資料庫的類型

William Wang

資料庫的類型

資料庫的類型

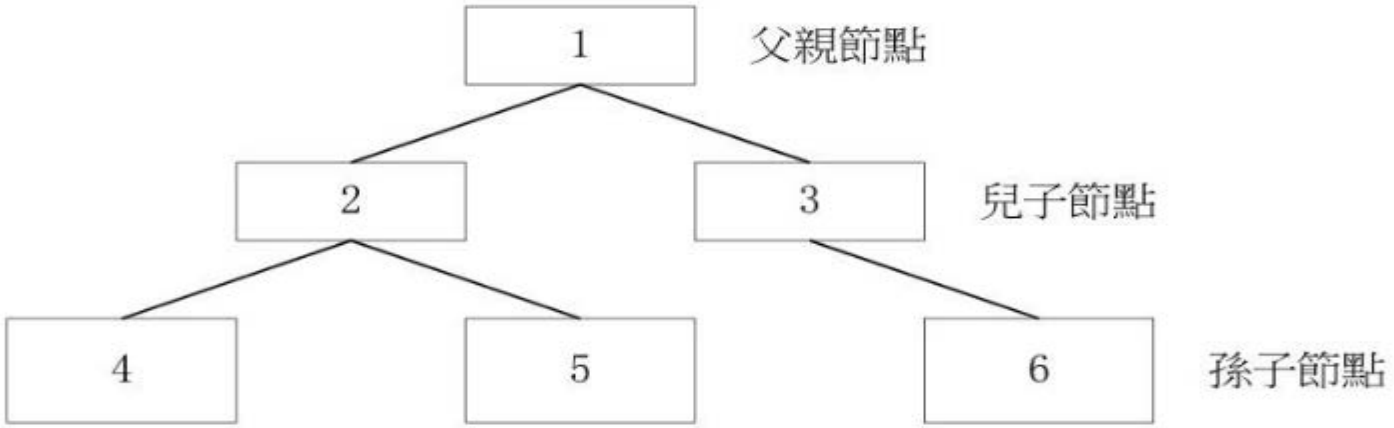
- 資料庫是儲存資料的地方, 就資料庫中儲存資料的架構來看, 又可分為多種類型
- 常見的有階層式資料庫、網狀式資料庫、關聯式資料庫及物件導向式資料庫

資料來源:旗標公司

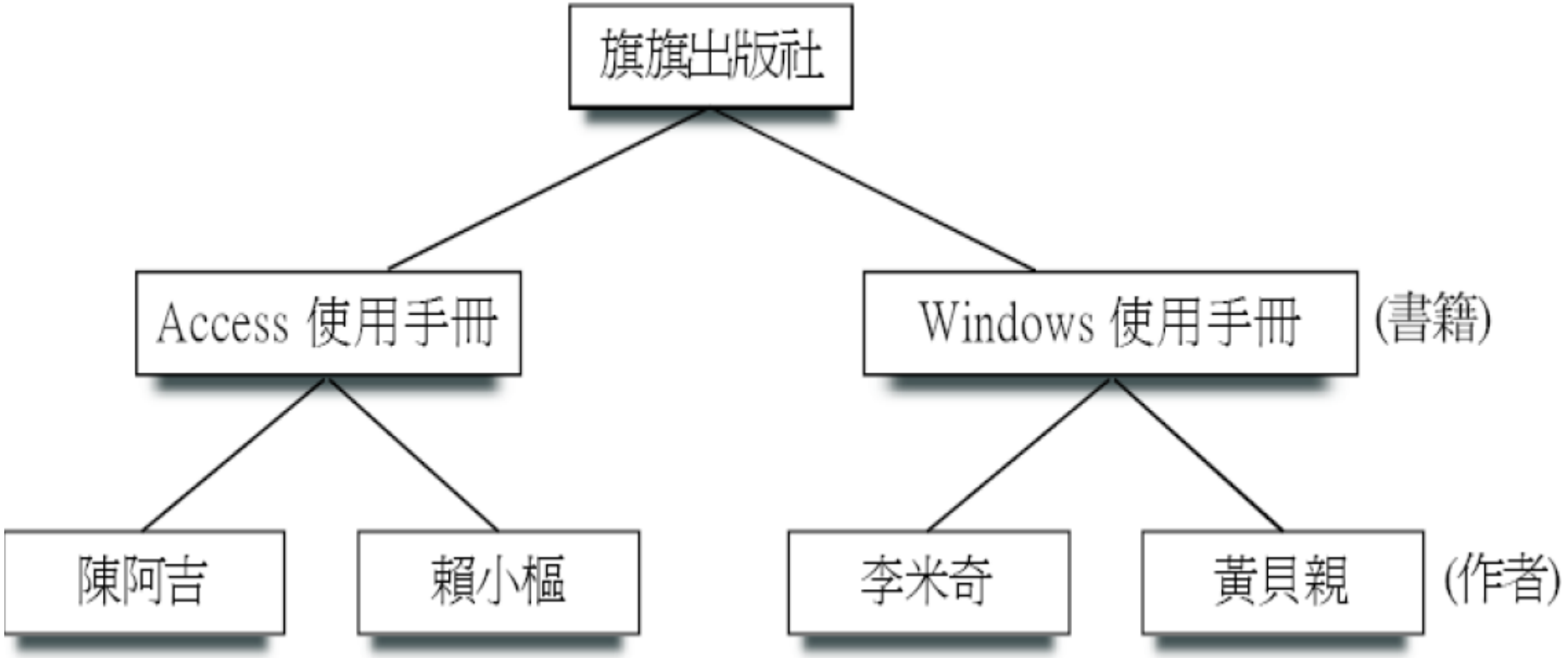
階層式資料庫(Hierarchical Database)

- 階層式資料庫是採用樹狀的結構, 將資料分門別類儲存在不同的階層下。
- 此類型的優點是資料結構類似金字塔, 對於在同一類型中不同階層資料的描述非常簡單且清楚。

■ 除了樹根以外的節點均只有一個直屬的「父親」節點



階層式資料庫(Hierarchical Database)



階層式資料庫(Hierarchical Database)

- 以上圖為例，可以很清楚地描述出版社和作者的關係。而它的缺點在於當資料的關係變得複雜時, 會造成管理及維護的不便。Why?
- 例如：我們要描述學生及老師的關係, 一個學生可受教於多個老師, 而一個老師又能教導多個學生, 此種情況下, 資料重複出現的機率很高, 會造成管理及維護上的不便。甚麼關係?
- 如IBM 公司的資料庫管理系統—IMS 即屬於此類。

網狀式資料庫(Network Database)

- 網狀式資料庫其實就是階層式資料庫的擴充, 我們可將每筆記錄想像成一個節點, 節點與節點間可以建立關聯(也就是建立記錄和記錄間的關聯), 形成一個複雜的網狀架構。
- 它的優點是避免了階層式資料庫中資料重複的問題, 缺點是關聯比較複雜, 尤其當資料庫的內容愈來愈多時, 要維護之間的關聯性就會變得非常複雜。
- 如 Computer Associates 發展的IDMS, 即是此類的資料庫管理系統。

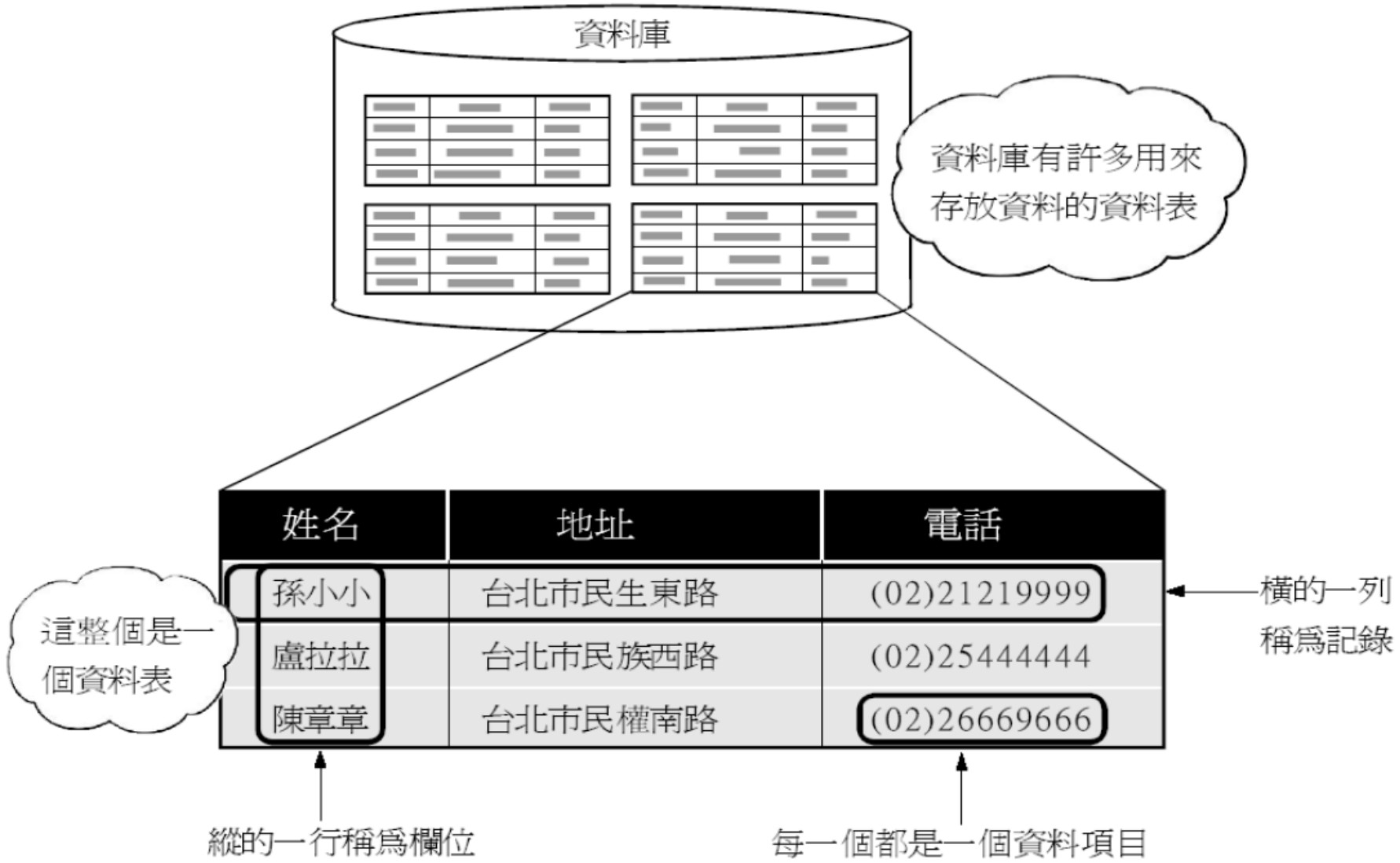
網狀式資料庫(Network Database)

- 圖中,我們利用作者姓名可查到他寫過的書, 這些書又由哪些出版社出版的關係, 當記錄的數量增加, 彼此的關係就容易變得牽扯不清。階層式資料庫如何劃?

關聯式資料庫(Relational Database)

- 關聯式資料庫是以2 維的矩陣來儲存資料(可以說是將資料儲存在表格的欄、列之中), 而儲存在欄、列裡的資料必會有所“關聯”, 所以這種儲存資料的方式才會稱為關聯式資料庫, 而儲存資料的表格則稱為“資料表”。
- □舉例來說, 通訊錄資料表的每一欄可以劃分為『姓名』、『地址』、『電話』：

關聯式資料庫(Relational Database)



關聯式資料庫(Relational Database)

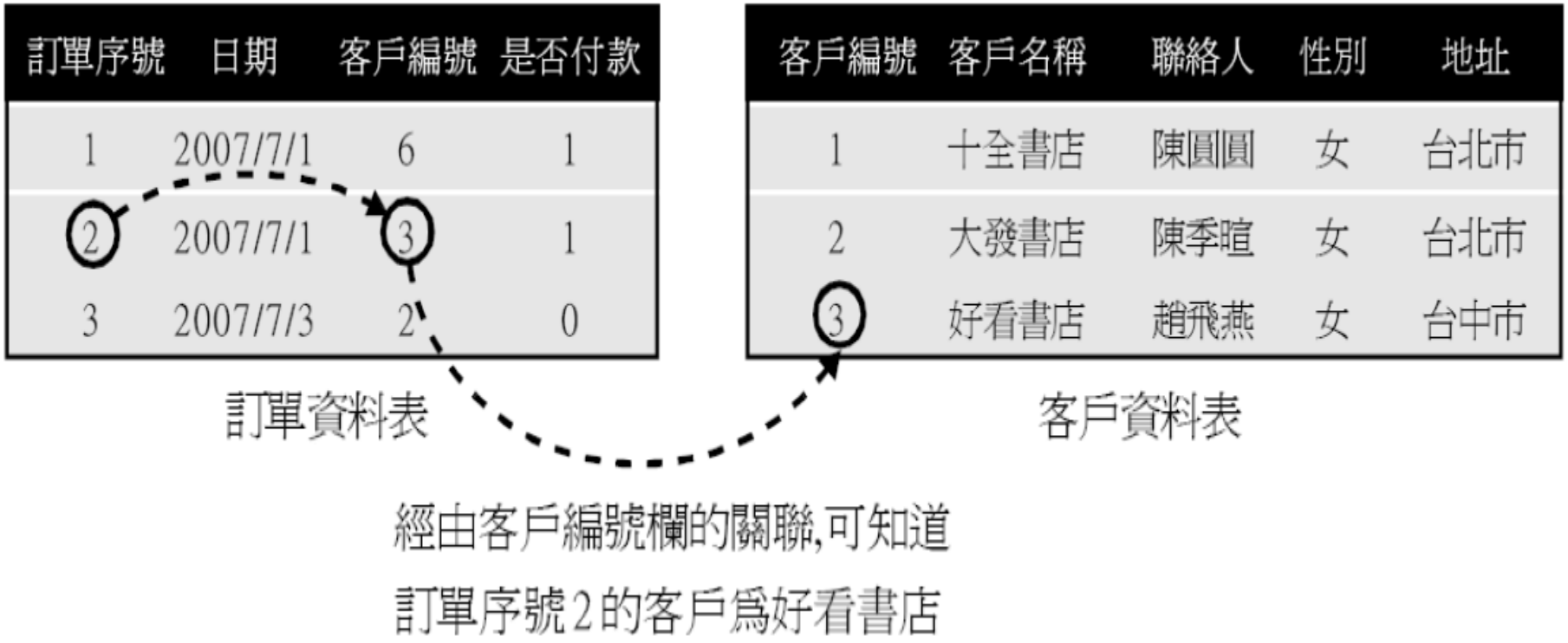
- 假如我們要從以上的資料表尋找“盧拉拉”的地址, 則是由橫向的『盧拉拉』與縱向的『地址』, 交相關聯而得來：

姓名	地址	電話
孫小小	台北市民生東路	(02)21219999
盧拉拉	台北市民族西路	(02)25444444
陳章章	台北市民權東路	(02)26669666

- 除了儲存在資料表行與列會有所關聯, 關聯式資料庫裡面的資料表之間通常也會互有關聯。

關聯式資料庫(Relational Database)

- 這種方式的優點是可以從一個資料表中的欄位, 透過資料表的關聯, 而找到另一個資料表中的資料：



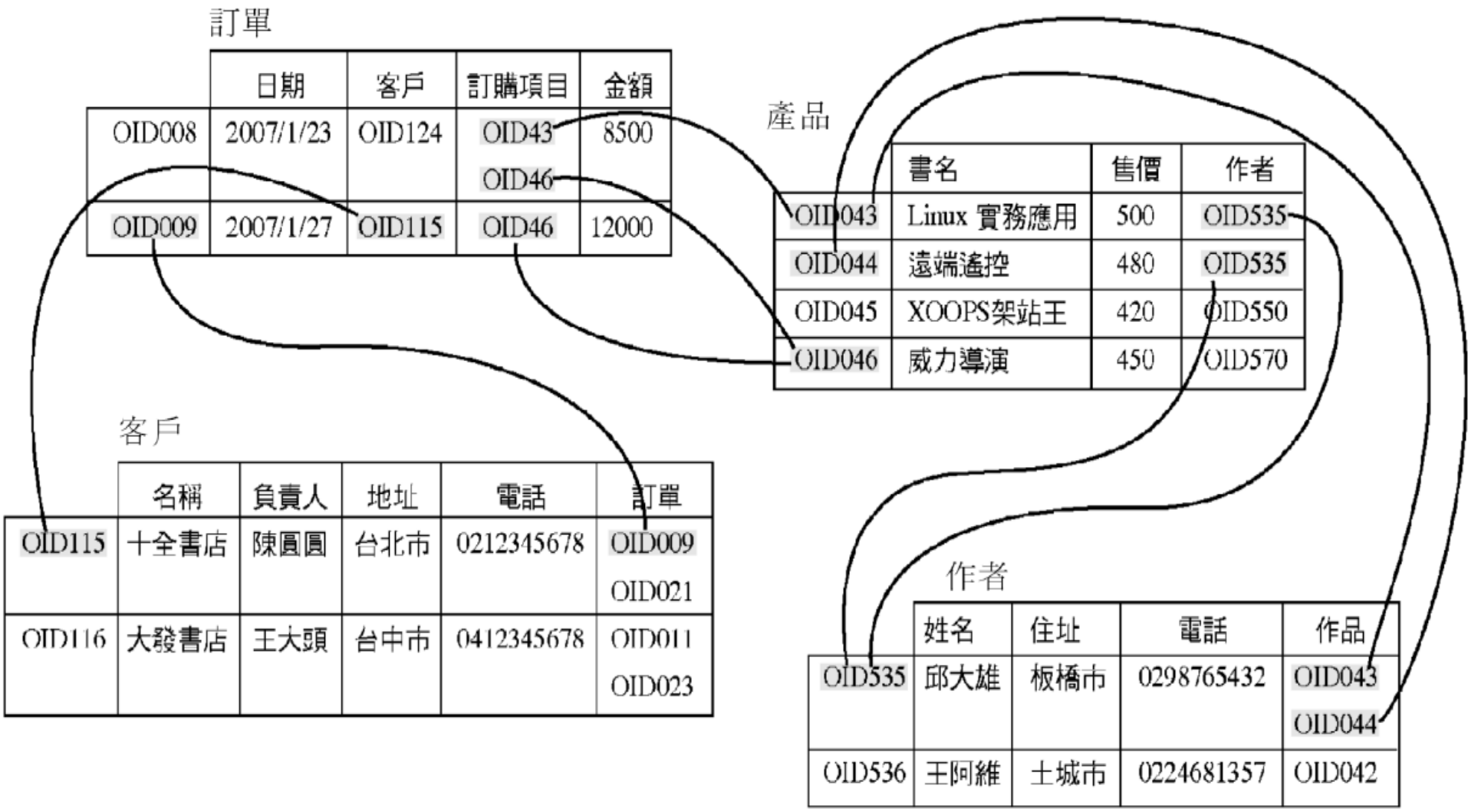
關聯式資料庫(Relational Database)

- 目前市場上是以關聯式資料庫使用最廣泛,
- 像Microsoft SQL Server、SyBase、Informix、MySQL、PostgreSQL、Access 、Maria 、Oracle...等, 都是屬於關聯式資料庫管理系統(Relational DBMS, RDBMS)。

物件導向式資料庫(Object-Oriented Database)

- 物件導向資料庫是以物件導向的方式來設計資料庫, 其中包含了物件的屬性、方法、類別、繼承等特性。
- 屬於這類的資料庫管理系統有Computer Associates 公司的Jasmine、Eastman Kodak 公司的Alltalk、Servio 公司的GemStone、O2 Technology 的O2 ...等資料庫管理系統。
- □此外也有關聯式資料庫為主, 再於其上架設物件導向概念的資料庫, 如PostgreSQL、Oracle。

物件導向式資料庫(Object-Oriented Database)



物件導向式資料庫(Object-Oriented Database)

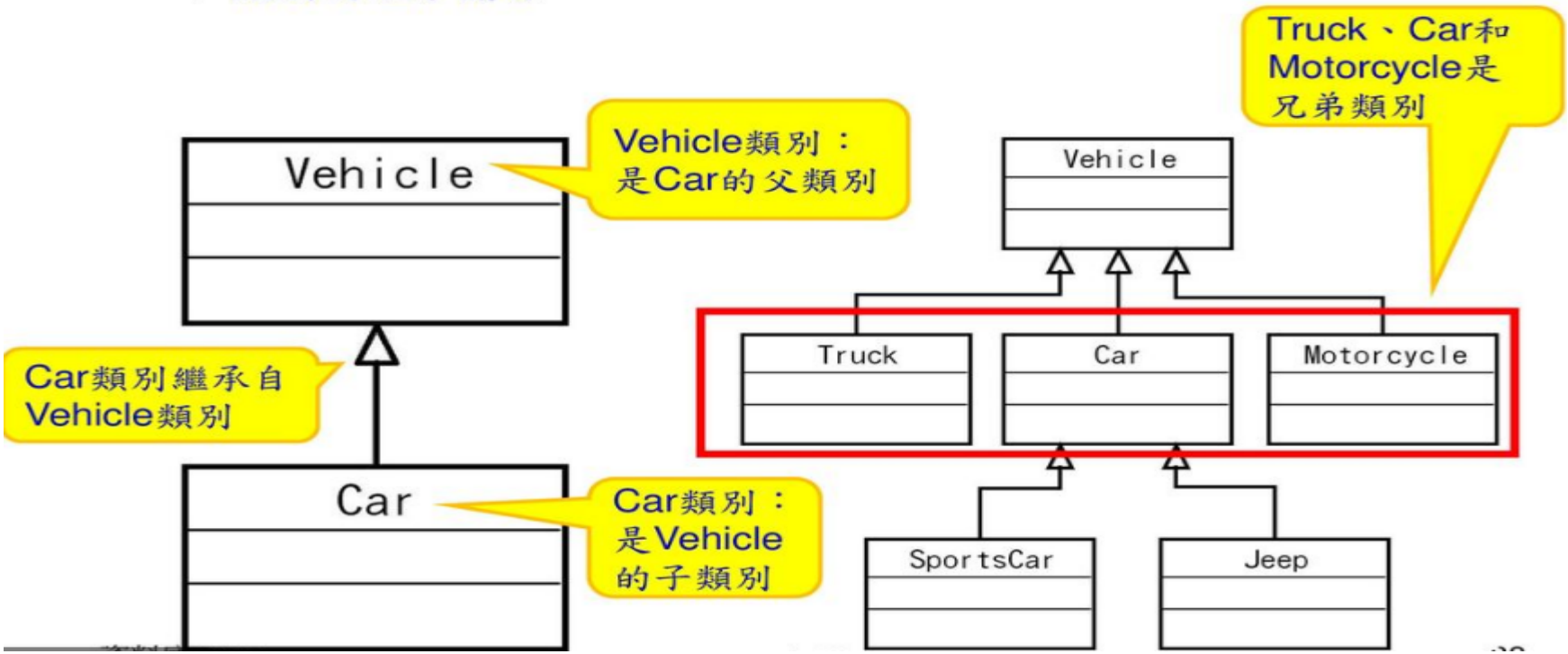
- 上列的示意圖中有幾個重點, 說明如下：
- □每一個橫列即為一個物件：
- □以訂單為例, 每一個物件包含了日期、客戶、訂購項目、金額等屬性(OID 是產生物件時的ID, 不是物件的屬性, 說明如後)。
- □這些屬性可以是文字資料、數值資料, 甚至是另一個物件, 而且一個屬性不必是唯一的值, 如上圖的訂單資料庫中, OID008 的物件, 其訂購項目屬性就包含OID43 及OID46 兩個物件。

物件導向式資料庫**(Object-Oriented Database)**

- 每個物件擁有唯一的Object IDentity (OID)：
- □同樣以訂單為例, 每個物件的第一欄就是物件的OID。
- □**OID 並不是資料庫設計者賦予的, 而是該物件成立時, 便自動產生一個OID**；要特別注意的是, OID 並不是物件的屬性, 實際上我們是看不到OID 的。
- □當物件內有包含其它物件時, 就能透過這個獨一無二的 OID 來快速找到對應用的物件。

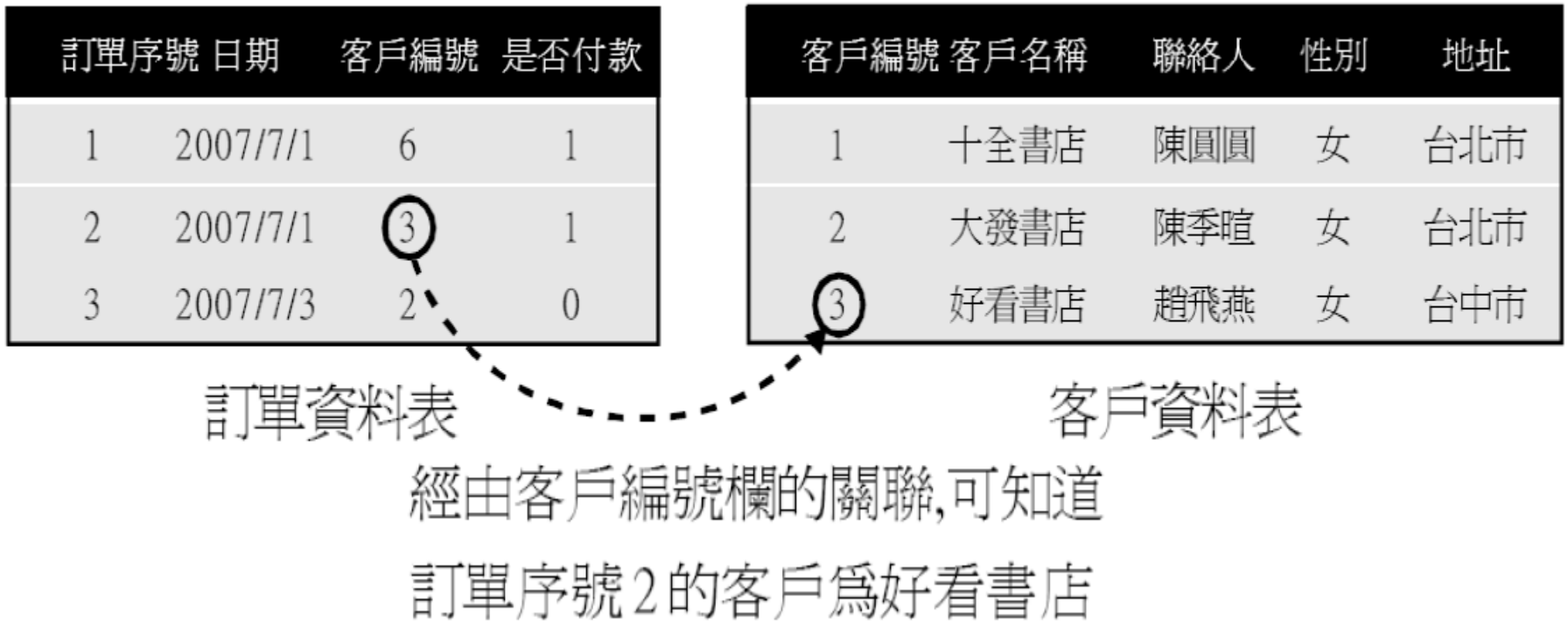
物件導向式資料庫(Object-Oriented Database)

- 若以關聯式資料庫和物件導向式資料庫來做比較, 關聯式資料庫必須由資料庫設計者來設計、建立、及管理關聯。
- □但物件導向式資料庫中, 物件和物件之間的連繫, 是因其屬性而必然發生的。



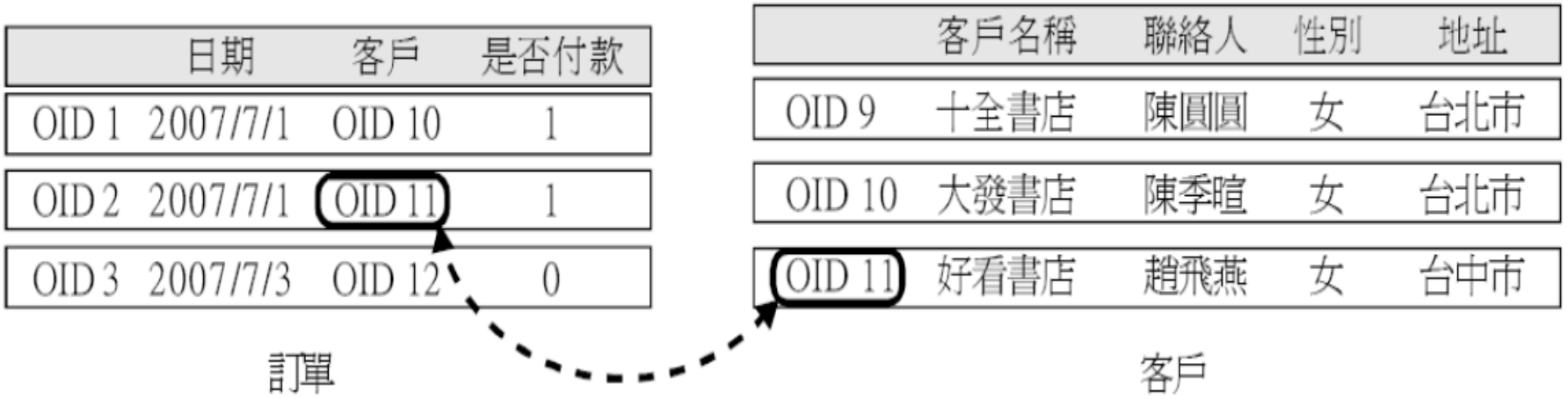
兩Databases差異

- 由下圖可知, 兩個資料表是藉由客戶編號來達成關聯的, 而這個關聯性在關聯式資料庫中, 必須由設計者自行建立才會真正產生關聯。



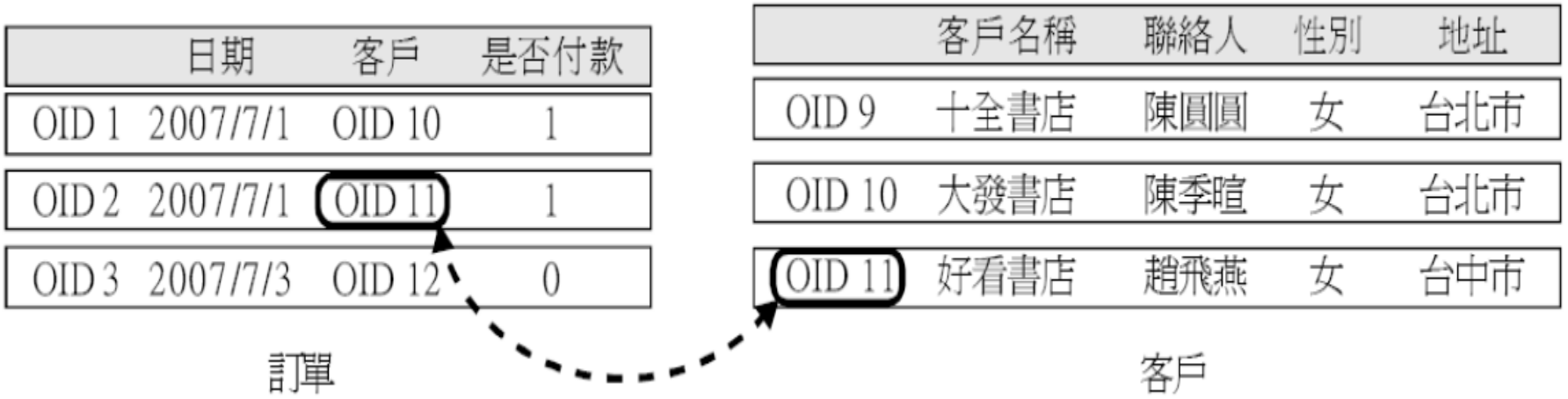
兩Databases差異

- 接著看下面的物件導向式資料庫：
- 下圖中, 兩個物件是透過OID 來連繫起來的。
- □簡單地說, 在關聯式資料庫中資料表間的關係必須靠**設計者**自行建立來產生關聯, 而物件導向式資料庫中, 各物件之間的關係則是在物件建立之時, 便會自行連繫起來。



兩Databases差異

- 接著看下面的物件導向式資料庫：
- 下圖中, 兩個物件是透過OID 來連繫起來的。
- □簡單地說, 在關聯式資料庫中資料表間的關係必須靠**設計者**自行建立來產生關聯, 而物件導向式資料庫中, 各物件之間的關係則是在物件建立之時, 便會自行連繫起來。



關聯式資料庫的表格+SQL

關聯式資料庫

- 關聯式資料庫是一組資料項目，項目之間具有預先定義的關係。這些項目會整理成由直欄和橫列構成的一組表格。
- 表格會儲存資料庫中所要表示的物件的相關資訊。表格的每一直欄儲存特定類型的資料，而每個欄位儲存某個屬性的實際數值。
- 表格中的橫列代表一個物件或實體的一組相關數值。表格的每一橫列可以用稱為主索引鍵的唯一識別符加以標記，而多個表格之間的橫列可使用外部索引鍵建立關聯。您不需要重新整理資料庫表格，即可用許多不同方法存取這些資料。

表格與資料的不一致

關聯式資料庫(Relation Database)

假設學校行政系統中有一個尚未分割的「學籍資料表」，如下表所示：

	學號	姓名	系碼	系名	系主任
#1	S0001	一心	D001	資工系	李春雄
#2	S0002	二聖	D001	資工系	李春雄
#3	S0003	三多	D002	資管系	李碩安
#4	S0004	四維	D002	資管系	李碩安
#5	S0005	五福	D002	資管系	李安

大量資料重複現象

由上表中，我們可以清楚看出多筆資料重複現象，如果有某一筆資料打錯，將會導致資料不一致現象。例如：在上表中的第5筆記錄的系主任，應該是「李碩安」卻打成「李安」。

解決方法之一: 關於資料不一致

因此，我們就必須要將原始的「學籍資料表」分割成數個不重複的資料表，再利用「關聯式資料庫」的方法來進行資料表的關聯。

何謂「關聯式資料庫」呢？它是由兩個或兩個以上的資料表組合而成。其目的：

1.節省重複輸入的時間與儲存空間。

2.確保異動資料(新增、修改、刪除)時的一致性及完整性。

因此，我們必須將各種資料依照性質的不同(如：學籍資料、選課資料，課程資料，學習歷程資料等.....)，分別存放在幾個不同的表格中，表格與表格之間的關係，則以共同的欄位值(如：「學號」欄位...)相互連結，以這種方式來存放資料的資料庫，在電腦術語中，稱為「關聯式資料庫(Relational Database)」。

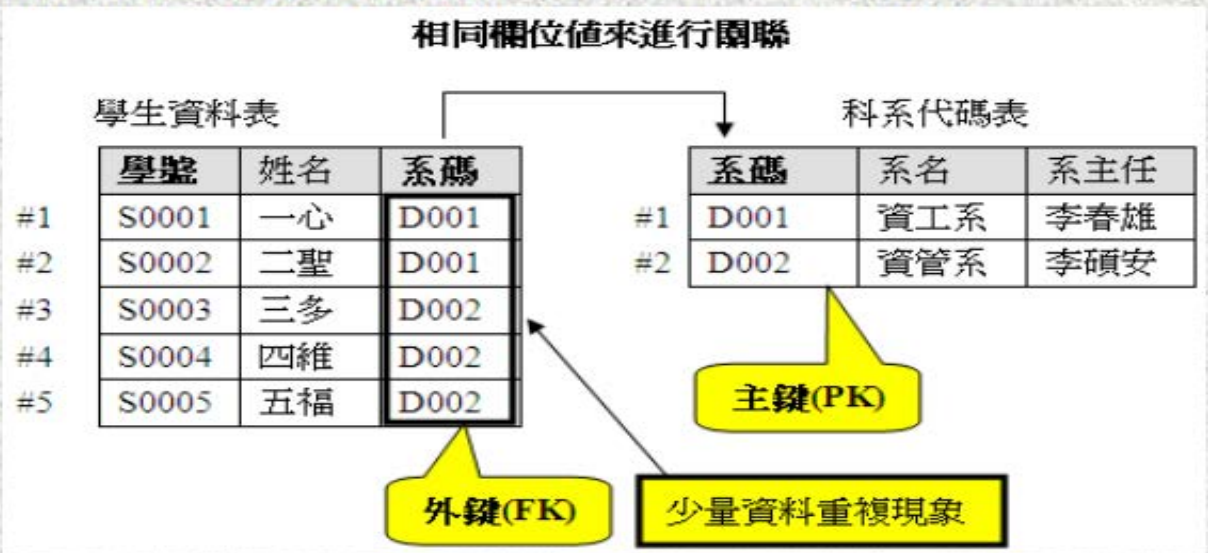
【定義】

1.是由一群相互關係的正規化關聯（表格）所組成。

2.關聯（表格）之間是透過相同的欄位值(即外鍵參考主鍵)來連繫。

3.關聯（表格）中的所有屬性內含值都是基元值(Atomic Value)。

因此，我們可以將上表中的「學籍資料表」分割為「學生資料表」與「科系代碼表」，如何產生關聯式資料庫呢？它是透過兩個資料表的相同欄位值(即系碼)來進行連結。如下所示：



註：「主鍵」與「外鍵」專有名詞會有後面章節中詳細介紹。

【優點】

1.節省記憶體空間

相同的資料記錄不須要再重複輸入。

2.提高行政效率

因為資料不須再重複輸入，故可以節省行政人員的輸入時間。

3.達成資料的一致性

因為資料不須再重複輸入，故可以減少多次輸入產生人為的錯誤。

	學號	姓名	系碼	系名	系主任
#1	S0001	一心	D001	資工系	李春雄
#2	S0002	二聖	D001	資工系	李春雄
#3	S0003	三多	D002	資管系	李碩安
#4	S0004	四維	D002	資管系	李碩安
#5	S0005	五福	D002	資管系	李安

大量資料重複現象

相同欄位值來進行關聯

學生資料表

	學號	姓名	系碼
#1	S0001	一心	D001
#2	S0002	二聖	D001
#3	S0003	三多	D002
#4	S0004	四維	D002
#5	S0005	五福	D002

科系代碼表

	系碼	系名	系主任
#1	D001	資工系	李春雄
#2	D002	資管系	李碩安

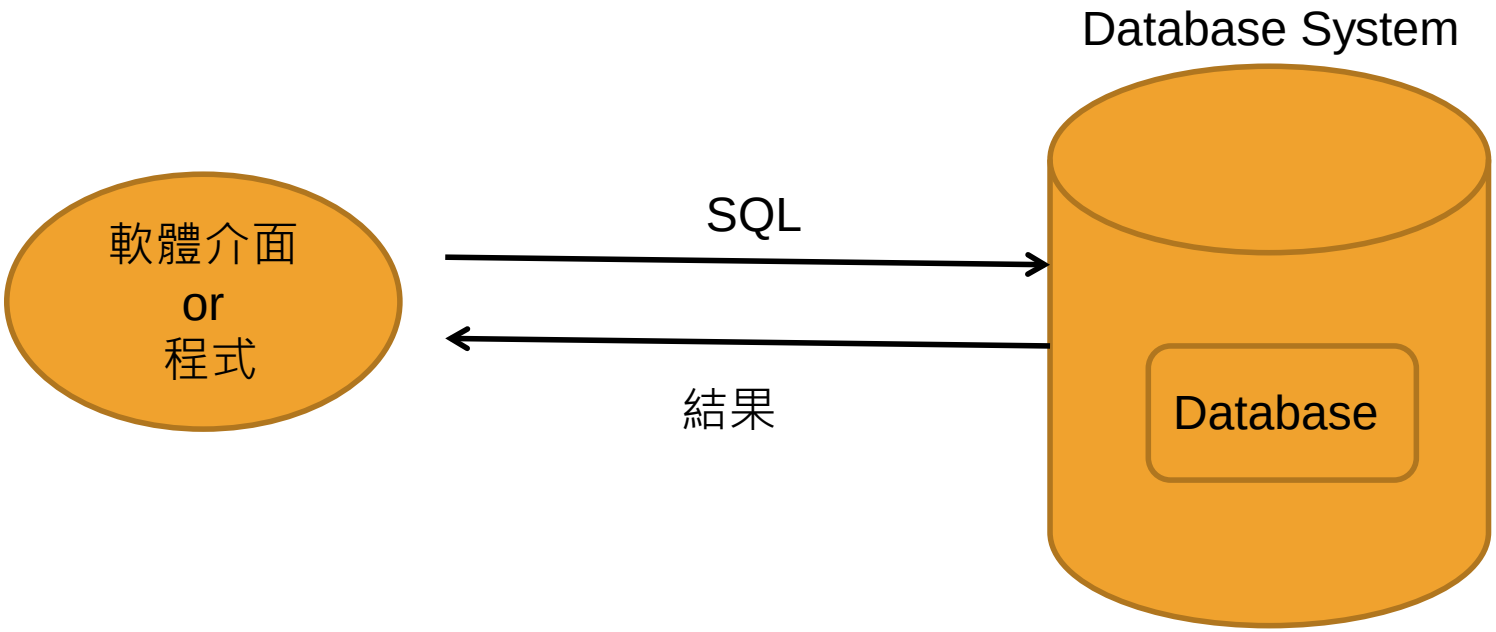
主鍵(PK)

外鍵(FK)

少量資料重複現象

SQL

- SQL是一種 用來從資料庫讀取與儲存資料的電腦語言
- https://www.1keydata.com/tw/sql/sql.html



SQL

- DML (資料操作性質的SQL)
  - 新增: Insert
  - 修改: Update
  - 刪除: Delete
  - 查詢: Select

Store_Information 表格

Store_Name	Sales	Txn_Date
Los Angeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Angeles	300	08-Jan-1999
Boston	700	08-Jan-1999

資料庫系統的處理架構

資料庫系統的處理架構

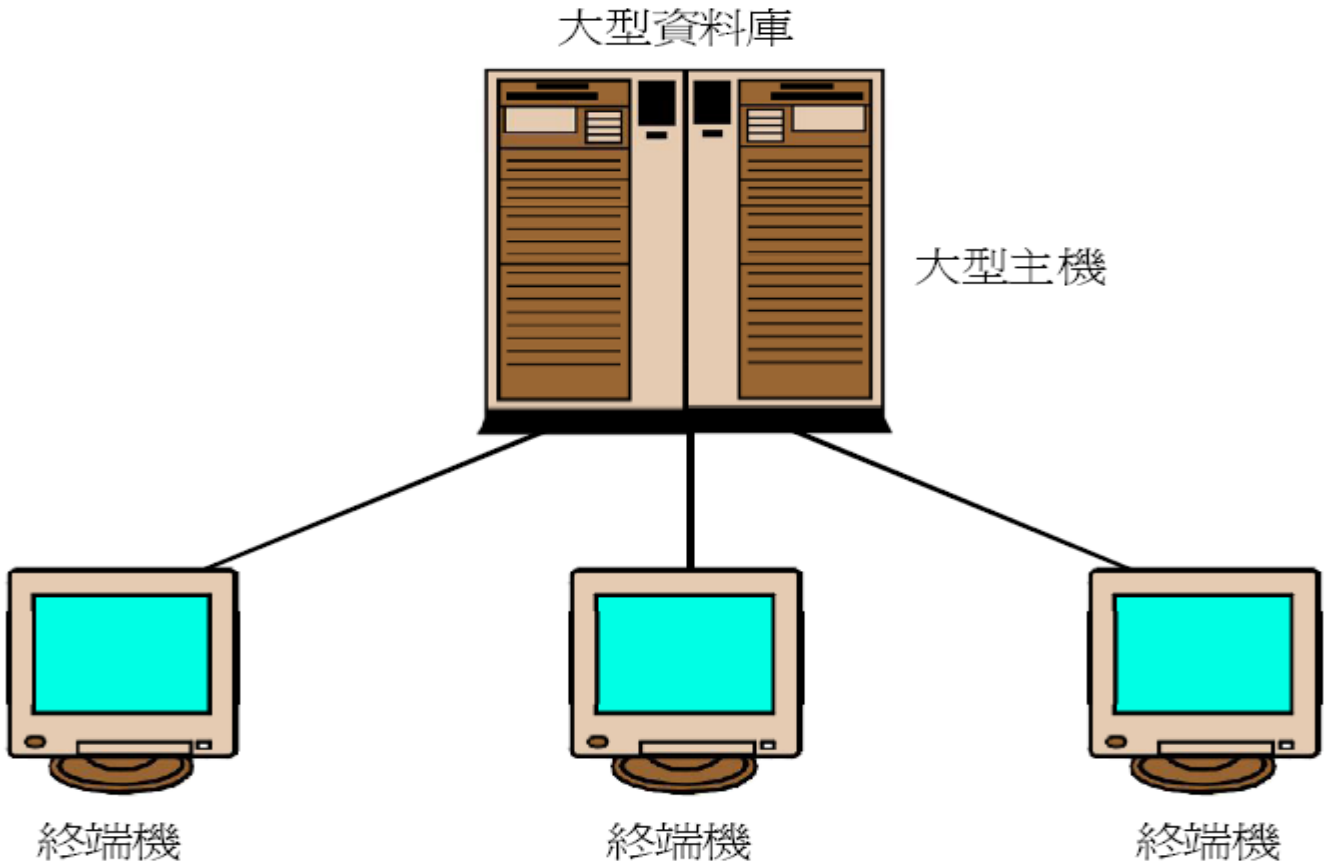
- 了解資料庫的類型後, 接下來我們要介紹如何部署您的資料庫系統, 通常我們會依照組織的規模、資料量的多寡、使用的人數及軟硬體的設備等條件來考量, 常見的有4 種架構。
- 單機架構
- □大型主機/ 終端機架構
- □主從式架構(Client / Server)
- □分散式架構

單機架構

- 在此架構中, 利用一台電腦完成所有的工作, 包含使用者存取資料、DBA 管理及維護資料庫...等, 適合在使用者少且資料量也不多的環境下使用, 例如小公司或個人使用者建立的資料庫。

大型主機/ 終端機架構

- 在這種架構中, 由一台大型主機負責儲存及處理資料, 所有的用戶端僅供操作, 沒有處理資料的能力, 只能透過鍵盤及終端機傳送和顯示大型主機的訊息。



大型主機/ 終端機架構

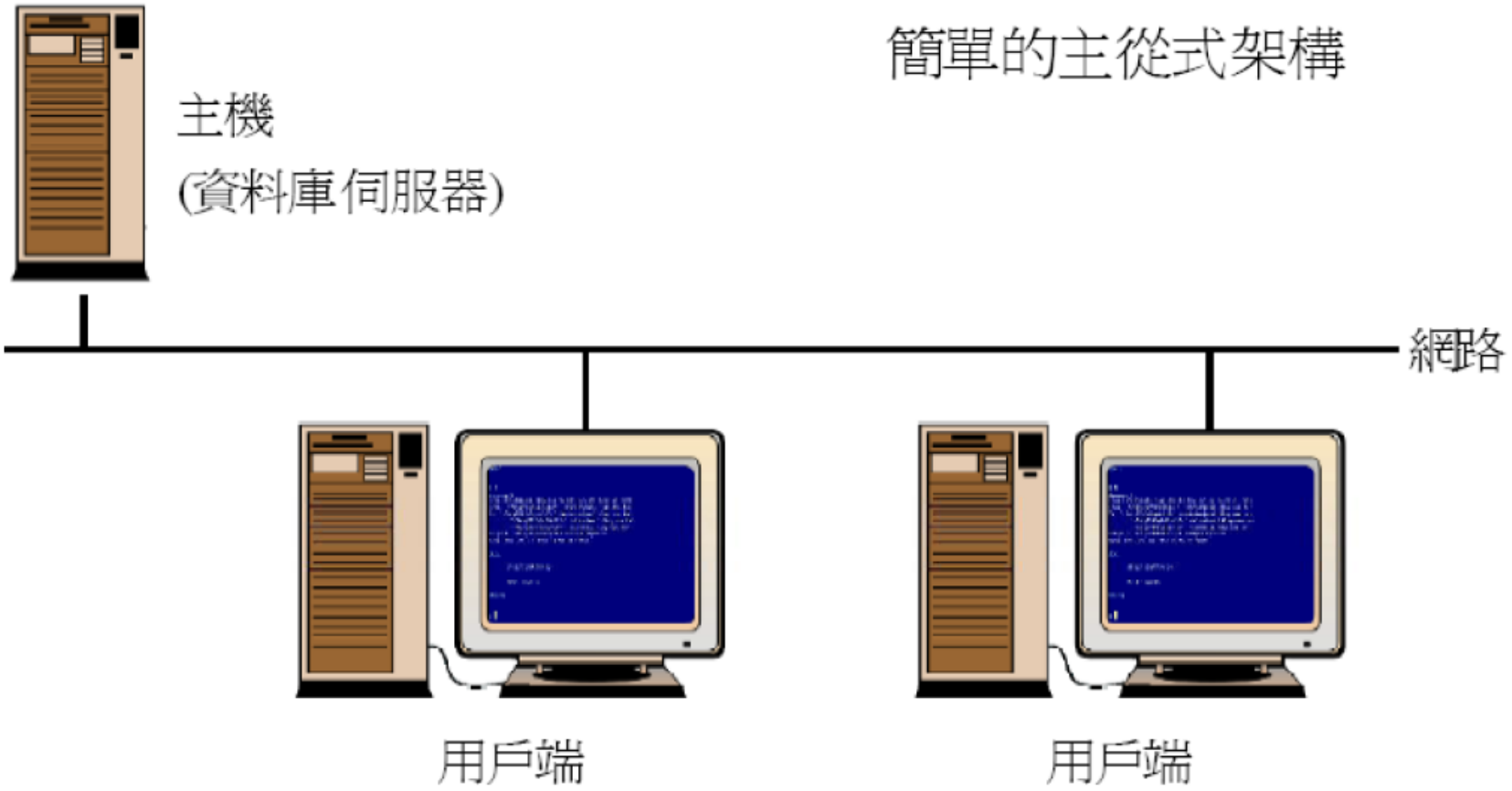
- 這種“集中式”管理的優點在於主機完全掌控系統的資源, 所有管理及維護工作都只要針對主機即可, 環境較單純；
- □但缺點則是只有一台主機執行工作, 當連線的使用者增加時, 會因為處理的工作增加而降低執行的效率。
- □目前除了一些大型的機構外, 中小企業甚少使用此架構, 且因為大型主機的價格都很昂貴, 一般較無能力負擔。

主從式架構(Client / Server)

- 近年來, 由於個人電腦的技術突飛猛進、軟體的功能愈來愈強以及網路的普遍, 因此不再由單一的大型主機來負責所有的工作。
- □基於分工的原則, 於是利用一台處理效能較強的電腦作為主機, 來維護資料庫及處理使用者提出的要求, 再利用使用者的個人電腦來分擔部分主機的工作(例如提供操作介面及應用程式), 這就是主從式架構。

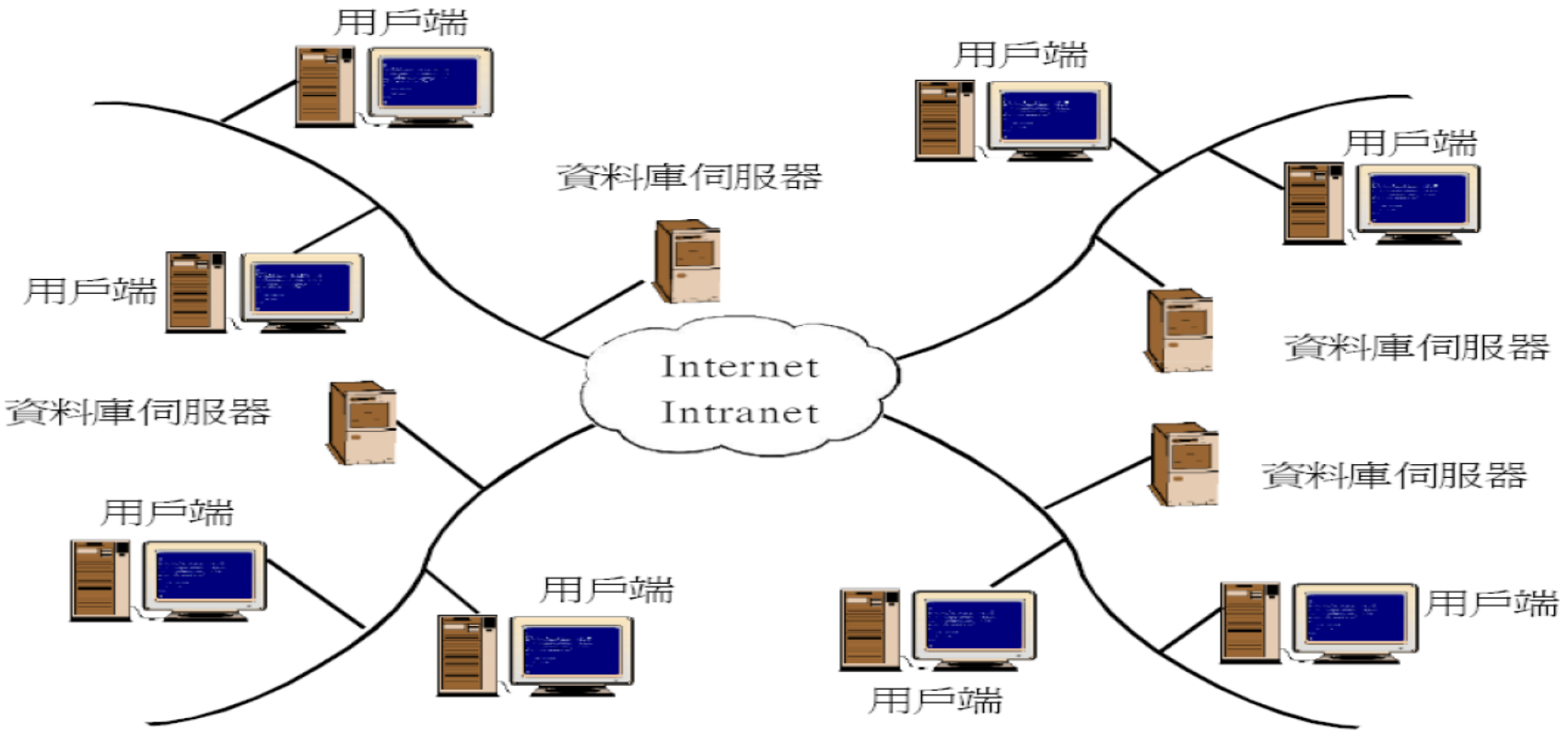
主從式架構(Client / Server)

- 在這種架構下, 主機能保留更多的效能來處理更多的使用者的連線。同時也不需要花費大量的金錢在購置大型主機上。



分散式架構

- 分散式架構是利用數台資料庫伺服器來分別處理使用者的連線, 其實這種架構也反映到現實社會中, 因為許多企業本身就是分散的, 不論是分散在各地的分公司或是總公司中的各部門。□基本上, 他們的資料都是各自獨立在各處。



分散式架構

- 以上圖為例, 使用者透過網路存取資料, 這些資料可能分別來自不同的主機, 如此分擔了一台主機的工作, 執行起來效能會更佳。
- □看完上述資料庫架構, 您或許會問Access 到底適合在哪一種架構中使用？
- □原則上Access 適合在單機架構、主從式架構或分散式架構中使用, 由於它只是小型的資料庫管理系統, 能處理的資料量有限, 且處理效能較低於一般大型資料庫, 所以並不適合資料量大的大型公司。

資料庫管理系統的基本功能

資料庫管理系統的基本功能

- 在之前中我們提過, 資料庫管理系統就是管理資料庫的軟體系統, 它們負責資料庫的建立、資料存取、權限設定、資料備份、操作的監督與記錄...等工作,
- □底下我們就再進一步詳述資料庫管理系統所應具備的基本功能。

資料庫管理系統的基本功能

- 資料定義：
  - DBMS 必須能夠充分定義並管理各種類型的資料項目。
  - 例如關聯式資料庫管理系統必須具備建立資料庫、資料表、定義各欄位的資料型別、限制, 以及資料表之間的關聯...等等的能力才行。
- 資料處理：
  - DBMS 必須提供使用者對資料庫存取的能力, 包括新增、修改、查詢與刪除的基本功能。
  - 有時DBMS 提供的功能雖然完善, 但並不適合一般使用者操作, 這時就需要設計師另外撰寫用戶端的應用程式, 以供一般使用者操作。

資料庫管理系統的基本功能

- 資料安全：
  - □DBMS 應該具備設定使用者帳戶、密碼及權限的功能, 讓每一個使用者只能存取授權範圍內的資料, 以防止機密資料外洩或資料遭受任何有意或無意的破壞。
- 資料備份：
  - □DBMS 必須提供方便的資料備份功能, 如此在資料庫不幸意外毀損時, 還可還原到備份資料時的狀況, 以減少損失。

資料庫管理系統的基本功能

- 此外, 維護資料庫的效率也非常重要, 尤其是在資料量很大或使用者很多的情況, 資料庫若因效率不佳而導致存取速度變慢, 亦會嚴重影響到操作人員的工作效率。

Q&A