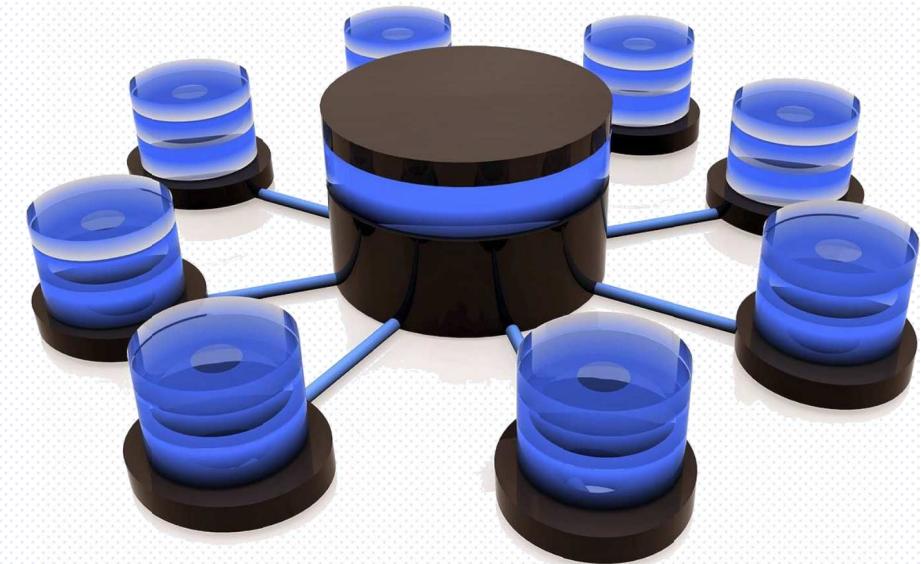


資料庫系統簡介

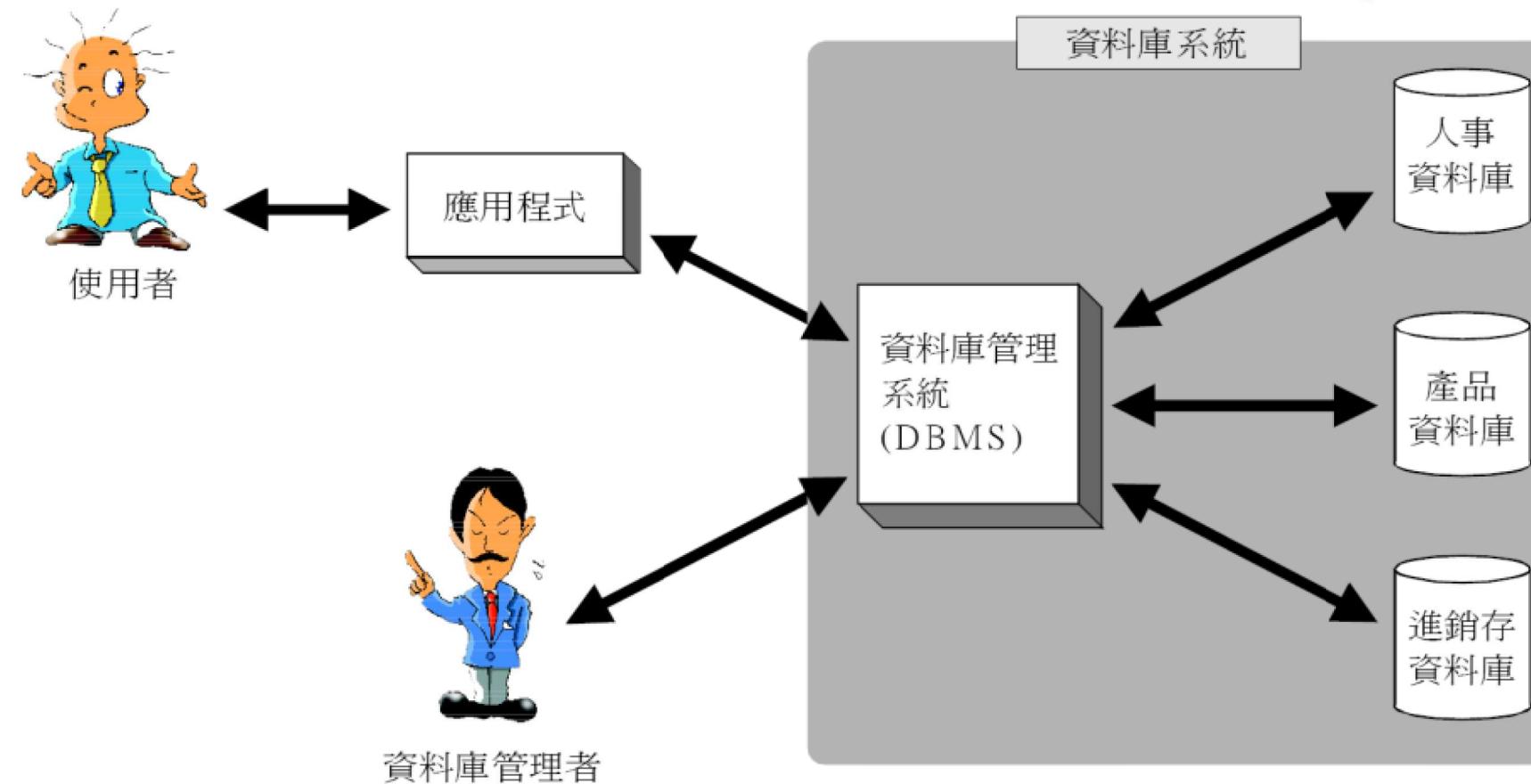
Introduction to Database





資料庫系統簡介

- ◆ 資料庫系統 (Database System) 是電腦化的資料儲存系統，使用者則透過各種應用程式來存取其中的資料。
- ◆ 資料庫系統又可分為兩個部份：資料庫 (Database) 與資料庫管理系統 (DataBase Management System, DBMS)。





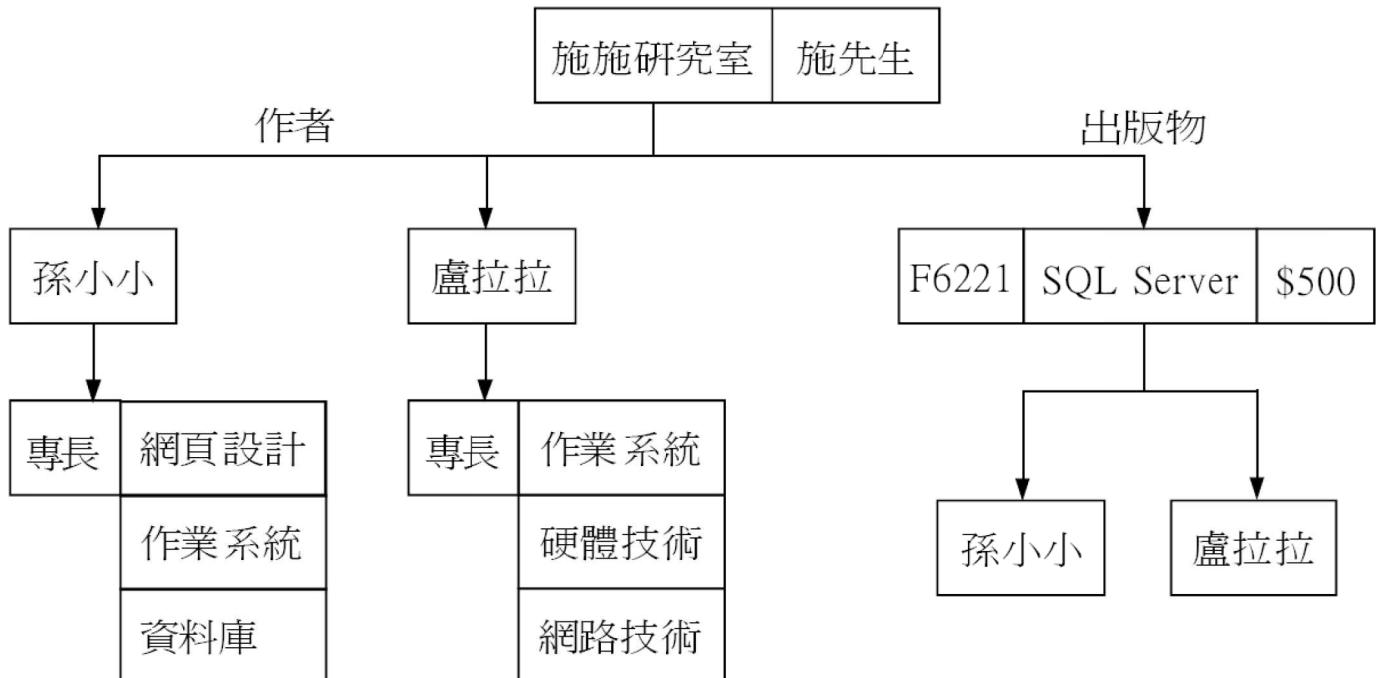
資料庫系統簡介

- ◆ 資料庫是儲存資料的地方。一個資料庫系統中可以有多個資料庫，每個資料庫都是一組經過整理好的資料集合。
- ◆ 一般，我們會將資料庫想像成是一個存放資料的容器，但資料庫的真實型態其實是一個個的電子檔案 (file)。
- ◆ 資料庫管理系統則是指管理資料庫的軟體，它們負責使用者與資料庫之間的溝通，如存取資料庫中的資料、以及管理資料庫的各項事務等。
- ◆ Microsoft 的 Access，許多用在大型資料庫系統上的 Microsoft SQL Server、Oracle、SyBase、Informix、MySQL、PostgreSQL ... 等皆是資料庫管理系統。



階層式資料庫 (Hierarchical Database)

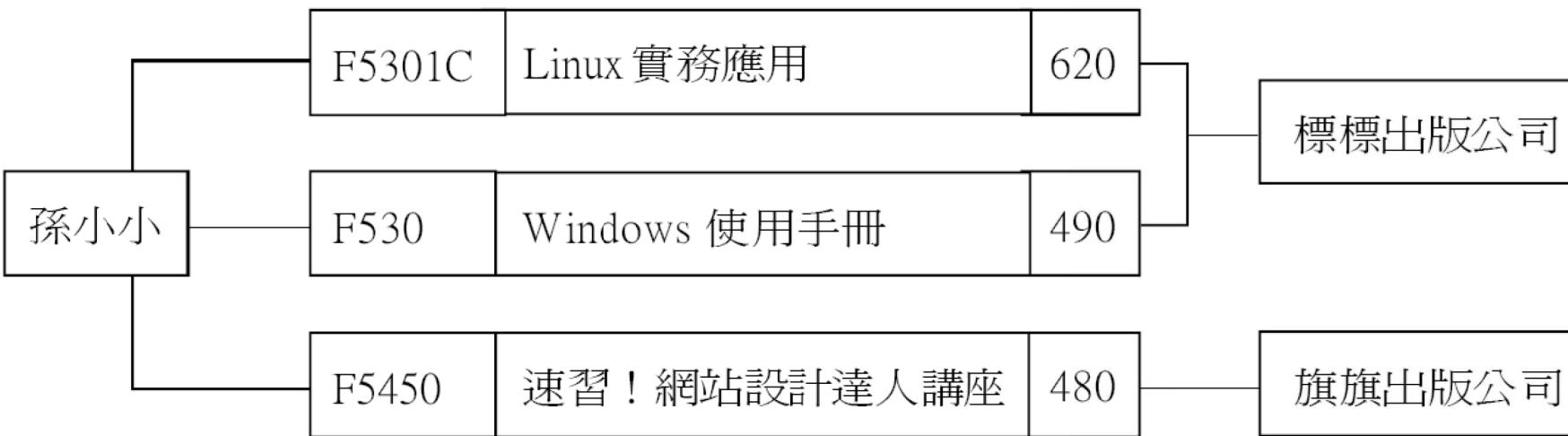
- ◆ 階層式資料庫採用樹狀結構，將資料分門別類儲存在不同的階層之下。
- ◆ 此類型的優點是資料結構很類似金字塔，不同層次間的資料關聯性直接且簡單；缺點則因資料以縱向發展，橫向關聯難以建立，所以資料可能會重複出現，造成管理維護上的不便。IBM 的 IMS 即是屬於此類的資料庫管理系統。





網狀式資料庫 (Network Database)

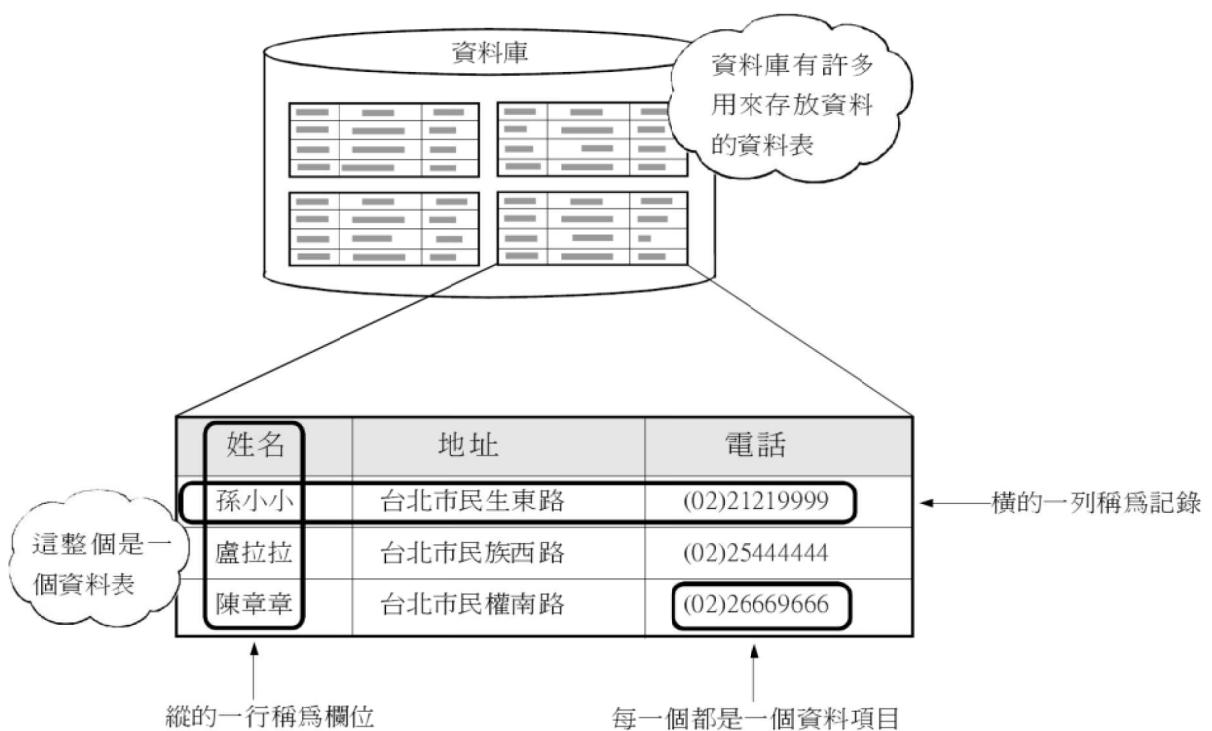
- ◆ 網狀式資料庫是將每筆記錄當成一個節點，節點與節點之間可以建立關聯 (也就是建立記錄與記錄間的關聯)，形成一個複雜的網狀架構。
- ◆ 優點是避免了資料的重複性，缺點是關聯性比較複雜，尤其是當資料庫的內容愈來愈多的時候，關聯性的維護會變得非常麻煩。
- ◆ Computer Associates 公司曾經推出的 IDMS 即是屬於此類的資料庫管理系統。
- ◆ 上圖表示從作者姓名可以查到他寫過的書，以及這些書是由哪些出版公司所出版的關係。當記錄的數量一多的時候，關聯就容易變得牽扯不清。





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)

◆ 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



關聯式資料庫 (Relational Database)

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

訂單序號	日期	客戶編號	是否付款
1	2005/7/1	6	1
2	2005/7/1	3	1
3	2005/7/3	2	0

訂單資料表

客戶編號	客戶名稱	聯絡人	性別	地址
1	十全書店	陳圓圓	女	台北市
2	大發書店	陳季暄	女	台北市
3	好看書店	趙飛燕	女	台中市

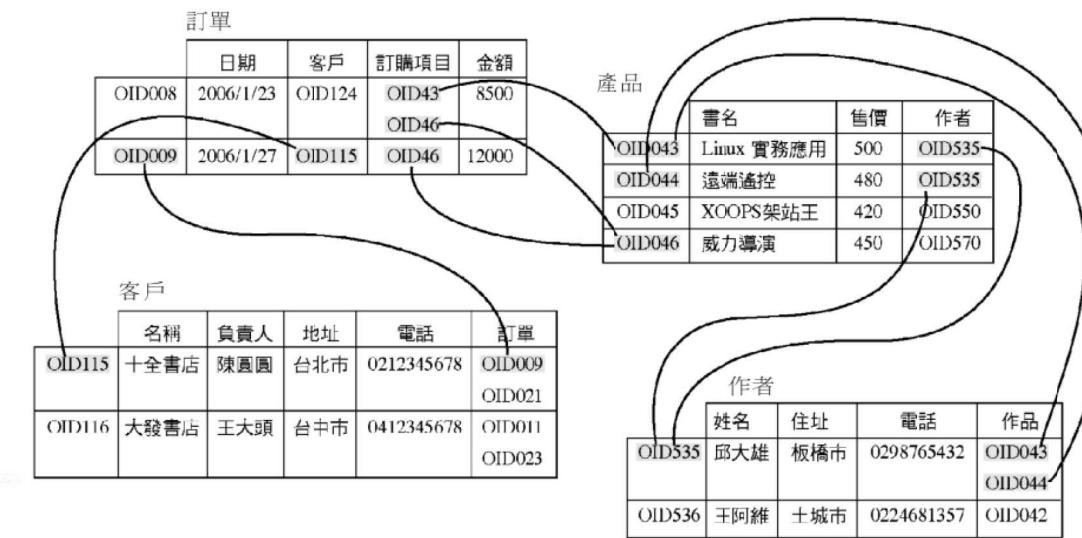
客戶資料表

經由客戶編號欄的關聯，可知道
訂單序號 2 的客戶為好看書店



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 。
- ◆ 底下是一個物件導向式資料庫的結構示意圖：

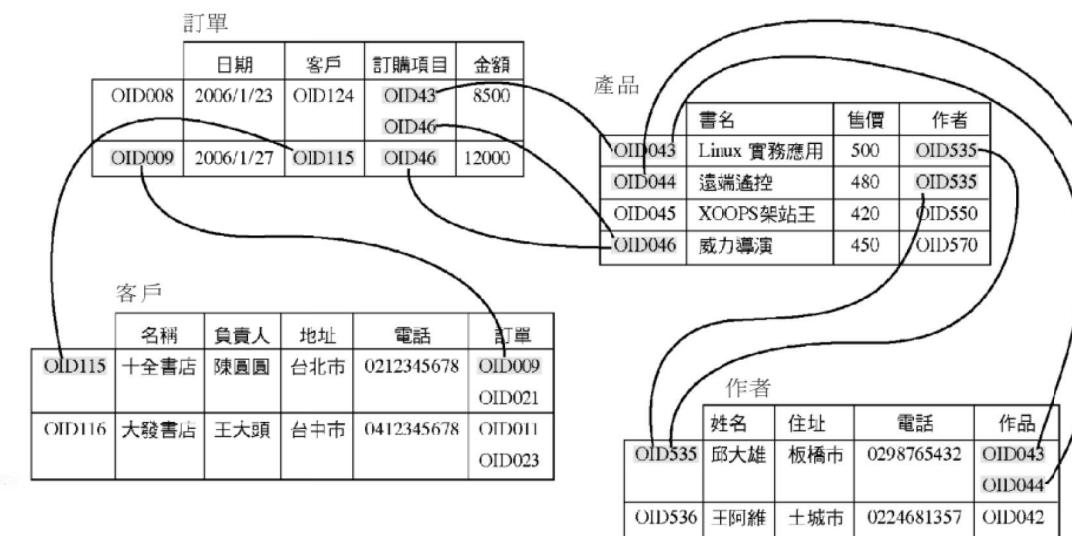




物件導向式資料庫 (Object-Oriented Database)

- ◆ 底下是一個物件導向式資料庫的結構示意圖：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。

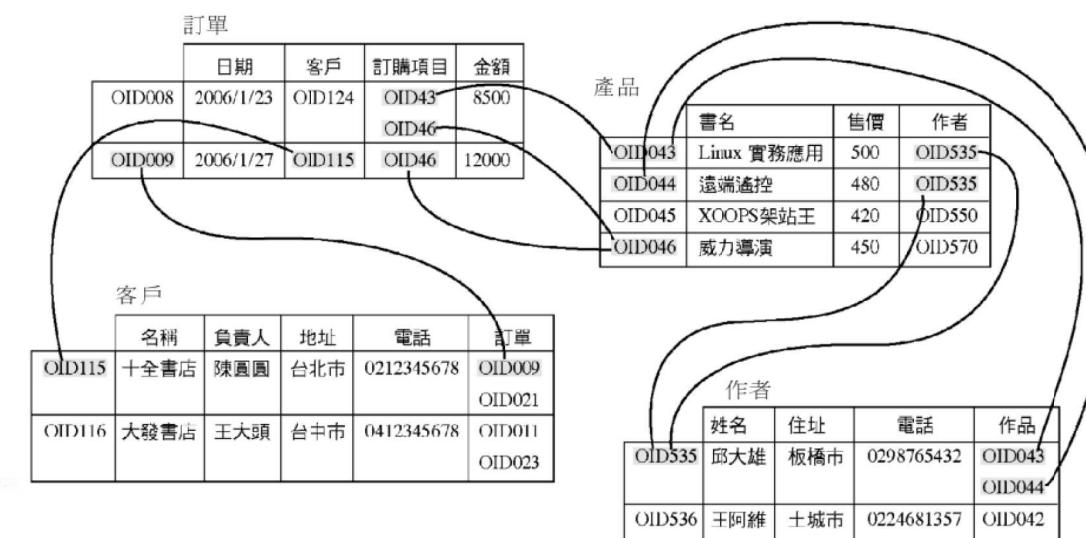




物件導向式資料庫 (Object-Oriented Database)

- ◆ 底下是一個物件導向式資料庫的結構示意圖：有幾個重點，說明如下：
- ◆ 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。
- ◆ 我們先看下面這張關聯式資料庫的資料表：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。

訂單序號	日期	客戶編號	是否付款
1	2005/7/1	6	1
2	2005/7/1	3	1
3	2005/7/3	2	0

訂單資料表

客戶編號	客戶名稱	聯絡人	性別	地址
1	十全書店	陳圓圓	女	台北市
2	大發書店	陳季暄	女	台北市
3	好看書店	趙飛燕	女	台中市

客戶資料表

經由客戶編號欄的關聯，可知道
訂單序號 2 的客戶為好看書店



物件導向式資料庫 (Object-Oriented Database)

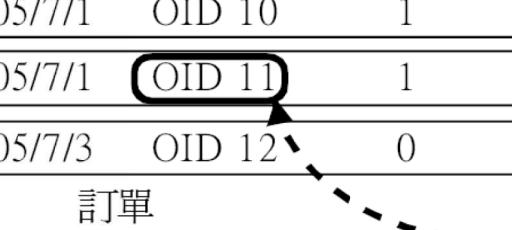
◆ 接著看下面的物件導向式資料庫：兩個物件是透過 OID 來連繫起來的。簡單地說，在關聯式資料庫中資料表間的關係必須靠設計者自行建立來產生關聯，而物件導向式資料庫中，各物件之間的關係則是在物件建立之時，便會自行連繫起來。

日期	客戶	是否付款
OID 1	2005/7/1	OID 10
OID 2	2005/7/1	OID 11
OID 3	2005/7/3	OID 12

訂單

客戶名稱	聯絡人	性別	地址
OID 9	十全書店	陳圓圓	女 台北市
OID 10	大發書店	陳季暄	女 台北市
OID 11	好看書店	趙飛燕	女 台中市

客戶





關聯式資料庫的內部結構

- ◆ 關聯式資料庫是由資料表 (Table) 所組成，其最大的特色便是將資料分類儲存在資料表中。
- ◆ 如下面的客戶資料表專門用來存放客戶方面的資料。其中第一列的項目，如客戶編號、客戶名稱、聯絡人...，是客戶資料中所具備的各項屬性 (attribute)，資料庫的用語稱為欄位 (Field 或 Column)；從第二列起則存放各欄位實際的值，例如十全書店便是客戶編號 1 的客戶名稱。

客戶編號	客戶名稱	聯絡人	地 址	電 話
1	十全書店	陳圓圓	台北市仁愛路二段 56 號	02-23219845
2	大發書店	陳季暄	台北市敦化南路一段 1 號	02-23334444
3	好看書店	趙飛燕	台北市忠孝東路四段 4 號	02-25984333
4	英雄書店	孟庭亭	台北市南京東路三段 3 號	02-27225652
5	娛人書店	劉金城	台北市北平東路 24 號	02-25786666
6	新新書店	黎國明	台北市中山北路六段 88 號	02-25557444

客戶資料表

還是一筆紀錄



關聯式資料庫的內部結構

- ◆ 將同一列各欄位的實際值集合起來，就稱為一筆記錄 (Record 或 Row)：
- ◆ 資料表的內部結構，說穿了就是欄位和記錄。在設計關聯式資料庫時，最重要的工作就是妥善規劃資料的配置，以避免產生資料重複儲存、資料不一致或資料表間的關聯不完整 ... 等等問題。

客戶編號	客戶名稱	聯絡人	地 址	電 話
1	十全書店	陳圓圓	台北市仁愛路二段 56 號	02-23219845
2	大發書店	陳季暄	台北市敦化南路一段 1 號	02-23334444
3	好看書店	趙飛燕	台北市忠孝東路四段 4 號	02-25984333
4	英雄書店	孟庭亭	台北市南京東路三段 3 號	02-27225652
5	娛人書店	劉金城	台北市北平東路 24 號	02-25786666
6	新新書店	黎國明	台北市中山北路六段 88 號	02-25557444

客戶資料表

還是一筆紀錄



資料庫系統的網路架構

- "網路架構" 要談的是資料庫系統要如何佈署的問題。通常，我們會依組織的規模、資料量的多寡、使用的人數、軟 / 硬體設備等條件來考量，常見的有下列 4 種網路架構：
 - 單機架構
 - 大型主機 / 終端機架構
 - 主從式架構
 - 分散式架構



單機架構

- ◆ 單機架構是由同一部電腦包辦所有資料庫系統的工作，包括保存資料、處理資料、管理及使用資料庫系統 ... 等等。適合使用者少、資料量也不多的資料庫系統使用，如小公司或個人使用者所建立的資料庫系統。
- ◆ 通常，用 Access、FoxPro 所設計的資料庫系統多採用這種架構。



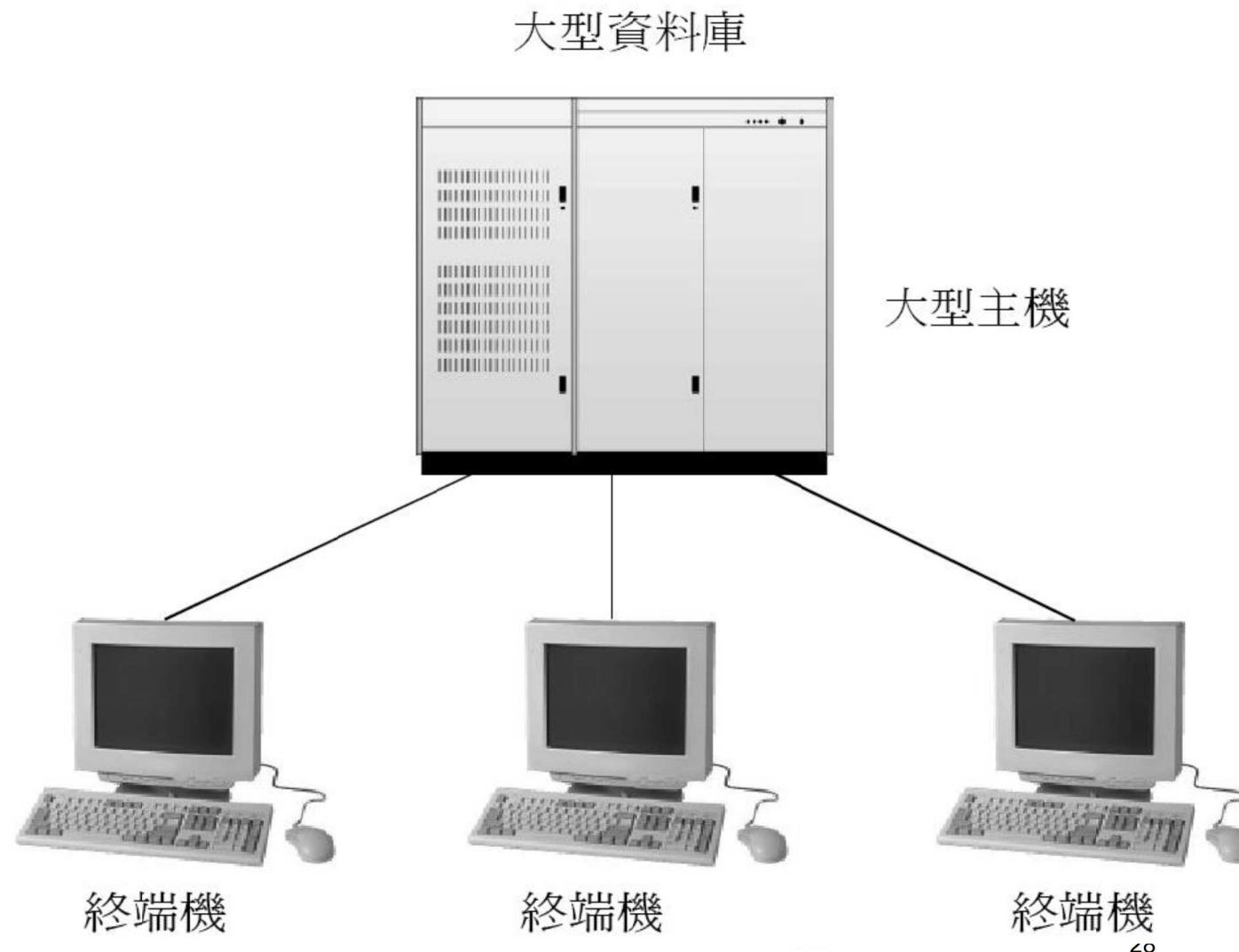
大型主機 / 終端機架構

- ◆ 大型主機 / 終端機架構是由一部大型主機負責儲存及處理龐大的資料，使用者則透過終端機與大型主機連線，以存取資料庫的內容。
- ◆ 這種架構的缺點在於，當多人同時使用時，由於所有的工作都要由大型主機來處理，因此會非常忙碌，易造成回應緩慢的問題。
- ◆ 目前除了一些大型機構外，已比較少使用這一類的架構了，而且此類的大型主機價格都相當昂貴，一般中小企業可能負擔不起。





大型主機 / 終端機架構



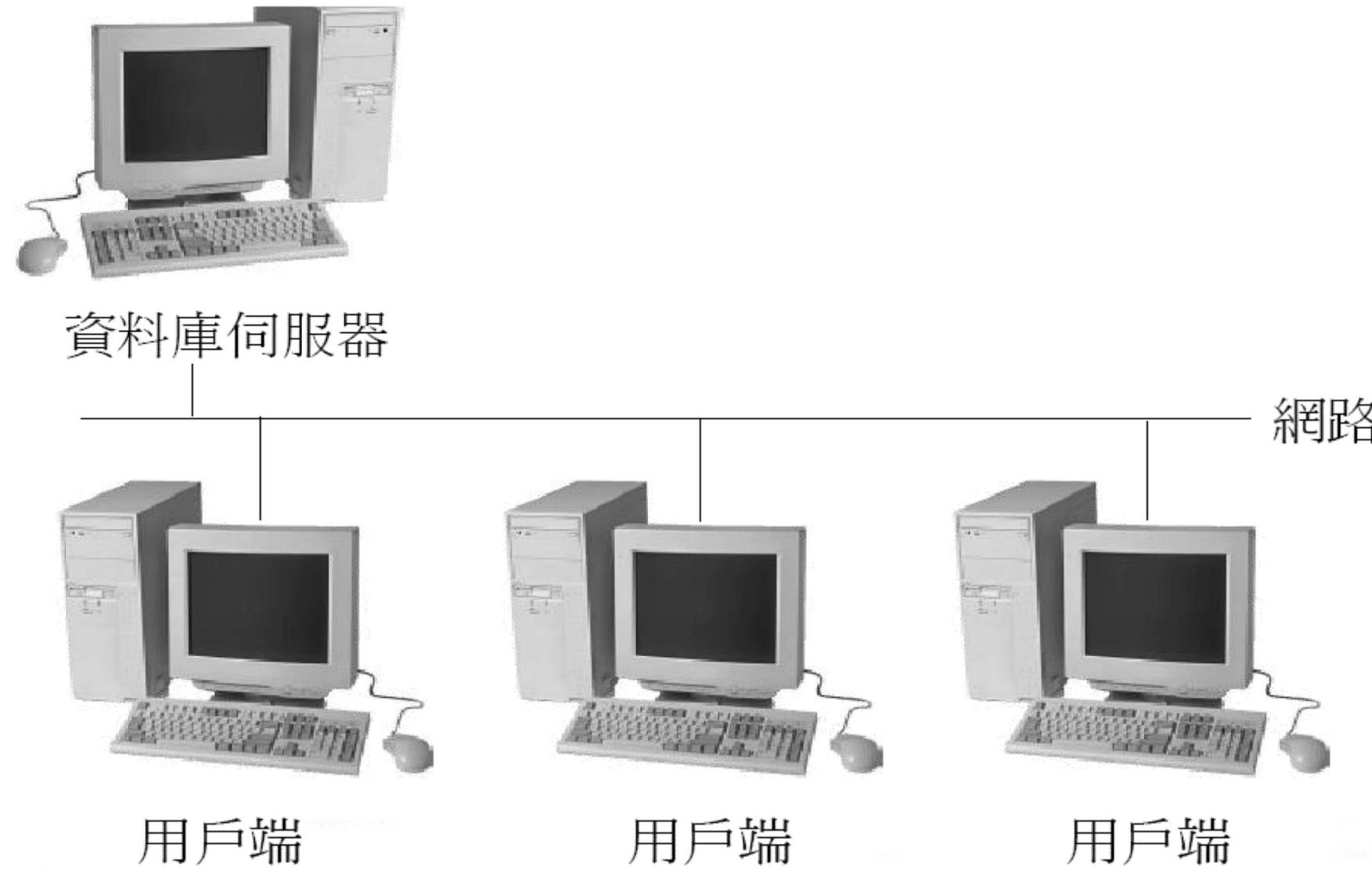


主從式架構

- ◆ 由於個人電腦的價格低廉，運算速度也不錯，利用網路互相連接之後，作為用戶端 (Client) 的各台電腦只要連結到做為資料庫伺服器端 (Server) 的電腦，就可以存取資料庫，而且部份的工作可由用戶端電腦來處理，分散資料庫伺服器的負荷，這就是主從式架構的佈署方式，同時也是目前一般公司中最普遍採用的方式。
- ◆ 若採用主從式架構，通常還會另外撰寫用戶端程式，以提供使用者易學易用的操作介面。



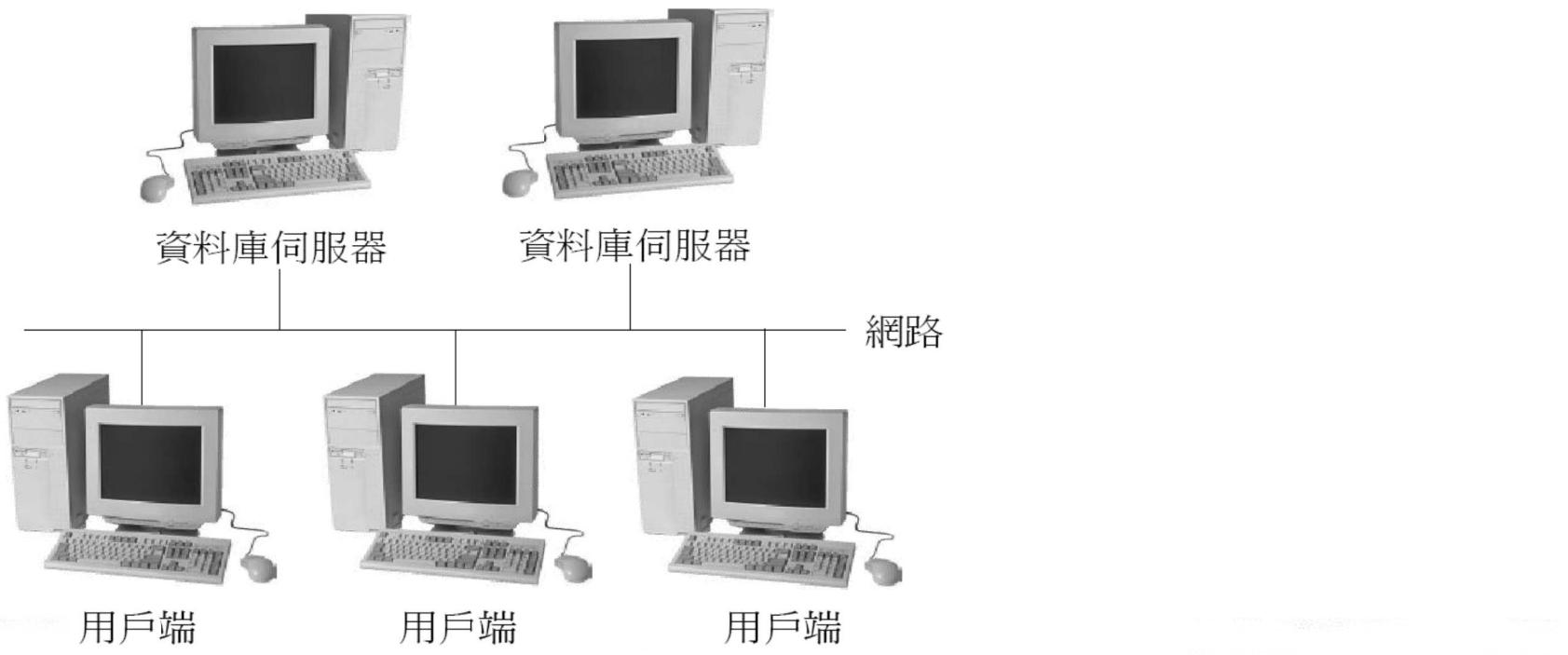
主從式架構





分散式架構

- ◆ 分散式架構是由數台資料庫伺服器所組成，使用者在存取資料時，資料可以來自於不同的伺服器中，如此在存取的效率上會比較好。
- ◆ 分散式架構的資料存取方式和主從式架構類似，只不過是多了幾台資料庫伺服器而已：





分散式架構





資料庫系統的網路架構

- ◆ 看過上述的網路架構介紹後，您可能會問 MS SQL Server 適合用在哪一種網路架構上？
- ◆ 原則上來說，MS SQL Server 是適用在主從式架構的環境，但其實除了大型主機 / 終端機架構之外，其它 3 種架構都可以。
- ◆ 我們可以視組織的規模及需要來選擇，例如在 5-10 人的小公司中，用單機架構或許就綽綽有餘了；而在規模龐大的大型企業中，則可建立多台 SQL Server (資料庫伺服器)，以建構出大容量、高效率的分散式工作平台。



資料庫管理系統的基本功能

- ◆ 資料庫管理系統其實就是管理資料庫的軟體系統，它們要負責整個資料庫的建立、資料存取、權限設定、資料備份、操作的監督與記錄 ... 等等工作。
- ◆ 底下我們就再進一步詳述資料庫管理系統 (DataBase Management System, DBMS) 所應具備的基本功能。
 - 資料定義：
 - 資料處理：
 - 資料安全：
 - 資料備份：



資料庫管理系統的基本功能

- ◆ 資料定義：DBMS 必須能夠充份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，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的存取能力，包括新增、修改、查詢、與刪除等基本功能。有時 DBMS 提供的功能雖然完善，但是並不是很適合一般的使用者操作，這時就需要程式設計師另外再撰寫用戶端的應用程式，以供一般使用者操作。
- ◆ 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，或資料庫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，還可以還原到備份資料時的狀況，以減少損失。
- ◆ 此外，維護資料庫的效率也是非常重要，尤其是在資料量很大或使用者很多的時候，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。



結構化查詢語言 SQL

- ◆ SQL (Structured Query Language, 一般習慣唸成 "sequel", 但正確的唸法應該是 "S-Q-L") 中文譯為結構化查詢語言，它是目前關聯式資料庫管理系統所使用的查詢語言，也就是說，大部份的關聯式資料庫管理系統都支援 SQL，所以使用者可以利用 SQL 語法直接對關聯式資料庫進行存取與管理的操作。
- ◆ SQL 的基本語法是由一些簡單的英文句子所構成，相當簡單易學，底下我們就來看個例子。假設要在訂單資料庫中建立一個客戶資料表，那麼可以執行以下的 SQL 敘述：建立如下型式的資料表：

```
USE 訂單資料庫
CREATE TABLE 客戶資料表
(
    客戶編號 int,
    聯絡人 char(10),
    送貨地點 varchar(50)
)
```

← 使用此資料庫
← 建立名稱為客戶資料表的資料表

指定資料表中的欄位名稱及資料型別

客戶編號	聯絡人	送貨地點



資料庫系統的使用者

◆ 最後, 讓我們來看看, 從資料庫系統的設計、建立、操作、到管理階段, 需要哪些使用者的參與, 各位並可試著從這些使用者類型中找出自己的定位：

- 資料庫設計者 (Database Designer)
- 資料庫管理者 (DataBase Administrator, DBA)
- 應用程式設計者 (Application Designer)
- 一般使用者 (End user)



資料庫設計者 (Database Designer)

- ◆ 資料庫設計者負責整個資料庫系統的設計，依據使用者的需求設計適當的格式來存放資料；同時對於整個資料庫的使用者存取權限也需要做規劃。
- ◆ 設計完成後就可交由資料庫管理者負責管理維護的工作。
- ◆ 在一般中小型企業中，資料庫的設計者與管理者有可能就是同一個人；若是大型企業，則可能設計者是一組人，而管理者又是另外一組人。



資料庫管理者 (DataBase Administrator, DBA)

- ◆ 資料庫建好之後，便可以交給資料庫管理者來負責管理及維護。
- ◆ DBA 最主要的任務，就是要維護資料庫的有效運作，並監督、記錄資料庫的操作狀況，必要時還得修改資料庫的資料結構或各項設定，以符合實際需求或提升運作效率。
- ◆ 由於資料庫中的資料對企業非常重要，而資料庫系統難免會碰到人為疏失、硬體或作業系統的問題而損壞，所以 DBA 必須設定資料庫備份的方法和時機，並且在資料庫受損時儘速讓資料庫回復原狀。
- ◆ 除此之外，DBA 也要負責資料庫的帳戶管理，決定哪些人有權利登入資料庫，哪些人有權執行哪些動作。
- ◆ 例如最基本的使用者可能只有查詢功能，需要輸入資料的使用者則具有寫入資料的功能，資料備份人員必須具有備份資料的權限 ... 等等。



應用程式設計者 (Application Designer)

- ◆ 應用程式設計者負責撰寫存取資料庫的用戶端應用程式，讓使用者用方便的操作介面來使用資料庫。
- ◆ 可用來開發應用程式的工具很多，早期的程式設計師可能用 C 或 PASCAL 等語言，現今的程式設計師則多採用 JAVA、Python、C++、.NET 或 PowerBuilder ... 等開發工具。



一般使用者 (End user)

- ◆ 一般使用者就是真正經常在存取資料庫的使用者，他們只需要學會用戶端的應用程式，不需要擔心資料庫的維護或管理方面的任可問題。若遇到問題，只要請 DBA 處理即可。

Q & A