



資料庫系統管理

Database System Management

資訊管理學系 陳士杰老師

E-R Model與延伸式E-R Model

E-R Model and EE-R Model

Outlines

- 資料庫系統的開發流程
- E-R Model (E-R Diagrams)
- Entity, Attributes, Relationships
- 繪製E-R Diagram
- 延伸性個體-關係模式 (Enhanced E-R Model)
- 類別 (Class)
- 一般化(Generalization)和特殊化(Specialization)

【講義：Ch. 1, Section 5; Ch. 8, Section 1】

【原文：Ch. 3; Ch. 4】

【補充講義 1】

資料庫系統的開發流程

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
- 選用合適的資料庫系統(DBS)
- 邏輯設計(Logical Design)
- 實體設計(Physical Design)
- 系統實作(Implementation)

■ 需求收集與分析(Requirements Collection and Analysis)

- 系統分析師與DBA需與使用者面談，以了解使用者需求
- 可利用如資料流程圖(Data Flow Diagram; DFD)等系統分析工具來輔助分析與描述。

■ 概念設計(Conceptual Design)

■ 選用合適的資料庫系統(DBS)

■ 邏輯設計(Logical Design)

■ 實體設計(Physical Design)

■ 系統實作(Implementation)

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
 - 讓分析師以概念性的工具來表達資訊的流程，以便於讓不熟悉電腦的使用者交換意見
 - 常用的概念性工具為個體-關係模式(Entity-Relational Model; E-R Model)
- 選用合適的資料庫系統(DBS)
- 邏輯設計(Logical Design)
- 實體設計(Physical Design)
- 系統實作(Implementation)

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
- 選用合適的資料庫系統(DBS)
 - 在此，我們選用MySQL 5資料庫系統
- 邏輯設計(Logical Design)
- 實體設計(Physical Design)
- 系統實作(Implementation)

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
- 選用合適的資料庫系統(DBS)
- 邏輯設計(Logical Design)
 - 將概念設計的產物轉換成真實資料庫管理系統的資料模式 (如:
關聯表格)-執行**正規化**
 - 需定義資料之間的**完整性限制**(Integrity Constraints)
- 實體設計(Physical Design)
- 系統實作(Implementation)

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
- 選用合適的資料庫系統(DBS)
- 邏輯設計(Logical Design)
- 實體設計(Physical Design)
 - 設計資料庫的內部儲存結構、檔案組織、索引...等
 - 評估的重點是**時間與空間的效率**
- 系統實作(Implementation)

- 需求收集與分析(Requirements Collection and Analysis)
- 概念設計(Conceptual Design)
- 選用合適的資料庫系統(DBS)
- 邏輯設計(Logical Design)
- 實體設計(Physical Design)
- 系統實作(Implementation)

E-R Model (E-R Diagrams)

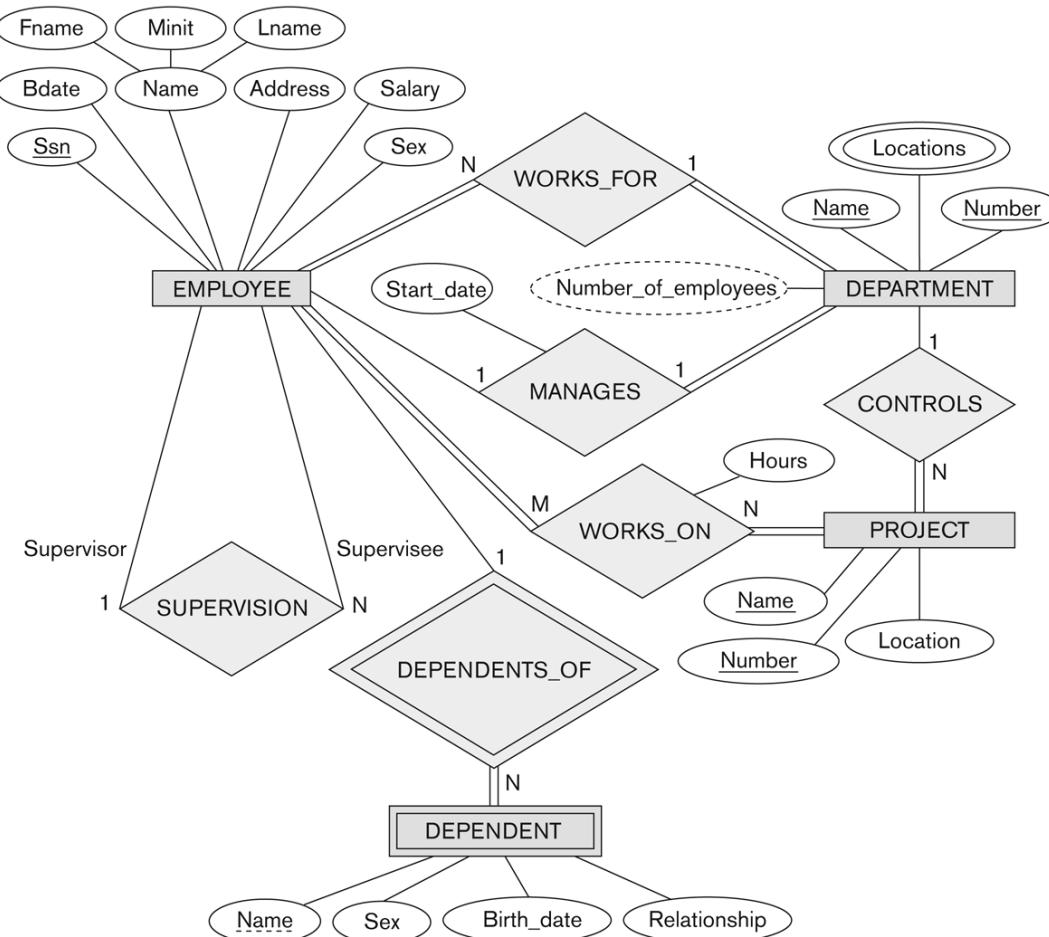


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

[

]

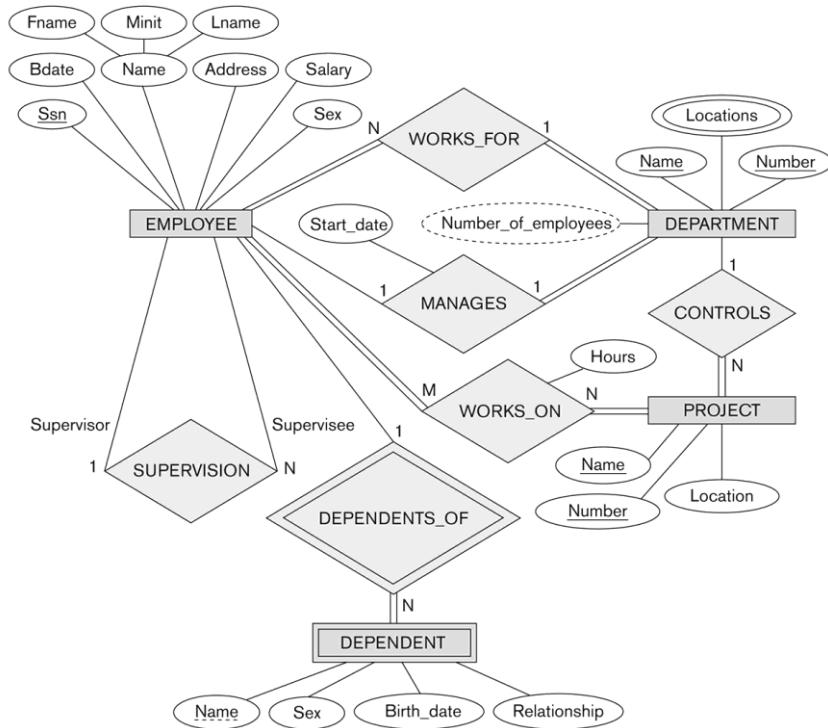
■ Def:

- E-R Model (Entity-Relationship Model; 個體關係模式)由M. E. Senko於1973年提出，1976年陳品山 (Peter P. S. Chen) 再度公開，為一高階的資料模式 (Data Model)，以圖形化的方式表示儲存於資料庫內的各資料項目之間的關係。
- 其產出的結果又可稱為E-R Diagram (ER圖)
- 可作為分析師與使用者溝通企業內部資料關係的工具。

Entity, Attributes, Relationships

E-R Model 主要包含：

- Entities (個體) – 方塊圖形表示
- Attributes (屬性) – 橢圓形表示
- Relationships (關係) – 菱形表示



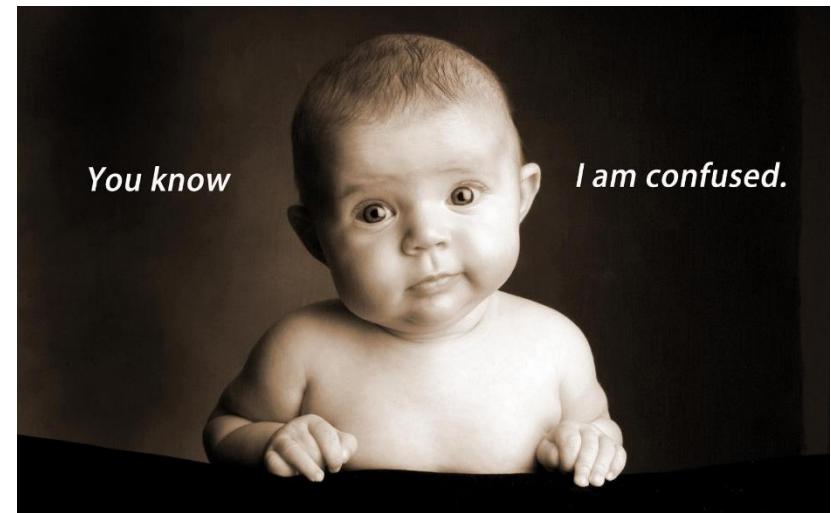
個體(Entities)

- 為E-R Model的最基本物件，是在真實世界中獨立存在的事物。
- 它可以是獨立存在的一個事物，或是一個概念上存在的物件。(如：學生，課程，...等)
 - 即：可以被明確識別之事物。
 - 在E-R Model中，以**方塊圖形**表示個體。
 - 個體中的每筆紀錄皆稱為一個**實例(Instance)**
 - 例如：“張三”的個人資料是“學生”這個Entity中的一個實例(Instance)

[

]

- 然而，有些講義會將Entity與Instance做不同的解讀：
 - Instance (實例) \Rightarrow Entity (個體)
 - Entity (個體) \Rightarrow Entity Type (個體型態)
- Instance和Entity，或是Entity和Entity Type一定是成對出現的。本講義是以Entity與Instance為主。



屬性(attributes)

- ❖ 用來描述個體的性質(Property)，而這些性質稱做**屬性**。
- ❖ 在E-R Model中，是以橢圓形符號表示。
- ❖ 屬性可分為下列五種型態：
 1. 一般屬性(General attributes)
 2. 複合屬性(Composite attributes)
 3. 多值屬性(Multivalued attributes)
 4. 衍生屬性(Derived attributes)
 5. 鍵值屬性(Key attributes)

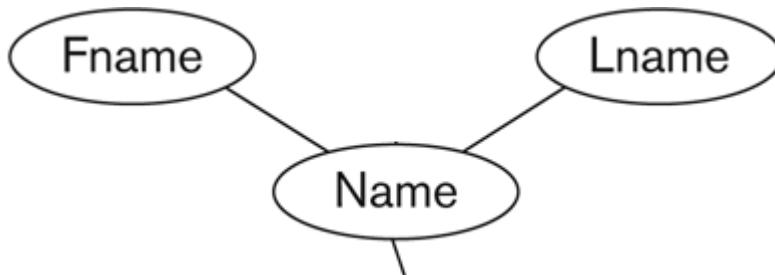
一般屬性(General attributes)

- 對於Entity而言，大多數的一般屬性是單一個值，像這樣的屬性又稱為**單值屬性 (Single-Value Attribute)**。
- 每個一般屬性的值不可以再分割，故又稱為**簡單屬性 (Simple Attribute)**或是**不可分割屬性 (Atomic Attribute)**
 - Ex.: 性別Sex
 - 是否為Atomic可由實際情況來判定
- 此屬性在E-R Model 的圖示為單環橢圓形：



複合屬性(Composite attributes)

- 若屬性中的值是可以分割成更小部份的屬性值，且分割的部份有自己獨立的意義，為更基本的屬性值。
 - 如：“學生”這個個體中包含一個“姓名”的屬性，而“姓名”這個屬性的值可以分別由“姓”、“名”這兩個屬性的值來組成。
- 此屬性在E-R Model 的圖示為：



多值屬性(Multivalued attributes)

- 該類屬性中，可包含一個以上的值。
 - 例如：“學生”這個個體中有一個“電話”的屬性。一個學生可能有很多支電話號碼，表示這個屬性有很多可能的值。
- 此屬性在E-R Model 的圖示為雙環橢圓形：



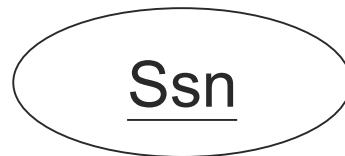
衍生屬性(Derived attributes)

- 這個屬性的值可以 “完全” 由其它屬性之值計算出來。因此可以不用實際存在於資料庫中。
 - 如：“學生”這個體中不用實際存放一個的“年齡”屬性，因為這個屬性的值可以完全由其“生日”計算出來。
 - 實作上而言，可以不用畫在E-R Model上。
 - 僅需記得當資料庫建構出來之後，若有使用者需要此資料時，需協助計算出來，並將此計算結果放入該使用者的View中。
- 此屬性在E-R Model 的圖示為虛線橢圓形：



鍵值(Key)或鍵值屬性(Key attributes)

- 鍵值是用來識別個體中每一個不同實例的屬性。
 - 該屬性具有唯一性，如：主鍵或候選鍵。
 - Key可由一個或一個以上的屬性構成，要滿足唯一性與最小性。
- 此屬性在E-R Model 的圖示為名稱下加底線之橢圓形：



- 外來鍵在繪製ER圖時，通常無法明確得知，因此不會被畫出；但是在未來從事資料庫正規化設計時即會被表示。

關係(Relationship)

- 是指不同個體(Entities)間的一種結合(Association)或關係集合(Relationship Set)，它表達了不同個體之間所隱含的關聯性



■ 有一些和關係相關的名詞要注意：

- 關係型態的階度 (Degree)
- 基數率 (Cardinality Ratio)
- 參與限制 (Participation Constraint)
- 關係型態的屬性
- 辨認關係 (Identifying Relationship)

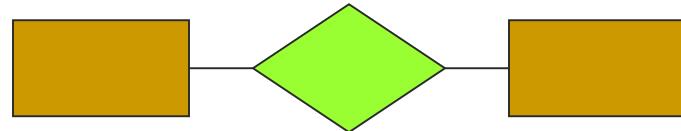
關係型態的階度 (Degree)

- 關係中所參與連接的個體的數目。

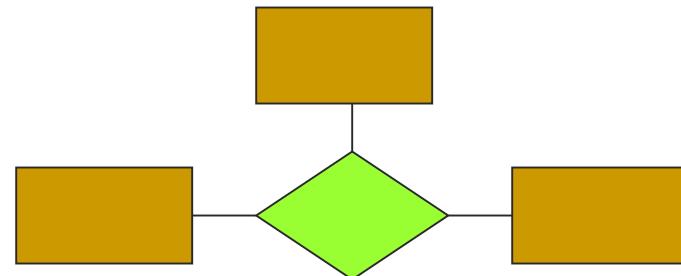
- 一元階度 (Unary Degree)



- 二元階度 (Binary Degree)



- 三元階度 (Ternary Degree)



- N元階度 (N-ary Degree)

基數率 (Cardinality Ratio)

- 代表不同個體之間在參與某一個關係時，個體間的實例相互參與之數量比。
- 常見的基數率有三種：

- 一對一關係(1:1)



- 一對多關係(1:N)



- 多對多關係(M:N)



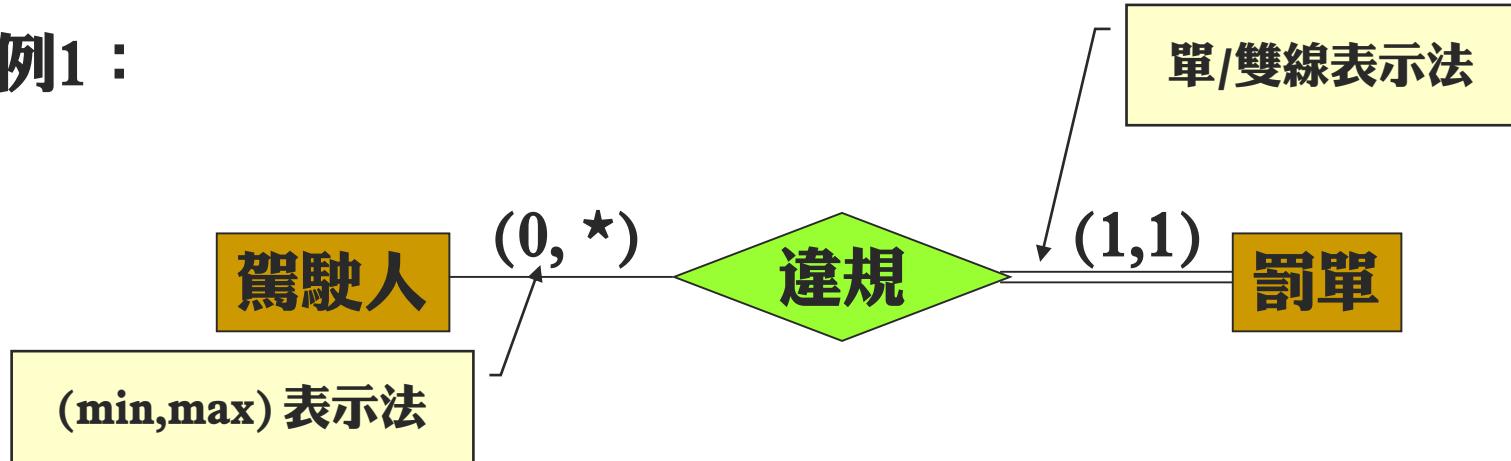


- **一對一(One-to-one, 1 : 1)**
 - 一對一關係是指參與關係的兩個個體之基數都為1。
- **一對多(One-to-many, 1 : N)**
 - 一對多關係是指關係一邊的一個個體實例與關係的另一邊的多個個體實例有參與關係。
- **多對多(Many-to-many, M : N)**
 - 多對多關係是指參與關係的兩個個體之基數都大於一。

參與限制 (Participation Constraint)

- 是指某個體中的所有實例，是否一定需要依靠參與某關係，並和另一個體產生關聯而存在。
- 參與限制分為兩種：
 - 全部參與 (Total Participation)
 - 是指某個體中的所有實例，需要依靠參與某關係，並和另一個體產生關聯而存在。即：必須全部參與該關係。
 - 部份參與 (Partial Participation)
 - 是指某個體中的實例，不需要全部依靠參與某關係，並和另一個體產生關聯而存在，只要部份參與即可。

■ 例1：



- “罰單”中的每一個實例必須**全部參與**“駕駛人”的“違規”關係。 \because 每一張罰單一定是因為駕駛人違反了交通規則所開出的!!
 \therefore 罰單中參與此關係的實例最少有一個，最多為1個
- “駕駛人” 中的每一個實例**僅須部份參與**“罰單”中的“違規”關係。 \because 不一定每一個駕駛人都會違規!!
 \therefore “駕駛人”中參與此關係的實例最少有0個，最多為 \star 個 (多個)

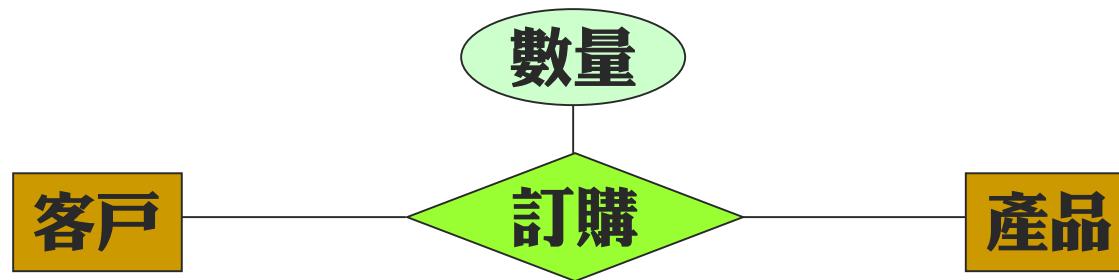
■ 例2：



- 本範例中，“客戶” 中的每一個實例僅須部份參與“試用產品”的“試用” 關係。 \because 客戶不見得每次都會執行試用的動作!!
 \therefore 客戶中參與此關係的實例最少有0個，最多為*個(多個)
- 本範例中，“試用產品” 中的每一個實例必須全部參與“客戶”的“試用” 關係。 \because 每一個試用產品一定是被客戶試用的!!
 \therefore 試用產品中參與此關係的實例最少有1個，最多為*個

關係型態的屬性

- 實務上，在關係型態上也可能有一些屬性存在。
 - 這些屬性當該關係進行時也會跟著改變。
 - 亦即關係進行時所使用到的屬性。
- 例如：



弱個體 (Weak Entity)

❖ 個體(Entity)依其存在條件又可分為兩類：

- 一般個體(Regular Entity) (或稱強勢個體 (Strong Entity))
- 弱個體(Weak Entity) (或稱附屬個體(Subordinate Entity))

Employee

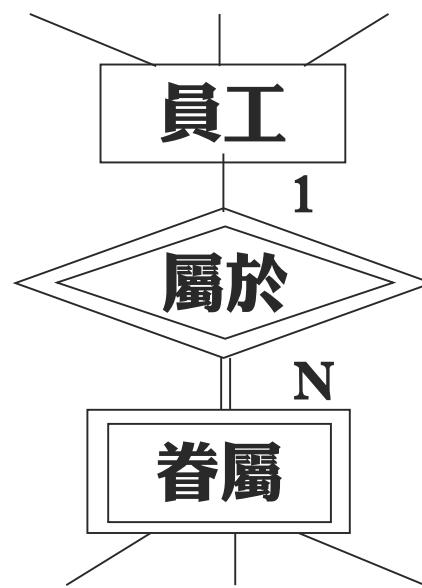
Dependent

一般個體

弱個體

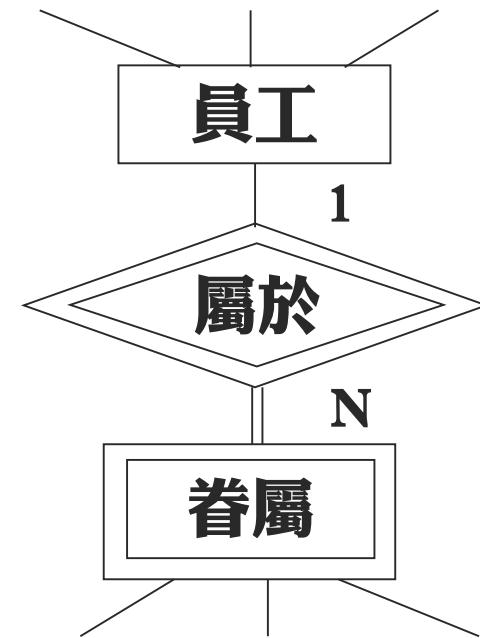
弱個體(Weak Entity):

- 弱個體必須倚賴其他的個體才能存在。如果某一弱個體所倚賴的個體消失了，該弱個體也會一併消失。
 - 例如：資料庫中有「員工」和「員工眷屬」兩個個體。



辨認關係 (Identifying Relationship)

- 弱個體與其所依附的個體之間的關係型態，稱為弱個體的辨認關係 (Identifying Relationship)。
 - 例如：員工與眷屬
- 弱個體與其識別關係一定為完全參與限制，因為弱個體在沒有擁有者個體的情況下是無法被識別的。



[

]

- ❖ 在E-R Model中，唯一可辨識與同一擁有者有關之弱個體實例的屬性，稱為**部份鍵 (Partial Key)**，其圖示為在該屬性名稱下方加上**虛底線** (實務上也可用**實底線**)
 - ❖ 假設同一個員工的眷屬，不會有相同的名字。

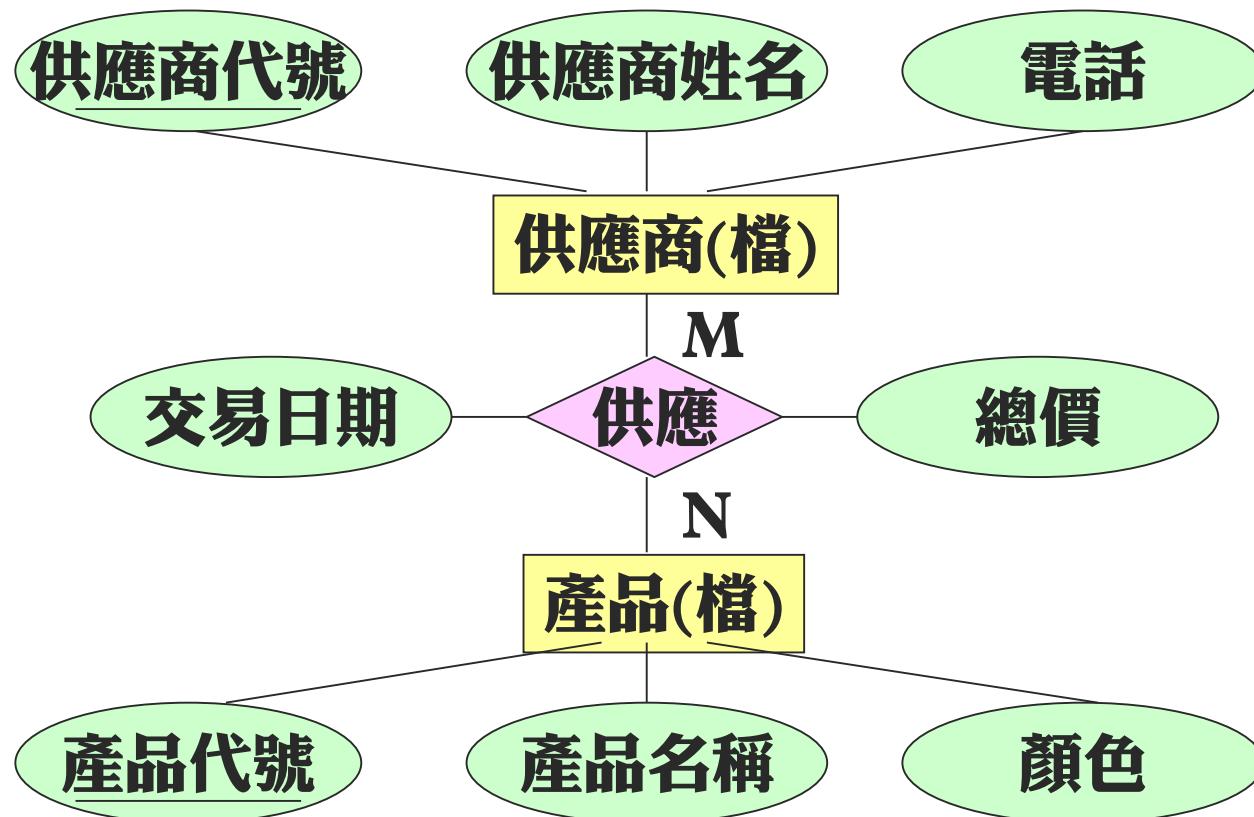


Figure 3.14
Summary of the notation for ER diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

■ 繪製E-R Diagram

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

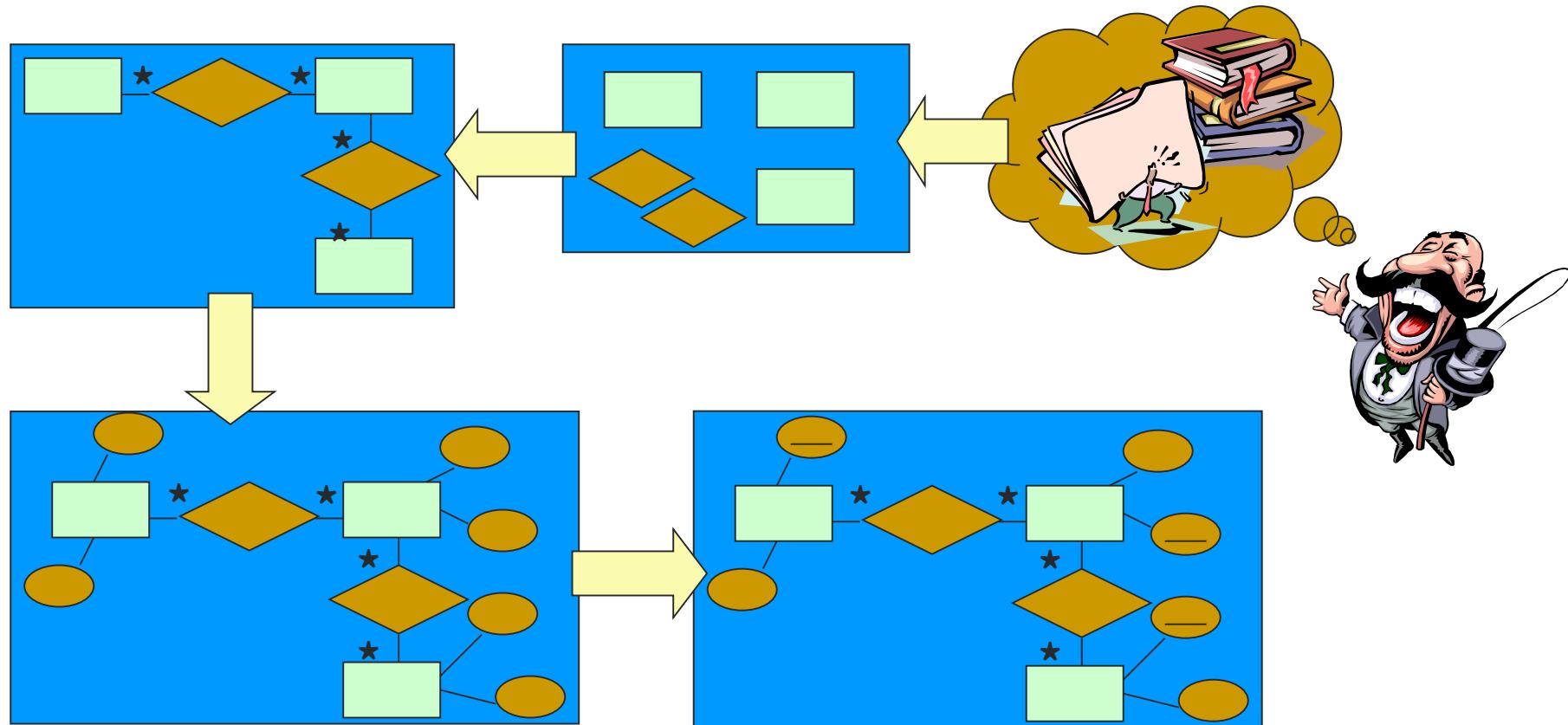


利用E-R Model從事資料庫概念設計的步驟

■ 為反覆改良的過程：

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(Attribute)**。
4. 定義**鍵值屬性 (Key Attribute)**。

E-R model 的設計步驟



如何繪製E-R圖

■ 繪製的資料依據：

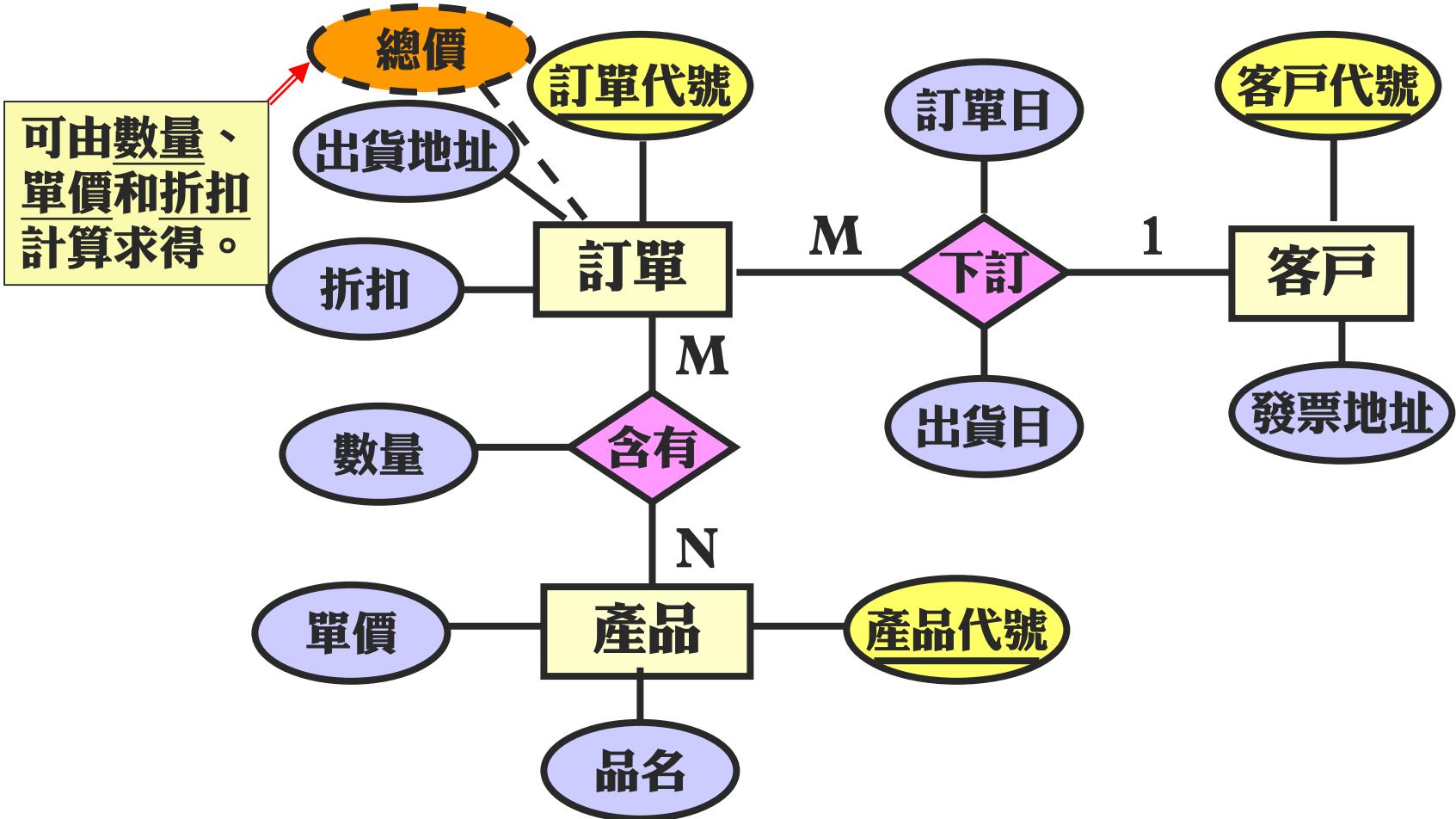
- 高階使用者需資訊化的**企業實際表單**
- 系統開發者以**訪談**的方式，將高階使用者所面對的問題及需求製成**文字型式**的記錄。然後**再把文字轉成E-R圖**。

以企業實際表單做為轉換依據

<input checked="" type="checkbox"/>	聯合科技公司			
<input checked="" type="checkbox"/>	訂單代號：	客戶代號：		
<input checked="" type="checkbox"/>	訂單日：	發票地址：		
<input checked="" type="checkbox"/>	出貨日：	出貨地址：		
<input checked="" type="checkbox"/>	折扣：	-----		
<input checked="" type="checkbox"/>	<u>產品代號</u>	<u>品名</u>	<u>單價</u>	<u>數量</u>
<input checked="" type="checkbox"/>	0002	隨身碟	1022	3
<input checked="" type="checkbox"/>	0018	印表機	3186	5
	⋮	⋮	⋮	⋮
		總價：	<input type="text"/>	
衍生屬性				

[

]



若有多張表單時如何處理？

公司名稱：[REDACTED]		退貨單		單據日期 097.09.14 單據編號 [REDACTED] 買 次 1-1	
品名規格	數量	單位	單價	金額	第二聯：客戶聯
台紐三麗奶粉	1.0	包	3,500.0	3,500.0	
公司名稱：[REDACTED]		出貨單		單據日期 097.07.05 單據編號 [REDACTED] 買 次 1-1	
品名規格	數量	單位	單價	金額	第二聯：客戶聯
台紐三麗奶粉	1.00	包	3,500.00	3,500	
梨加工草莓	2.00	罐	165.00	330	
大雞巧米	1.00	罐	170.00	170	
安佳PZ赫	1.00	KG	270.00	270	
麻糬15入	1.00	盒	35.00	35	
美國瓜子仁	1.00	斤	130.00	130	
小哈雷	1.00	支	100.00	100	
數錯(50入)?	1.00	盒	80.00	80	
1.請確認上列貨物數量無誤，並蓋章簽效。 2.上列貨物若有疑問，請速連絡TEL: [REDACTED] 3.買方在貨款未付清之前，貨物所有權歸賣方所有。				合計金額 4,615	答 案
會計	審核	經手人	審 核		

以訪談資料做為轉換依據

- ❖ 文字紀錄轉換成E-R圖通常有以下幾個對應原則：
 - 名詞：
 - 句子中的「名詞」可對應成「實例」，「名詞的集合」可對應成「個體」。
 - 動詞：
 - 「動詞」可對應成「關係」。
 - 形容詞：
 - 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」。
 - 副詞：
 - 句子中描述動詞的「副詞類」可對應成「關係上的屬性」。

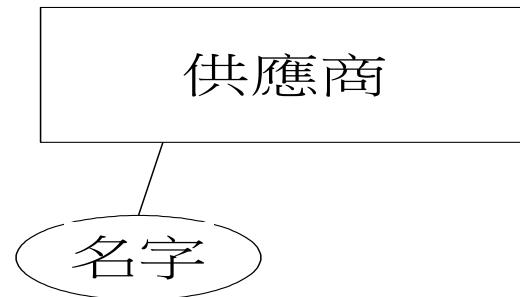
- 句子中的「名詞」可對應成「實例」，「名詞的集合」可對應成「個體」。
- 「動詞」可對應成「關係」
- 例如：“**供應商 A, B, C可以分別供應我們產品 P, Q, R**”
 - A, B, C是供應商的「名詞」，P, Q, R是產品的「名詞」
 - 「供應商」和「產品」為名詞集合
 - 「供應」為動詞



[

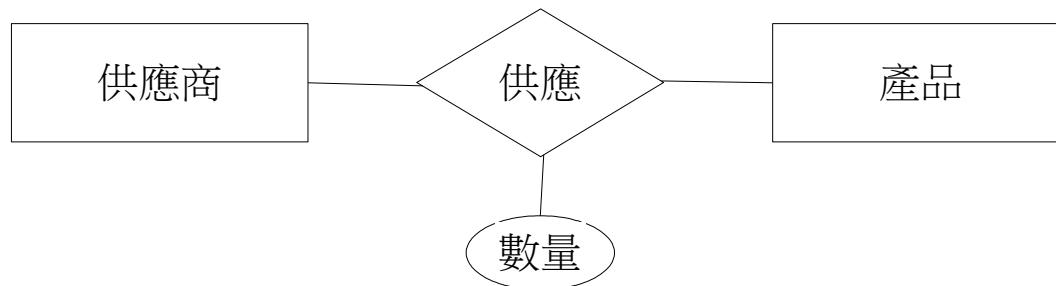
]

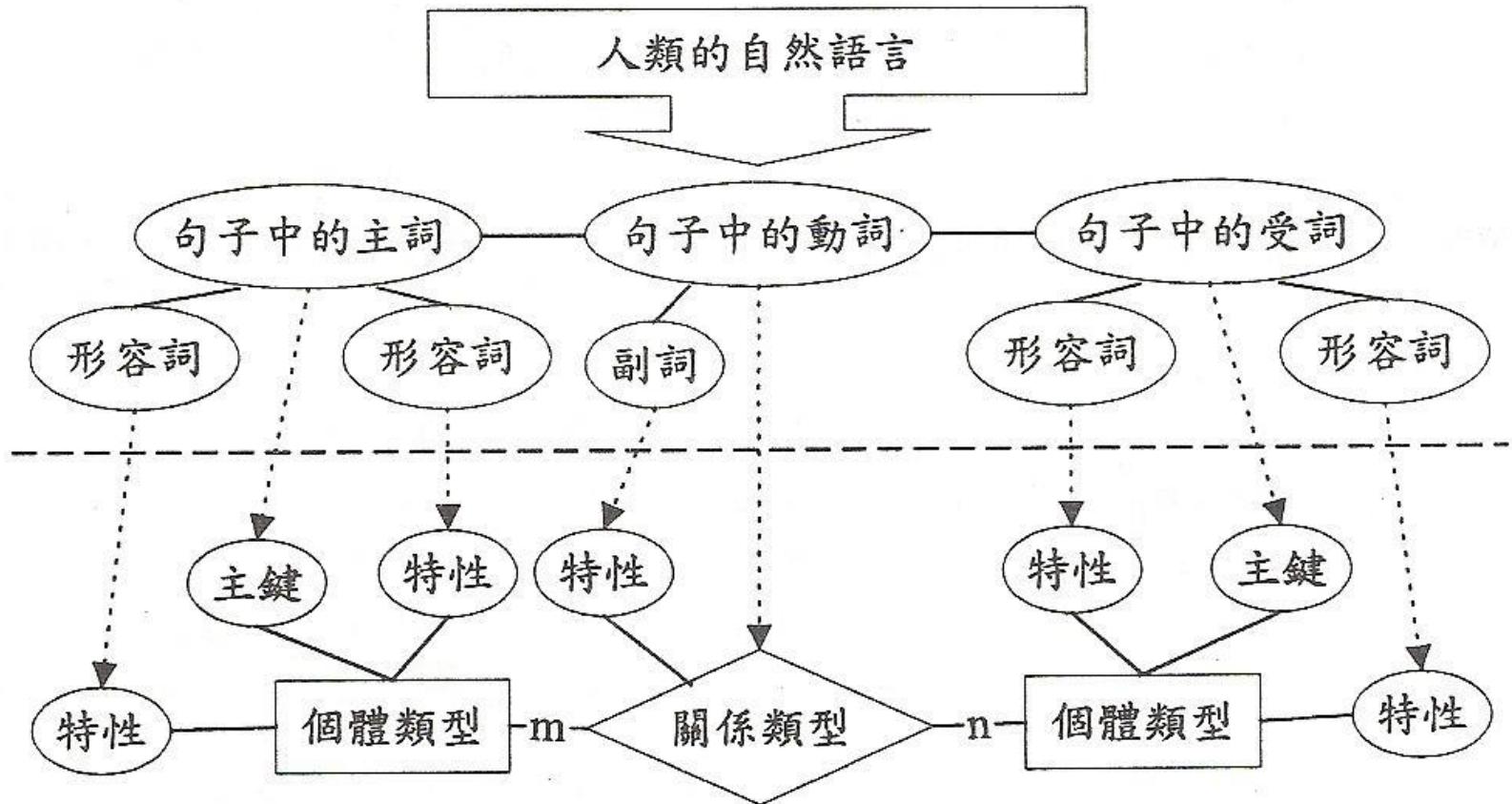
- 句子中描寫名詞的「形容詞」或「所有格」可對應成個體型態的「屬性」。
 - 例如：“供應商 C 的名字為華北企業公司”
 - 「供應商C」為名詞（名詞集合），「名字」為所有格



■ 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 A 供應數量 W 的產品 P”
 - 「數量W」為副詞





自然語言與個體關係模式中各元素的對應關係

如何說明所繪製的E-R圖

- 找出所有個體，敘述每一個個體之屬性
 - 包含一般屬性、鍵值屬性、複合屬性、多值屬性等。
- 找出各個關聯，以及參與此關聯的個體
 - 依照關聯之敘述，判斷個體間之一對一、一對多、或多對多的邏輯關係、主受詞等，用白話加以說明。
 - 可補充說明數量關係。

※練習範例※

- 請說明slide 40的E-R Model。

Ans:

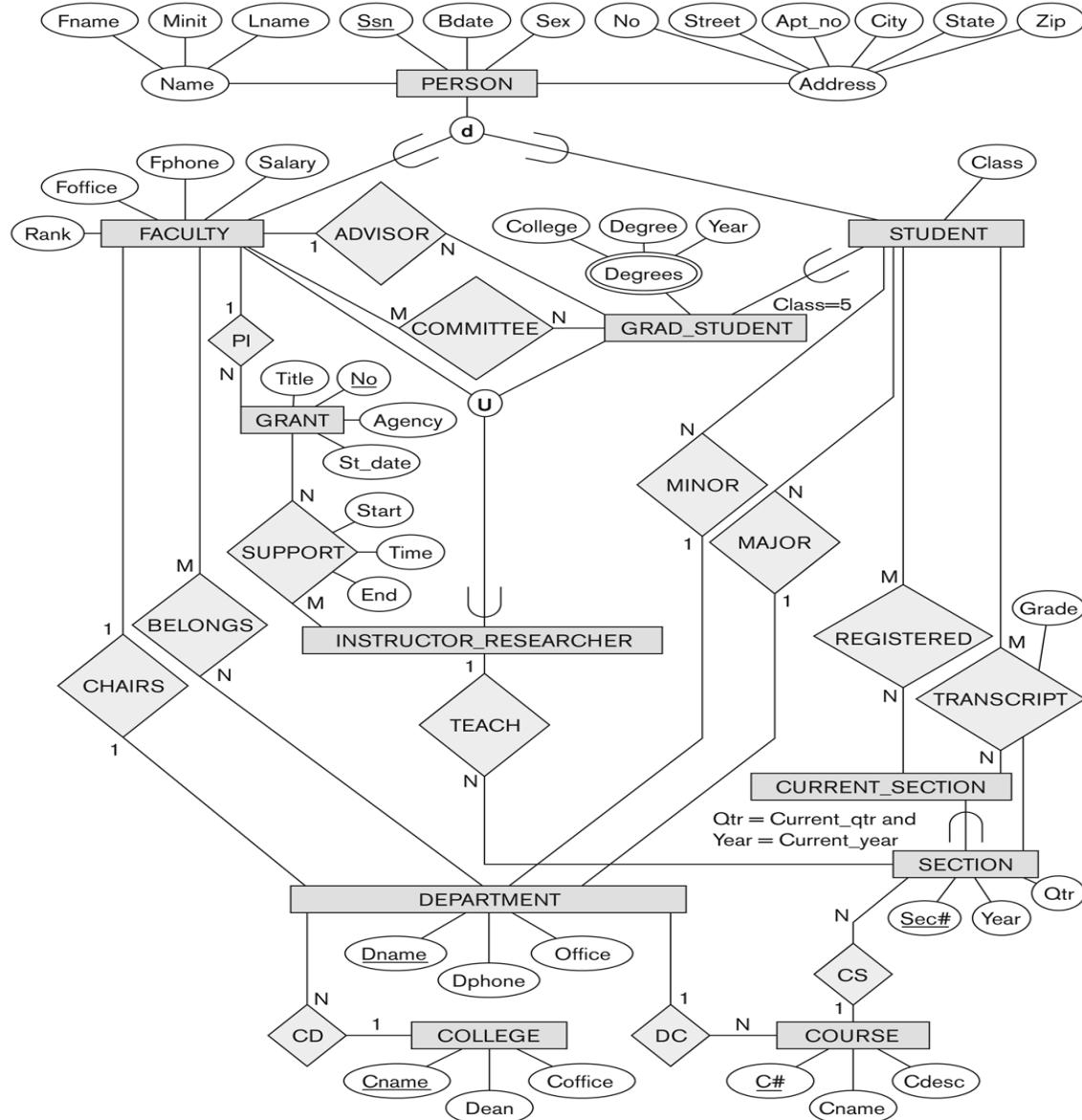
- 有三個個體：訂單、客戶、產品
- 訂單有四個屬性：訂單代號、出貨地址、折扣、總價。其中訂單代號是唯一的，而總價是衍生屬性。
- 客戶有兩個屬性：客戶代號、發票地址。其中客戶代號是唯一的。
- 產品有三個屬性：產品代號、品名、單價。其中產品代號是唯一的。
- 一個訂單只能屬於一個客戶，但一個客戶可下多個訂單。
- 一個訂單中可能有多個產品，且一個產品可以出現在多個訂單。
- 每張訂單在客戶下訂時需列出訂單日與出貨日。
- 訂單中的每項產品需列出數量。

延伸性個體-關係模式 (Enhanced E-R Model; EE-R Model)

- 先前所討論的E-R Model，已足以表示許多傳統資料庫中、資料個體間的關係。然而，仍然有些關係E-R Model無法表示(如：**繼承關係**)。因此需加以補強。
- 延伸性個體-關係模式也可以稱為擴充性個體-關係模式 (Extended E-R Model；EE-R Model)。EER模式為E-R Model之強化，除了ER模式所能表達的個體關係外，亦多加入了一些重要性質。

Figure 4.9

An EER conceptual schema for a UNIVERSITY database.

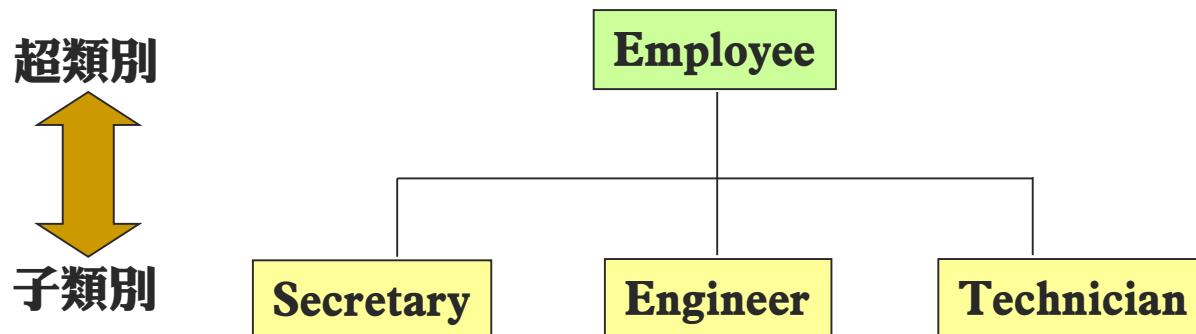


[

]

■ EER模式主要包含：

- E-R Model
- 類別 (Class) 概念：超類別(Superclass)與子類別(Subclass)
- 一般化(Generalization)與特殊化(Specialization)



■ 類別 (Class)

■ Def:

- 類別為一組個體的集合(A set of entity)。該集合中的每個個體可能具有不同的類型，而這些不同類型的子個體集合本身是有意義的。這時我們可以用類別來表達之間的關係。

■ 如：

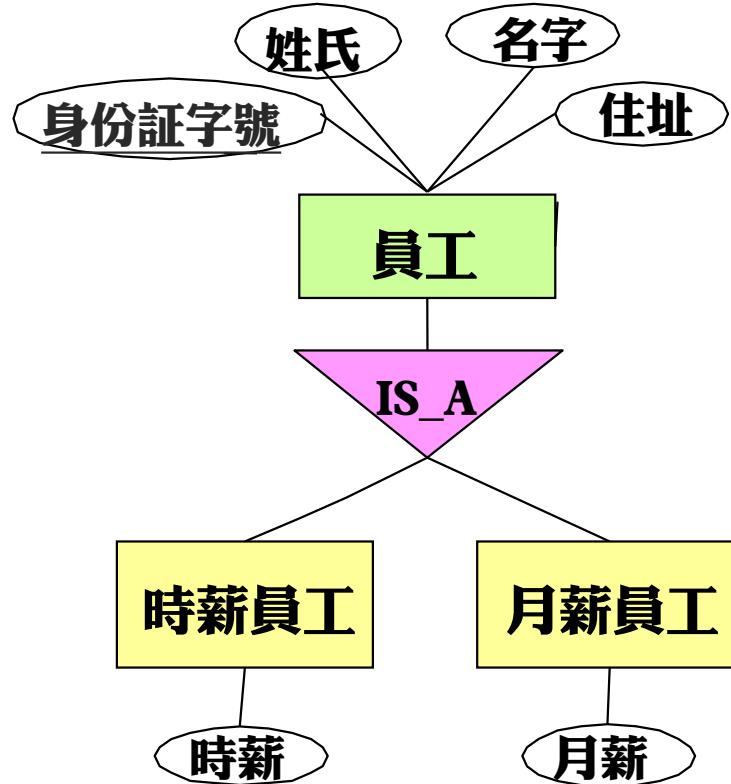
- Employee可以再分成Secretary, Engineer, Manager, Technician, Hourly_Employee, Salaried_Employee, ...等。

超類別(Superclass)與子類別(Subclass)

- 子類別(Subclass)為其超類別(Superclass)之子集合。
 - 如上頁例子中，Employee為Secretary, Engineer, Technician的超類別；反之，這三個類別為Employee的子類別。
 - 一個個體可能有任意數目的子類別。

類別(Class)的繼承

- 超類別/子類別擁有屬性繼承 (Attribute Inheritance) 的關係，即子類別可以繼承其超類別所擁有的所有屬性。
 - 例：
 - Employee可能擁有的屬性：姓名，員工編號，住址，電話，性別
 - Secretary可能擁有的屬性：(Employee所有屬性)，打字速度
 - Engineer可能擁有的屬性：(Employee所有屬性)，專長類別
 - Technician可能擁有的屬性：(Employee所有屬性)，証照等級
 - 由於屬於子類別中的每個實例，同時也必定屬於其超類別，因此超類別所具有的所有屬性，子類別也會擁有相同的屬性，而子類別會擁有其它獨特的屬性。



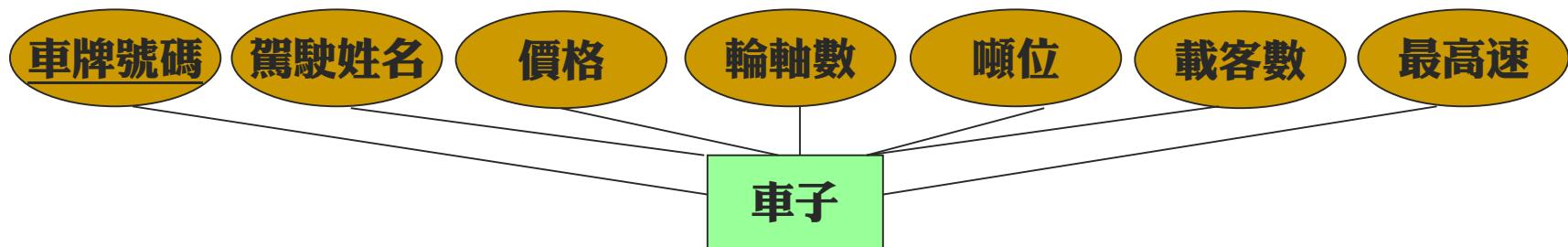
- 超類別/子類別的關係又稱為**IS-A關係**。

■ 一般化(Generalization)和特殊化(Specialization)

- ❖ 「**一般化**」是用來強調各個個體間的**共同性**。
- ❖ 「**特殊化**」是用來強調各個個體間的**差異性**。

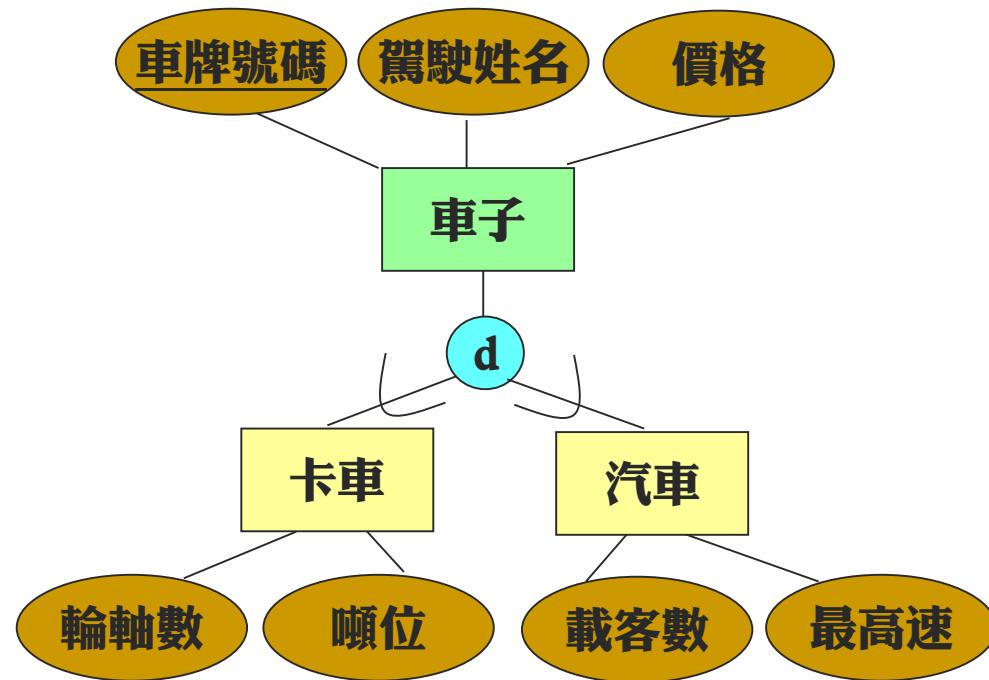
特殊化(Specialization)

- 特殊化是定義某個體中一群子類別的程序。此個體稱為該特殊化後的超類別。
- 範例：在E-R Model中，公司的“車子”個體如下：



車牌號碼	駕駛姓名	價格	輪軸數	噸位	載客數	最高速
1010	陳小明	2,500,000	NULL	NULL	4	200
2233	王大頭	960,000	4	100	NULL	NULL
:	:	:	:	:	:	:

- 上述例子中，“車子”個體中的所有實例可區分為兩大類：“卡車”、“汽車”。
- 因此，我們可以將車子特殊化 (Specialization) 為卡車和汽車兩個子類別。

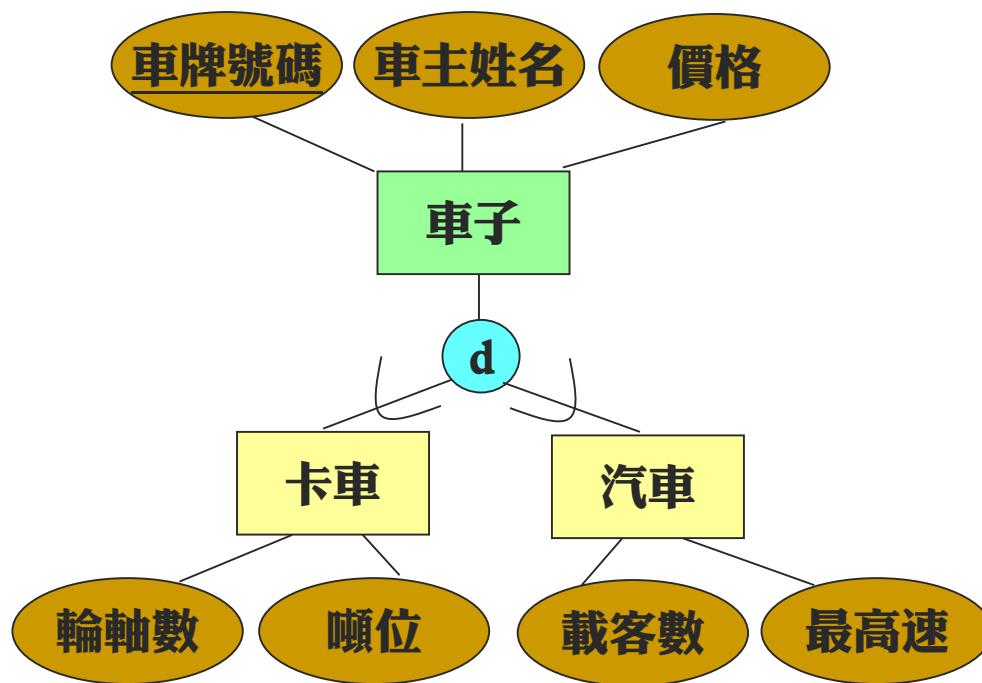


特殊化關係的限制

- 分離限制(Disjoint Constraints)
- 重疊限制(Overlap Constraints)
- 完整性限制(Completeness constraints)

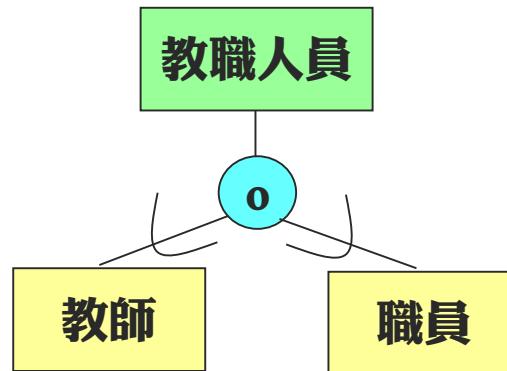
分離限制(Disjoint Constraints)

- 所有子類別中的個體實例**不可重複**，即一個個體實例最多只能屬於一個子類別。以符號 d 表示



重疊限制(Overlap Constraints)

- 子類別中的個體實例可能重複屬於多個特殊化的子類別。以符號 \circ 表示。



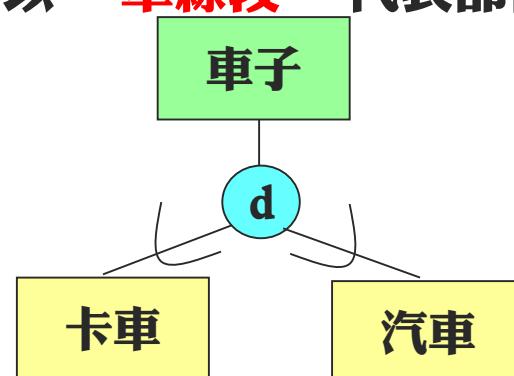
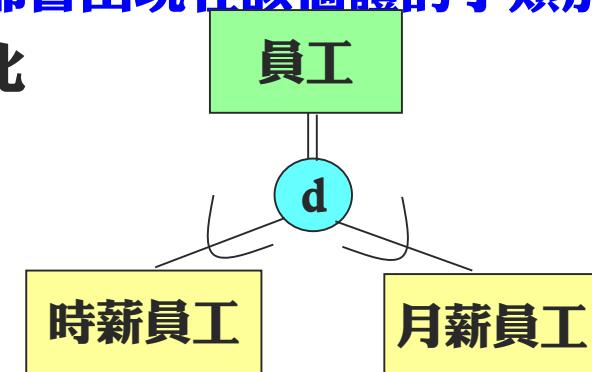
完整性限制(Completeness constraints)

「全部特殊化」(Total specialization) :

- 對某一個個體做特殊化處理時，在該個體中的每個實例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的子類別個體中。以“**單線段**”代表部份特殊化



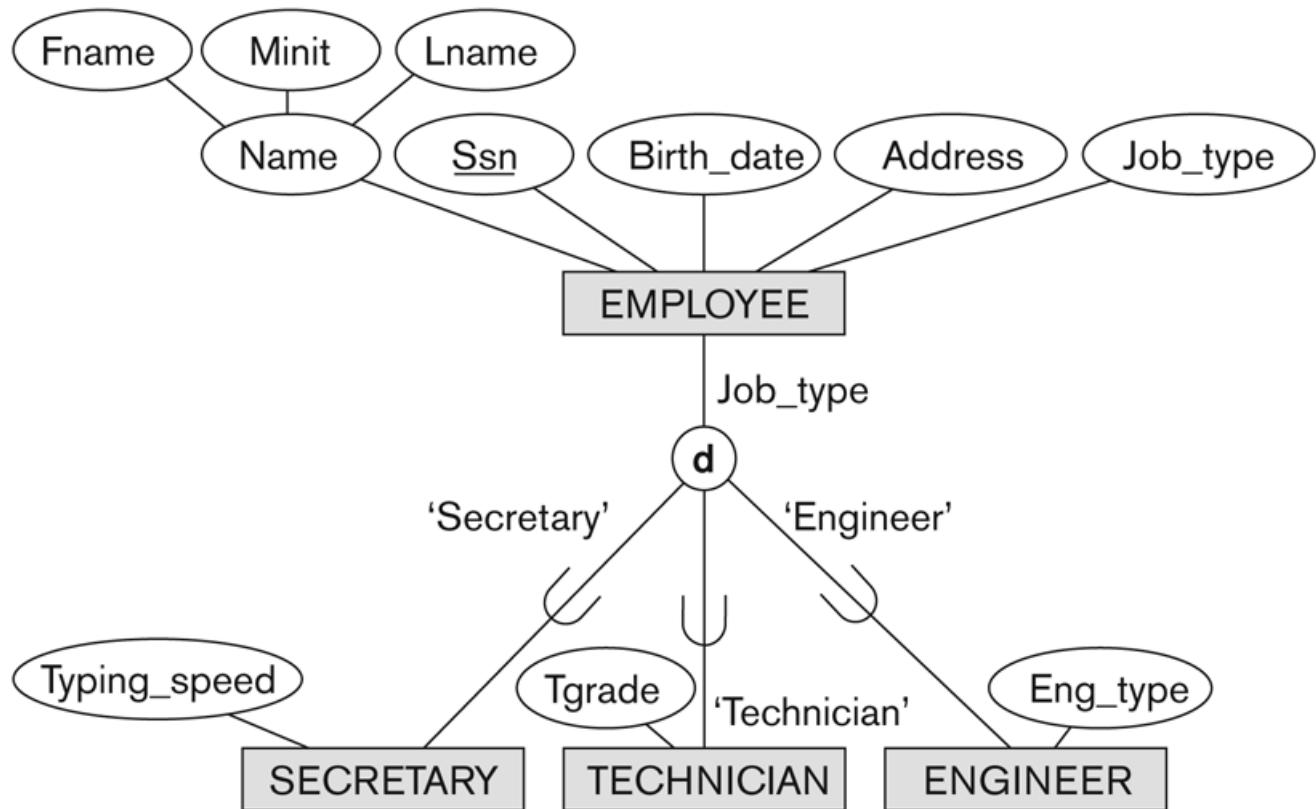
[

]

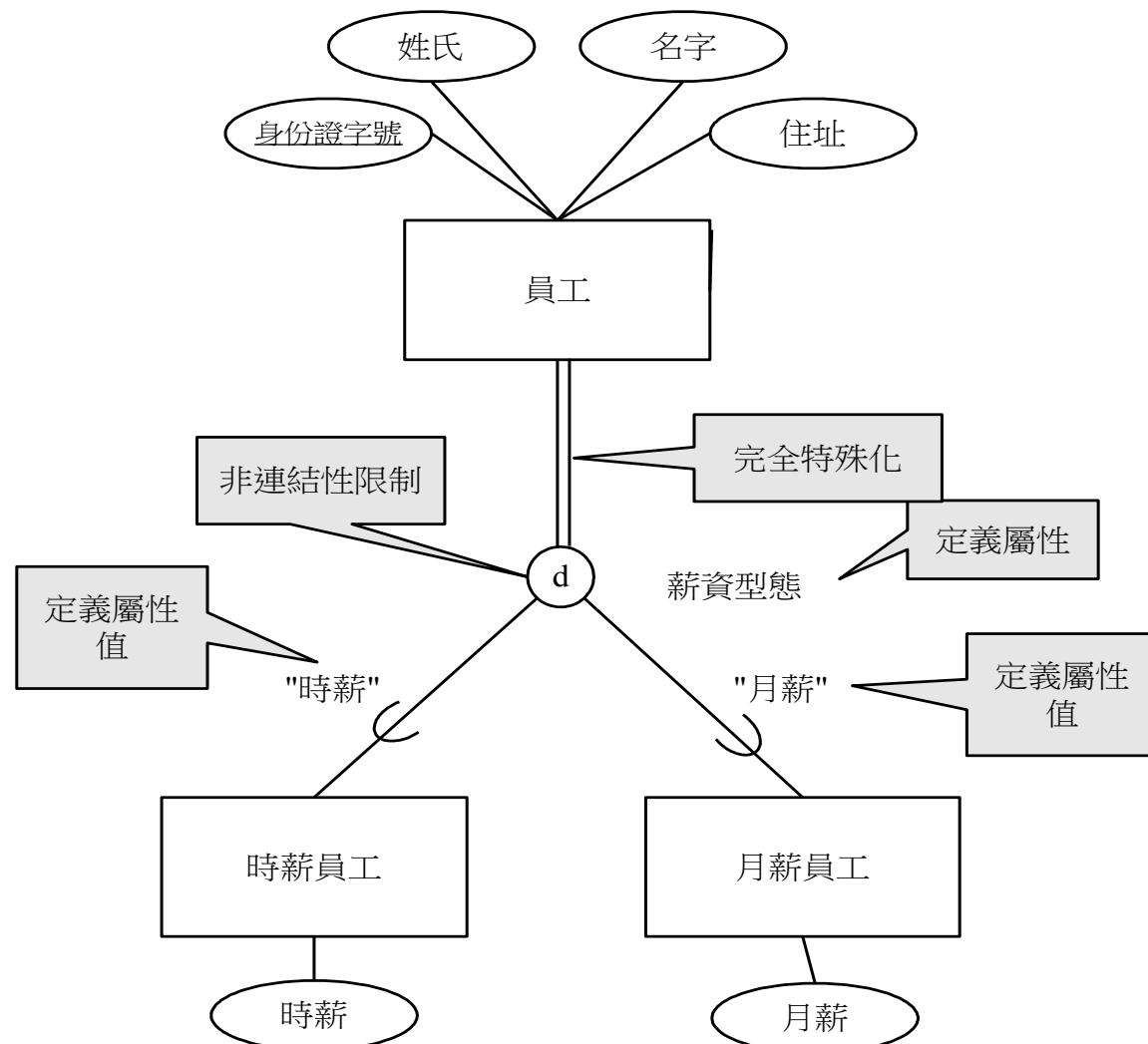
- 有時，我們是根據**某個特定屬性**來做特殊化。此時通常會在EE-R Model上註明特殊化的屬性為何。

Figure 4.4

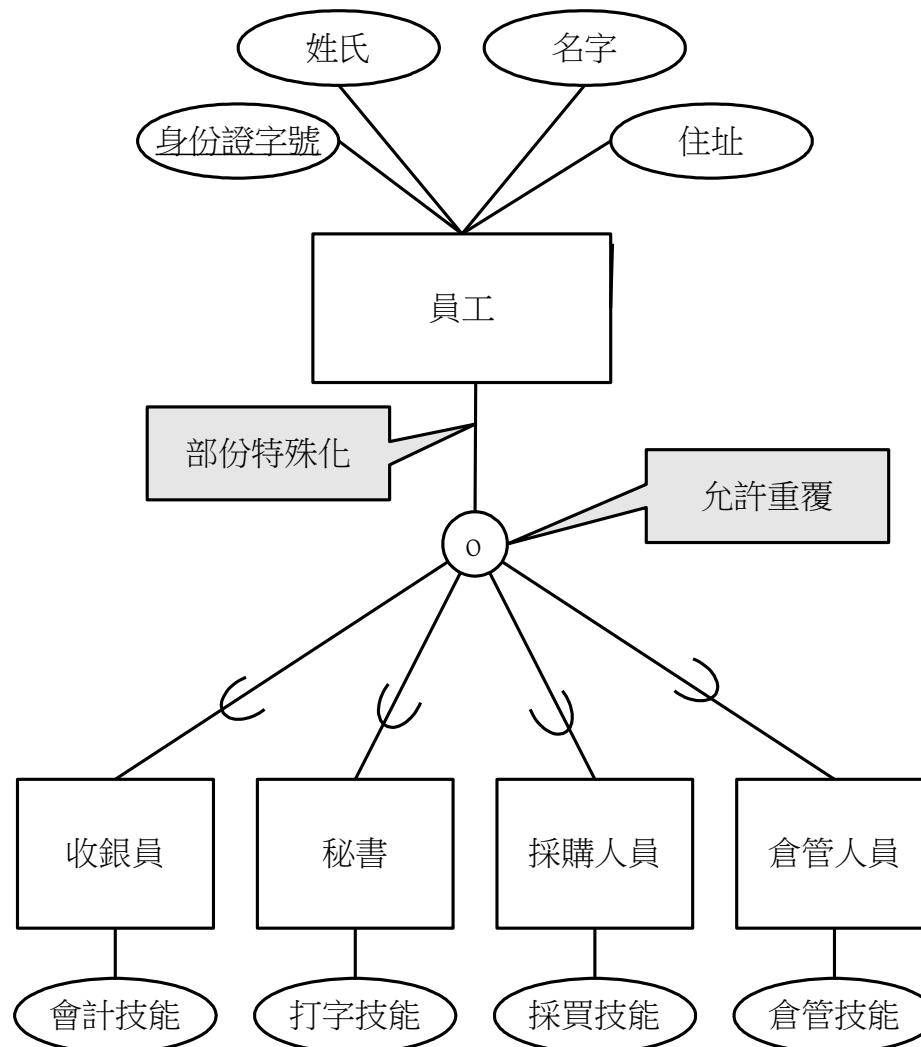
EER diagram notation
for an attribute-defined specialization
on Job_type.



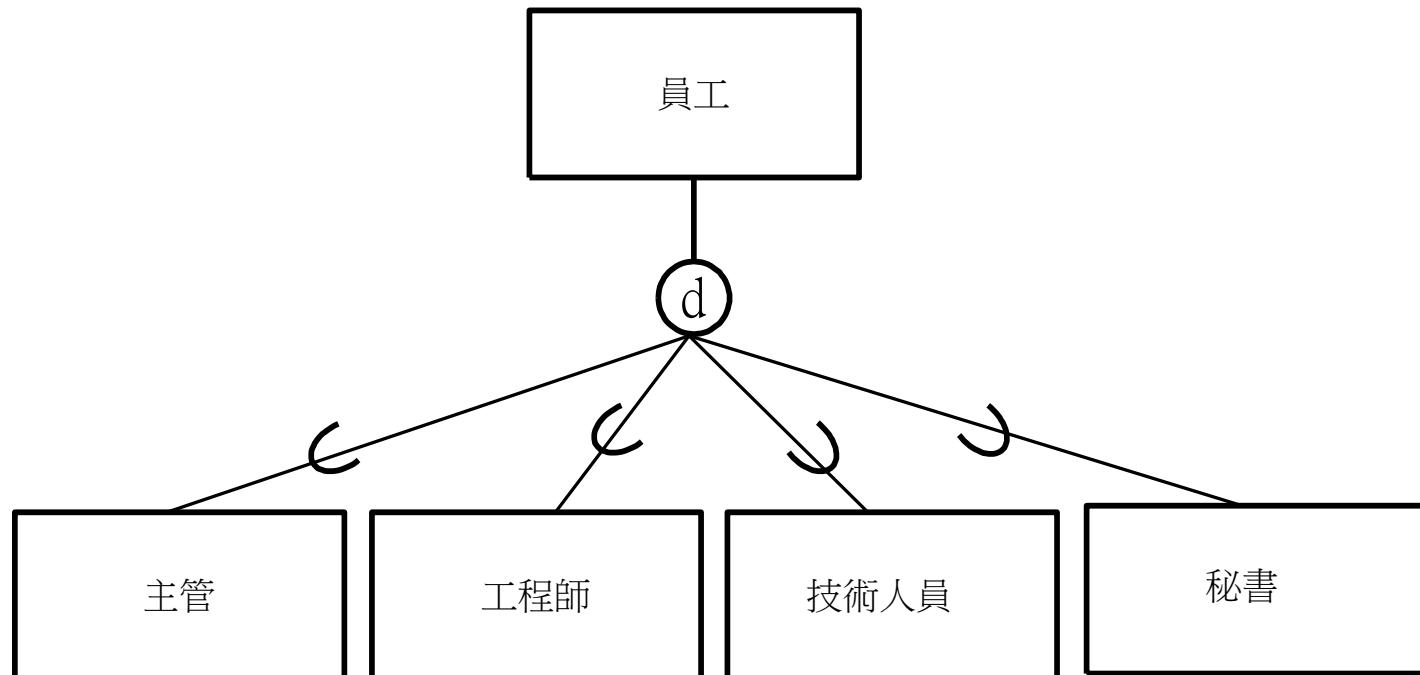
「分離限制」與「全部特殊化」的例子



「部份特殊化」與「重疊限制」的例子



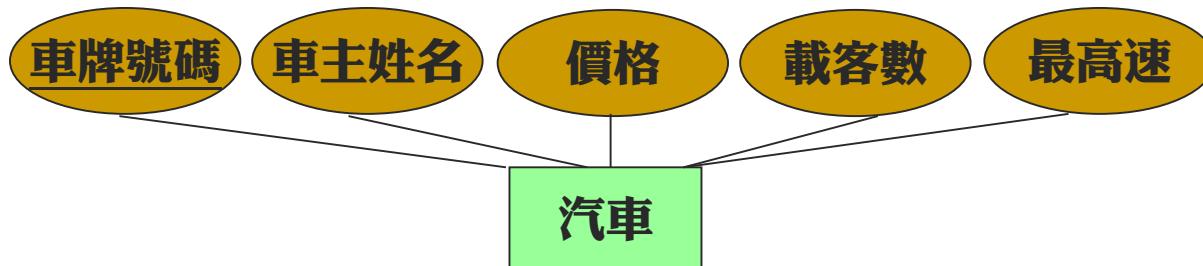
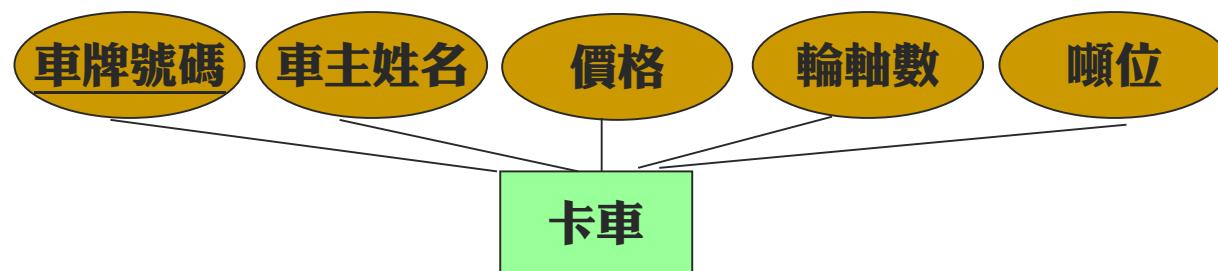
「部份特殊化」與「分離限制」的例子



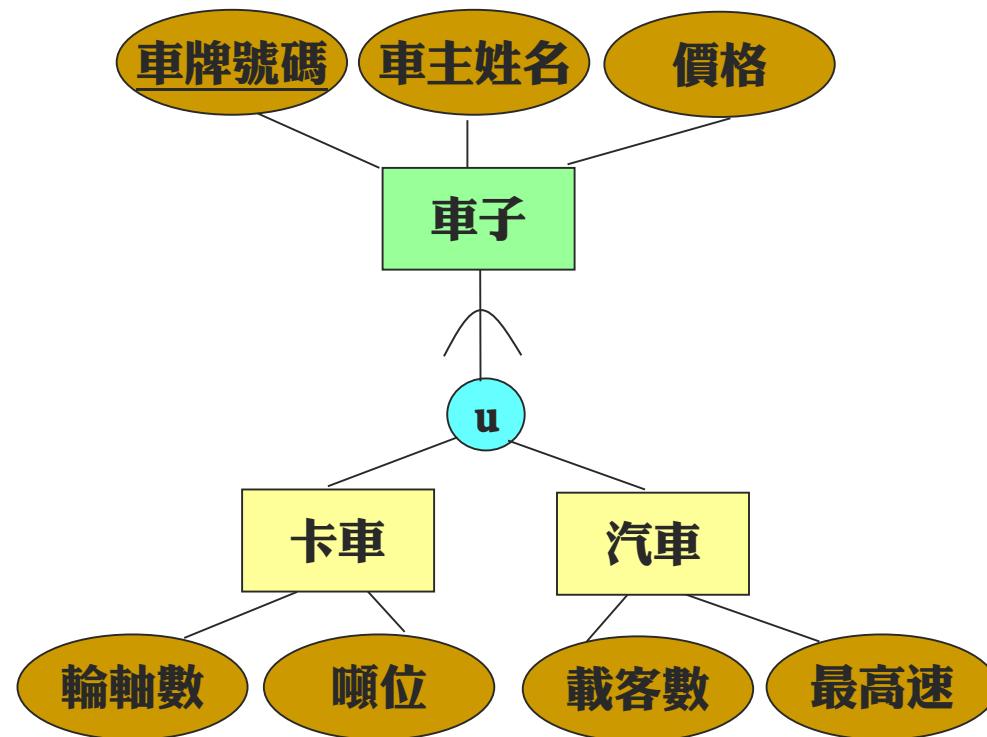
「部份特殊化」與「分離限制」的例子

一般化(Generalization)

- 將數個不同的個體抽離出相同的屬性，再把這些屬性**一般化為超類別**。即多個子類別可一般化為一個超類別。
- 範例：在E-R Model中，公司有兩個個體如下：



- 我們可以將上述兩個個體一般化 (Generalization) 為車子這一個超類別。
- 符號 **u** 表示一般化的**聯集(Union)**關係。



完整性歸類(Completeness Category)

■ 完整性歸類分為**全部 (Total) 歸類**與**部份 (Partial) 歸類**。全部歸類以雙平行線段表示，部份歸類採單一線段表示。

- 某些個人戶、某些國內客戶及某些國外客戶成為公司真正的客戶。
- 學校所有的職員、學生、教師都是借閱人，一律發給圖書借閱証。
- 若僅有部份子個體為全部歸類，則雙平行線繪於聯集符號與該子個體之間；若所有子個體為全部歸類，則雙平行線可繪於聯集符號與超個體之間。

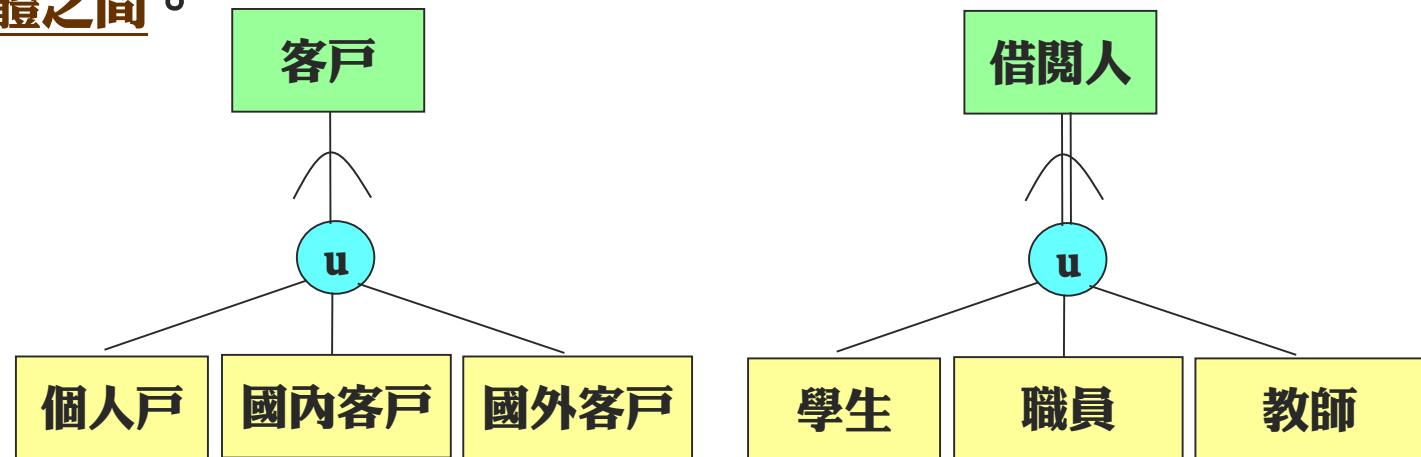
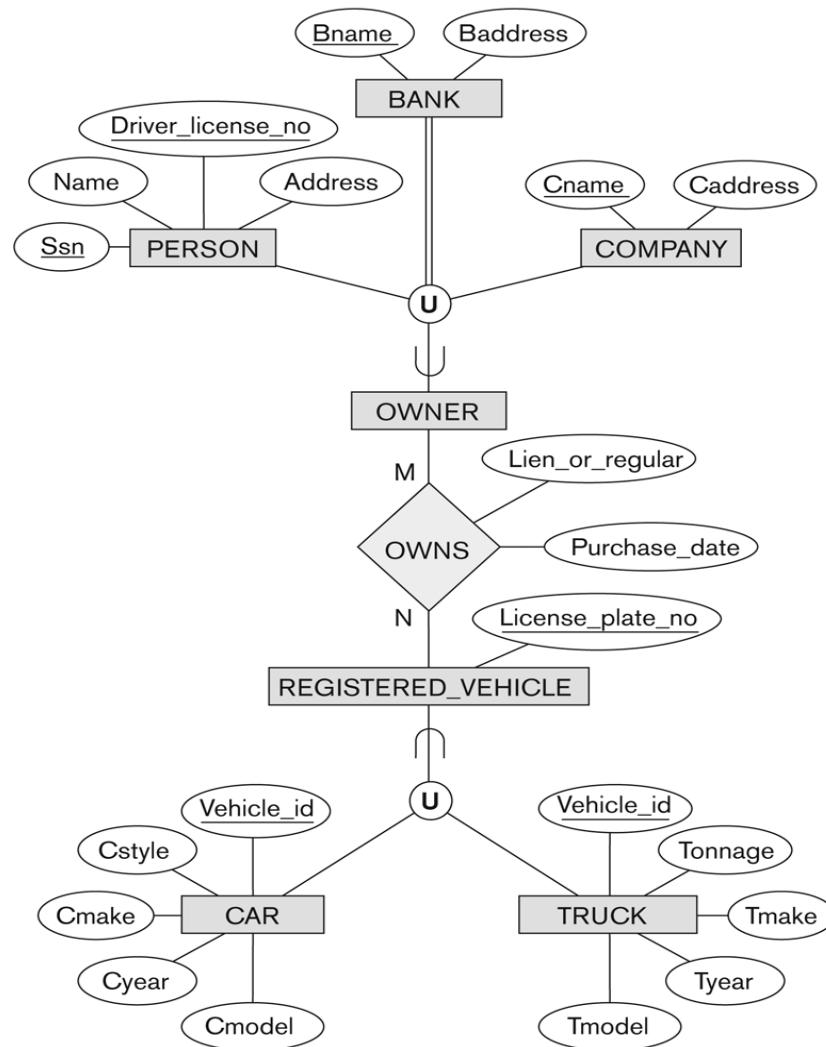


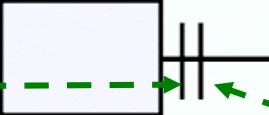
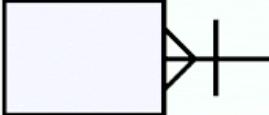
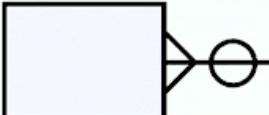
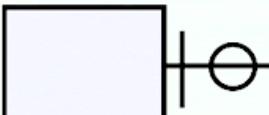
Figure 4.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.



補充

■ ER Model的其他表示法： 雞爪圖(Crow's Foot Notation)

- 實務上，ER Model的繪製除了常用的陳品山版本外，尚有其它的繪製方法。各方法主要不同點在於關係型態的表達。
- 常見的另一種ER Model表示法為**雞爪圖**，又稱**魚尾圖**。
 - 對應至陳品山版本中的 **(min, max)** 表示。

SYMBOL	MEANING	UML REPRESENTATION
	One and only one	1
	One or many	1..*
	Zero, or one, or many	0..*
	Zero, or one	0..1

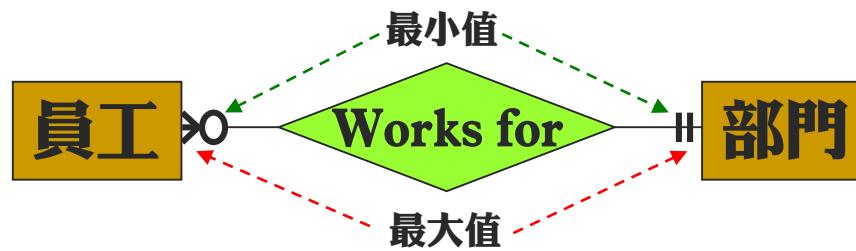
[

]

- 陳品山版：



- 雞爪圖：



- 在雞爪圖中，Entity的(min, max)所在位置與陳品山版的不同。

- 每位老師最少可分配0個車位，最多1個車位；每一個車位最少可被分配給1位老師，最多也僅1位。



- 每位客戶至少要下1張訂單，最多可多張；每一張訂單只能被一位客戶下單，最多也僅1位。

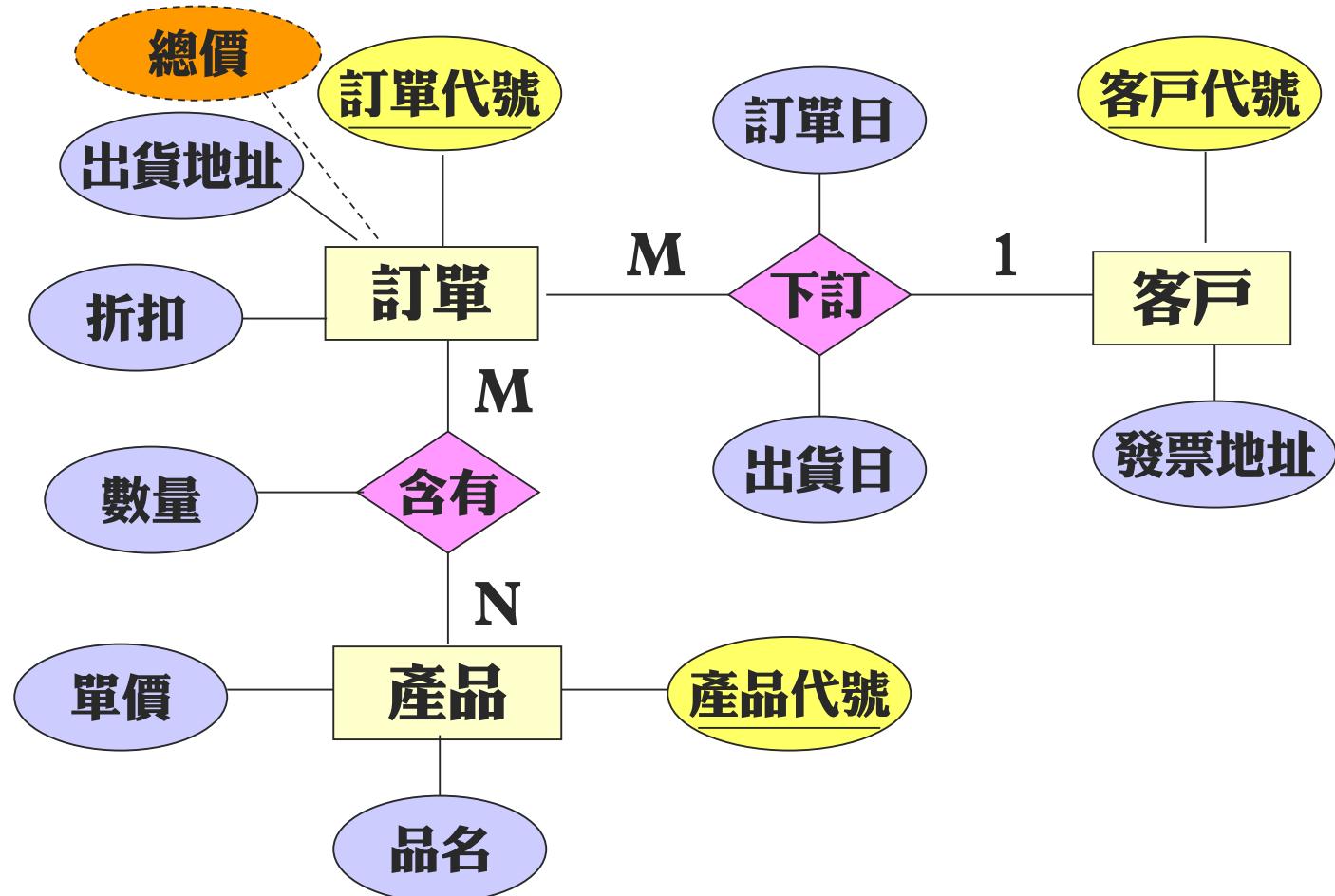


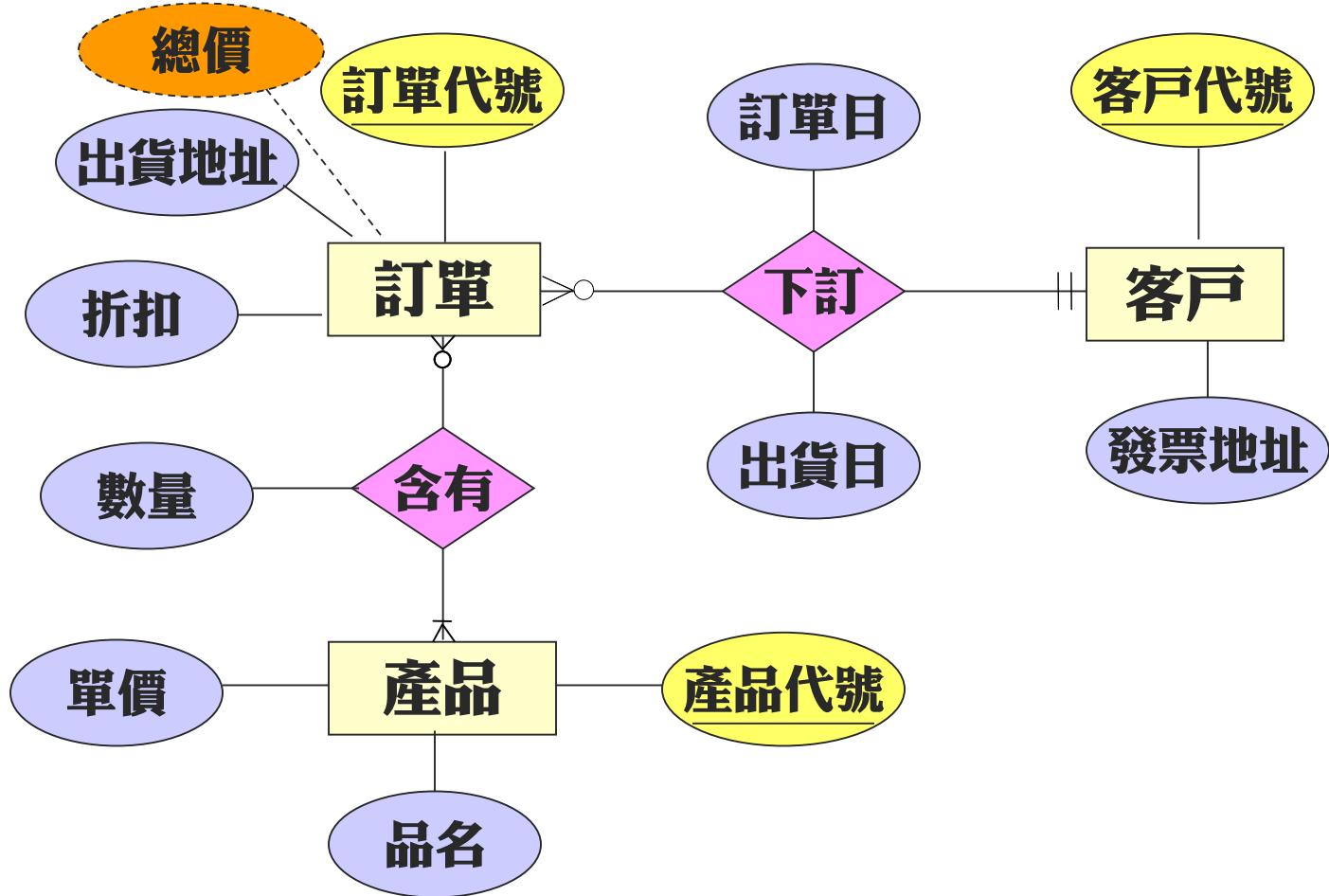
- 每位學生最少需選1門課，最多可多門；而每一門課至少要有一位學生選修，最多可多位。



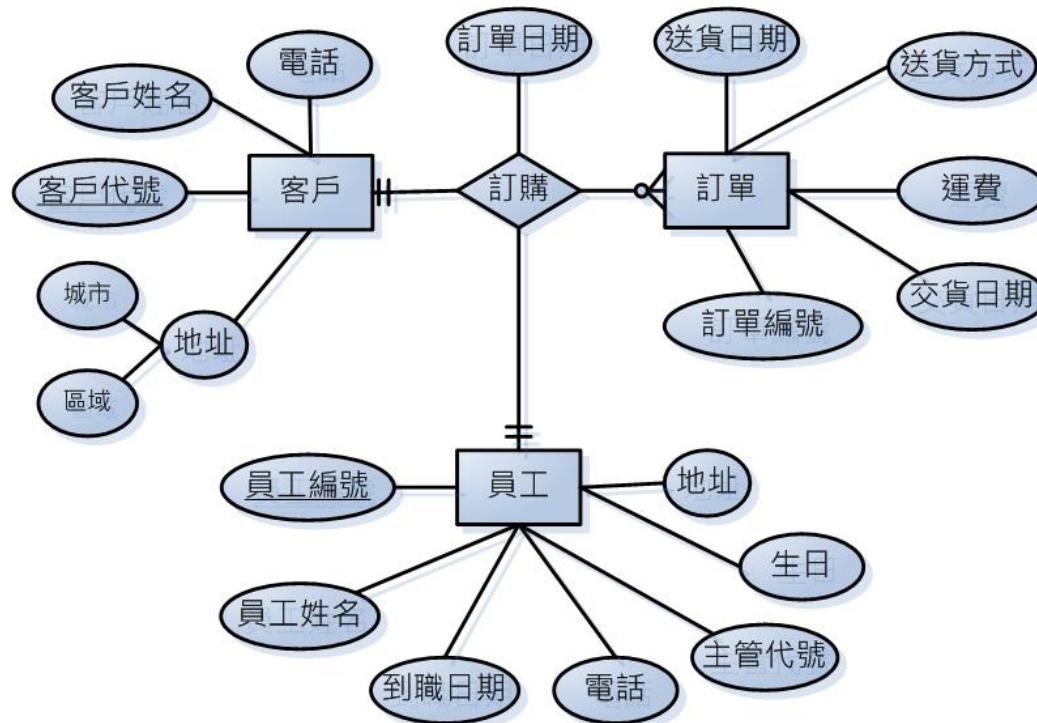
以企業實際表單做為轉換依據

<input checked="" type="checkbox"/>	聯合科技公司			
<input checked="" type="checkbox"/>	訂單代號：	客戶代號：		
<input checked="" type="checkbox"/>	訂單日：	發票地址：		
<input checked="" type="checkbox"/>	出貨日：	出貨地址：		
<input checked="" type="checkbox"/>	折扣：			
<hr/>				
<input checked="" type="checkbox"/>	<u>產品代號</u>	<u>品名</u>	<u>單價</u>	<u>數量</u>
<input checked="" type="checkbox"/>	0002	隨身碟	1022	3
<input checked="" type="checkbox"/>	0018	印表機	3186	5
	⋮	⋮	⋮	⋮
總價： <input type="text"/>				
衍生屬性				



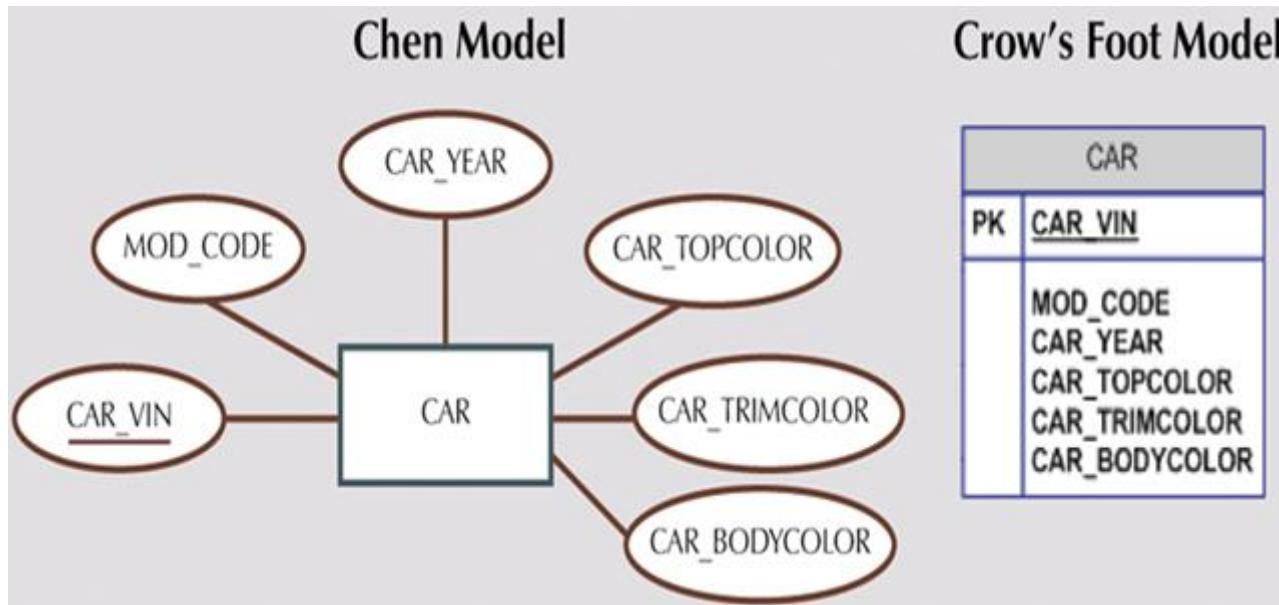


■ 每一位客戶可以訂購一張以上的訂單，也可以沒有下任何訂單；但是，每一張訂單必須會有一位客戶的訂購資料。並且每一張訂單必須要有一位員工負責客戶的訂購資料。

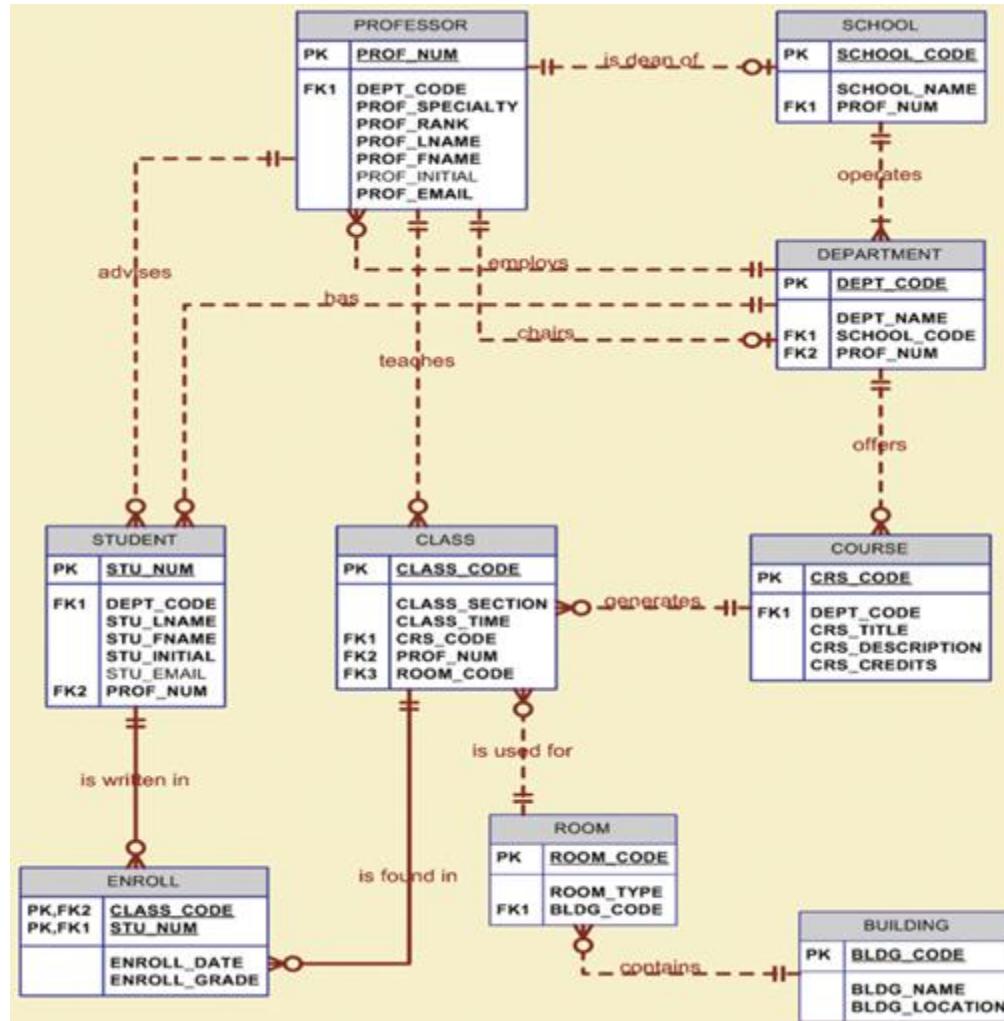


變化1：屬性表示方式

- 通常在使用雞爪圖時，原本陳品山版的各種屬性，皆會歸屬到相對應的Entity之方框下，並以列表形式表示。

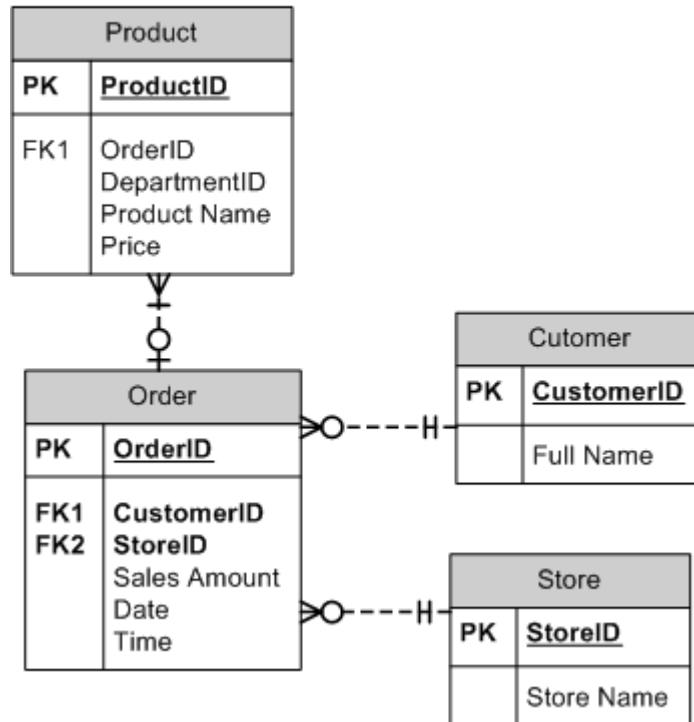


變化2：無菱形符號，但關係項目保留



變化3：無菱形符號，且無關係項目

ER Diagram of Online Sales System



變化3：無菱形符號，且無關係項目

ER Diagram of Online Sales System

