

第 4 章

建立關聯式資料庫

本章重點

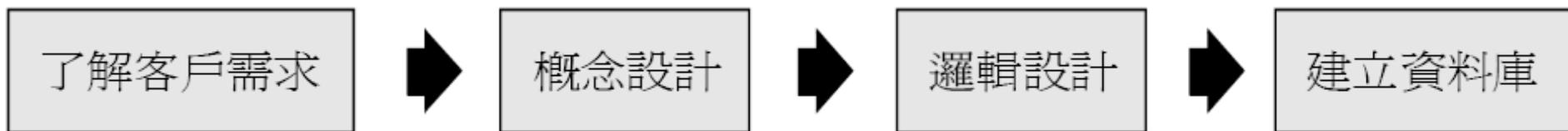
- 如何才能建立一套完善的關聯式資料庫？
- 本章要為您介紹資料庫的設計步驟，及利用前面所學的實體-關係圖，協助設計出資料庫的結構，進而轉換為真正的 SQL Server 資料庫。
- 如何設計一個完善的資料庫
- 建立實體-關係圖
- 將實體-關係圖轉換為關聯式的資料表

4-1 如何設計一個完善的資料庫

- 資料庫設計包含兩大部分：一是操作介面設計；另一則是結構設計。
 - **結構設計：**
 - 結構設計是指設計出適當且最佳化的資料表。一個結構良好的資料庫可提升其整體的存取效率及儲存效率。
 - **操作介面設計：** (程式設計師的工作)
 - 以程式語言 (例如 Visual Basic、ASP.NET) 撰寫操作介面, 讓使用者不必接觸資料庫的結構, 就能操作資料庫, 如新增、刪除資料... 等等工作。

資料庫的設計流程

- 為了避免設計者閉門造車，直到規劃後期才發現錯誤，所以在規劃的過程中，應分為數階段分別執行，並隨時與使用者溝通，方可設計出符合需求的資料庫。
- 資料庫的規劃過程大致可分為 4 個階段：



需求說明書

- 事務流程圖
- 資料流程圖(DFD)
- UML

實體關係圖
E-R Model

資料表

實際建立資料庫

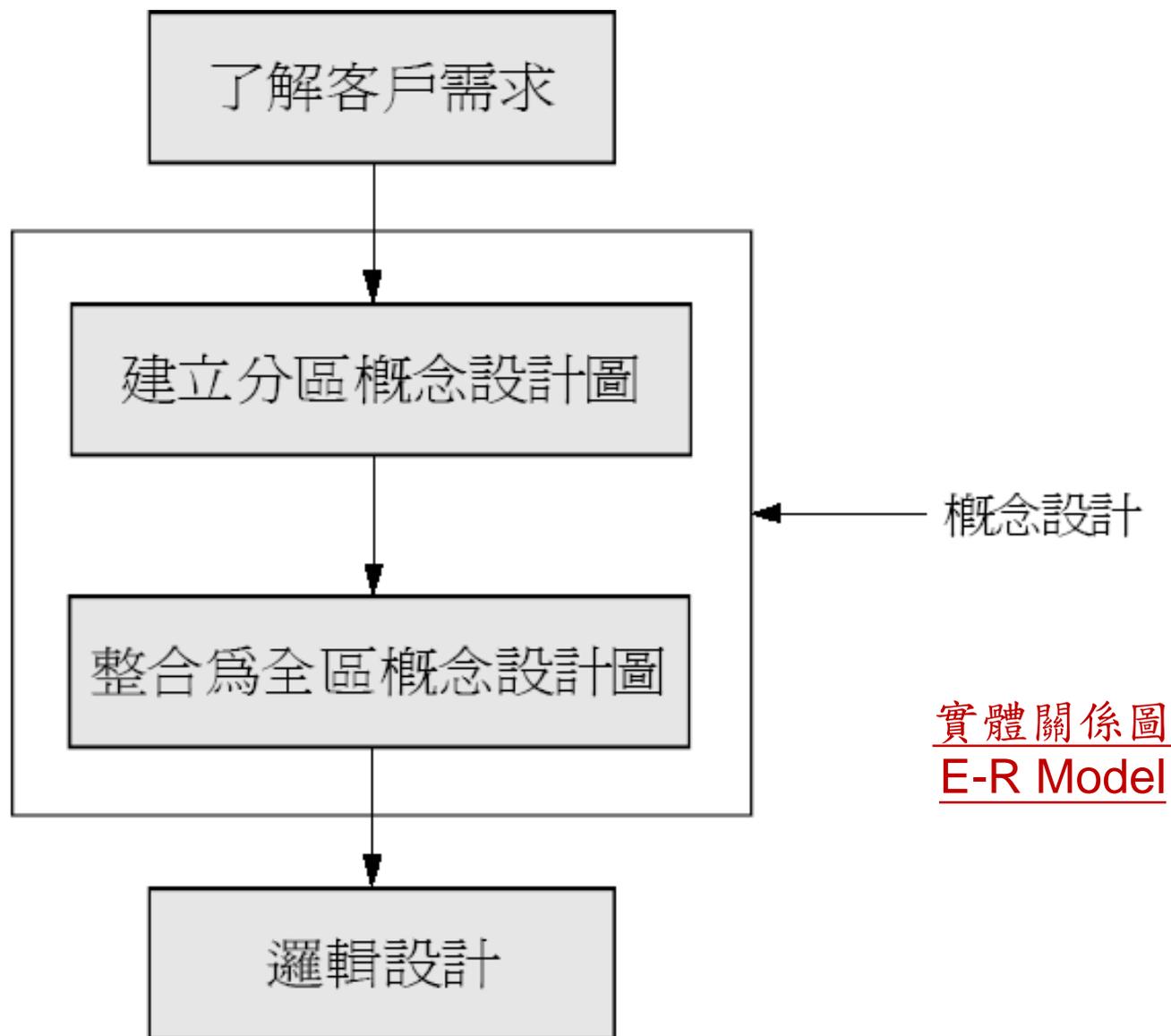
了解客戶需求

- 在此階段，設計者最主要的工作是收集建立資料庫所需的資訊，做為後續設計的基礎。
- 本階段的主要工作包含以下兩項：
 - **針對客戶需求，確定設計範圍**：拜訪客戶，了解實際工作流程、各部門執掌範圍及資料處理方式。
 - **收集和分析資料**：盡量收集客戶平時使用的各類表單、報表、檔案...，這些都是規劃資料庫的重要依據。
- 進行電腦化後可能會產生一些新的需求，例如每個月各產品的銷售分析，或改變部分現行的作業流程，這些都要事先和客戶討論，看看是否有此需求。

概念設計

- 概念設計是將收集的資料，經過分析及整理後，產生一個能符合使用者需求的資料庫模型，並以簡單的形式表現出來（例如實體 - 關係圖）。
- 通常概念設計分為兩個階段：
 - 建立分區的概念設計
 - 整合為全區概念設計圖

概念設計



建立分區概念設計圖

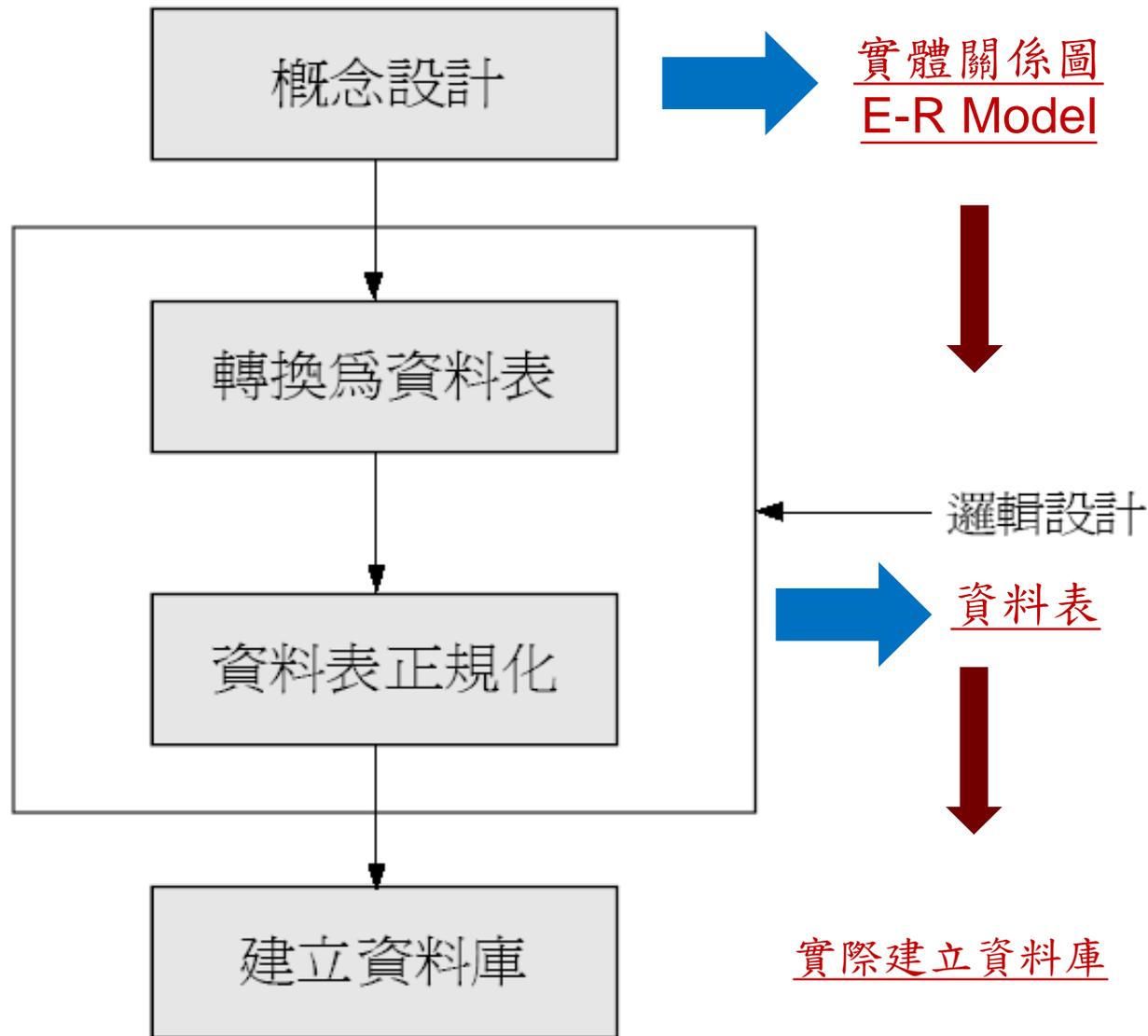
- 概念設計的第一個步驟要分別針對不同需求的使用者，確定使用範圍。
- 例如公司的資料庫系統必須面對**業務部**、**財務部**、**產品部**...等不同部門的使用者，這些使用者牽涉到資料庫中的資料及處理的方式各不相同，所以應針對不同的需求，設計不同的概念模型。

整合為全區概念設計圖

- 完成分區的概念模型後，便要將它們整合為一個全區概念模型。整合的過程必須注意下列幾點：
 - 解決各分區概念設計之間不一致的情形：
 - 由於分區概念設計所面對的使用者不同，所以對於共用資料的看法及重要性有時會出現差異，
 - 而此步驟最主要的工作就是要消弭各分區模型之間的不一致。
 - 刪除概念設計中重複或多餘的物件，以免造成後續設計時的困擾。

邏輯設計

- 邏輯設計階段的主要工作，是將概念設計階段產生的結果，轉換為實際使用的資料表。



邏輯設計

- 以實體 - 關係圖來說, 此階段的工作可分為以下兩項：
 - **轉換為資料表**：遵循規則, 將 實體 - 關係圖 正確無誤地轉換為實際使用的 資料表
 - **資料表正規化**：依照正規化的步驟重新檢驗一次, 最好讓每一個資料表都能符合 BCNF (Boyce-Codd Normal Form) 的規範

建立資料庫

- 經過邏輯設計階段之後，紙上的分析工作即已完成。
- 最後便要將結果建立成資料庫 (例如以 SQL Server、MariaDB 建立)。

4-2 從客戶分析到建立實體 - 關係圖

- 我們以書籍銷售的實際範例來說明如何建立實體 - 關係圖。
 - Step 1：收集資料, 確定設計範圍
 - Step 2：依照不同的使用者訂出分區的設計範圍
 - Step 3：列出系統中的實體及其屬性
 - Step 4：建立實體之間的關係
 - Step 5：加入屬性

Step 1：收集資料，確定設計範圍

- 假設某圖書公司要開發書籍銷售系統，經過調查整理後，規劃出該系統的主要工作為：
 - (業務部門)處理客戶訂單，並產生出貨單交由倉庫(發行部門)出貨。
 - (發行部門)將出貨單中詳列的書籍產品包裝後，運送到客戶手中。
 - (財務部門)依照出貨單上的書籍產品產生請款單，送到客戶手中。
 - 客戶依照請款單上的金額繳付，公司收到客戶的帳款後，便開立發票寄送到客戶處。

Step 2：依照不同的使用者訂出分區的設計範圍

- 此工作範圍內牽涉到 3 個不同部門的使用者：
 - 業務部門：負責處理訂單事宜。
 - 發行部門：負責依照業務人員開立的出貨單，將書籍送到客戶手上。
 - 財務部門：負責處理開立發票及催收帳款事宜。

Step 2：依照不同的使用者訂出分區的設計範圍

部門	處理工作
業務部門	<ul style="list-style-type: none">○處理客戶的訂單,並填寫出貨單○將處理後的訂單送交財務部門○每張訂單由訂單編號、客戶資料及訂購的書籍所組成
財務部門	<ul style="list-style-type: none">○依照訂單填寫請款單○收到客戶的帳款後,開立發票寄送到客戶處
發行部門	<ul style="list-style-type: none">○將出貨單中詳列的書籍產品包裝後,運送到客戶手中

Step 3：列出系統中的實體及其屬性

- 規劃出設計範圍後，就要先確定實體。
- 實體通常是整理資料中的名詞，例如地點、人物、概念、事件及設備等。

Step 3：列出系統中的實體及其屬性

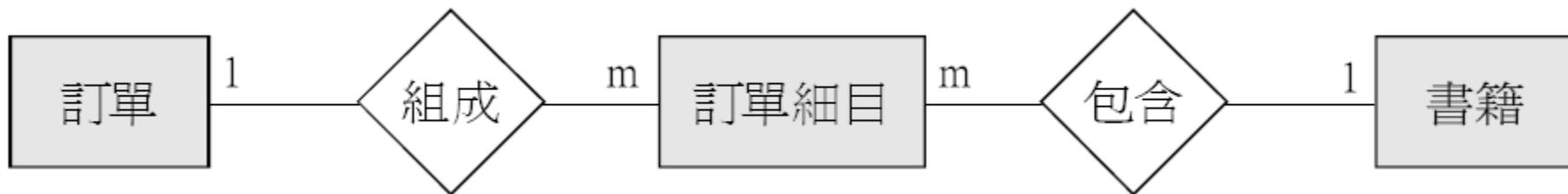
- 實體通常是整理資料中的名詞，例如地點、人物、概念、事件及設備等。
- 業務部門實體為：
 - **訂單**：包含 (* 訂單編號)、客戶名稱、聯絡人、地址、電話、訂單日期、訂單細目、總金額及備註等屬性。
 - **書籍**：包含 (* 書籍編號)、書籍名稱及單價等屬性。
 - **客戶**：包含 (* 客戶編號)、客戶名稱、聯絡人、地址及電話等屬性。
 - **出貨單**：包含 (* 出貨單編號)、書籍名稱、地址、電話、客戶名稱及聯絡人等屬性。

Step 4：建立實體之間的關係

- 實體及屬性確定後，接下來就要討論各實體之間的關係。以業務部門為例，實體之間的關係：
 - **訂單**和**書籍**的關係：
 - **訂單**是由**訂單編號**、**客戶資料**及**訂單細目**組成。
 - 我們必須透過**訂單**上的**訂單細目**才能了解客戶訂購了哪些書籍，所以**訂單**和**書籍**的關係必須細分為**訂單**和**訂單細目**、以及**訂單細目**和**書籍**的關係(亦即將**訂單細目**視為一個實體，屬性包括**細目編號**、**書籍編號**、**書籍名稱**、**單價**及**數量**)。

Step 4：建立實體之間的關係

- **訂單細目**與**訂單**之間的關係：一個訂單可包含多個訂單細目，但一份訂單細目只能屬於一份訂單，所以是**一對多**的**組成**關係。
- **訂單細目**與**書籍**之間的關係：一個訂單細目只可訂購一種書籍，但一種書籍卻能屬於多個訂單細目，所以是為**一對多**的**包含**關係。



Step 4：建立實體之間的關係

- **客戶**和**訂單**的關係：
 - 一個客戶可填寫多份訂單，但一份訂單只能屬於一個客戶，所以**訂單**與**客戶**之間的關係是**一對多**的**填寫**關係。

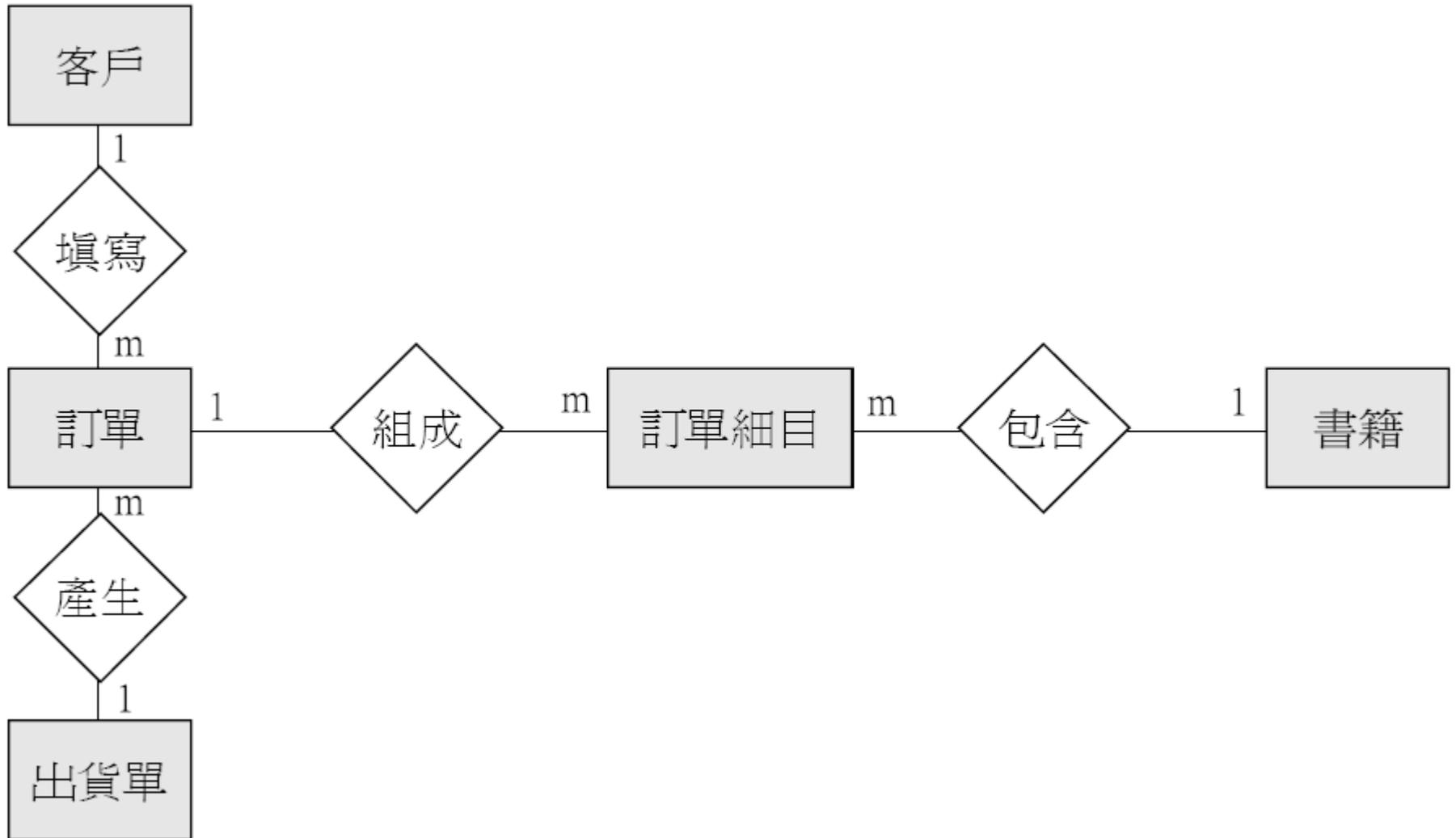


Step 4：建立實體之間的關係

- **出貨單**和**訂單**的關係：
 - 一份出貨單可由多份訂單組成，但一份訂單只能屬於一張出貨單，否則客戶就會收到好幾份相同的書籍。
 - 所以**訂單**與**出貨單**之間的關係，也是一對多的**產生**關係。



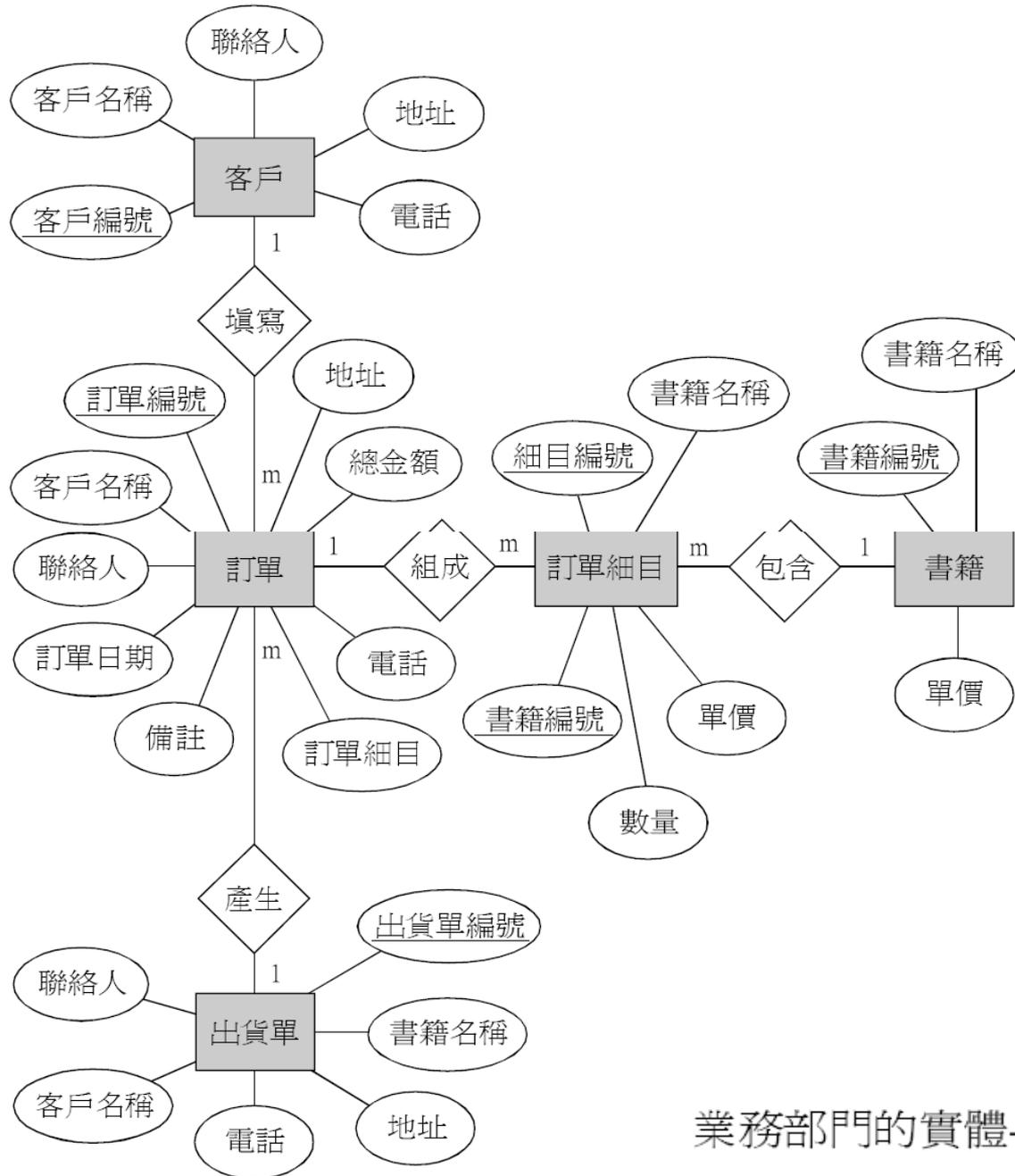
Step 4 : 建立實體之間的關係



Step 5：加入屬性

- 實體和關係確認後，只要將屬性加入，即完成了業務部門的實體 - 關係圖。

加入屬性

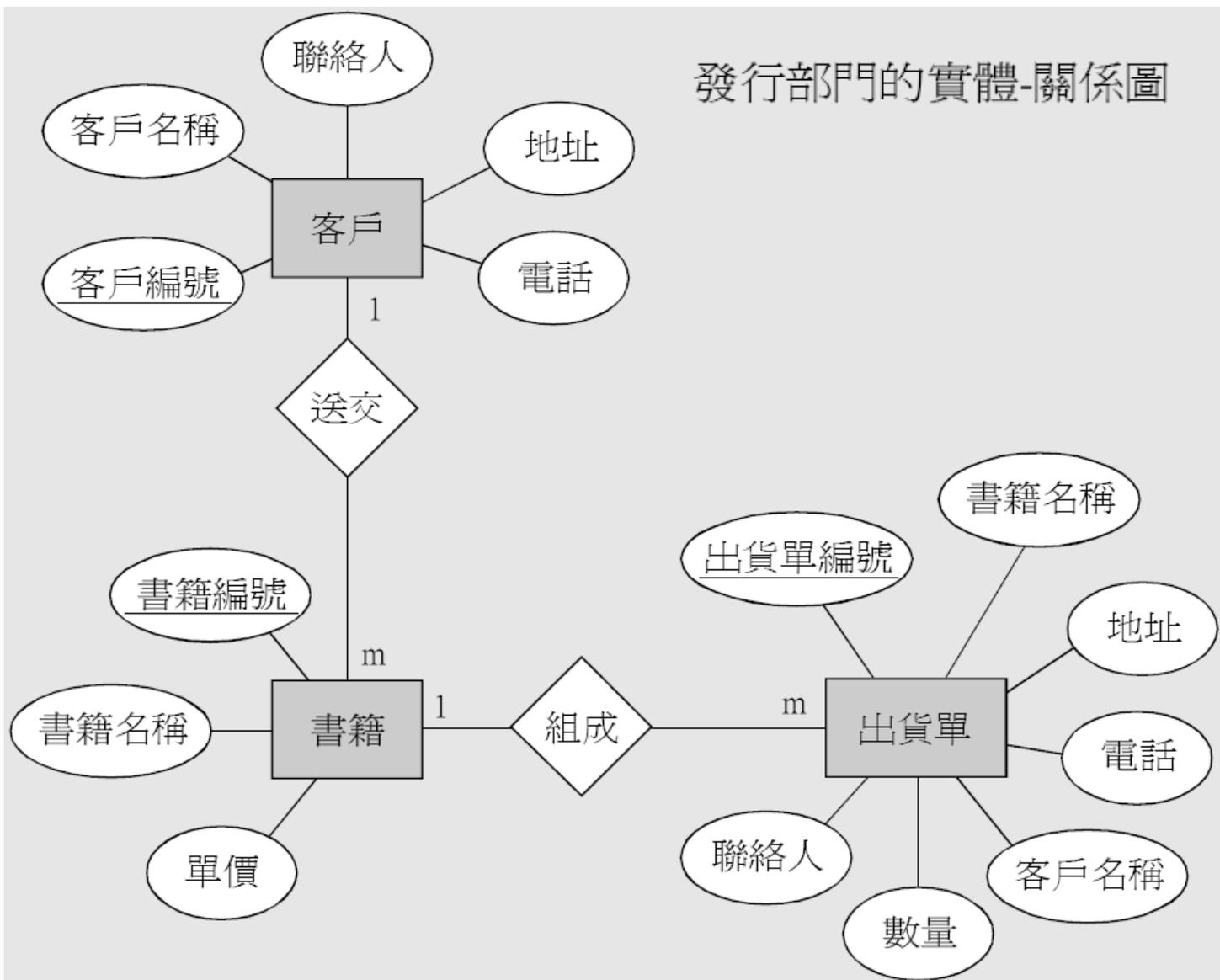


業務部門的實體-關係圖

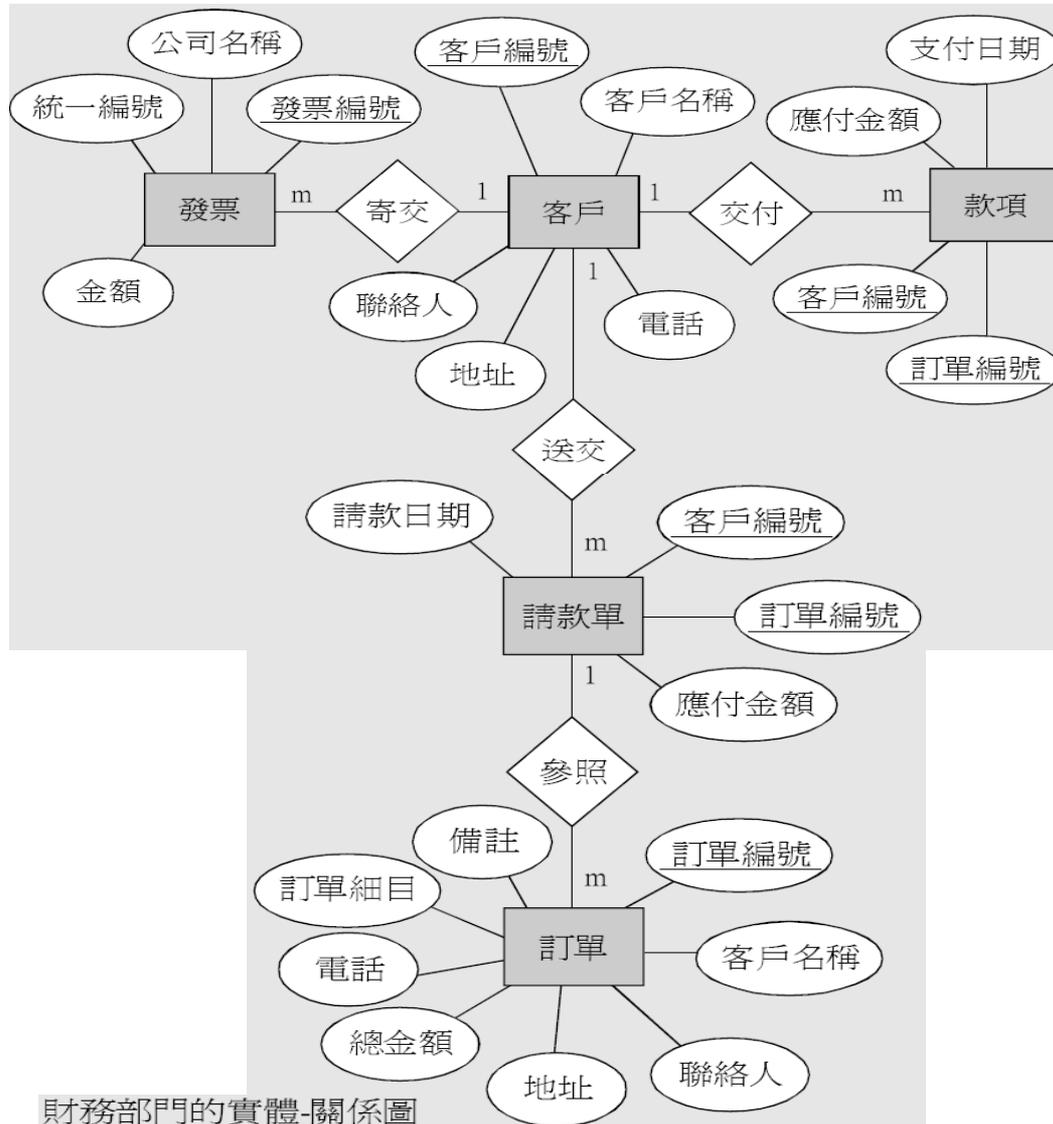
將其他部門的工作流程圖轉變為 實體 - 關係圖

- 依照相同的方法，將發行部門及財務部門的工作描述分別轉換為實體 - 關係圖，如下圖所示：

將其他部門的工作流程圖轉變為 實體 - 關係圖



將其他部門的工作流程圖轉變為 實體 - 關係圖



財務部門的實體-關係圖

4-3 整合為全區的概念模型

- 由於各分區實體 - 關係模型所面對的問題不同，且通常由不同的設計者設計，如此容易導致各個分區圖存在許多不一致的地方。
- 因此在整合時，最重要的工作就是要消除各分區圖的不一致，產生一個能被所有使用者接受的概念模型。

整合分區實體 - 關係圖時可能遭遇的問題

- 整合實體 - 關係圖時可能遇到的問題有 3 類：
 - 屬性不一致
 - 命名不一致
 - 結構不一致

屬性不一致

- 有時各分區圖，對於相同實體的屬性類型、範圍、單位...會有不同。
- 例如不同部門，對於相同零件的編號方式不同；員工的年齡屬性，有的部門以出生日期表示，有的則直接用數字來表示；或產品的重量，有的部門以公斤為單位，有的卻以公克為單位。
- 若遇到屬性上的不一致，通常較容易解決，只要各部門討論協商，訂出共同的標準即可。

命名不一致

- 命名不一致可分為兩種：
 - 同名不同義：不同的物件，在不同的分區圖中，具有相同的名稱。
 - 同義不同名：相同的物件，在不同的分區圖中，具有不同的名稱。
- 命名不一致可能發生在實體或關係上，也可能發生在屬性上。一般來說，只要各部門討論協商，訂出一個共同的標準即可。

結構不一致

- 結構上不一致，常見的有以下幾種：
 - 同一物件在不同的分區圖中的表現的方式不同。
 - 例如員工在某個分區圖中被視為實體，但在另一個分區圖中卻被視為屬性。
 - 解決之道：必須檢查該物件在實體 - 關係圖中是否擁有屬性，或和其他實體產生關係。若是，則可將之設定為實體。
 - 同一個實體在不同的分區圖中，所包含的屬性不相同。
 - 這類情形相當常見，因為每個分區圖所重視的物件屬性並不相同，解決的方式是將所有的屬性都納入。

結構不一致

- 實體間的關係在不同的分區圖中不相同。
 - 例如甲實體和乙實體在某一個分區圖中是多對多的關係，但是在另一個分區圖中卻是一對多的關係；或是甲實體和乙實體在某一分區概念圖中具有關係，但在其他分區圖中卻沒有關係。
 - 例如有的系所規定學生只能參加一個社團，而有的系所可讓學生參加多個社團。
 - 則學生和社團的關係上就產生了一對多及多對多的關係。
- 為了滿足不同的需求，在整合時，就可使用多對多的關係來表示。

以『書籍銷售系統』為例, 說明整合全區的概念設計

- 實體 - 關係圖的整合可分為實體、關係及屬性等 3 部分的整合。
 - Step 1 : 整合實體
 - Step 2 : 整合關係
 - Step 3 : 整合屬性
 - Step 4 : 消除不必要的實體、關係及屬性

Step 1：整合實體

- 實體的整合主要是檢討各分區圖中，實體的名稱及其代表的意義是否有衝突。在**書籍銷售**系統中，可整合得到以下的實體：

客戶

訂單

出貨單

訂單細目

書籍

發票

請款單

款項

Step 2：整合關係

- 關係的整合主要是針對不同分區圖中相同實體之間的關係整合,看看是否有不一致的地方。
- 經過整合後,可得到以下的關係：

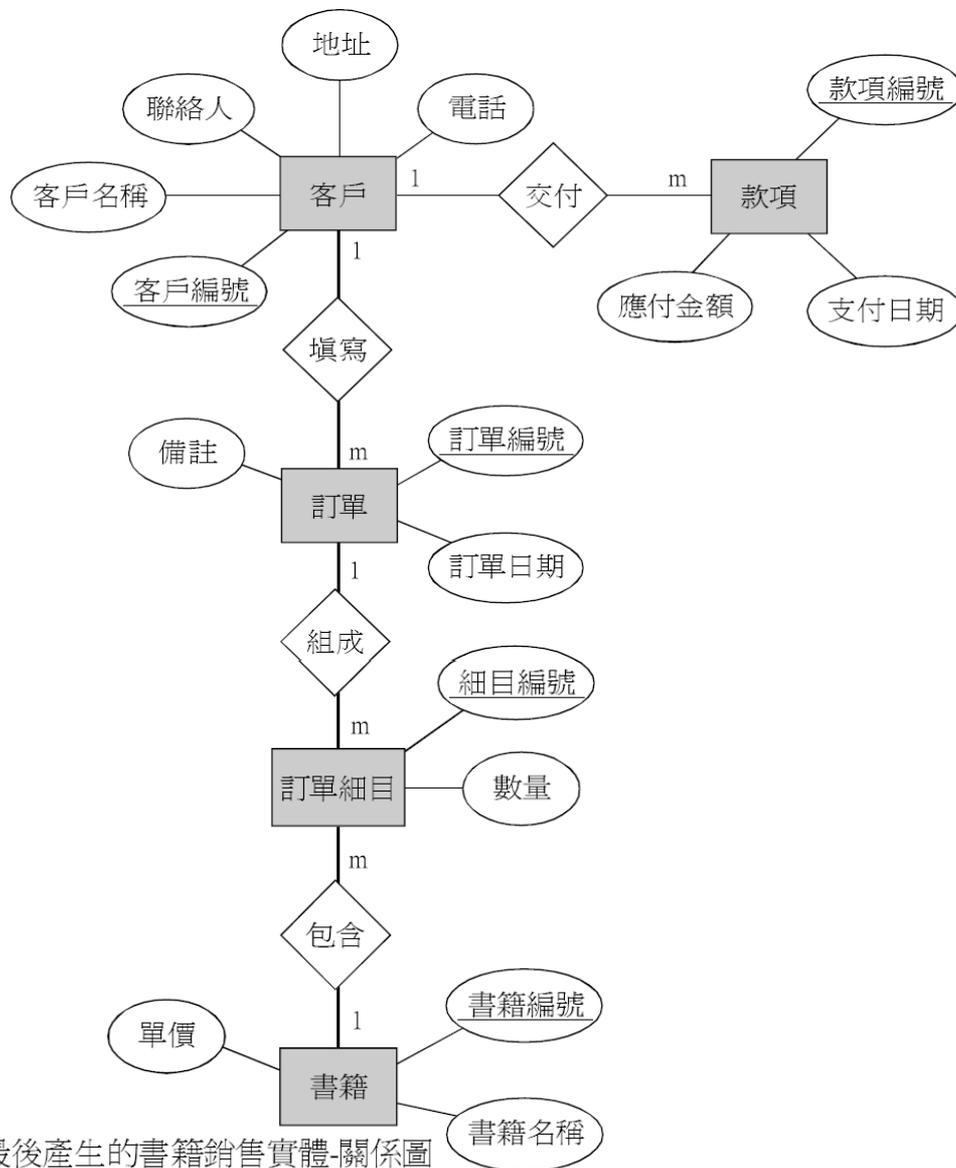
Step 3：整合屬性

- 屬性的整合主要是檢討各實體中的屬性名稱及其代表的意義是否有衝突。

Step 4：消除不必要的實體、關係及屬性

- 初步整合的實體 - 關係圖中，可發現實體、關係及屬性可由其他的實體、關係及屬性推導出來。
- 以下從**書籍銷售**圖中列舉幾個例子來說明：
 - **出貨單**及**請款單**都是依據**訂單**所產生，所以我們可利用**訂單**來產生**出貨單**及**請款單**。因此，便可將這兩個實體刪除，當然它的關係也就跟著刪除了。
 - **訂單**中的客戶名稱、聯絡人、地址...等屬性亦可由**客戶**實體中取得，因此可將它們刪除。
- 經過了一番整理後，我們便可得到一個最終結果的實體 - 關係圖：

消除不必要的實體、關係及屬性

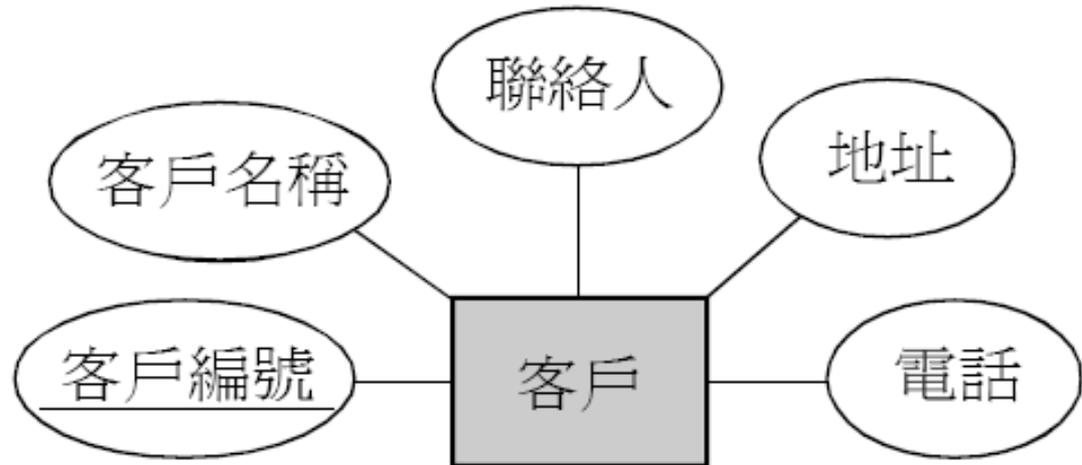


4-4 將實體 - 關係圖轉換為 關聯式的資料表

- 以下介紹將實體 - 關係圖，轉換為關聯式的資料表時常用的規則，以便設計者能依循這些規則，轉換為實際使用的關聯式資料表。

實體及屬性的轉換

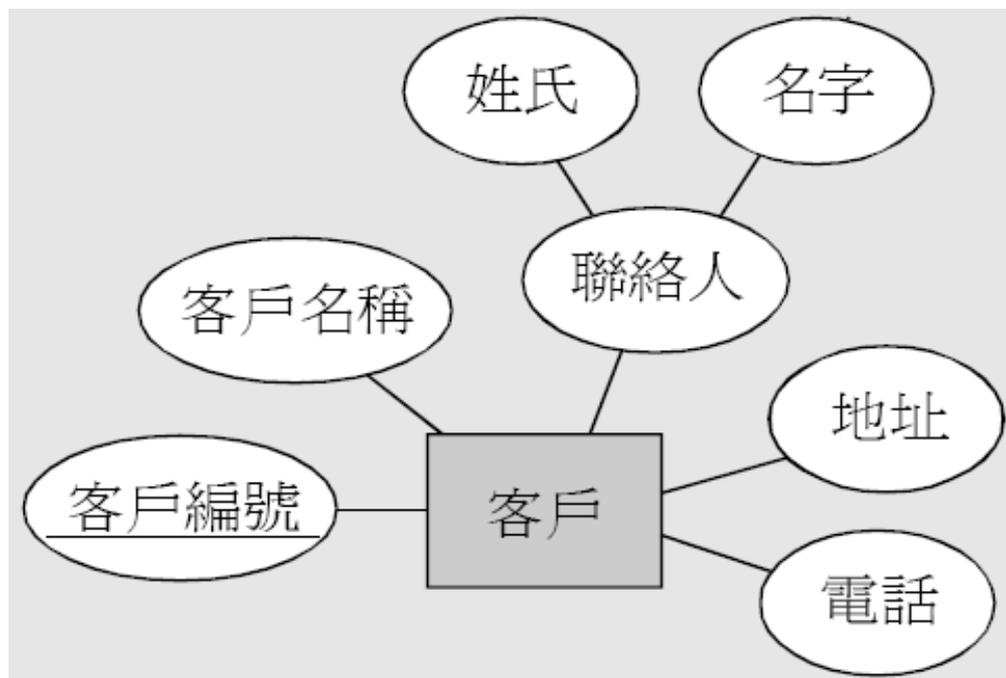
- **實體 - 關係圖**中所有的**實體**都用**資料表**來表示；而**屬性**則轉換成為資料表的**欄位**，若為**鍵屬性**，則會成為該資料表的**主鍵**。
- 如下圖所示：



客戶(* 客戶編號, 客戶名稱, 聯絡人, 地址, 電話)

複合屬性的轉換

- 若該屬性為**複合屬性**時，必須將複合屬性中的所有屬性，都轉換為該資料表的欄位。
- 如下圖所示：



複合屬性的轉換

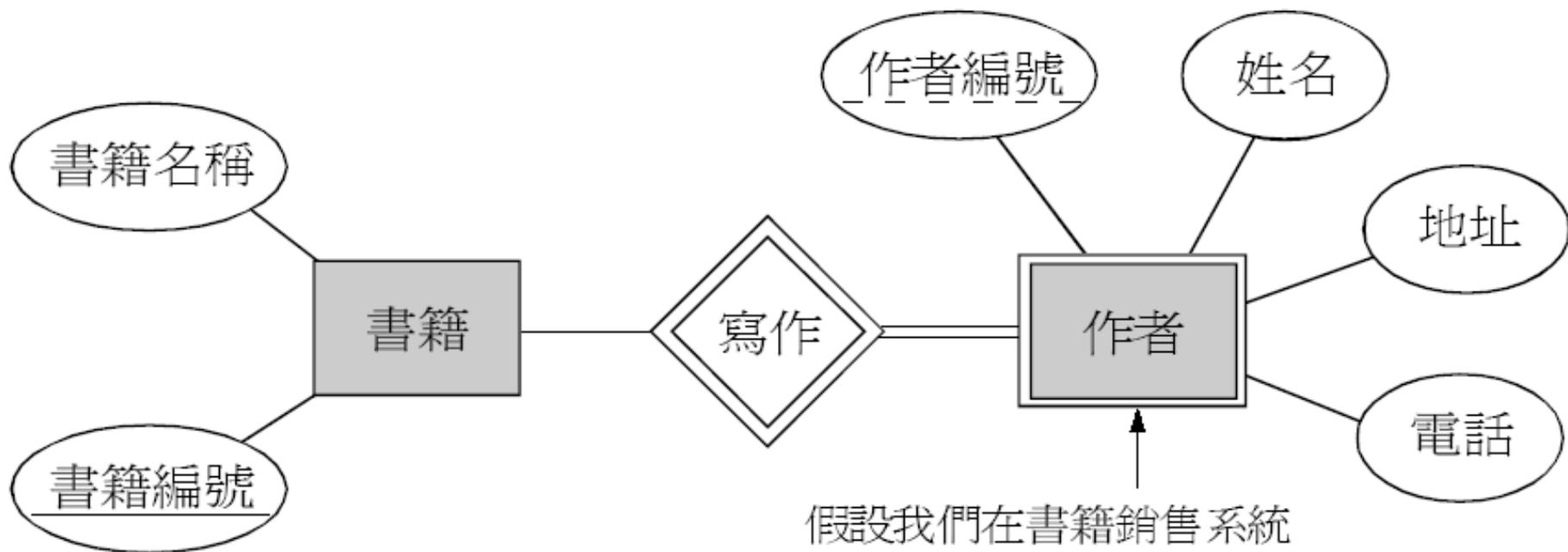
- 在轉換時，必須將**聯絡人**屬性中的**姓**及**名字**屬性當作是**客戶**實體的屬性之一。
- 轉換後的資料表如右：

客戶
* 客戶編號
客戶名稱
聯絡人姓氏
聯絡人名字
...

弱實體的轉換

- 在上一章中提過，若某一實體沒有鍵屬性，或是必須依附其他實體才能存在，則該實體即稱為『弱實體』。
- 如果是弱實體，轉換時，必須將其依賴實體的鍵屬性加入，做為該弱實體的連外鍵，並與該弱實體的識別屬性合起來，成為弱實體的主鍵。
- 如下圖所示：

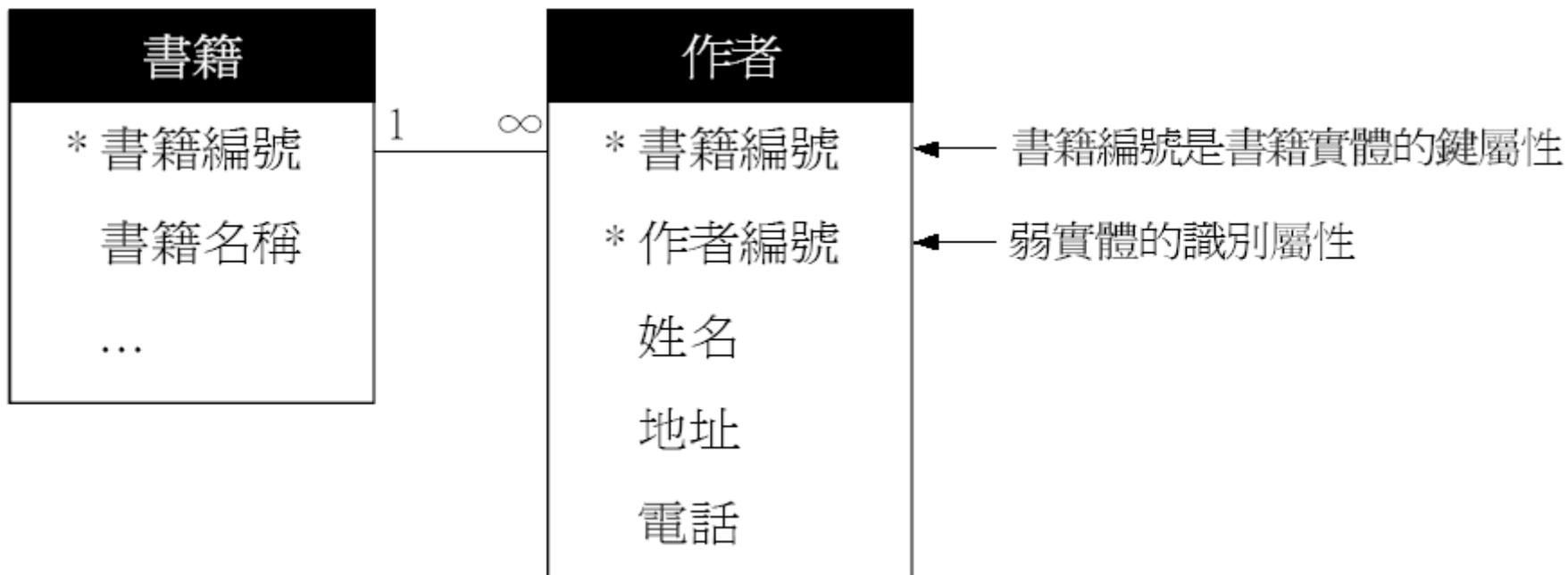
弱實體的轉換



假設我們在書籍銷售系統中又增加了作者弱實體

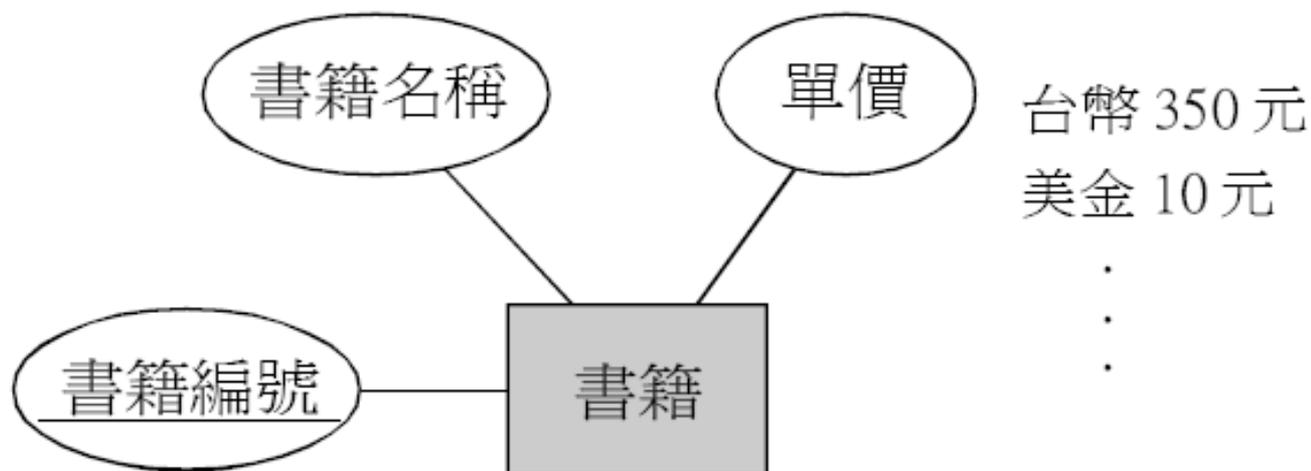
弱實體的轉換

- **作者**弱實體在轉換過程中，必須加入**書籍**實體的鍵屬性作為連外鍵，並且結合**作者編號**成為**作者**資料表的主鍵，如下表所示：



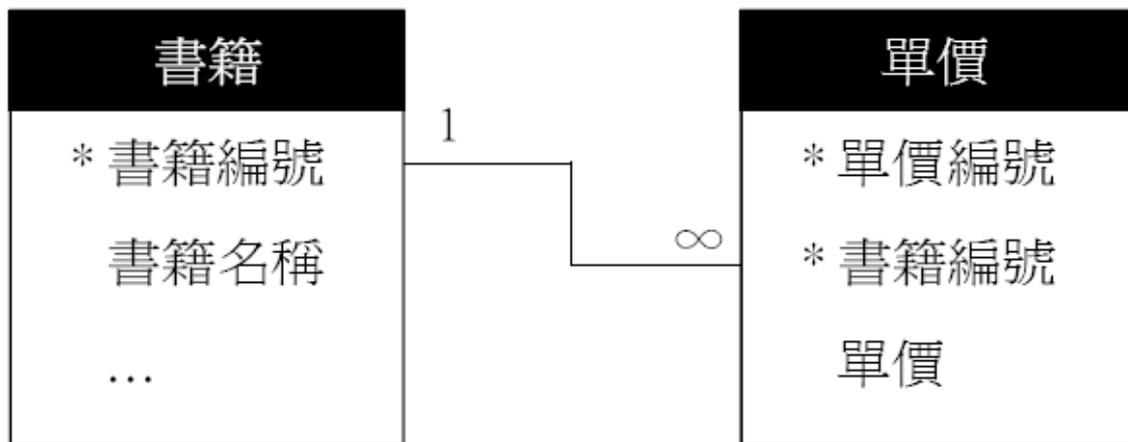
多值屬性的轉換

- 若實體中的屬性為**多值屬性**，則在轉換時，必須為該屬性另外**建立資料表**，假設**書籍**實體中的**單價**屬性為多值屬性。
- 如下圖所示：



多值屬性的轉換

- 我們必須為**單價**屬性另外建立一個**單價**資料表，然後將**書籍**實體的鍵屬性(書籍編號)加入**單價**資料表中，成為該實體的連外鍵，並且與單價編號(識別屬性)合起來成為**單價**資料表的主鍵。



單價編號	書籍編號	單價
P01	A001	NT350
P02	A001	US10

實體間關係的轉換

- 以下我們分別討論一對一、一對多及多對多關係的轉換。
 - 『一對一』關係的轉換
 - 『一對多』關係的轉換
 - 『多對多』關係的轉換

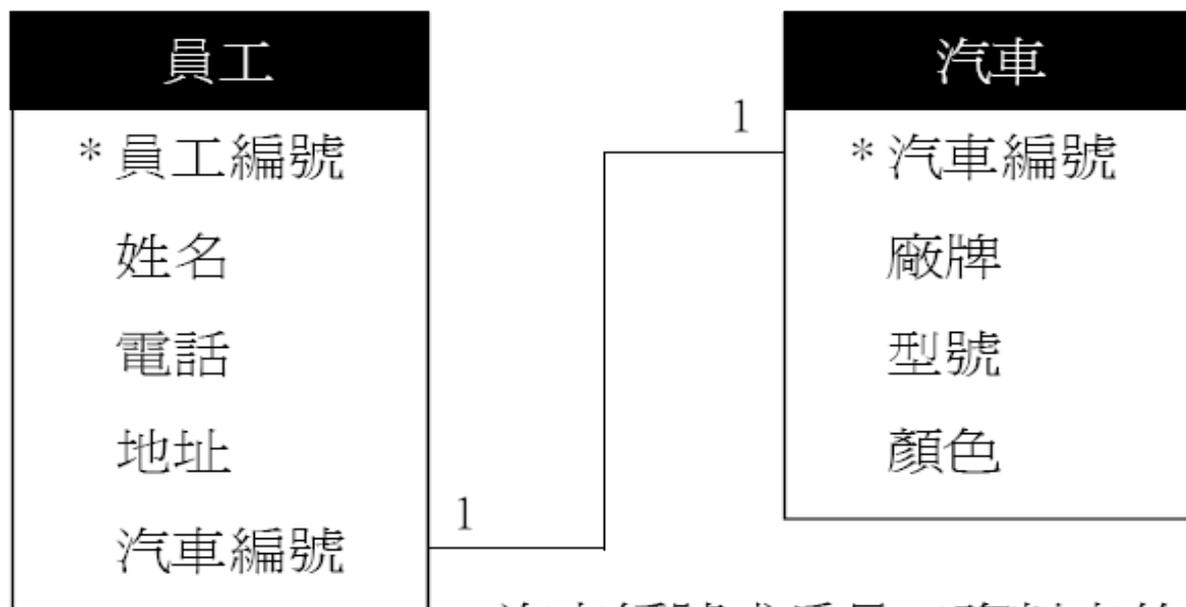
『一對一』關係的轉換

- 將部份參與實體的主鍵放入全部參與的實體中，做為連外鍵。
- 例如公司員工 (全部參與，假設所有的員工都會分配到汽車) 和汽車 (部分參與，並非所有的汽車都分配給員工) 之間為一對一的關係：



『一對一』關係的轉換

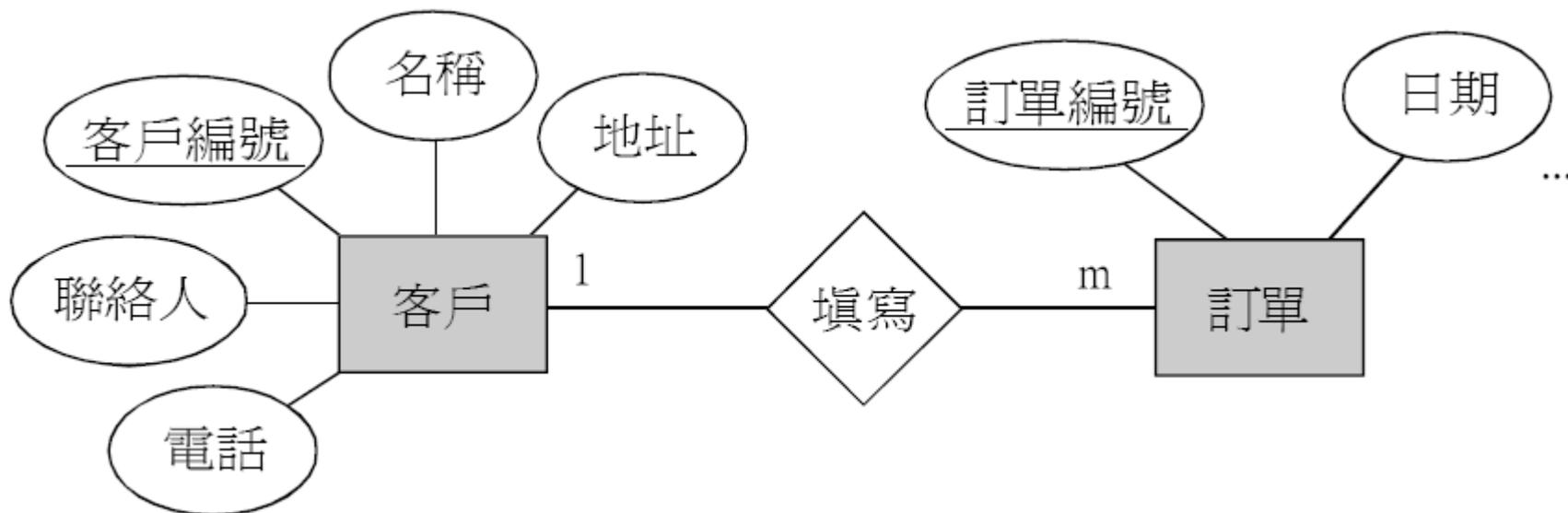
- 首先，將**員工**及**汽車**實體分別轉換成資料表，然後再將**汽車**資料表的主鍵放到**員工**資料表中，做為連外鍵，如下所示：



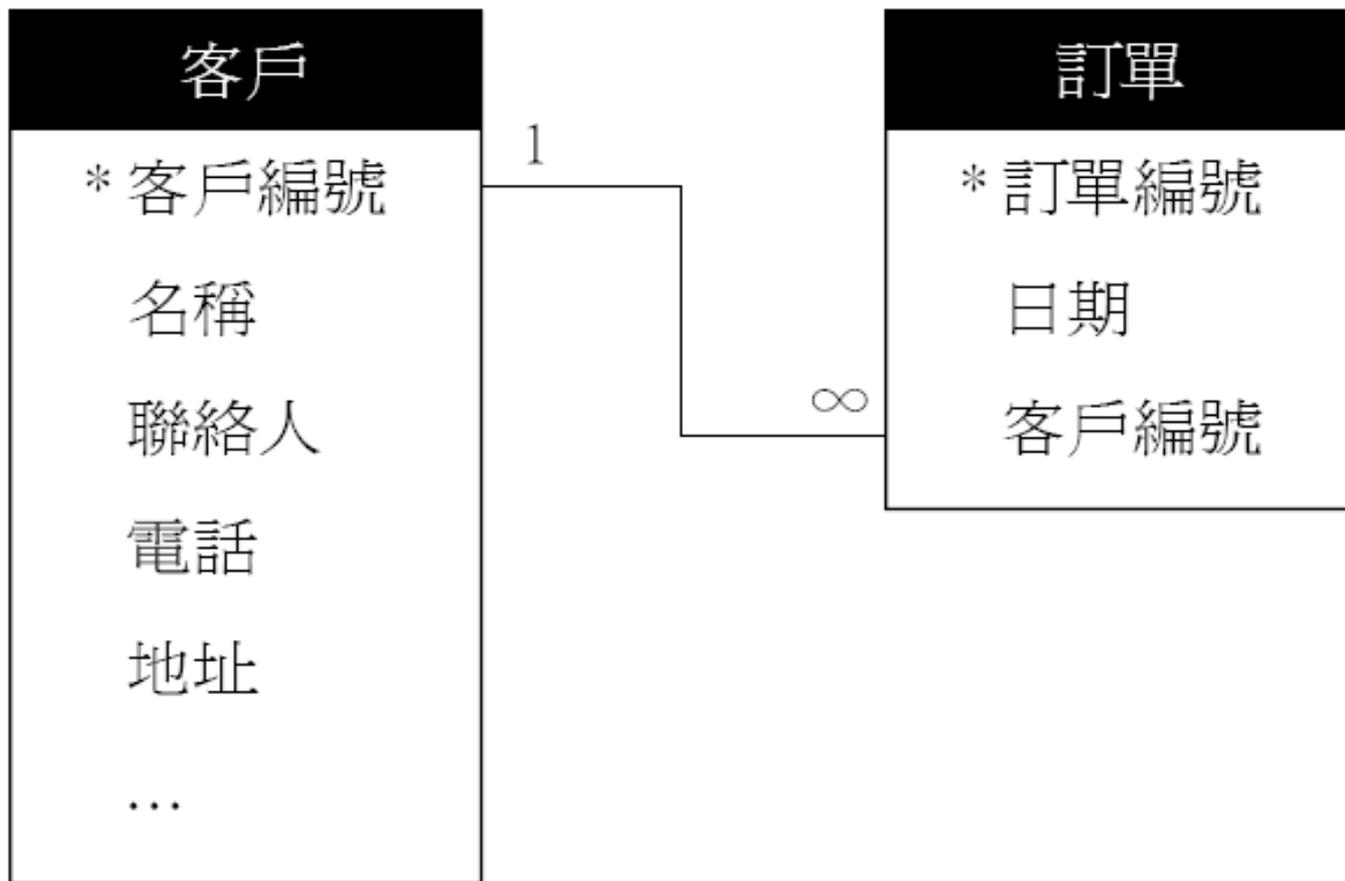
汽車編號成爲員工資料表的連外鍵

『一對多』關係的轉換

- 在實體 - 關係圖中，一對多的關係是將一個父資料表 ("一" 的這一方) 中的主鍵放入子資料表 ("多" 的這一方) 中，做為子資料表的連外鍵。
- 我們將**客戶**及**訂單**的一對多關係轉換後，可產生下列的結果：



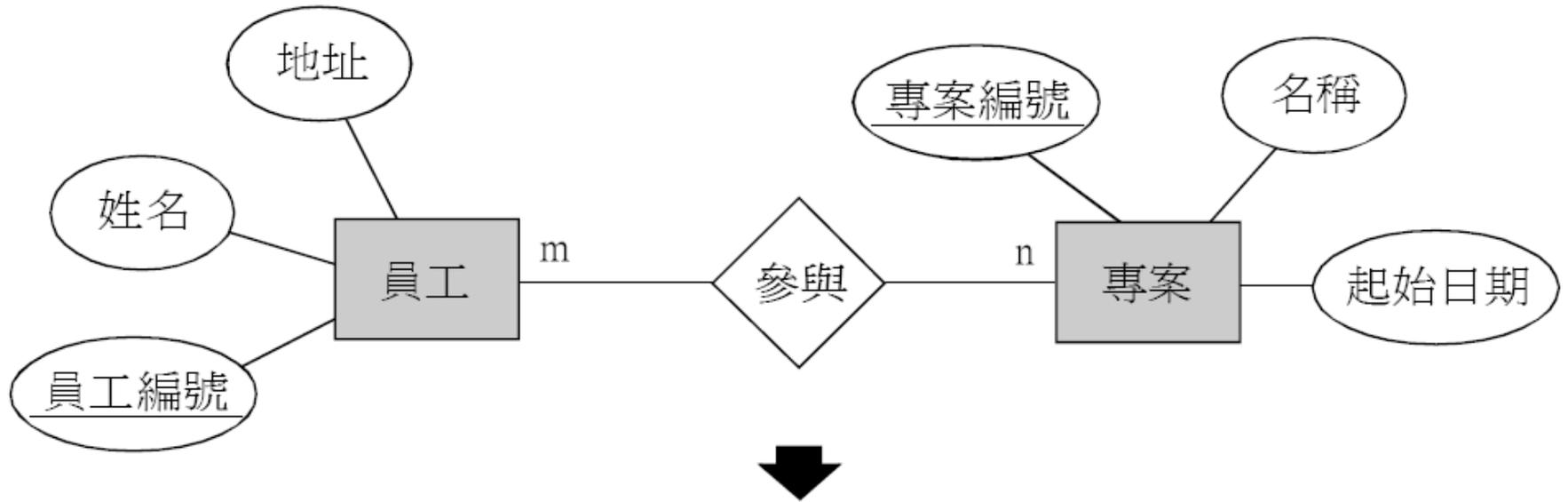
『一對多』關係的轉換



『多對多』關係的轉換

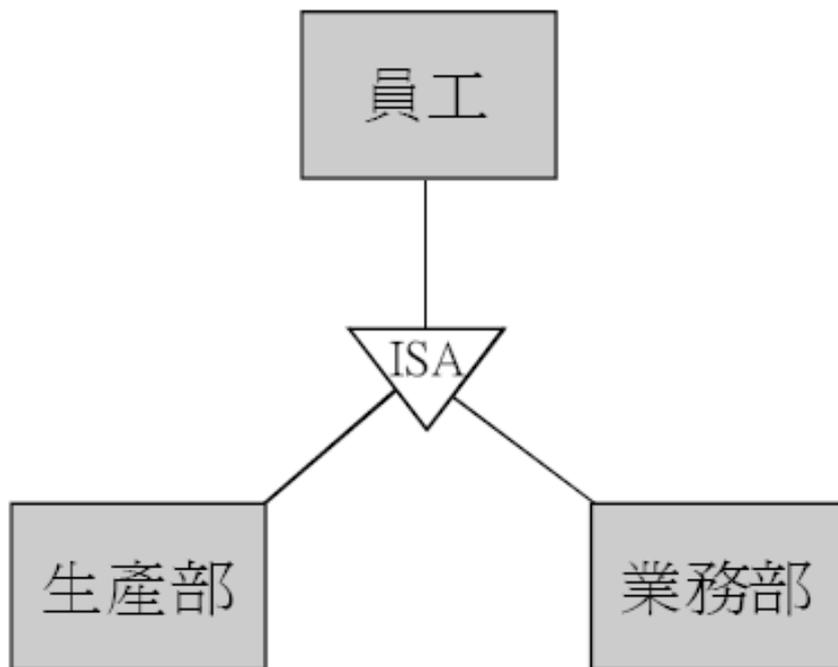
- 當實體間是多對多的關係時，就必須為**關係**另外**建立一個資料表**，且該資料表要包含它所關聯的實體的主鍵。
- 例如**員工**和**專案**的關係，可得到如下的結果：

『多對多』關係的轉換



超類型和子類型的轉換

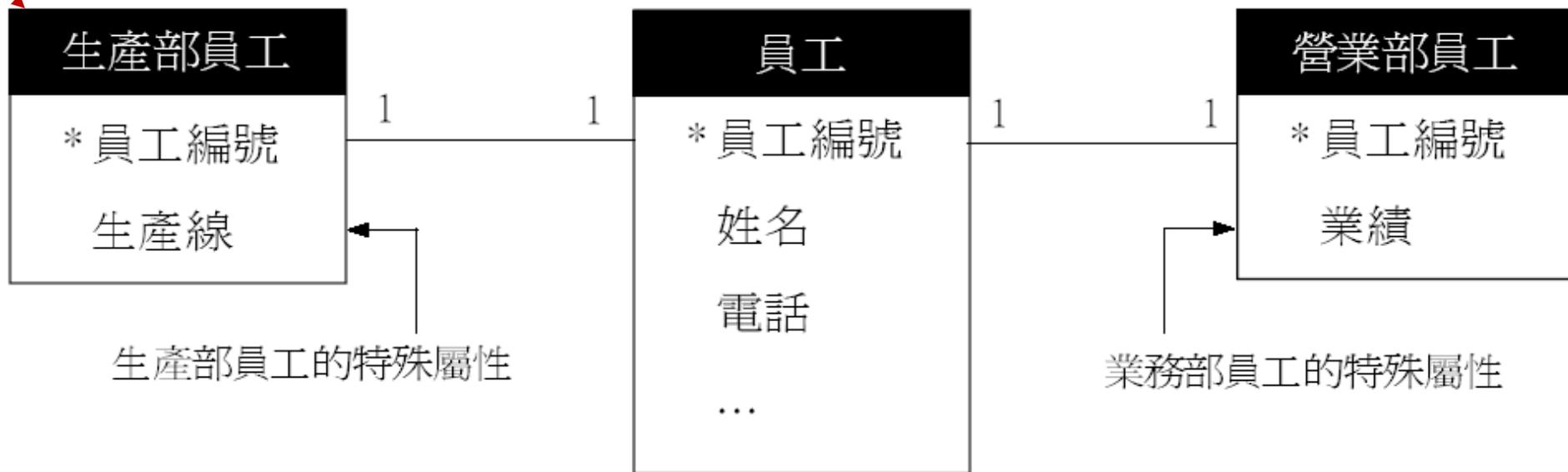
- 假設公司的員工依照特殊的屬性，再細分為**業務部員工**及**生產部員工**兩個子類型：



超類型和子類型的轉換

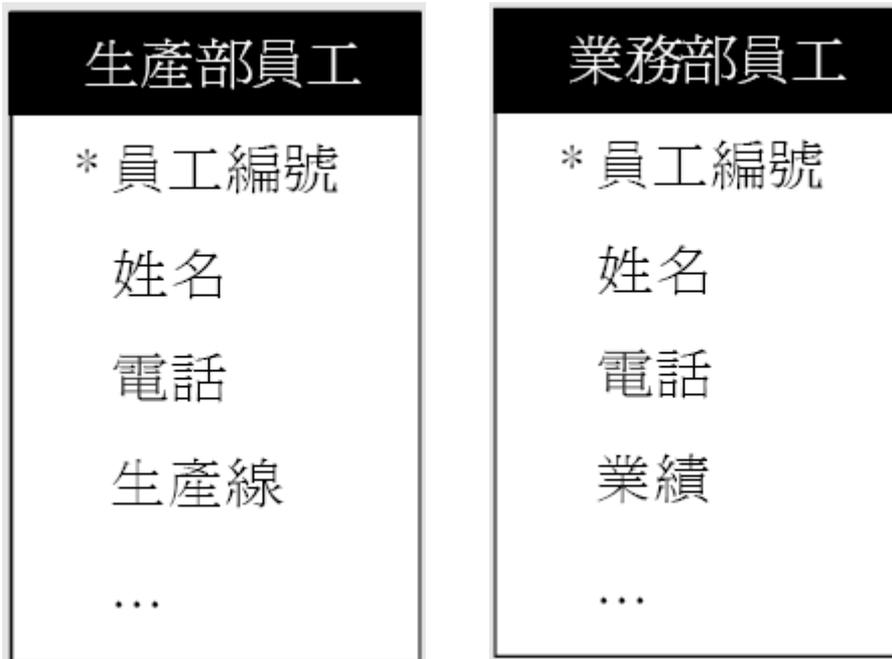
- 則在轉換時，必須將**員工**實體轉換為**員工**資料表，所有**員工**實體的屬性轉換為**員工**資料表的欄位；
- 再將**生產部員工**及**業務部員工**實體轉換為**生產部員工**及**業務部員工**兩個資料表，最後將**員工**的主鍵分別加入**生產部員工**及**業務部員工**資料表做為該資料表的主鍵。
- 如下圖所示：

超類型和子類型的轉換



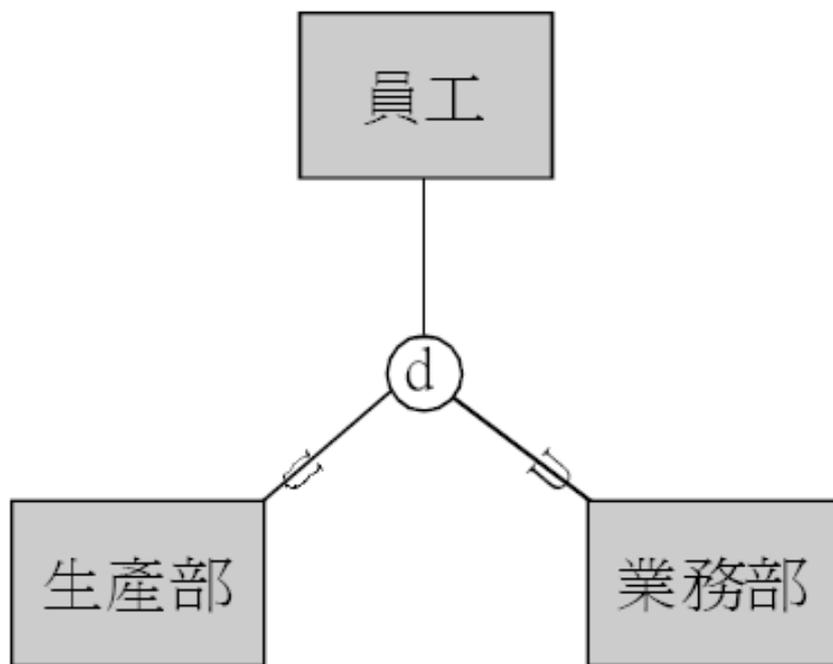
超類型和子類型的另一種轉換模式

- 若該公司員工只有生產部員工及業務部員工 (意即所有員工非生產部就是業務部), 則可將**員工**資料表中的所有屬性全部移至**生產部員工**及**業務部員工**兩個資料表中, 做為這兩個資料表的欄位。
- 如下圖所示：



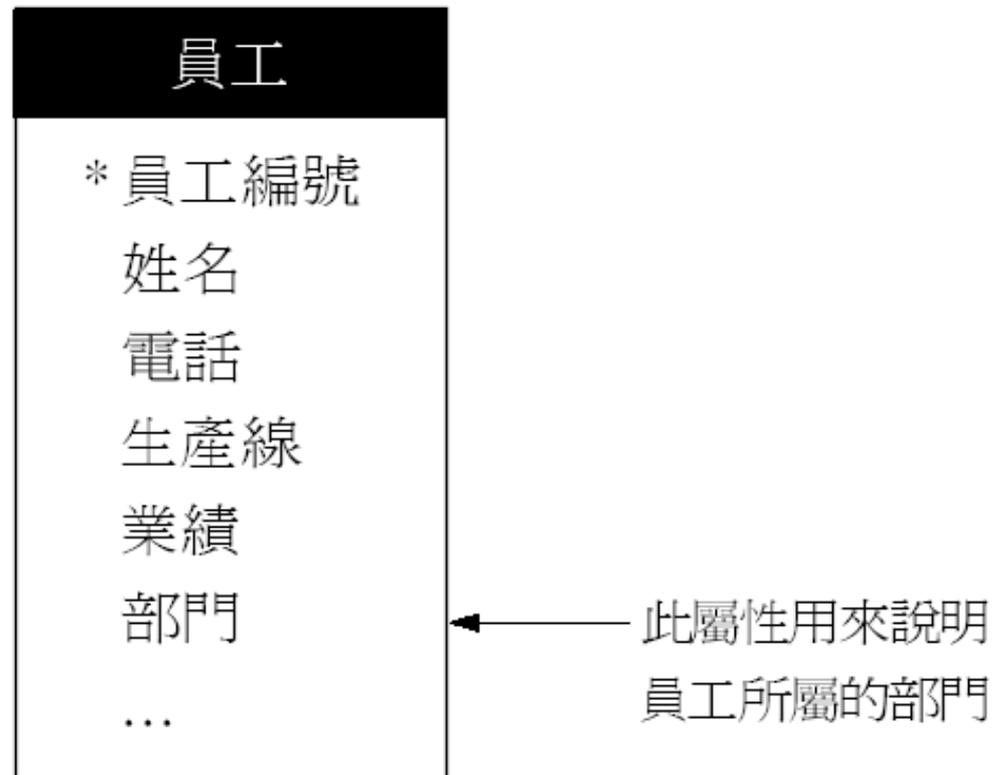
Disjoint 子類型的轉換

- 如果子類型為兩個不相交的情況，如下圖所示：



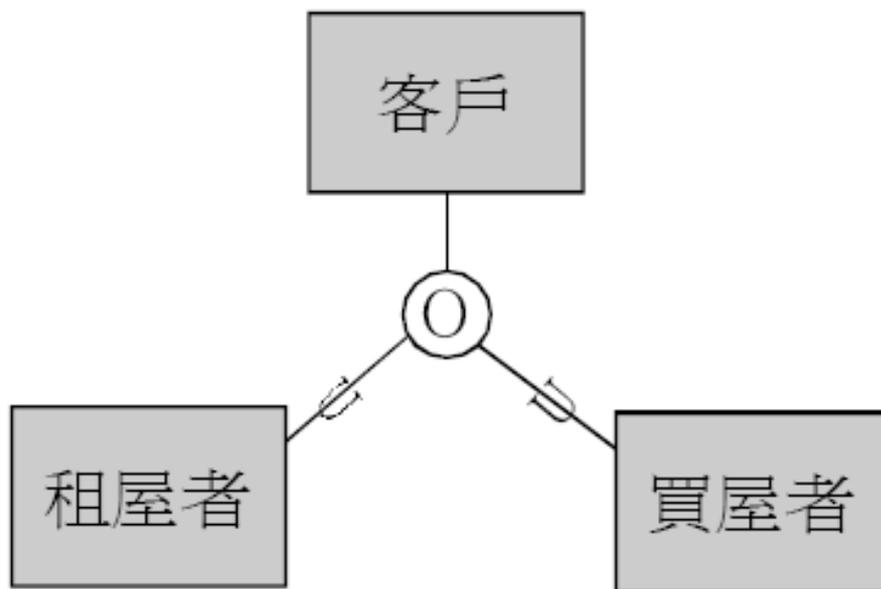
Disjoint 子類型的轉換

- 則我們會建立一個**員工**資料表，其中包含**員工**、**生產部員工**及**業務部員工**所有屬性的資料表，不過，在**員工**資料表中還需要建立一個特殊屬性，來說明員工隸屬的部門：



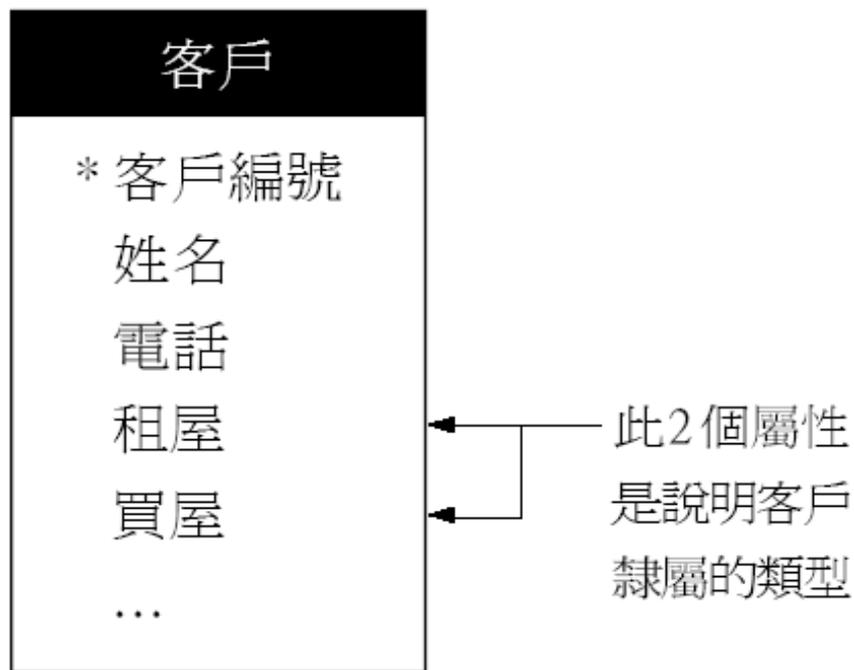
Overlap 子類型的轉換

- 如果子類型是屬於 Overlap (重疊) 的關係, 如右圖所示:



Overlap 子類型的轉換

- 則我們會建立一個**客戶**資料表，其中包含**客戶**、**租屋者**及**買屋者**所有屬性的資料表。
- 不過，在**客戶**資料表中還需要建立二個特殊屬性，來說明客戶隸屬的類型：



以『書籍銷售系統』為例，說明如何轉換為關聯式的資料表

- 了解上述轉換規則後，我們再來看看如何將**書籍銷售系統**轉換為實際運用的資料表。
 - Step 1：將實體轉換為資料表、屬性轉換為欄位
 - Step 2：建立資料表之間的關聯

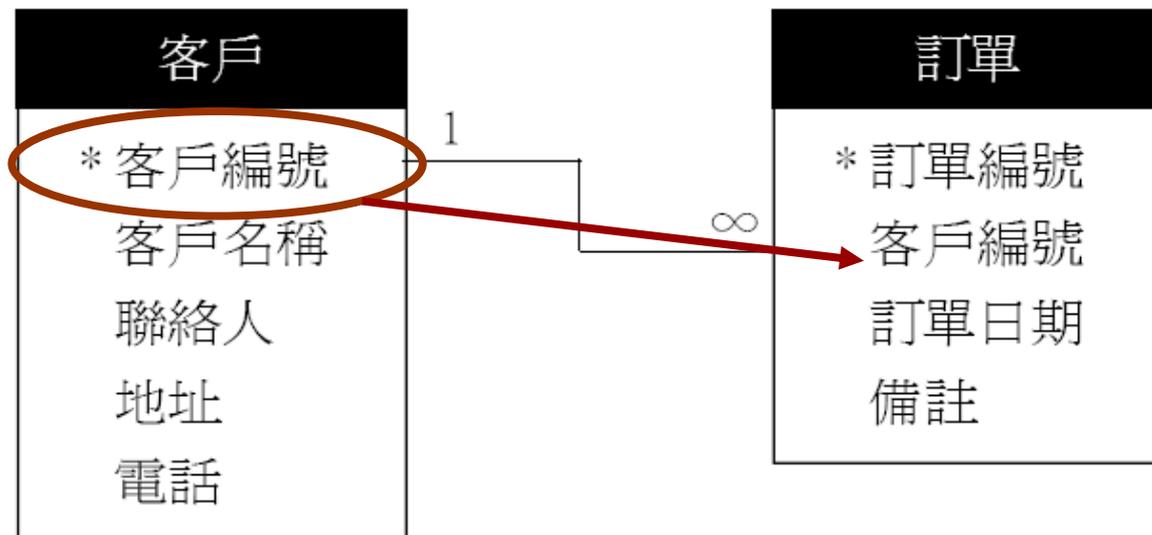
Step 1：將實體轉換為資料表、 屬性轉換為欄位

- 對於實體 - 關係圖中的所有 實體 和 屬性，分別轉換為 資料表 及 欄位，鍵屬性 則轉換為該資料表的 主鍵。
- 所以從 6-19 頁的圖，可得到以下資料表：

客戶	訂單	訂單細目	書籍	款項
* 客戶編號 客戶名稱 聯絡人 地址 電話	* 訂單編號 訂單日期 備註	* 細目編號 數量	* 書籍編號 書籍名稱 單價	* 款項編號 支付日期 應付金額

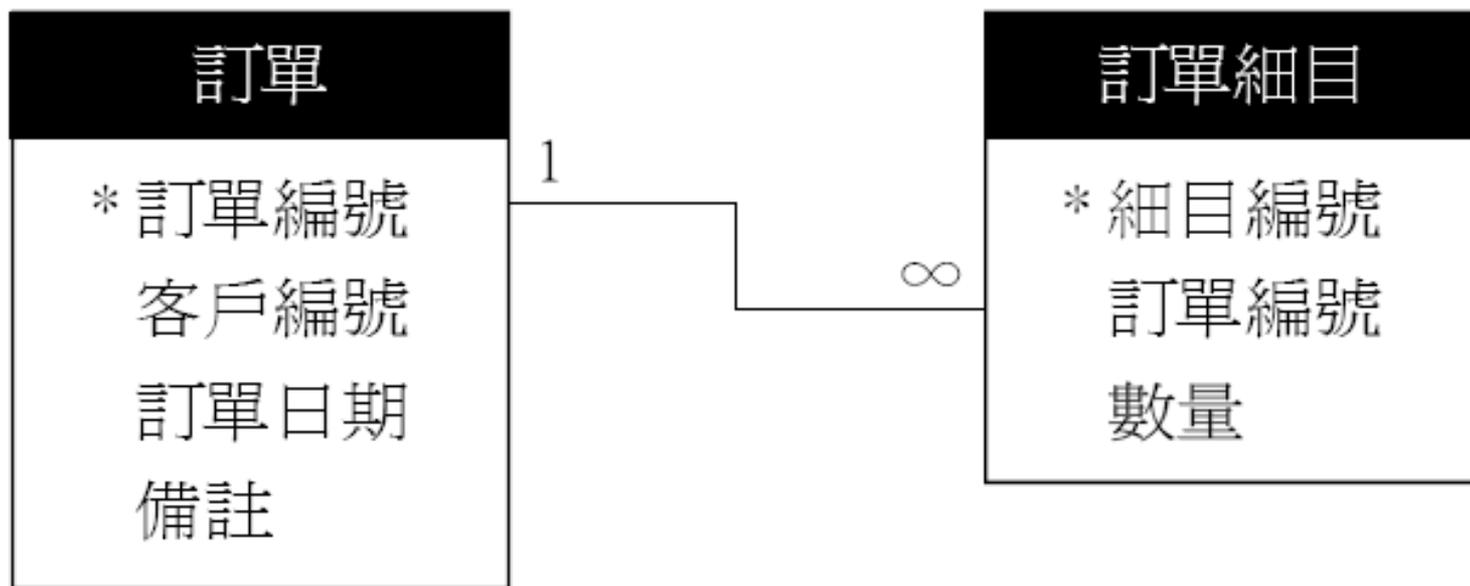
Step 2：建立資料表之間的關聯

- 接下來分別根據各實體之間的關係，轉換成資料表之間的關聯：
 - **客戶**與**訂單**實體：兩者的關係是一對多關係，所以要將**客戶**資料表的主鍵加入**訂單**資料表中，做為**訂單**資料表的連外鍵。



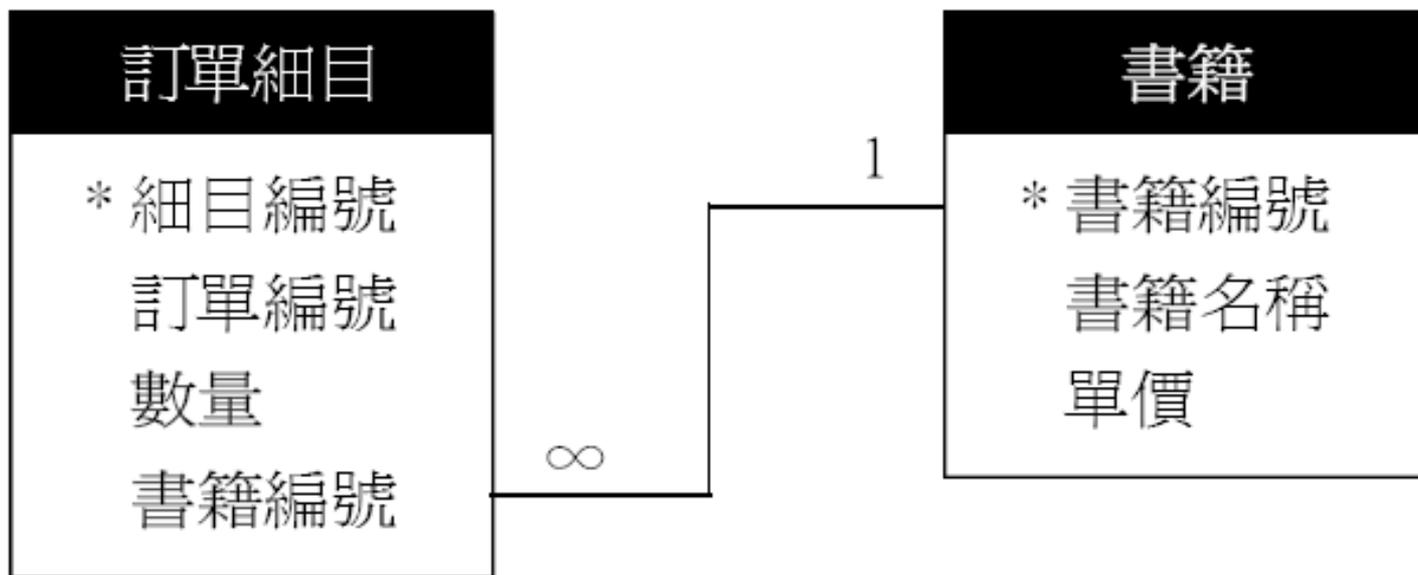
Step 2：建立資料表之間的關聯

- **訂單**與**訂單細目**實體：兩者的關係是一對多關係，所以要將**訂單**資料表的主鍵加入**訂單細目**資料表中，做為連外鍵。



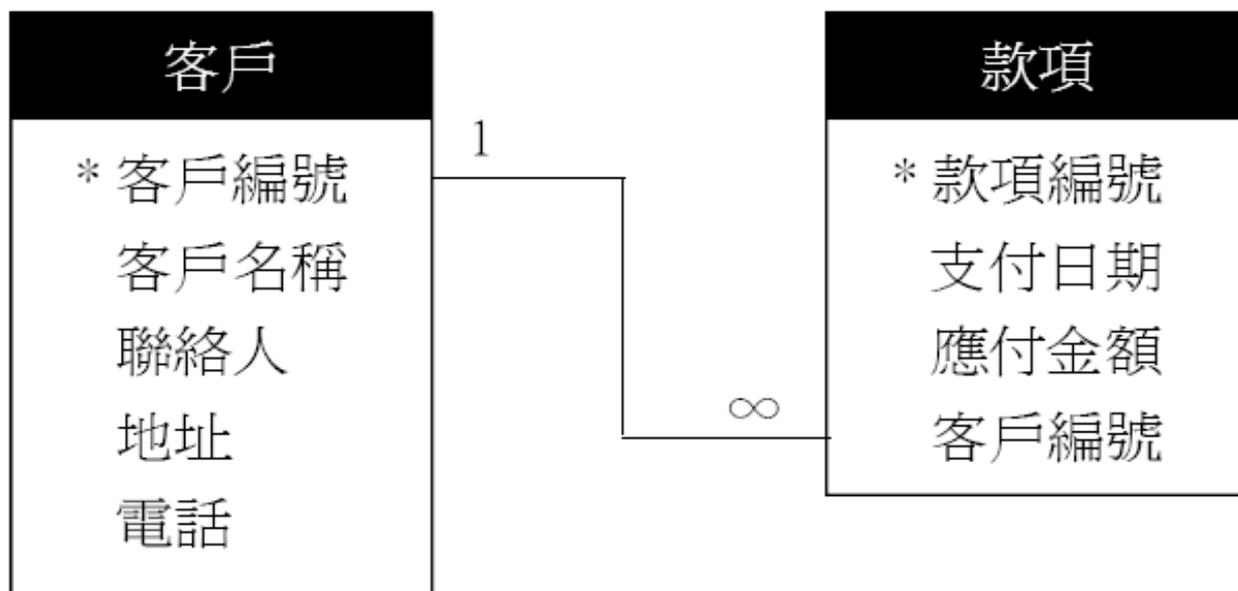
Step 2：建立資料表之間的關聯

- **訂單細目**與**書籍**實體：兩者的關係是多對一關係，所以要將**書籍**資料表的主鍵加入**訂單細目**資料表中，做為連外鍵。



Step 2：建立資料表之間的關聯

- **客戶**與**款項**實體：兩者的關係是一對多關係，所以要將**客戶**資料表的主鍵放入**款項**資料表中，做為連外鍵。



Step 2：建立資料表之間的關聯

- 總合這些關係圖，我們即可得到**書籍銷售**資料庫的資料表之間的關聯圖：

