第二章
概念設計與關聯式資料模式

黃仁鵬

2-1 個體-關係模式(E-R Model)

- 資料庫結構包括：
  1. 資料型態(Data Types)。
  2. 資料關係(Relationships)。
  3. 整合限制(Integrity Constraints)。
  4. 運算(Operations)。

資料模式種類

- 概念性資料模式(Conceptual Data Model)
  - 「個體-關係模式」
  - 「物件資料模式」
- 表示性資料模式(Representational Data Model)
  - 階層式資料模式(Relational Data Model)
  - 網路式資料模式(Network Data Model)
  - 關聯式資料模式(Relational Data Model)
- 實體資料模式(Physical Data Model)

2-1-1 個體(Entities)

- 資料庫中的資料描述了真實世界的抽象化情形，每個物件在個體-關係模式中被表示成個體。依其存在條件又可分爲兩類：
  1. 弱個體(Weak Entity)。
  2. 一般個體(Regular Entity)。

2-1-2 屬性(attributes)

- 屬性可分爲下列五種型態：

1. 單一屬性(Single attributes)。
2. 多值屬性(Multivalued attributes)。
3. 複合屬性(Composite attributes)。
4. 衍生屬性(Derived attributes)。
5. 虛值(Null Value)屬性可分爲:
   - 可適用的虛值(Applicable Null Values)。
   - 不可適用的虛值(Inapplicable Null Values)。
   - 完全不知道的虛值(Totally Unknown)。

2-1-3 鍵值(Key)：

鍵值是用來識別每一個個體的屬性，其屬性是唯一的，兩個不同的個體不可以有相同的鍵值。

2-1-4 關係(Relationship)

- 關係爲不同個體(Entities) 間的一種結合性(Association)
- 依參與者參與的關係實例數目，可分爲:
  - 一對一(One-to-one, 1：1)
  - 一對多(One-to-many, 1：N)
  - 多對多(Many-to-many, M：N)

一對一(One-to-one, 1：1)

- 一對一關係是指參與關係的兩個個體之基數都爲一。

一對多(One-to-many, 1：N)

- 一對多關係是指關係一邊的一個個體實例與關係的另一邊的多個個體實例有參與關係。

多對多(Many-to-many, M：N)

- 多對多關係是指參與關係的兩個個體之基數都大於一。

2-1-5 子型態(Subtypes)

- 子型態會繼承所有父型態的屬性與關係，除此之外，各個子型態可以產生比父型態多出數個特有的屬性與關係。

2-1-6 個體-關係圖

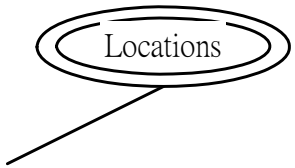
個體-關係模式可以用“個體-關係圖(E-R Diagram)”來加以完整描述。

個體(Entity)

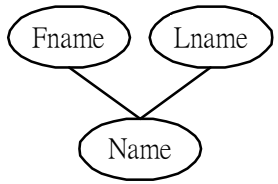
一般個體　　弱個體

屬性(attribute)

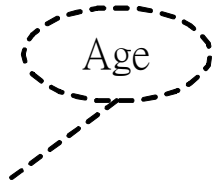
一般屬性



多值屬性

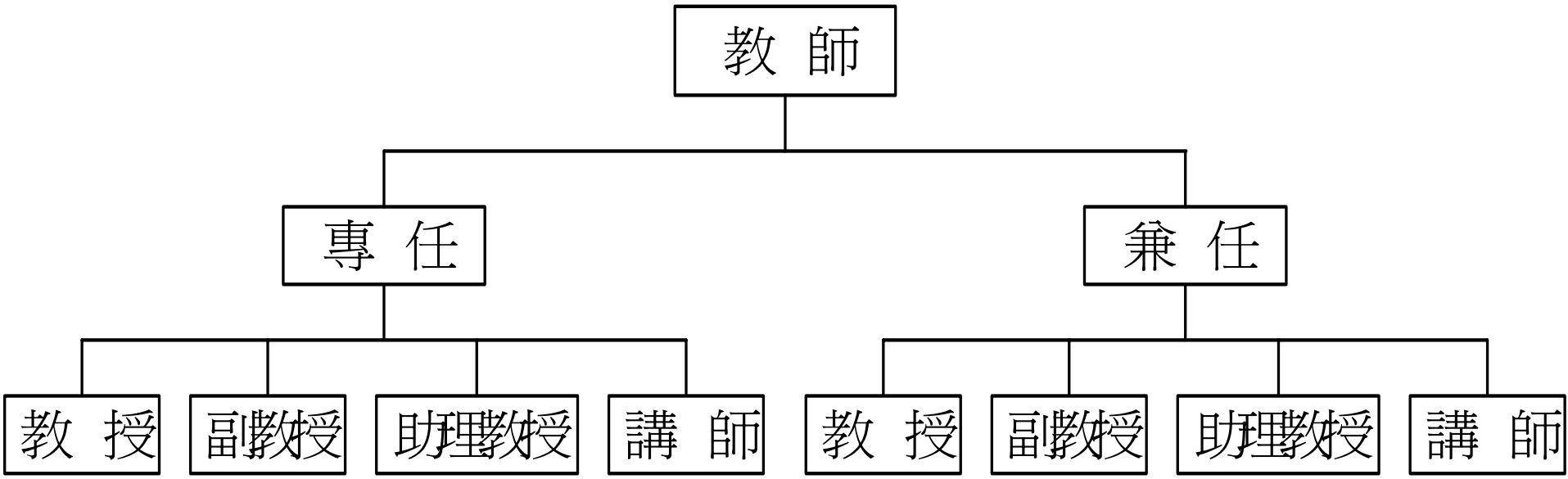


複合屬性

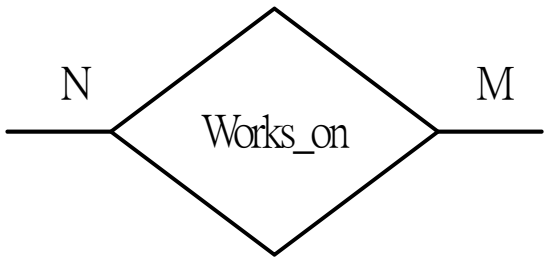


衍生屬性

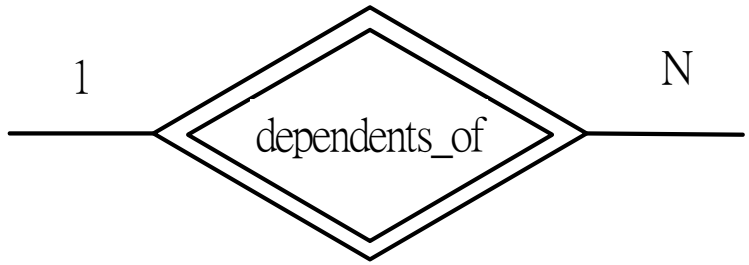
子型態(Subtype) 與父型態(Supertype)



關係(Relationship)

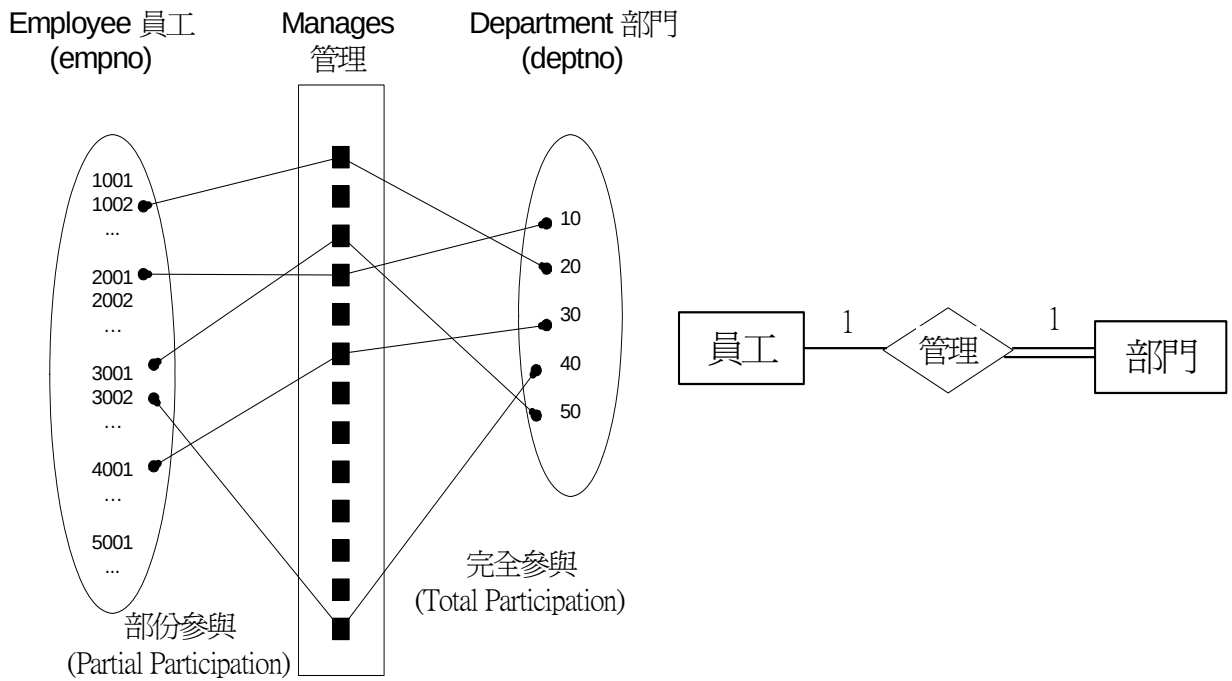


一般關係

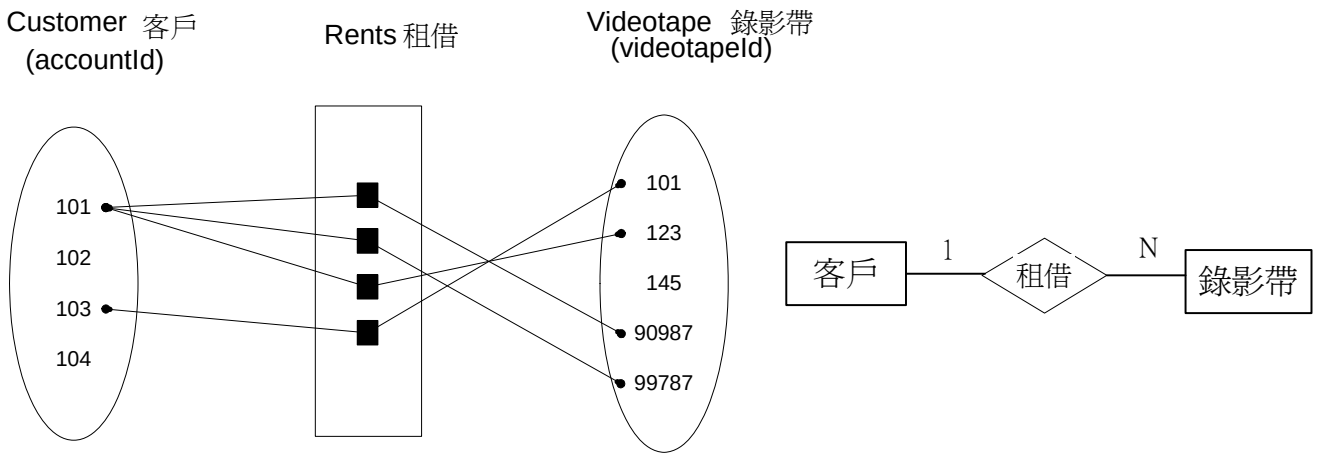


弱關係

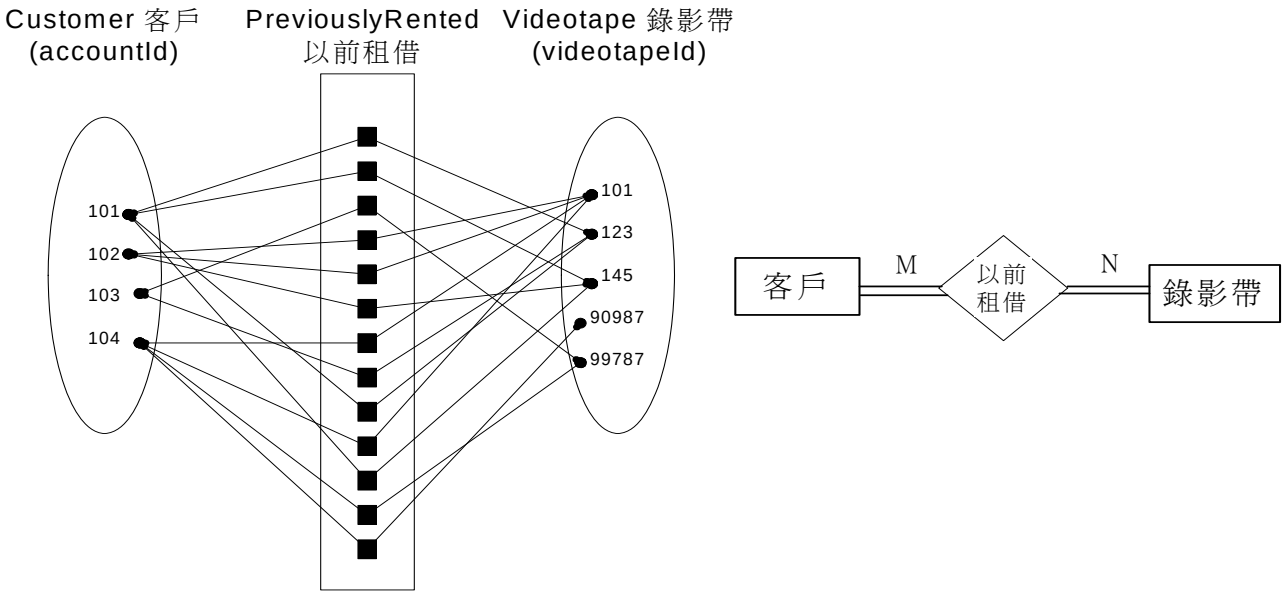
一對一「完全參與」與「部分參與」的 E-R 圖



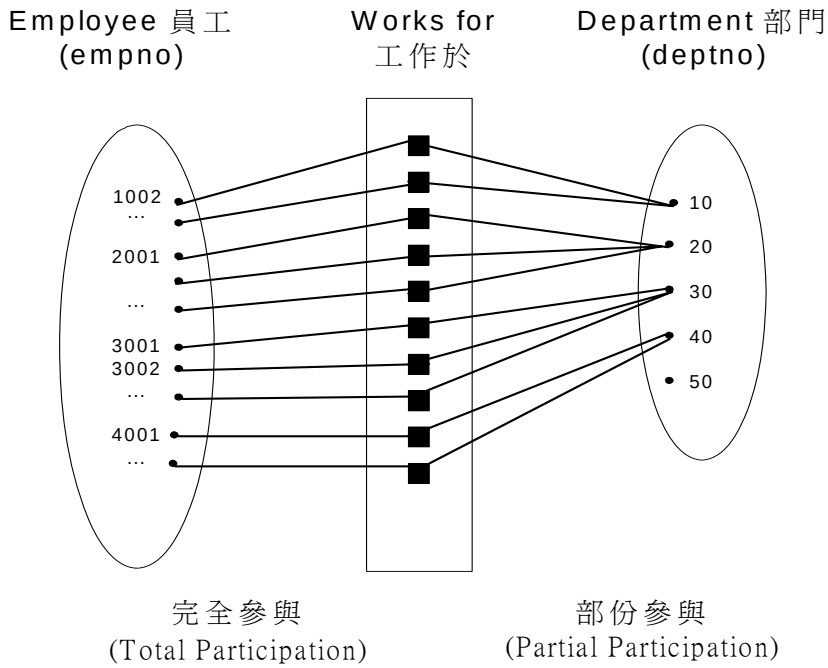
一對多「部分參與」的 E-R 圖



多對多「完全參與」的 E-R 圖



多對一「完全參與」與「部分參與」的 E-R 圖



2-1-7 利用ER Model概念設計的步驟

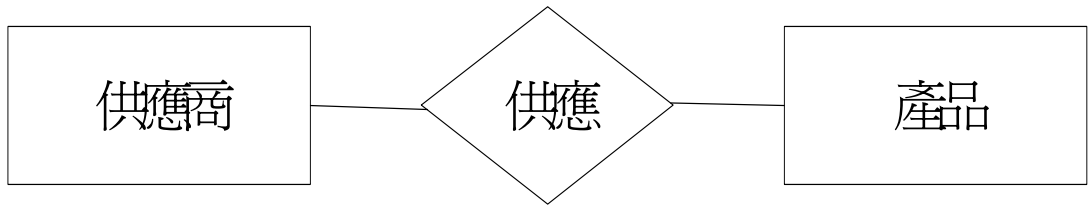
1. 根據客戶的需求定義有興趣的個體與關係。
2. 定義這些關係的基數(Cardinality)。
3. 細部定義這些個體與關係的屬性。
4. 定義這些屬性的基數(Cardinality)。
5. 定義這些屬性的資料型態。

2-1-8 如何繪製E-R圖

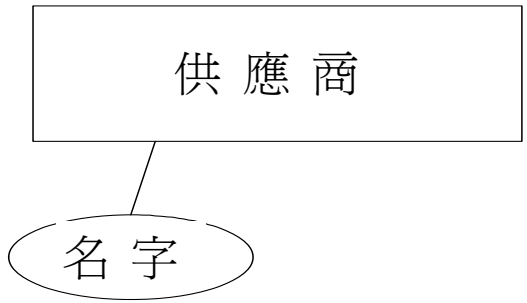
系統開發者通常是以訪談的方式，將問題及需求作成文字型式的記錄。然後再把文字轉成E-R圖。

- 繪製E-R圖通常有以下幾個原則：
  1. 句子中的「名詞」可對應成「個體」，「動詞」可對應成「關係」，「名詞的集合」可對應成「個體型態」。
  2. 句子中描寫名詞的「形容詞」或「所有格」可對應成個體型態的「屬性」。
  3. 句子中描述動詞的「副詞類」可對應成「關係上的屬性」。

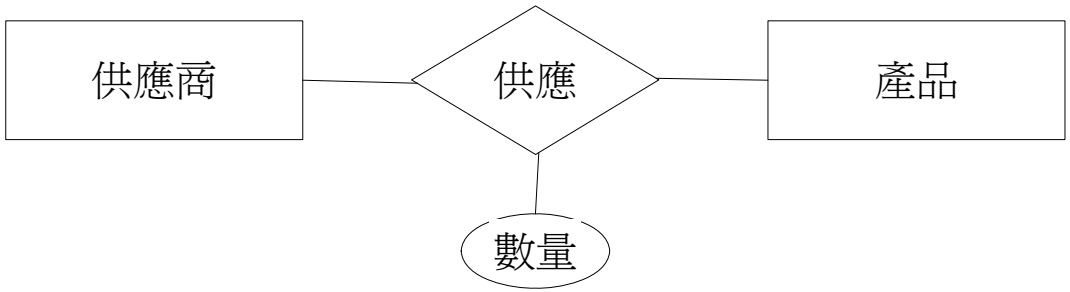
- 例如：”供應商 S 供應產品 P”，其中「供應商」和「產品」爲名詞，「供應」爲動詞，因此可轉成以下的E-R圖：



- 例如：”供應商 S 的名字爲華北企業公司”，其中「供應商」爲名詞，「的名字」爲所有格，因此這句子可以轉成以下的E-R 圖

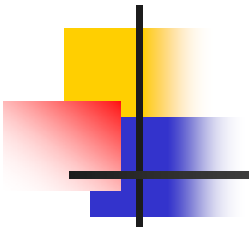


- 例如：”供應商 S 供應數量 Q 的產品 P”，其中「數量」爲副詞，因此這句子可以轉成以下的E-R 圖：



2-2 關聯式資料模式簡介

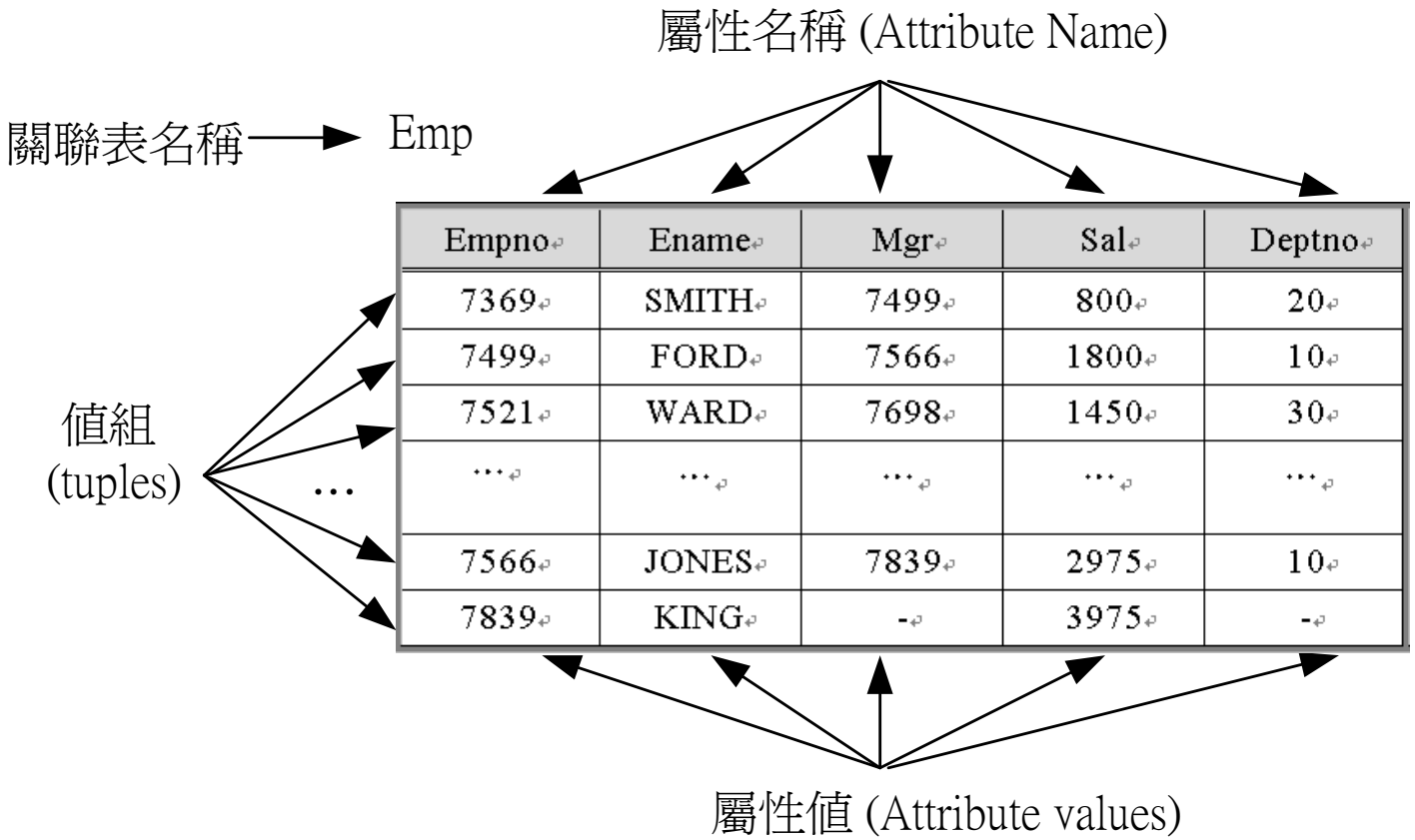
- 1970 年 E.F. Codd 提出關聯式資料模式理論，關聯式資料模式的理論基礎來自數學的集合論(Set Theory)，藉由關聯式資料模式達到以下的目標，並且達到減少資料的重覆性與資料的一致性：
  1. 資料獨立(Data Independence)。
  2. 便於溝通(Communicability)。
  3. 集合處理(Set Processing)。



- 關聯式資料庫系統的基礎理論基於資料的關聯式模式理論，使得關聯式資料庫系統具有以下的特性：

1. 資料結構(Data Structures)。
2. 整合限制條件(Integrity Constraints)。
3. 資料操作(Data manipulation)。

2-2-1 資料結構


屬性名稱 (Attribute Name)
關聯表名稱
Emp
值組
(tuples)
Empno Ename Mgr Sal Deptno
7369 SMITH 7499 800 20
7499 FORD 7566 1800 10
7521 WARD 7698 1450 30
7566 JONES 7839 2975 10
7839 KING - 3975 -
屬性值 (Attribute values)

鍵值有關的名詞：

1. 超鍵(Superkey)。
2. 候選鍵(Candidate Key)。
3. 主鍵(Primary Key)。
4. 替代鍵(Alternate Key)。
5. 外來鍵(Foreign Key)。

關聯表具備四個性質

1. 關聯表的值組是沒有順序的。
2. 關聯表的屬性是沒有順序的。
3. 關聯表中的屬性值是單元值。
4. 關聯表中不含重複的數值。

2-2-2　整合限制條件(Integrity Constraints)

- 其目的是爲了確保資料庫內部資料的一致性與完整性，以避免因爲新增、刪除與修改等操作而所造成的異常現象。

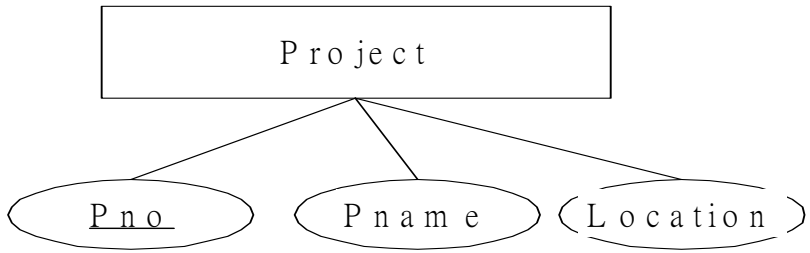
關聯表格內的整合限制的種類

1. 個體整合限制(Entity Integrity Constraints)
2. 值域整合限制(Domain Integrity Constraints)，
3. 參考整合限制(Referential Integrity Constraints)

個體中「屬性」(attributes) 的轉換

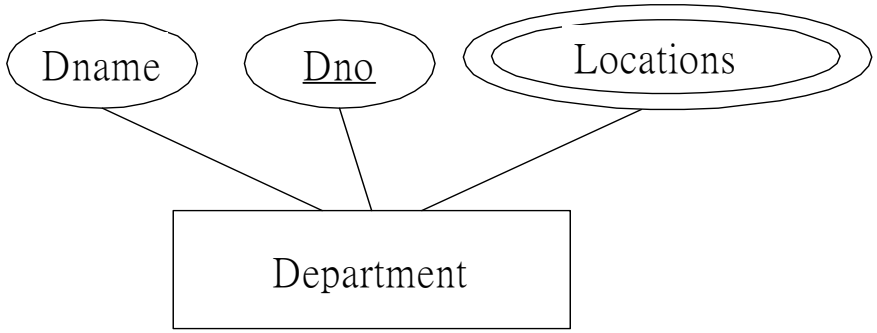
- 單值(Single-valued)
- 多值(Multivalued)
- 複合屬性(Composite attributes)
- 衍生屬性(Derived attributes)

單值屬性(Single-valued attributes)



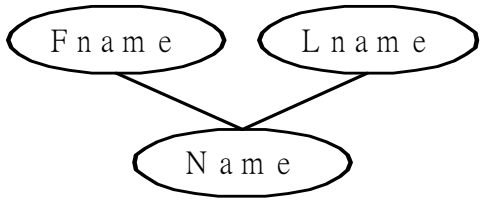
Project(Pno, Pname, Location)

多值屬性(Multivalued attributes)



Dept_Location(*Dno*, Location)

複合屬性(Composite attributes)

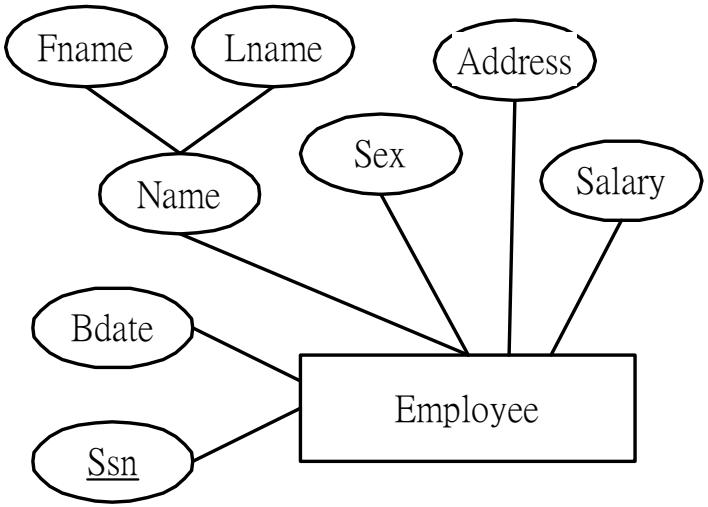


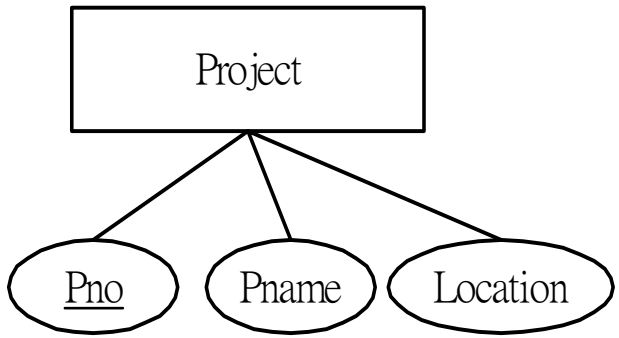
(Fname, Lname)

衍生屬性(Derived attributes)

因爲衍生屬性會因時間的改變而改變，因此一個設計良好的資料庫並不會儲存衍生屬性。

「一般個體」的轉換

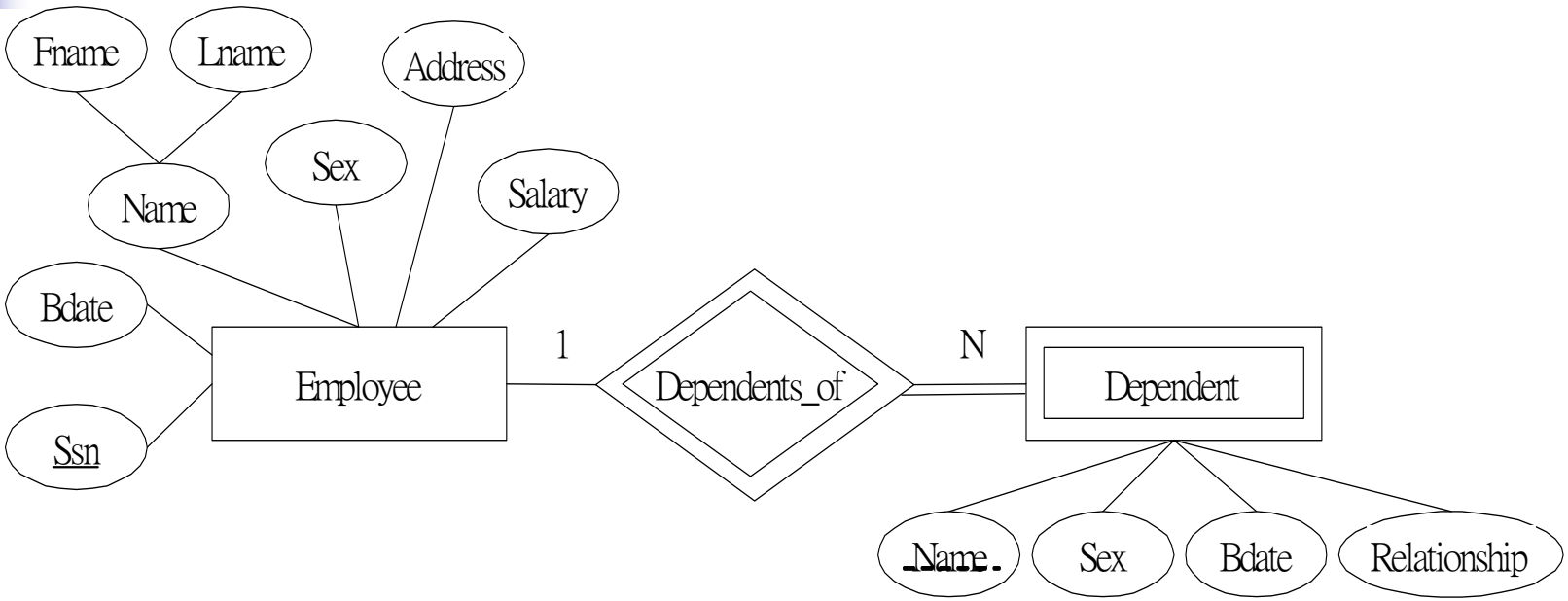




Employee(Ssn, Bdate, Fname, Lname, Sex, Address, Salary)

Project(Pno, Pname, Location)

「弱個體」的轉換

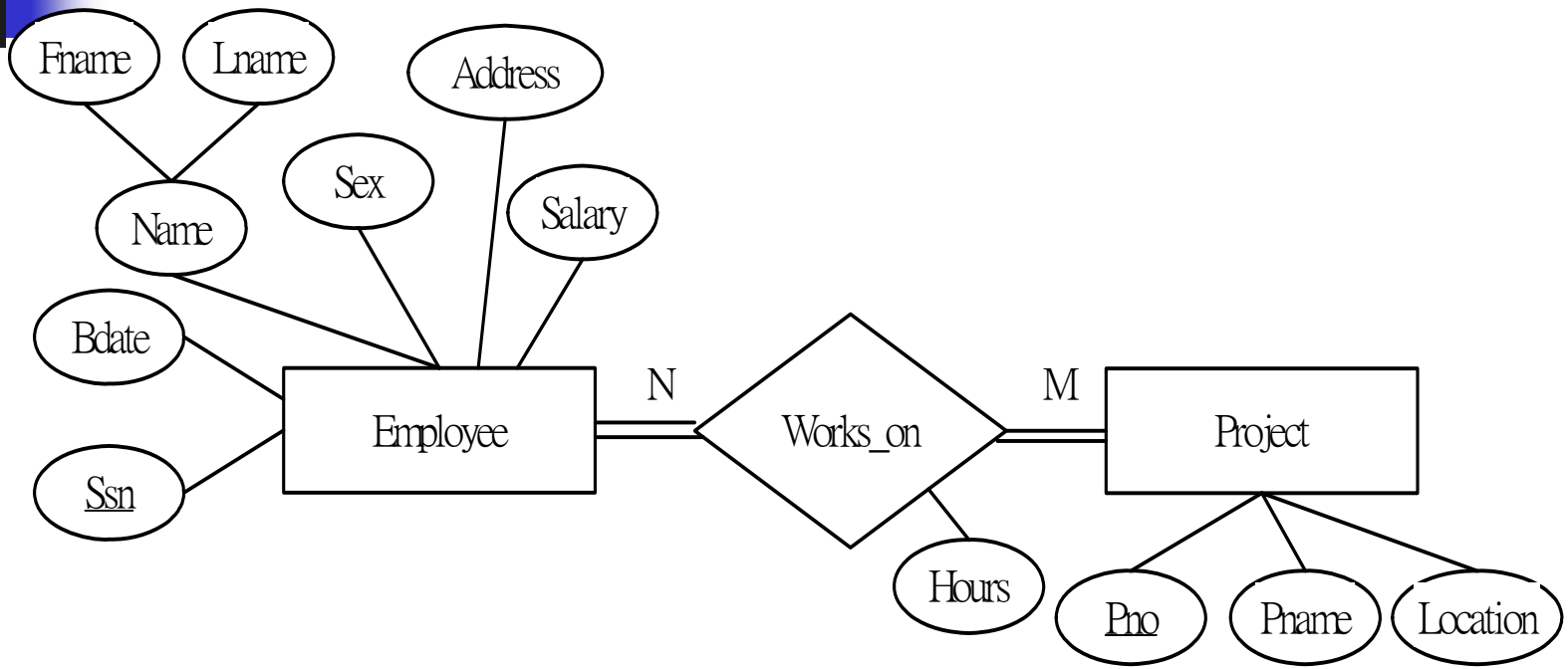


Dependent(*Ssn*, Name, Sex, Bdate, Relationship)

參考

Employee(Ssn, …, …)

「多對多關係」的轉換

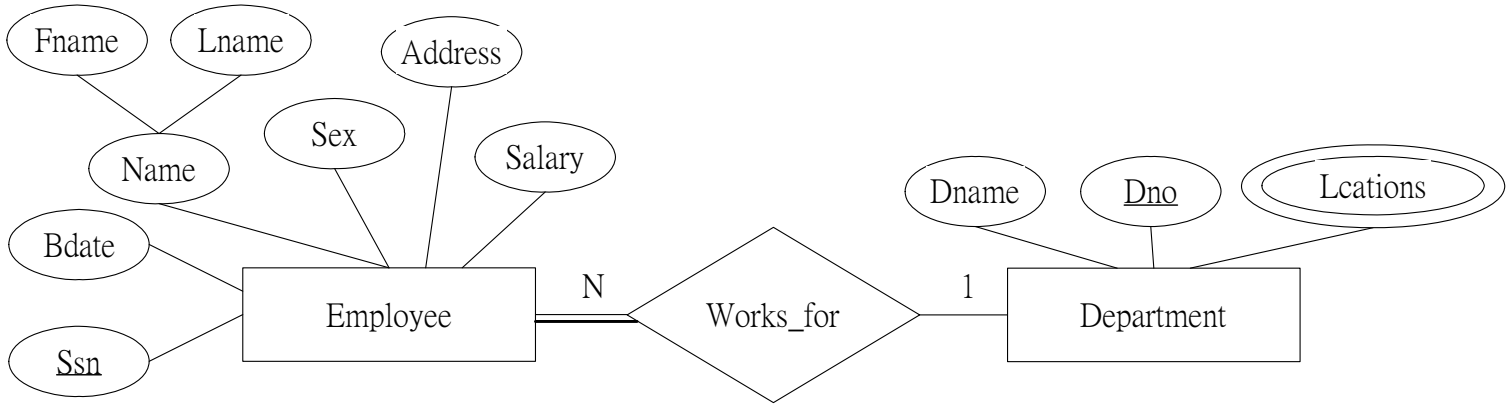


Works_on(*Ssn*, *Pno*, Hours)

參考 參考

Employee(Ssn, …, …) Project(Pno, …, …)

「多對一」關係轉換

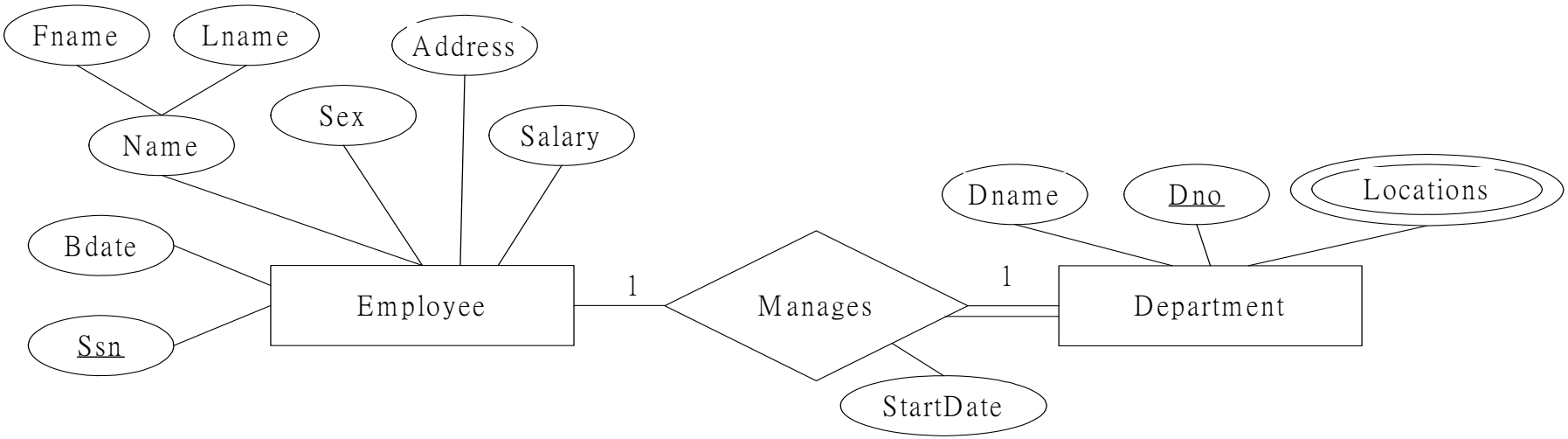


Employee(<u>Ssn</u>, Bdate, Fname, Lname, Sex, Address, Salary, *Dno*)

參考

Department(<u>Dno</u>, …, …)

一對一關係的轉換



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

「子型態」與「父型態」的轉換

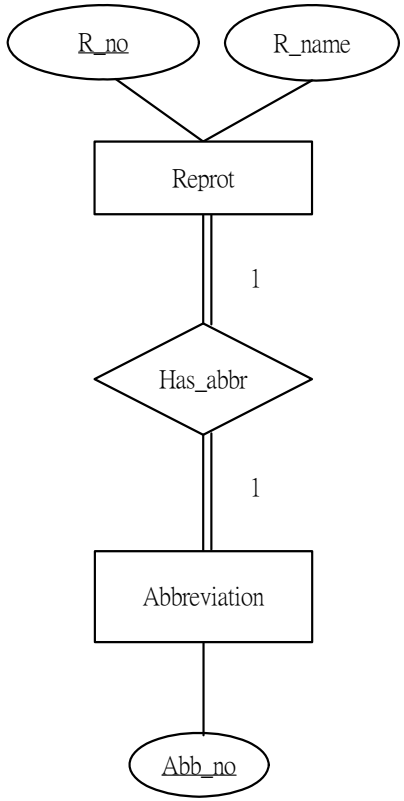
- 「子型態」(Subtype) 與「父型態」(Supertype) 轉換成「外來鍵參考」。子型態與父型態的個體均會產生一個關聯表格。

2-4 關係轉換

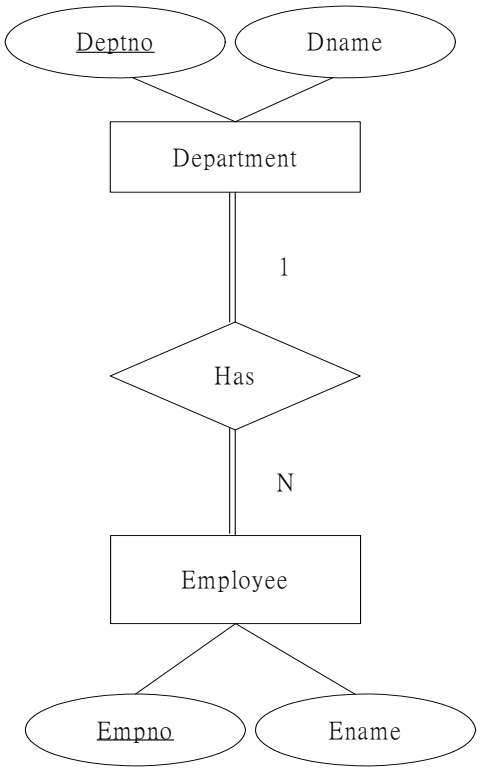
- 除了可依關係分類外，又可以依關係的兩端的個體是否爲「完全參與」(Total Participation) 或「部份參與」(Partial Participation)，可以分爲 (1)關係兩端都爲「完全參與」、(2)一端爲「完全參與」而另一端爲「部份參與」、(3)關係兩端都爲「部份參與」等三種對應方式。

2-4-1 二元關係(Binary relationships)

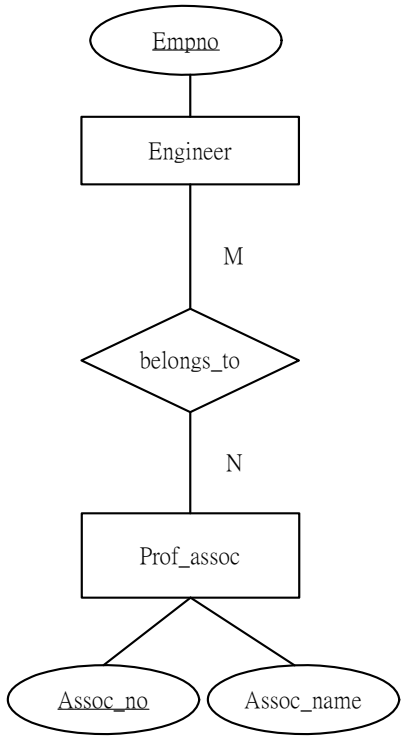
(1)一對一關係(one-to-one relationship)



(2)一對多關係(one-to-many relationship)



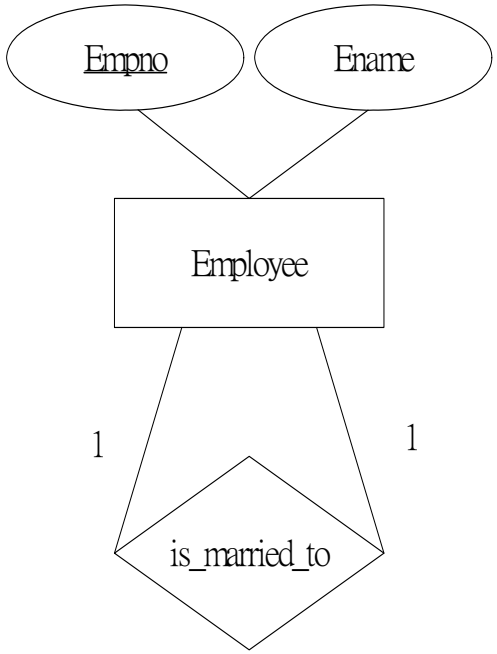
(3) 多對多關係(many-to-many relationship)



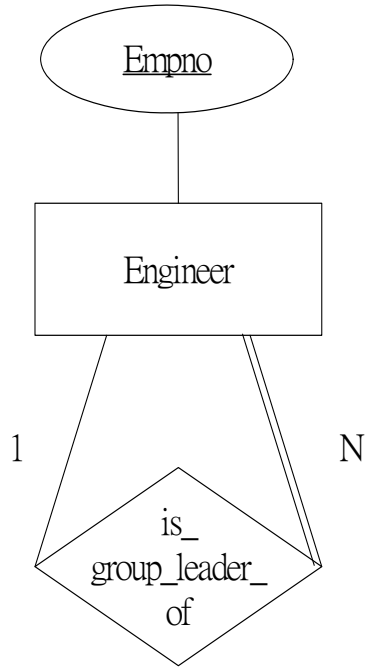
(4)遞迴關係(recursive relationships)

- 二元關係中如果關係兩端的個體爲同一個體，則我們稱爲遞迴關係(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。

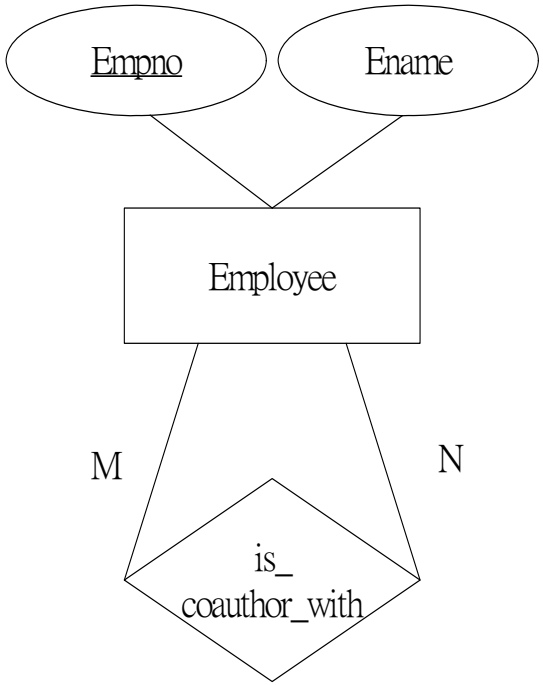
一對一遞迴關係



一對多遞迴關係



多對多遞迴關係



2-4-2 三元關係(Ternary relationships)

- 轉換方式爲將關係中所有的個體都轉換成關聯表格，再將三元關係轉成一個關聯表格表示，並以三元關係端的主鍵加入新產生的表格，而新表格的候選鍵與主鍵由表格中的屬性(attribute) 中產生。

2-4-3 多元關係(N-ary relationships)

- 將關係中所有的個體都轉換成關聯表格，再將多元關係轉成一個關聯表格表示，並以多元關係端的主鍵加入新產生的表格，而新的表格的候選鍵與主鍵由表格中的屬性(attribute) 中產生。

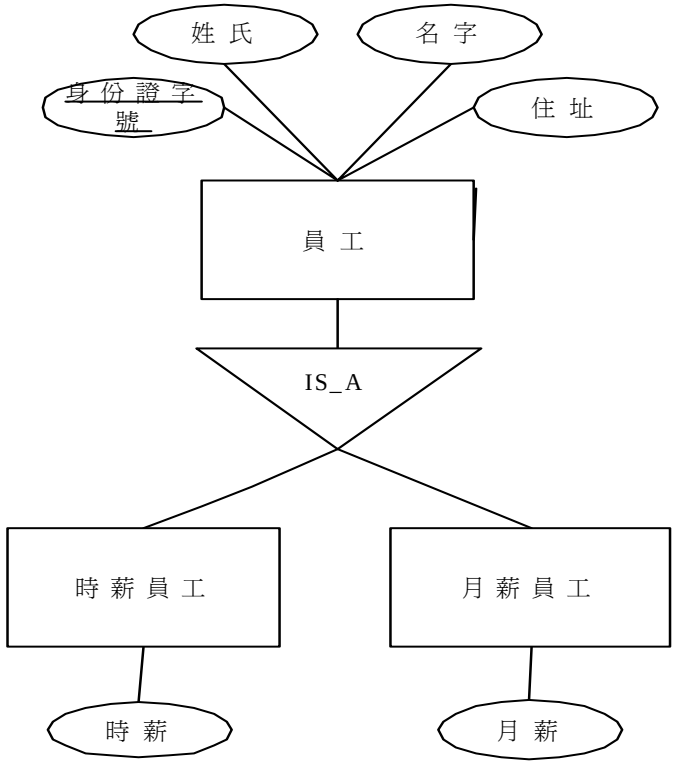
2-5 擴充性個體-關係模式

- 擴充性個體-關係模式也可以稱爲補強性個體-關係模式(Enhanced E-R Model)。

2-5-1 特殊化和一般化

- 「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。

「特殊化」或「一般化」的個體關係圖



「特殊化」關係的限制

- 非連結性限制」(Disjointness constraints)
- 「完整性限制」(Completeness constraints)

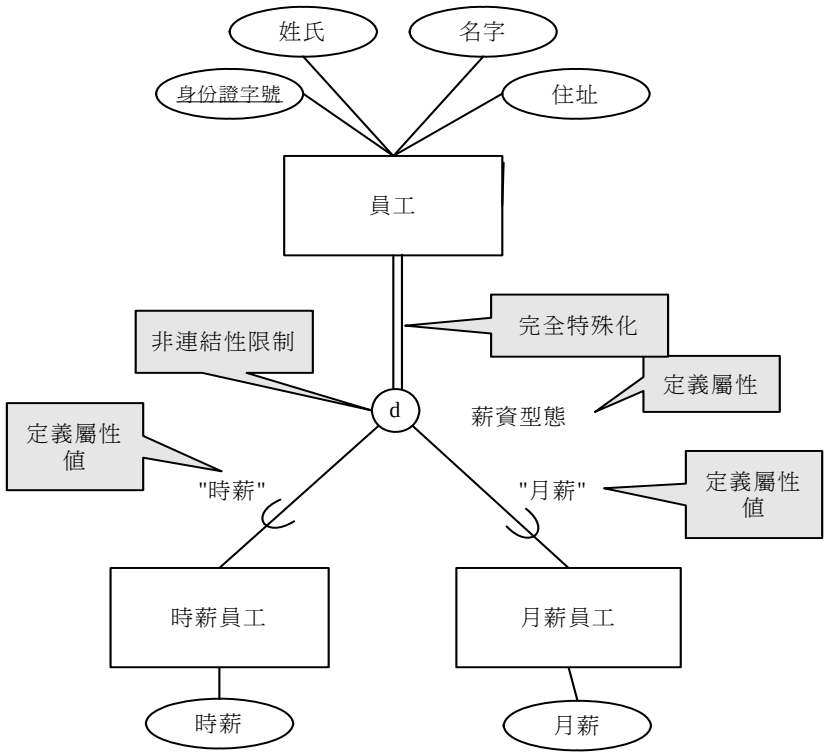
「非連結性限制」

- 滿足「非連結性限制」：做「特殊化」處理後，在低層次的子類別個體類型間，交集的結果一定是空集合。 d 來表示所有子類別皆滿足「非連結性限制」
- 若不滿足「非連結性限制」時，用 o 來表示所有子類別間有發生重疊的情形 (Overlap allowed)。

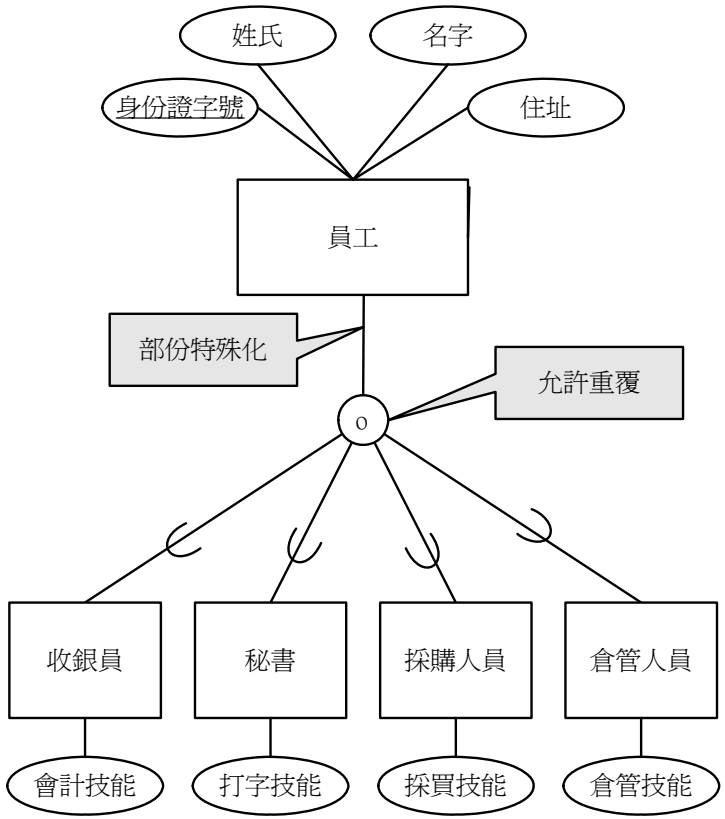
「完整性限制」

- 「完全特殊化」 (Total specialization) ：對某一個個體類型做「特殊化」處理時，在該個體類型中，每一個個體都會出現在該個體類型的子類別個體類型中。用”雙線段” 代表「完全特殊化」。
- 「部份特殊化」 (Partial specialization) ：對某一個個體類型做「特殊化」處理時，在該個體類型中，並非每一個個體都會出現在該個體類型的子類別個體類型中。以 “單線段” 代表「部份特殊化」

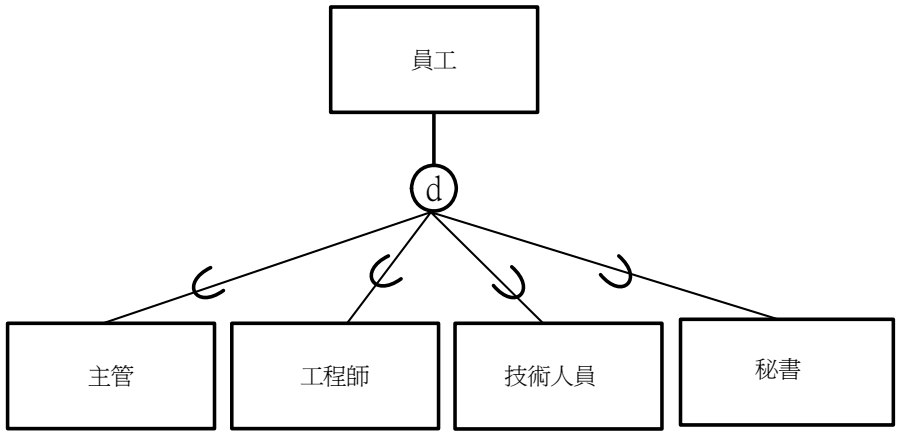
「非連結限制」與「完全特殊化」的例子



「部份特殊化」與「允許重覆」的例子

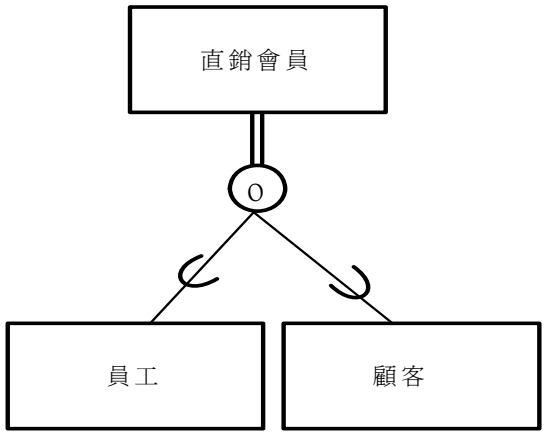


「部份特殊化」與「非連結限制」的例子



「部份特殊化」與「非連結限制」的例子

「完全特殊化」與「允許重覆」的例子



2-5-2 特殊化和一般化的轉換

1. 對上層的父類別個體類型建立一個關聯式綱要R；此時，關聯式綱要R包含了上層個體類型中所有的屬性。而對於每一個低層的子類別個體類型而言，都另外再建立一個新的關聯式綱要 R_i。而在 R_i 中，除了包含該低層個體類型中所有的屬性外，同時亦包含了該個體類型的父類別個體類型中的主鍵。

2. 第二種轉換法與前一種轉換方式最大不同的地方是直接對上層的父類別個體類型建立一個關聯式綱要 R，而不另外分別再對下層子類別個體類型建立其所屬的關聯式綱要 R_i。而在關聯式綱要 R 中，除了包含該類別個體類型中所有的屬性外，尚須包含該個體類型的子類別個體類型中所有的屬性。

3. 第三種轉換法與前兩種轉換方式最大不同的地方是直接對每一個低層的子類別個體類型都建立一個關聯式綱要 R_i，而不另外再對上層父類別個體類型建立其所屬的關聯式綱要 R。而在關聯式綱要 R_i 中，除了包含該個體類型中所有的屬性外，尚須包含該個體類型的父類別個體類型中所有的屬性。