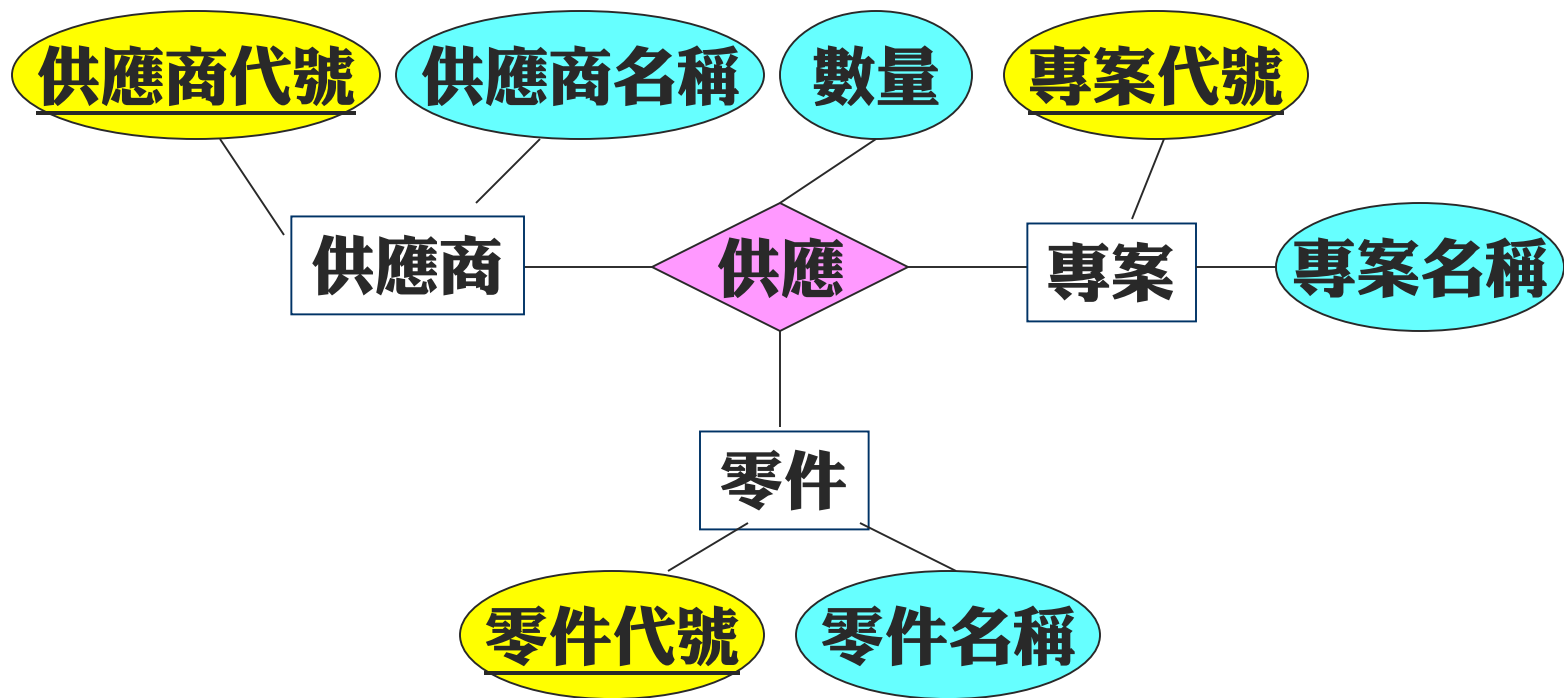
**專案表**(專案代號，專案名稱，部門代號)

**專案員工表**(員工代號，專案代號，工作時數)

**部門位置表**(部門代號，部門位置)

## ■ 以上六個Relations皆符合1NF, 2NF, 3NF, BCNF

## 範例 2



## ■ Step 1: Mapping of Regular Entity Types

- 針對一般的個體，為其建立**新的關聯表格**
- 該關聯表格的**Key**即為個體的**主鍵(Primary Key)**
- 該個體的所有一般Attributes均需納入表格中
- 此時**不考慮**Foreign Key; 關係上的屬性; 衍生屬性; 多值屬性

### Step 1：供應商表

供應商代號      供應商名稱

### 專案表

專案代號      專案名稱

### 零件表

零件代號      零件名稱

## Step 7: Mapping of N-ary Relationship Types

- 針對此多元關係，建立一個**新的關聯表格**
- 將所有關連到此關係的個體之主鍵，做為此新關聯表格的**複合主鍵**。而通常這些主鍵同時也是**外來鍵**，分別與其所屬的一般個體有所關連。

### Step7：供應表

供應商代號

專案代號

零件代號

數量

■ 本範例的結果為：

**供應商表**(供應商代號，供應商名稱)

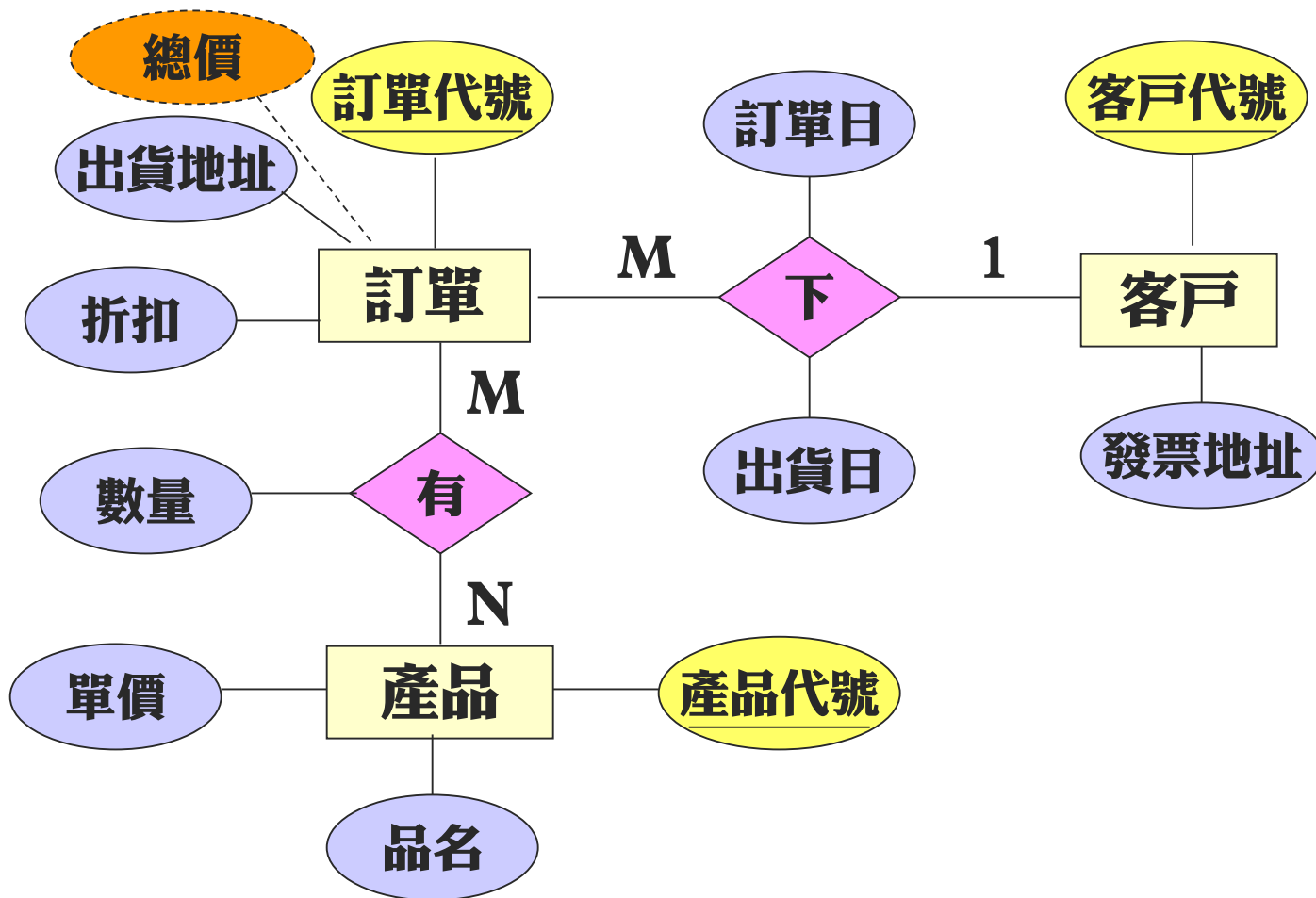
**專案表**(專案代號，專案名稱)

**零件表**(零件代號，零件名稱)

**供應表**(供應商代號，專案代號，零件代號，數量)

■ 以上四個Relations皆符合1NF, 2NF, 3NF, BCNF

# 範例 3



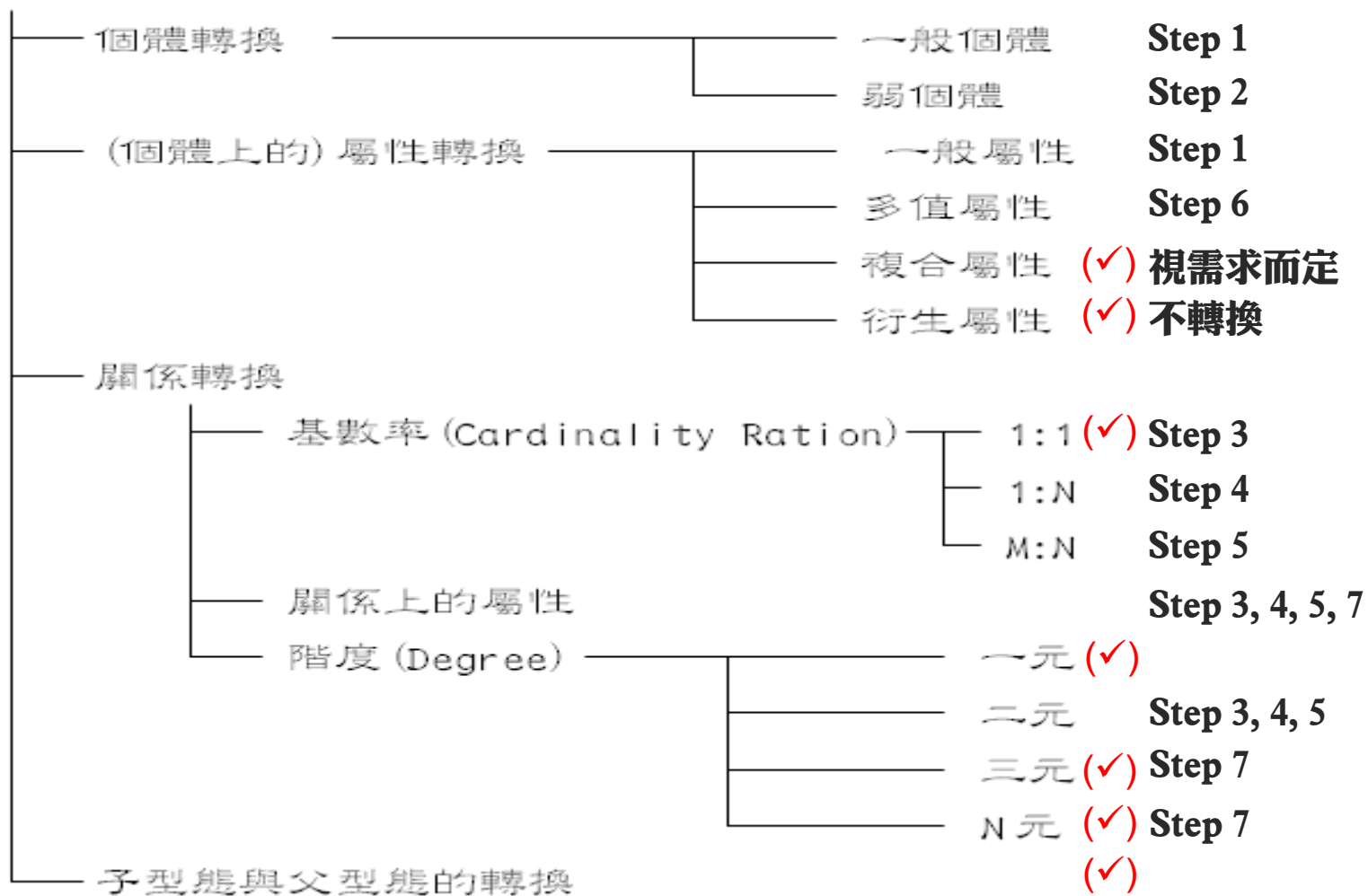




- 根據前面所介紹的7個轉換步驟，本範例的轉換結果如下所示：
  - 產品表：產品代號、品名、單價
  - 產品訂單表：產品代號、訂單代號、數量
  - 訂單表：訂單代號、訂單日、出貨日、折扣、出貨地點、客戶代號
  - 客戶表：客戶代號、發票地址

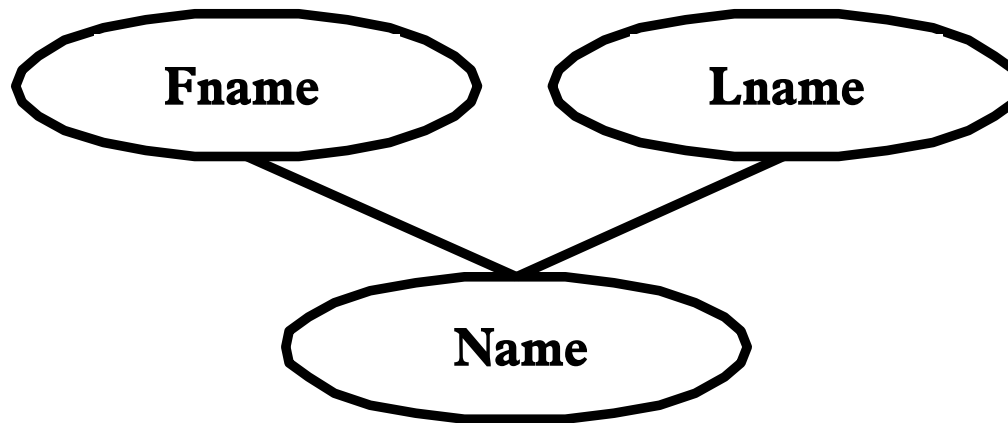
# [ ■轉換細則 ]

※ 要點：



## 複合屬性(Composite Attributes)

- 此屬性處理與否則視需要而定。

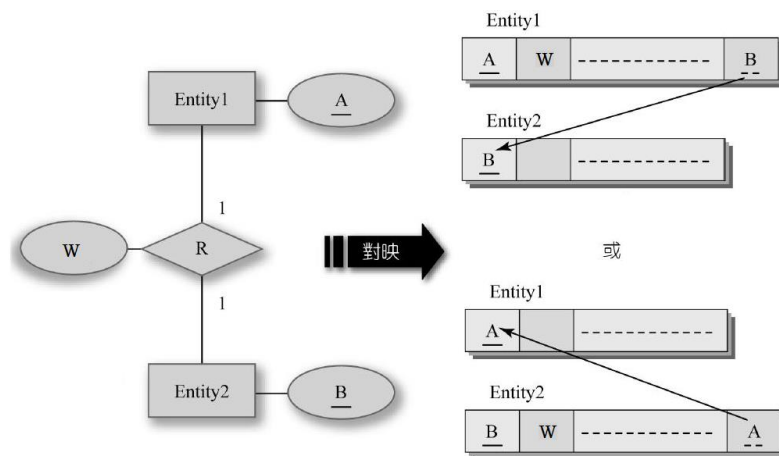


## 衍生屬性(Derived Attributes)

- 因為衍生屬性**可完全由其它屬性求得**，一個設計良好的資料庫**並不會儲存衍生屬性的資料**。因此，在ER Model轉換到關聯表格的過程不會處理衍生屬性。

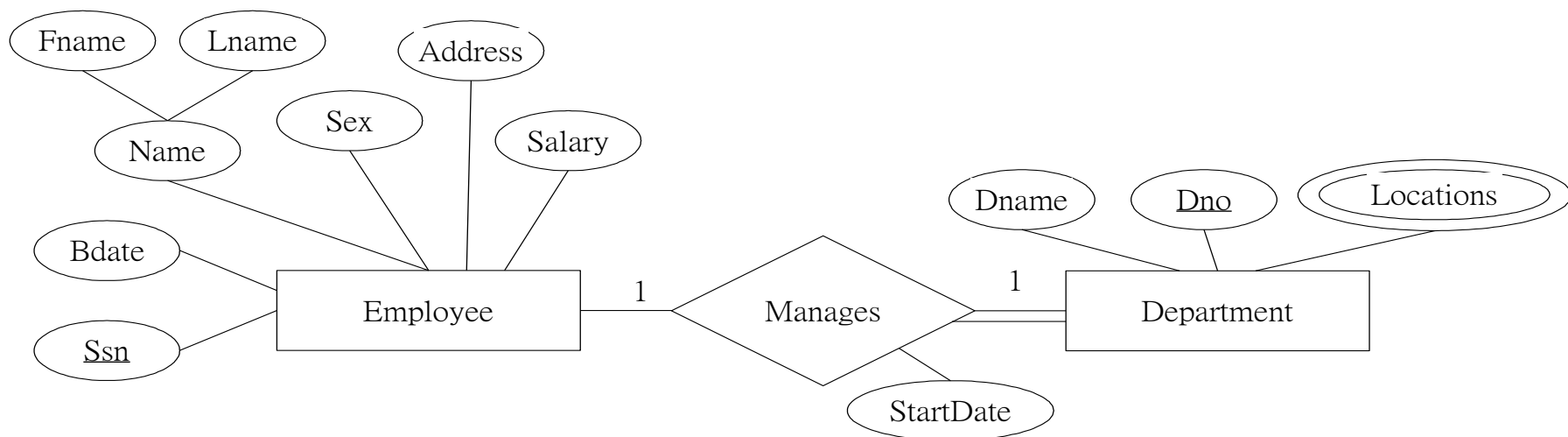
## 「一對一關係」的轉換

- 此關係在Step 3處理，主要是將**某一端個體所形成之表格的主鍵**，當成**另一端個體所形成之表格的外來鍵**
- 轉換時，會受兩端的個體是否為「完全參與」(Total Participation ; T) 或「部份參與」(Partial Participation ; P)之影響而分成三種情況：
  - 一端為「完全參與」，另一端為「部份參與」
  - 兩端均為「部份參與」 (✓)
  - 兩端均為「完全參與」 (✓)



# 一端為「完全參與」，另一端為「部份參與」

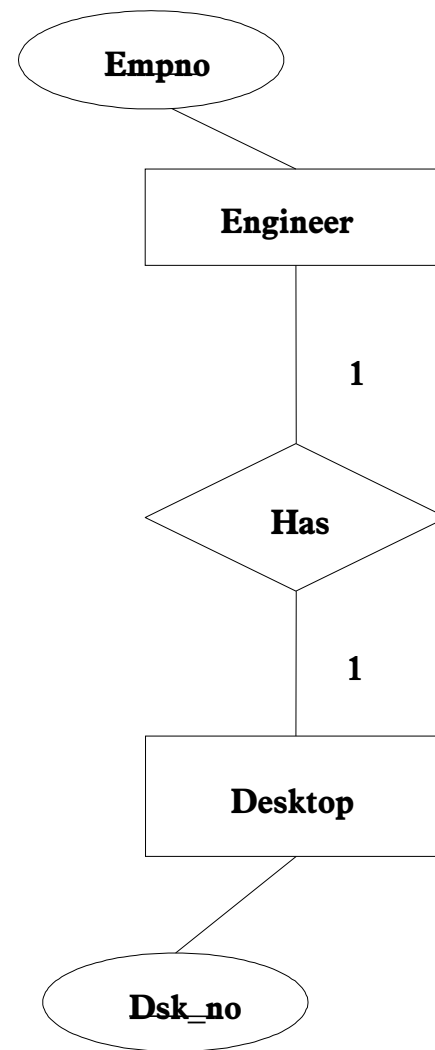
- 在**完全參與的個體 (T)** 中，加入**部份參與個體 (P)** 之主鍵做為**Foreign Key**。
- 若有**關係上的屬性**，亦加入至**完全參與的個體**



**Department(Dno, Dname, Ssn, StartDate)**

## 兩端均為「部份參與」

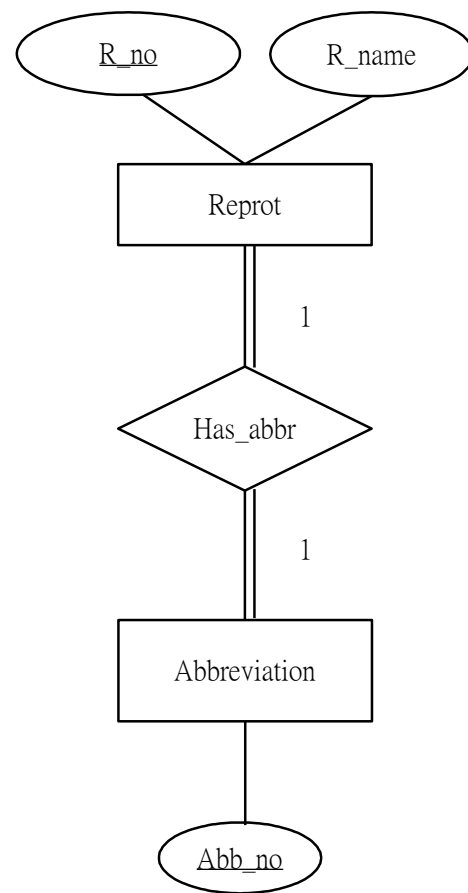
- 在**實例(Instance)數目較少**的個體中，加入**實例數目較多**的個體之**主鍵為Foreign Key**。
- 若有**關係上的屬性**，亦加入至**實例數目較少**的個體。
- 右圖為一些桌上型電腦被授予某些工程師，但不是每個工程師都必須有電腦。
- 轉換結果：
  - Engineer(Empno)
  - Desktop(Dsk\_no, Empno)



## 兩端均為「完全參與」

- 將兩個體與關係合併成一個表格。
- 若有關係上的屬性，亦加入至該關係表格。
- 右圖假設每一個報告皆有一個簡介，且每一個簡介都只對應一個報告(兩端為完全參與)。
- 轉換結果：

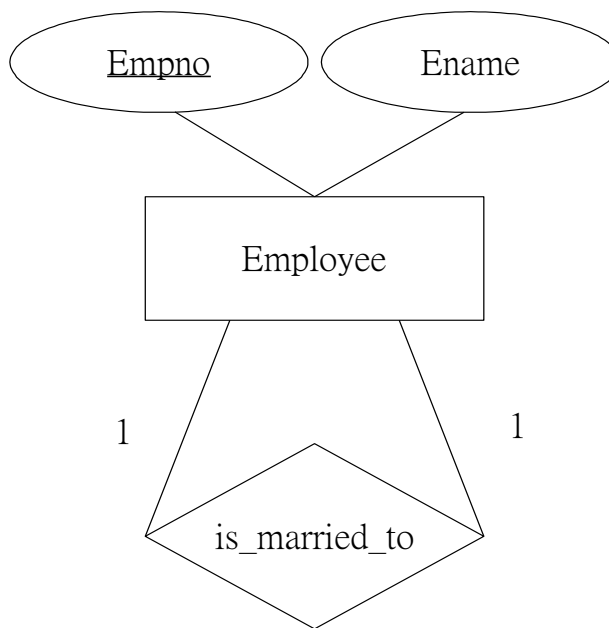
Report\_Abbreviation(R\_no, Abb\_no, R\_name)  
(誰當主鍵都O.K.!!)





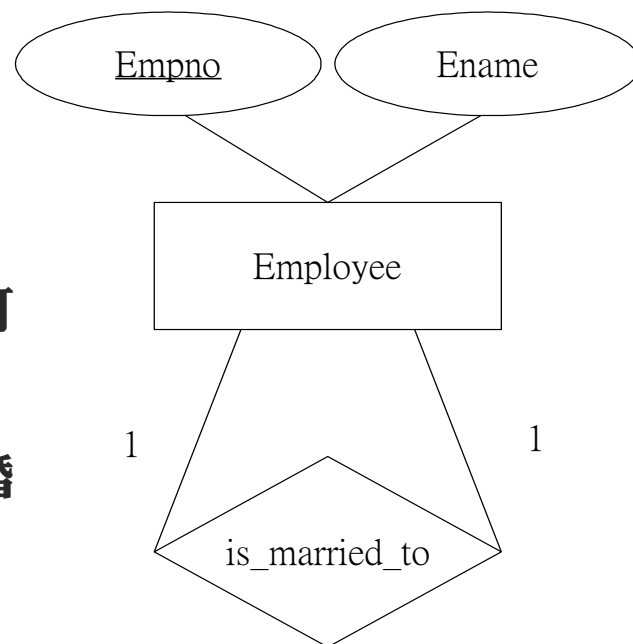
## 遞迴關係(Recursive Relationships)

- 二元關係中如果關係兩端的個體為同一個體，則我們稱為**遞迴關係(Recursive relationship)**。
- 無論是「完全參與」(Total Participation)或是「部分參與」(Partial Participation)都**不會影響**其轉換方法。



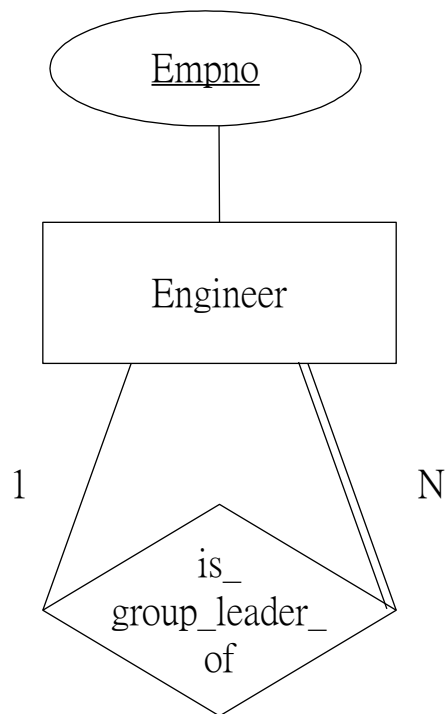
# 一對一遞迴關係

- ❖ 可視為一對一關係中的特例
- ❖ 相似於一端為「**完全參與**」、一端為「**部份參與**」的一對一關係
  - ❖ 因為**兩端是相同的個體**，所以用一個表格即可表示。
  - ❖ 右圖為某公司中任一員工允許與其他員工結婚
- 轉換結果：
  - Employee(Empno, Ename, Spouse\_id)
  - 為避免與原來的Empno混淆，將外來鍵Empno依其關係意義更名為Spouse\_id(配偶的員工代碼)來取代。



# 一對多遞迴關係

- 可視為一對多關係中的特例，轉換方法與原一對多關係的轉換方法相同。
  - 將1端個體的主鍵當作N端個體所轉成的表格中之外來鍵
  - 因為兩端的個體相同，因此用一個關聯表格即可表示
  - 右圖為某公司中的工程師團隊，此團隊有一個領導者
- 轉換結果：
  - `Engineer(Empno, Leader_id)`
  - 為避免與原來的Empno混淆，將外來鍵Empno依其關係意義更名為Leader\_id(領導者的員工代碼)來取代。



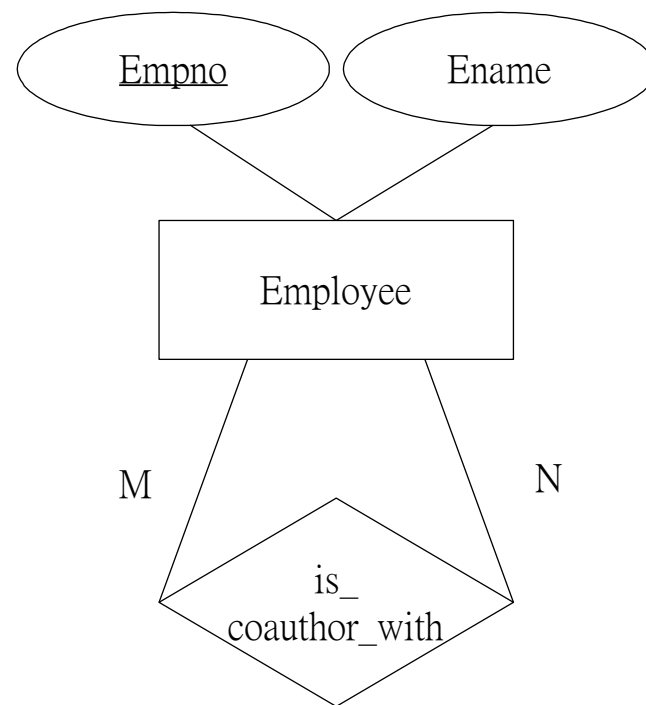
# 多對多遞迴關係

- ❖ 可視為**多對多關係中的特例**，轉換方法與原多對多關係的轉換方法相同

- 將多對多關係轉成一個新的表格
- 右圖為某公司中的員工可以與另一位或多位員工合寫一篇報告，或是自己獨立完成

## ■ 轉換結果：

- **Employee(Empno, Ename)**  
**Coauthor(Author\_id, Coauthor\_id)**
- 新表格的主鍵為多對多關係兩端的個體所轉換成之表格的主鍵所組成。因為關係的兩端都是同一個體Employee，為了避免與原來的Empno混淆，將依其意義更名為 (Author\_id, Coauthor\_id)。

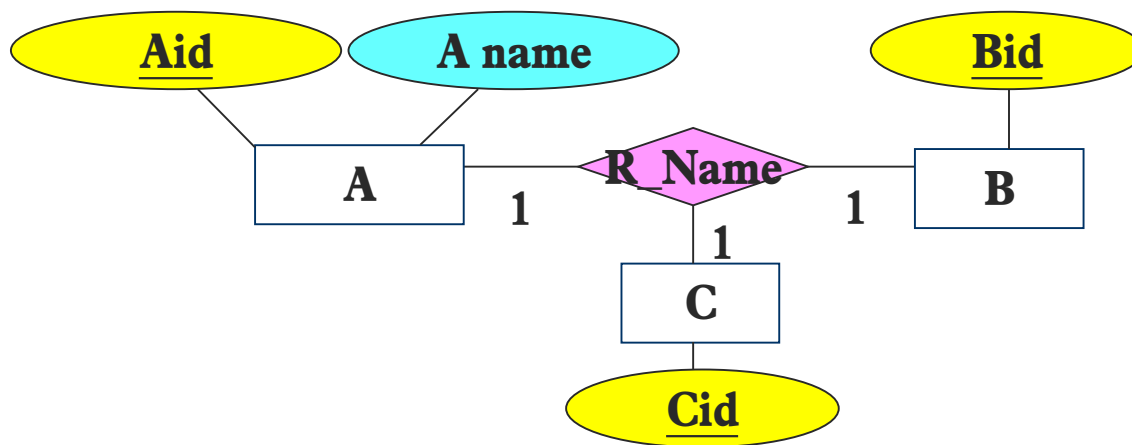


# 三元關係(Ternary Relationships)

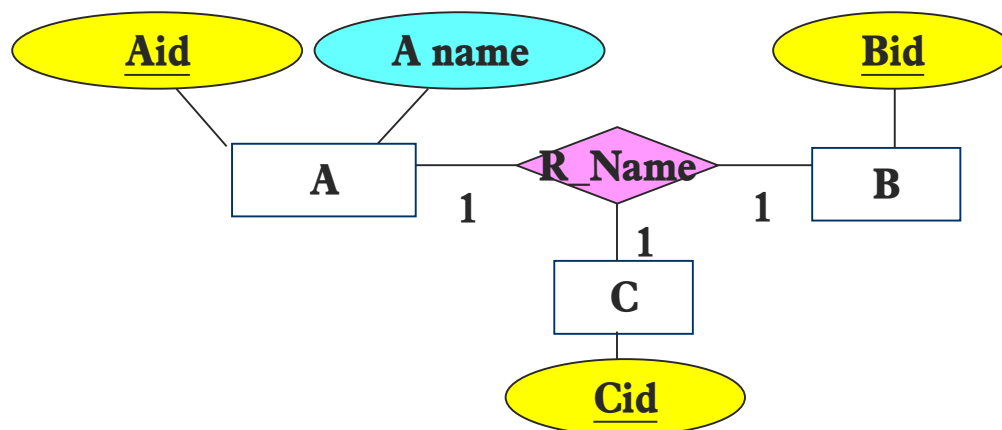
❖ 此關係主要是在Step 7處理。其處理步驟如下：

1. 首先，將三元關係中的所有一般個體都轉換成表格
2. 再將三元關係轉成一個關聯表格，並將原本的三個一般個體之主鍵加入新產生的表格，這三個欄位在新表格中扮演外來鍵的角色，分別與其所屬的一般個體的表格有所關連。而新表格的主鍵則由這三個欄位來決定。

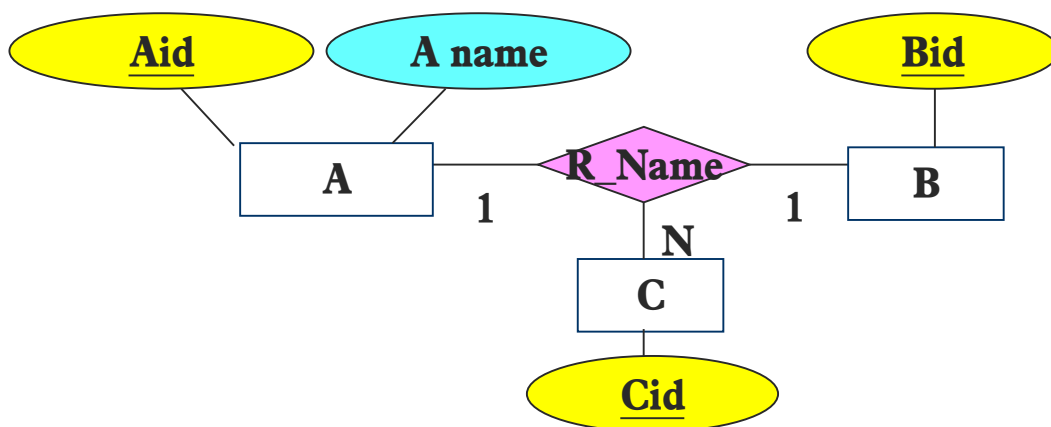
❖ 簡單作法：所有關連到此關係的個體，它們的主鍵做為此新關聯表格的複合主鍵。而這些主鍵同時也是外來鍵，分別與其所屬的一般個體有所關連。



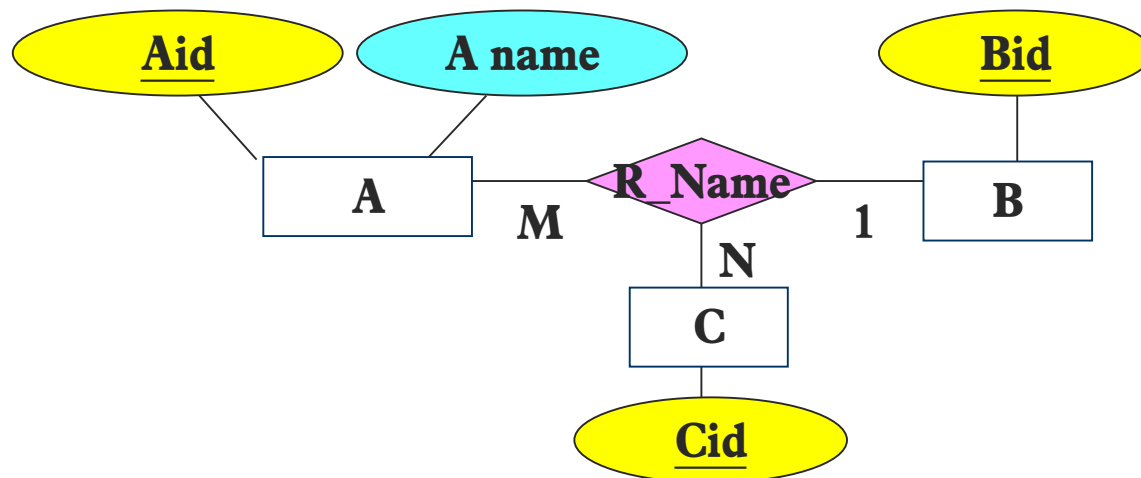
- 1:1:1 的關係可決定出 **3組** 新表格的候選鍵
  - 由三個“1”端個體中的主鍵分別搭配組合而成
  - 新建立的表格為：**R\_Name(Aid, Bid, Cid)**。如果有關係上的屬性，則一併放入新表格中
- 由下圖得知可構成新表格的主鍵之候選鍵有：(Aid, Bid), (Aid, Cid), (Bid, Cid)，可任選其一做為主鍵，其餘的為替代鍵(仍需具備**唯一性**與**Non-Null**)。這三個欄位在新表格中同時也是外來鍵，分別與其原個體的表格有所關連。



- 1:1:N 的關係可決定出 2組 新表格的候選鍵
  - 由兩個“1”端個體中之主鍵與N端個體中之主鍵分別組成
  - 新建立的表格為：**R\_Name(Aid, Bid, Cid)**。如果有關係上的屬性，則一併放入新表格中
- 由下圖得知可構成新表格的主鍵之候選鍵有：(Aid, Cid), (Bid, Cid)，我們可任選其一做為主鍵，其餘的則為替代鍵(仍需具備唯一性與Non-Null)。這三個欄位在新表格中同時也是外來鍵，分別與其原個體的表格有所關連。

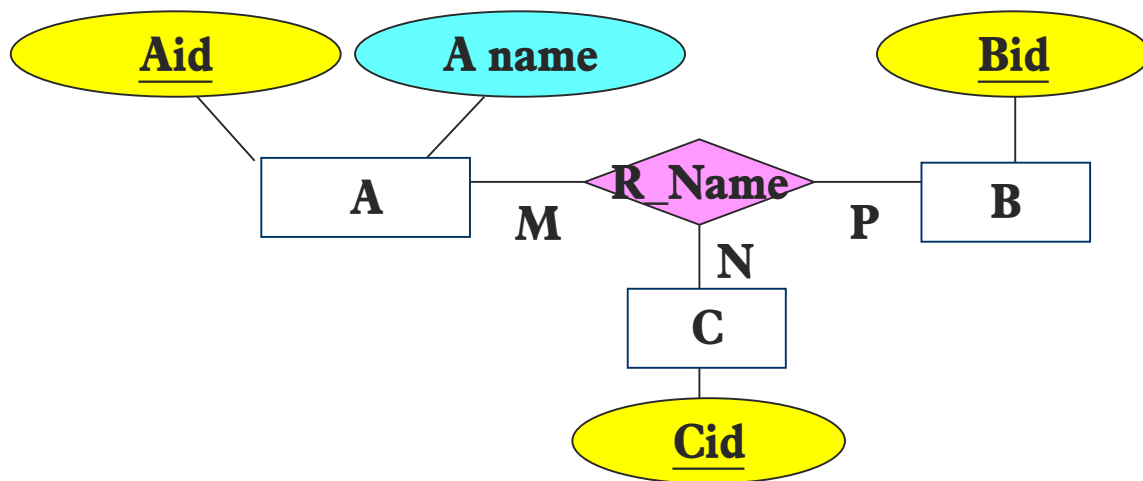


- 1:M:N 的關係可決定出 **1組** 新表格的候選鍵
  - 由兩個“多”端個體中之主鍵組成
  - 新建立的表格為：**R\_Name(Aid, Bid, Cid)**。如果有關係上的屬性，則一併放入新表格中
- 由下圖得知可構成新表格的主鍵之候選鍵有：(Aid, Cid)，我們選擇它做為主鍵。這三個欄位在新表格中同時也是外來鍵，分別與其原個體的表格有所關連。





- P:M:N 的關係可決定 **1組** 新表格的候選鍵
  - 由三個“多”端的原始表格中之主鍵組成
  - 新建立的表格為：**R\_Name(Aid, Bid, Cid)**。如果有關係上的屬性，則一併放入新表格中
- 由下圖得知可構成新表格的主鍵之候選鍵有：(Aid, Bid, Cid)，我們選擇它做為主鍵。這三個欄位在新表格中同時也是外來鍵，分別與其原個體的表格有所關連。



# 多元關係(N-ary Relationships)

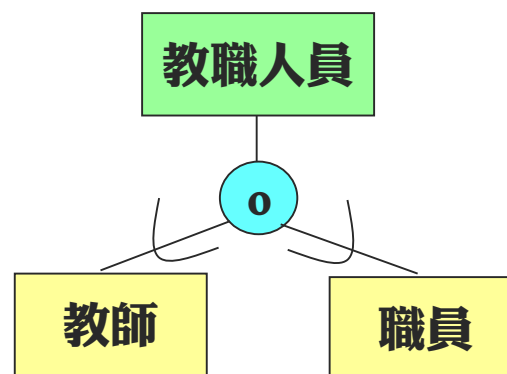
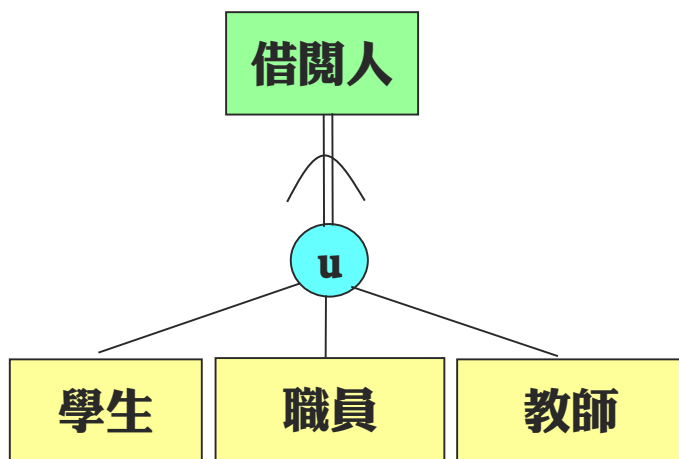
❖ 此關係在Step 7處理。其處理步驟如下：

1. 將關係中所有的一般個體都轉換成表格
2. 再將**多元關係轉成一個新的關聯表格**，並將各個一般個體之主鍵加入此新的表格中，這些欄位在新表格中扮演**外來鍵**的角色，分別與其所屬的一般個體的表格有所關連。而新表格的候選鍵(主鍵)則是由這些新加入之欄位產生(同三元關係)。

❖ 簡單作法：所有關連到此關係的個體，它們的主鍵做為此新關聯表格的**複合主鍵**。而這些主鍵同時也是外來鍵，分別與其所屬的一般個體有所關連。

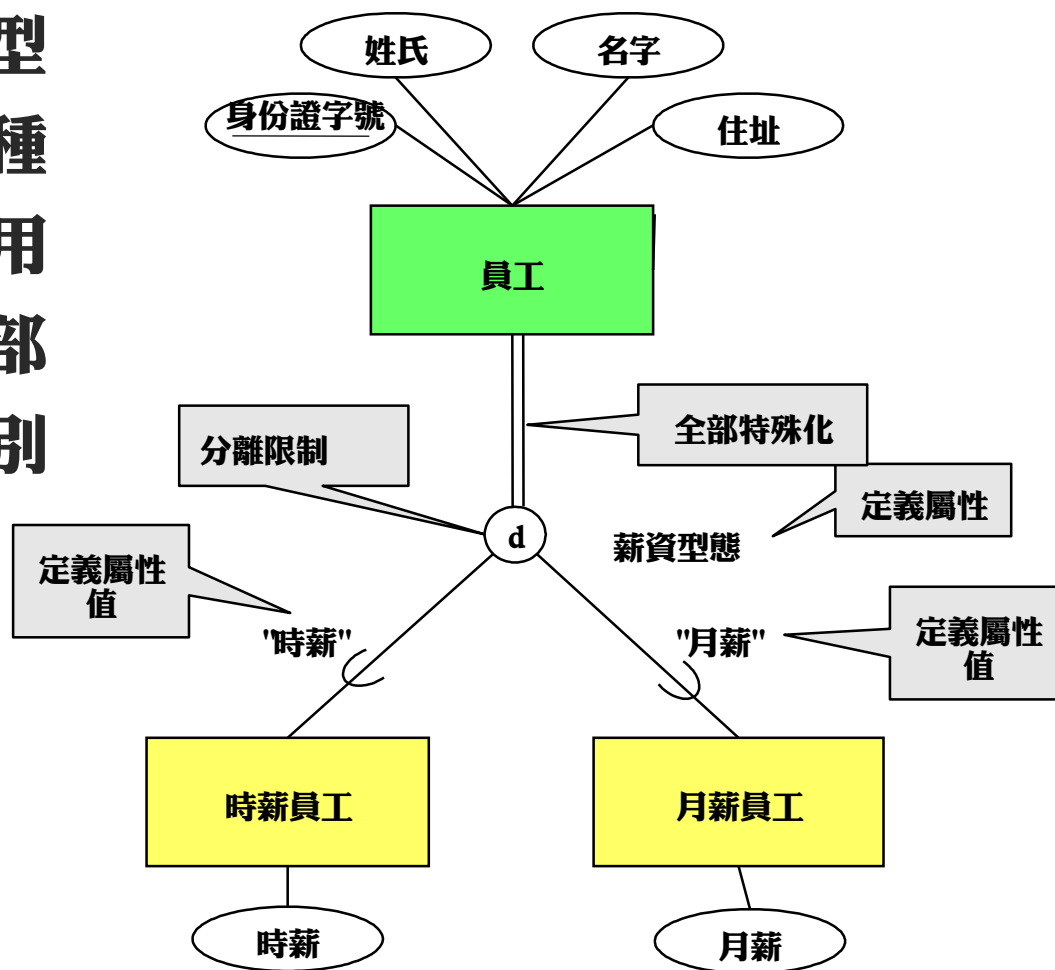
## ■ 將EER Model對映到關聯

- ❖ 子類別 (Subclass) v.s. 超類別 (Superclass)
- ❖ 「一般化」是用來強調各個體類型間的**共同性**
- ❖ 「特殊化」是用來強調各個體類型間的**差異性**



## 範例

- EER Model的關係類型轉換成關聯表格有三種常見的方法，以下利用「分離限制」與「全部特殊化」的例子來分別說明這三種方法。



# 方法1

## ■ 作法：

1. 對上層的超類別個體建立一個關聯表格R；此時，該表格R包含了上層個體類型中所有的屬性。
2. 對於每一個低層的子類別個體而言，分別建立新的關聯表格 $R_i$ 。而在  $R_i$  中，除了包含該低層個體中所有的屬性外，同時亦包含了該個體所屬的超類別個體中的主鍵。

## ■ 轉換結果：

- 員工(身份証字號，姓氏，名字，住址，薪資型態)
- 月薪員工(身份証字號，月薪)
- 時薪員工(身份証字號，時薪)

## ■ 說明：

- 好處：不論子類別是否為“允許重疊” 或是是否為“全部特殊化” 皆適合!!
- 本例子中“薪資型態” 可不用放到超類別的表格中。但一般而言，若EER Model中有註明特殊化屬性時，通常會選擇將該屬性加入。

☞ 一般情況下，最常採用這一種轉換方法 ☞

## 方法2

### ■ 作法：

- 直接對上層的超類別個體建立一個關聯表格  $R$ ，而不另外分別再對下層子類別個體建立其所屬的關聯表格  $R_i$ 。而在關聯表格  $R$  中，除了包含該類別個體中所有的屬性外，尚須包含該個體所屬的子類別個體中所有的屬性。

### ■ 轉換結果：

- 員工(身份証字號，姓氏，名字，住址，薪資型態，月薪，時薪)

### ■ 說明：

- 一個員工的薪資不可能同時為月薪與時薪，因此會造成部份欄位空間的浪費。

✎ 此方法易造成部份欄位空間的浪費 ✎

## 方法3

### ■ 作法:

- 直接對每一個低層的子類別個體都建立一個關聯表格  $R_i$ ，而不另外再對上層超類別個體建立其所屬的關聯表格  $R$ 。而在關聯表格  $R_i$  中，除了包含該子類別個體中所有的屬性外，尚須包含該子類別個體所屬的超類別個體之所有屬性。

### ■ 轉換結果：

- 月薪員工(身份証字號，姓氏，名字，住址，薪資型態，月薪)
- 時薪員工(身份証字號，姓氏，名字，住址，薪資型態，時薪)

### ■ 說明：

- 本例子中“薪資型態”可不用放到表格中。但一般而言，若EER Model 中有註明特殊化屬性時，通常會選擇將該屬性加入。

☞ 此方法只限用在全部特殊化限制與具分離限制情況 ☞

➤ 若**允許子類別間有重疊時** (具重疊限制，違反分離限制)：

- 會造成所有的關聯表格中**資料重複出現**
- 如：若某一員工的薪資型態既有時薪又有月薪 (即：此EER Model具**重疊限制**)，則在這兩個表格中會重複出現兩次該員工的多項個人記錄。

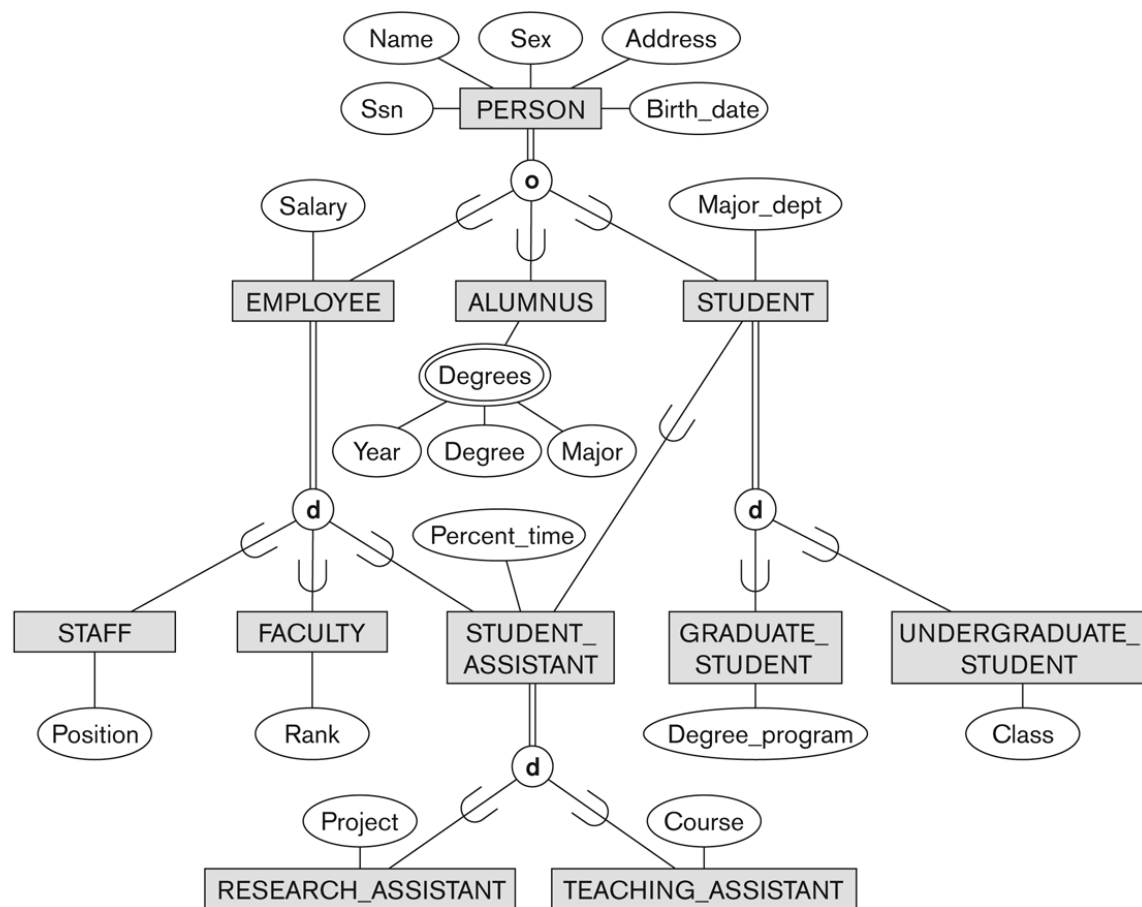
➤ 若為**部份特殊化限制** (具部份特殊化，違反全部特殊化限制)：

- 會造成關聯表格中**資料的遺失**
- 如：若該公司的薪資型態不只時薪、月薪兩種 (如：算週薪...等)，且這些其它型態的薪資未出現於EER Model中 (即：此EER Model為**部份特殊化**)，則這些薪資型態不屬於這時薪、月薪兩類之員工的個人資料將不被存入資料庫中。



# Summary

- 一般情形下，通常會採用**第一種方法**
- 對於同時有數個超類別的共享子類別而言，其轉換方式可採用前述方法的**任何一種**



**Figure 4.7**

A specialization lattice with multiple inheritance for a UNIVERSITY database.