基礎篇

Chapter 01

資料庫概念

學習目標

本章重點提示

歡迎進入資料庫的世界，這個世界像是一本有組織、有系統的電子書，它可以讓使用者透過檢索、排序、計算、查詢等方法，來有效率的管理並轉換成有用的資訊，因此，在本章節中將對資料庫概念作基本的介紹，相信各位讀者「按圖施工、保證成功」。

1.1 資料庫的意義

隨著資訊科技的進步，電腦網路帶給我們極大的便利，例如：我們要借閱某一本書，想知道該本書是否正放在某一圖書館中，並且尚未被預約借出。此時，我們只要上網就可以立即查詢到這本書的相關訊息，而這主要的**幕後工程**就是**圖書館中有一部功能強大的資料庫**。那麼，『資料庫』到底是個什麼東西？它能夠幫我們做什麼？在以下的內容中，我們就要一一來探討這些問題，期望在我們了解資料庫後，能夠進一步應用在日常生活中。

1.1.1 什麼是資料庫(Database)

簡單來說，**資料庫**就是儲存資料的地方。但比較正式的定義為：**資料庫**是**由一群相關資料的集合體**。就像是一本電子書，資料以不重覆的方式來儲存許多有用的資訊，讓使用者可以透過檢索、排序、計算、查詢等方法，來有效率的管理並轉換成有用的資訊。

例如：我們將日常生活中的**學生成績單**、親朋好友的**聯絡電話**、**各式的帳單**或公司的**人事資料表**、**顧客資料表**等等，分門別類的將這些資料加以數位化的儲存。因此存放這些資料的電腦檔案，就是所謂的資料庫了。

1.1.2 資料庫有什麼好處

將傳統資料分門別類的加以數位化，並使用資料庫來儲存，能對我們產生許多好處，我們可以歸納下面十項：

1. 無紙作業，有效利用空間(Reduce Paper)

傳統的資料儲存作業需要大量的空間來存放，像學校每年學生的**學籍資料表**、醫院每年病人的**病歷資料表**等，規模大一點的話，沒有特別蓋個檔案室來存放還真不行。若是利用資料庫來儲存，需要時只要利用電腦來觀看，如此，**每年節省的紙張與存放的空間是非常驚人的。**

2. 避免資料的重覆(Redundancy)

資料庫**最主要的精神**就是，在「**相同的資料**」情況下，**只須儲存一次**。其作法為透過**資料集中化**(Data Centralized)及**存取界面標準化**來減少資料的重覆性，有效地改善傳統檔案系統產生大量資料重覆的問題。例如：學校中的「學務處」與「教務處」都有學生的基本資料，其內容及格式如果不相同時，這種現象將會導致大量資料的重覆性。因此，在關聯式資料庫中，是**利用正規化**(Normalization)**來減少資料的重覆性問題**。

3. 達成資料的一致性(Consistency)

由於相同的資料在資料庫中是大家共用的，該項資料一旦更新，大家所得到的當然同時都是最新的資料，不會發生不一致的現象。例如：學生的姓名由「李安」改為「李碩安」時，則「學務處」與「教務處」兩處的相關姓名全部都會被修改。而資料庫系統中具有「傳播性更新(Propagating updates)」的特性，也就是說，**資料庫系統能夠確保當一筆記錄的資料更新時**（含新增、修改及刪除），**另一筆記錄的資料也會自動更新**。它是利用關聯式資料庫中的外鍵(Foreign Key; FK)來連接各種表格（關聯），來有效避免資料不一致的現象。

4. 達成資料共享(Data Sharing)

指同一份資料在同一時間可以提供給多位使用者同時來存取。它是透過資料庫系統中的SQL（Structured Query Language；結構化查詢語言）中的DML（Data Manipulation Language；資料操作語言）與同步控制(Concurrency Control)機制，來存取(Insert、Delete、Update)及共享(Query)同一份資料。

5. 制定的標準化(Standardize)

藉由**資料集中化及存取界面標準化**來達成。因此，使用者可以不知道檔案的真正位址和格式，也能使用資料庫。SQL目前已經有三代標準了，分別為第一代的SQL/1、第二代的SQL/2及第三代的SQL/3。

6. 資料的安全性(Security)

由於**資料庫內的資料**是屬於**企業組織中最重要的資產**，因此，為了防止無關人員獲知機密資料或更改資料，資料庫均有完善的保密系統，它是透過DBA（資料庫管理師）來管理與管制，使非高階級人員或非職權範圍的使用者，無法取得或修

改資料。除了人為因素之外，資料庫還必須考慮硬體故障的問題，因此，為了防止因機器故障而毀掉資料庫中的資料，資料庫隨時有最新的備份(Back-up)，以保障資料的安全性。

7. 資料的完整性(Integrity)

用以確保資料正確無誤，即利用整合性規則，來檢查使用者將錯誤及不合法的資料值存入資料庫中。例如：學生的年齡為200歲時，這顯然是一種錯誤性的資料。因此，藉由DBA的集中控制可以定義資料更新時完整性檢查。

8. 資料獨立性(Data Independent)

指資料與應用程式間無關或獨立。資料庫中每一項資料都具有「獨立性」(Independence)，也就是資料的結構和存取方法是固定的，不因用途不同而有所不同，任何程式都可取得，不專屬於某些用途。資料的獨立性是在建立資料庫時的主要目標。

9. 性能的優越性(Superiority)

資料庫的建立是為了提高資訊系統的功能，所以必須有最短的存取時間和反應時間，以最佳的形態供給最多人同時使用，以快速反應企業組織的各項需求。

10. 降低資料處理的成本(Cost)

由於制定的標準化，減少了檔案、程式和系統的維護成本。所以建立資料庫比起建立傳統的檔案，只需較低的資料處理成本(Minimum Cost)。

1.2 資料庫與資料庫管理系統

Databases in Theory and Practice

在上一節中，我們已經了解『資料庫』的意義與好處，接下來，那資料庫與資料庫管理系統有何不同呢？

資料庫是個儲存資料的地方，但是如果資料只是儲存到電腦的檔案中，其效用並不大，因此，我們還需要一套能夠讓我們很方便地管理這些資料庫檔案的軟體，這軟體就是所謂的『資料庫管理系統』。一個資料庫管理系統可以同時管理

數個資料庫。因此，資料庫加上資料庫管理系統，就是一個完整的『資料庫系統』了。所以，一個資料庫系統(Database System)可分為資料庫(Database)與資料庫管理系統(Database Management System, DBMS)兩個部份。如圖1-1所示：

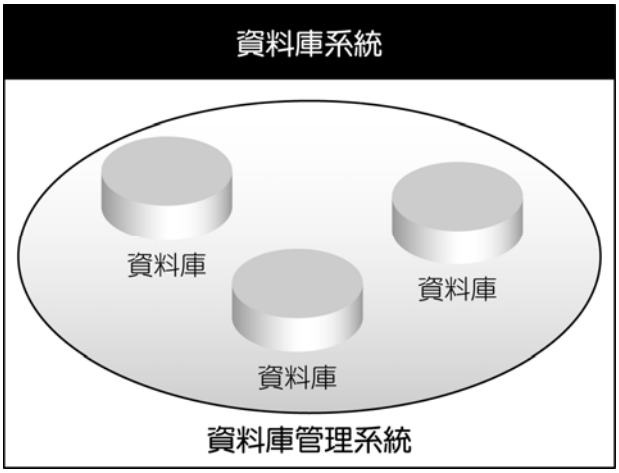

圖1-1　資料庫、資料庫管理系統及資料庫系統關係圖

說明：1. 資料庫(DB)：是指一群相關資料的集合體。
2. 資料庫管理系統(DBMS)：管理資料庫的軟體。
3. 資料庫系統：DB+DBMS。

1.2.1　資料庫管理系統的功能

它是用來管理資料庫的軟體，以作為使用者與資料庫之間的界面。如圖1-2所示。讓使用者可以透過資料庫管理系統來資料新增(Insert)、資料更新(Update)、資料刪除(Delete)、資料查看(View)及資料列印(Print)等功能。

一個資料庫系統主要組成包括：資料、硬體、軟體及使用者。

1. 資料：係由許多相關聯的檔案所整合而成。資料庫內的個別資料，可由多人共同使用，或是讓不同使用者在同一時間使用同一筆資料。
2. 硬體：即磁碟、硬碟等輔助儲存設備，或一切的週邊設備。
3. 軟體：資料庫管理系統(Data Base Management System, DBMS)。
4. 使用者：一般使用者、程式設計師及資料庫管理師(DBA)。

其中資料庫管理師(DBA)的主要職責如下：

(1) 定義資料庫的內容、架構及存取方法。

(2) 協助使用者使用資料庫，並授權不同使用者存取資料。

(3) 維護資料安全及資料完整性。

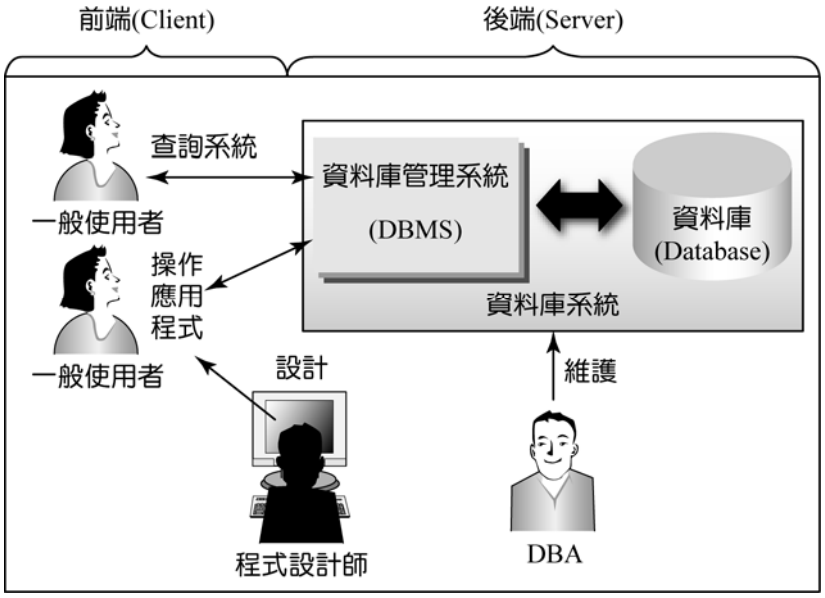

圖1-2　使用者與資料庫關係圖

(4) 資料庫備份(Backup)、回復(Recovery)及並行控制(Concurrency control)作業處理。

(5) 提高資料庫執行效率，並滿足使用者資訊需求。

在如圖1-2中，一般使用者在前端(Client)的介面中，操作應用程式及查詢系統，必須要透過DBMS才能存取資料庫中的資料。而要如何才能管理後端(Server)之DBMS與DB的資料存取及安全性，則必須要有DBA（資料庫管理師）來維護之。在圖1-2中DBMS是一套程式，用以溝通使用者程式與資料庫之間關係的界面，其主要的功能如下：

1. 資料的定義(Data Define)

儲存在資料庫中的資料類型有很多種，例如文字、數字或日期等資料型態，資料庫管理系統必須要能讓DBA建立資料庫格式及儲存這些資料格式的能力。

2. 資料的操作(Data Manipulation)

資料儲存在資料庫之後，必須能夠讓使用者方便的存取，包括新增、刪除、修改、查詢等基本功能。

3. 重複性的控制(Redundancy Control)

指可以避免資料的重複性，並且達成資料的一致性及節省儲存空間。

4. 限制未授權的存取(Constraint Access)

在考慮資料安全性的情況之下，必須要提供不同使用者有不同的存取權限，並且能夠防止資料庫受到外來或使用者無意間的破壞，設定權限以讓合法的使用者操作。

5. 程式物件與資料結構的永久儲存(Data Durability)

DBMS必須要提供「程式物件與資料結構」永久儲存於資料庫的功能。換言之，當交易中所有的寫入動作在已確認(Committed)之後，會永久的存入資料庫中。

6. 提供多重使用者介面(Multi-User Interface)

不同的使用者有不同的使用需求，所以DBMS應該提供多樣的使用者介面。

7. 表示資料之間的複雜關係(Multi-Relationship)

DBMS必須要有能力來表示資料之間的複雜關係，才可以輕易且有效率的擷取與更新相關資料。例如：一對一、一對多及多對多的關係。

8. 實施完整性限制(Integrity Constraint)

用來規範關聯中資料的儲存、更新、刪除及插入等運算，以避免使用者將錯誤或不合法的資料值存入資料庫中。

9. 並行控制(Concurrency)

針對多位使用者同時存取資料庫時進行必要的控制。

10. 提供「備份」與「回復」的能力(Backup and Restore)

讓使用者能方便的備份或轉移資料庫內的資料，以防在系統毀損時，還能將資料還原回去，減少損失。

1.2.2 常見的資料庫管理系統

目前市面上常見的資料庫管理系統，大部份都是以關聯式資料庫管理系統為主，例如Microsoft的Access和SQL Server及Oracle、Informix、MySQL等等，功能強大，皆有自己一套處理資料的方式。目前一般使用者常用的Microsoft Office中就含有Access這套資料庫管理軟體，適合初學者學習資料庫使用。

1. 常見的商業資料庫系統：
   - SQL Server
   - Access
   - DB2
   - Oracle
   - Sybase
   - Informix

2. 常見的免費資料庫系統為：
   - MySQL
   - MySQL MaxDB
   - PostgreSQL

重要觀念：DBMS一般依儲存的規模大小可分為兩種
大型DBMS：SQL Servcer, Oracle, Informaix, Sybase
小型DBMS：DB2, Access, Fox Pro

1.3 資料庫與檔案系統比較

Databases in Theory and Practice

目前有兩種常見資料處理系統：一是檔案系統，另一是資料庫系統。檔案系統是以「檔案為導向」的方法，一次只能處理一個檔案，無法同時處理多個檔案，所以它適用在「不複雜」的場合使用。在複雜的環境中，若是每個應用系統都有自己的所屬檔案，那麼資料便有重覆存放、不一致的問題發生。因此，就必

須要使用資料庫系統，來避免以上的問題。現在我們就來比較兩種資料處理的方式。

1.3.1 檔案系統

在以往，電腦皆採用「檔案處理系統(File processing system)」的方法來處理資料。其處理方式是依據每一個企業組織各部門的需求來設計程式，再根據所寫的程式去設計所需要的檔案結構，而**不考慮企業組織整體的需求**。所以在此發展模式下，**每一套程式和檔案皆自成一個系統，彼此互為獨立**。例如：生產部門與行銷部門都有自己部門的程式與檔案。同時，由於各檔案處理系統彼此之間互不相關，所以各系統所使用的程式語言與檔案結構可能會不同，也增加了系統維護的困難度。而且在此發展模式下，當一有新的需求產生時，便須撰寫新的程式和建立新的檔案，所以，**往往會造成資料重覆與資料不一致的問題**，因此，**檔案系統逐漸為資料庫所取代**。

【舉例】

在以前學校中，各處室的資料檔案是單獨儲存該處室的資料，並且使用不同的程式語言來開發應用系統。如圖1-3所示：

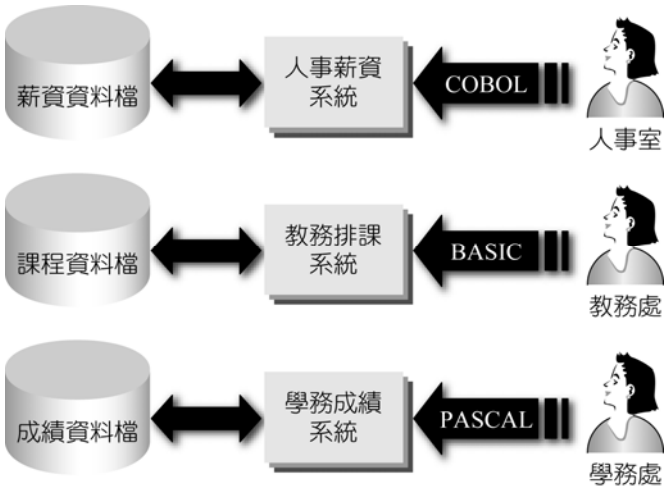

圖1-3 傳統校務檔案系統示意圖

【優點】

1. 程式的設計方式相當單純。
2. 檔案系統較容易滿足各部門或應用系統之要求。

【缺點】

1. 資料之重覆性高

各部門檔案各自獨立，例如：教務處與學務處會重覆儲存學生的基本資料。

2. 導致資料不一致性

當某一位學生的姓名更改時，必須要同時到教務處與學務處更改資料。

3. 資料無法整合及共享

當學生要查詢成績單時必須要查詢兩個處室，一次要到教務處查詢智育成績，另一次則要到學務處查詢德育成績。

4. 資料保密性和安全性非常低

在檔案系統中沒有安全機制，而資料庫系統則有。

5. 資料與程式高度相依

每一個程式有它們使用的每個檔案維護metadata。

6. 漫長的開發時間

程式設計師必須設計他們自己的檔案格式。

7. 大量的程式維護工作

佔據資訊系統預算的80%。

1.3.2 資料庫系統

由於傳統的檔案系統缺點實在太多而不容易解決，於是資料庫及資料庫管理系統乃應運而生。因此，現在我們則是採用「資料庫系統」來處理資料。對一個企業組織而言，當我們採用資料庫方式來發展一個系統時，我們會**先依據一個企業組織的整體需求做分析**考量，將所有**相關的資料**都利用一樣的資料結構**儲存於資料庫中，讓不同的使用者皆可利用資料庫的資料來發展所需的應用程式**。所以在此發展模式下，即使有新的需求產生，當所需的資料已存在原資料庫時，則使用者便可直接利用原資料庫的資料以發展所需要的程式，而**不需要另外再建立新的檔案**。

在資料庫系統中強調資料處理應該是「**集中化**」且「**整合性**」的來存取資料，並可讓來自不同處室的多位合法使用者透過電腦系統對資料做集中控制，以及對資料做「安全性」、「整合性」等管理。

舉例：現在學校中，各處室透過資料庫管理系統來加以整合。如圖1-4所示。

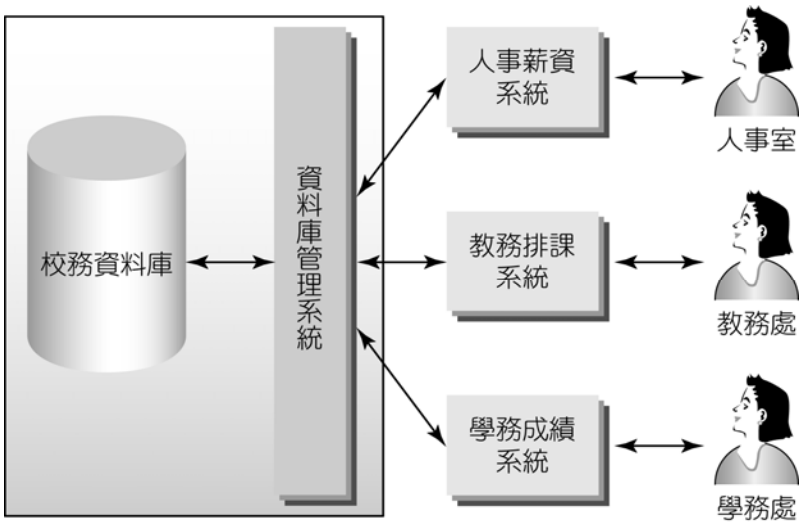

圖1-4　校務行政資料庫示意圖

【優點】

1. 避免資料之重覆
2. 達成資料一致性
3. 達成資料整合及共享
4. 加強制定之標準化
5. 完成資料之保密性和安全性
6. 資料獨立性

【缺點】

1. 初期投資成本較高，例如DBMS昂貴。
2. 資料庫管理師(DBA)人才難尋。
3. 採集中控制，在意外狀況時，容易造成重大災害。

4. 作業成本較高。

5. 為了提供安全性、同步控制、復原機制與整合性，比較花費大量資源。

1.4 資料庫的儲存資料結構

Databases in Theory and Practice

資料庫的儲存資料結構是有循序的關係，也就是由小到大的排列，其最小的單位是Bit（位元），而最大的單位則是DataBase（資料庫）。為了資料儲存與處理上的方便，並使我們能有效使用資料，因而將資料依其單位的大小與相互關係分為幾個層次，說明如下：

Bit（位元）→Byte（字元）→Field（資料欄）→Record（資料錄）→Table（資料表）→Data Base（資料庫）。如圖1-5所示：

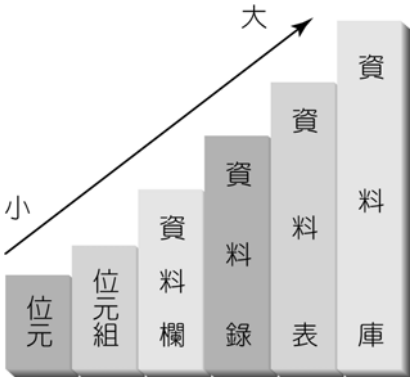

圖1-5　資料層次示意圖

1.4.1　資料庫的組成分析

由圖1-5資料層次示意圖中，我們可以了解資料庫是由數個資料表組成，而資料表是由數個記錄組成，而每一筆記錄又是由多個欄位所組成，並且每一個欄位是由字元所組成。其詳細說明如圖1-6所示：

1. 「Byte（字元）」可以說是資料庫的最小單位了。

2. 「Field（欄位）」是由許多個「字元」組成的。

3. 「Record（資料記錄）」是由好幾個「欄位」所組成。
4. 「Table（資料表）」則是由許多個「資料記錄」所組成的。
5. 「DataBase（資料庫）」是由許多個「資料表」所組成的。

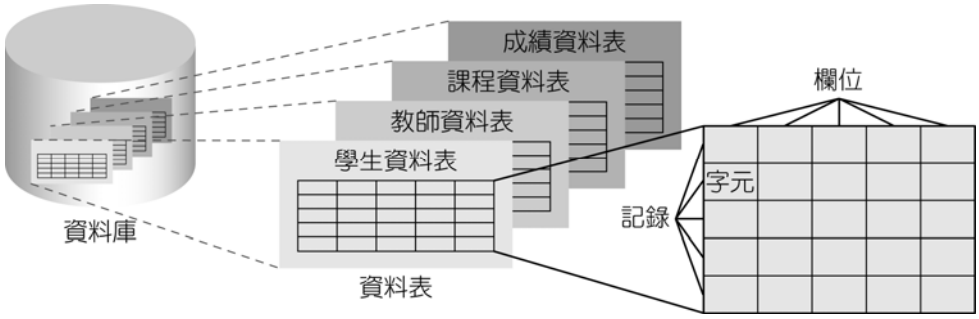

圖1-6　資料庫的組成

1.4.2　位元(Bit)及位元組(Byte)

1. 位元(Bit)是儲存資料的最小單位，即二元型態資料「0」與「1」來表示。
2. 位元組(Byte)則是用以代表一個字元(Character) 所需的位元之組合，通常為八個位元。計算機的記憶體容量均以位元組(Byte)為單位，如：主記憶容量為64MB，其中的B就是位元組(Byte)。

1.4.3　欄位(Field, Data Item)

字元很少被單獨處理，通常是將幾個字元組成一項資料來處理，這種多個字元所組成而表示某種意義的資料單元組合，稱為欄位(Field)或資料項目(Data Item)。如人事薪資資料中的員工代碼，姓名及月薪……等，都是資料項目。資料項目包含三項特性：資料名稱、資料內容和資料表示法。例如人事薪資表內的姓名是資料名稱，而在此資料項目的內容為「李碩安」。資料表示法是指資料項目的最大長度與資料型態，那麼人事薪資表內的姓名欄位的最大長度為四個中文字。

1.4.4　資料錄(Record)

資料錄是由一群相關的資料欄所組成。例如：客戶資料錄可由公司名稱、地址、電話、負責人姓名等欄位所組成。一張原始的憑證可視為一個資料錄、一張

卡片上的資料也可視為一個資料錄，印在報表上的一行字亦是一個資料錄。一般所說的資料量就是指資料錄數量。

1.4.5 資料表(Table)；檔案(File)

資料表是由相關記錄(Record)所組成。例如：學生基本資料表即為所有學生資料記錄的集合；或學生選課表為學生的選課記錄的集合。

1.4.6 資料庫(Database)

資料庫是由相關的資料表所組成。例如：學校的校務行政系統的資料庫是由數十個資料表所組成，其中包括：學籍資料表、老師資料表、課程資料表、選課資料表、成績資料表…等等。

綜合上述，如表1-1所示：

[表1-1] 資料的層級表

資料層級	層級描述	資料範例
位元(Bit)	1.數位資料最基本的組成單位 2.二進位數值	0或1
位元組(Byte)	1.由8個位元所組成 2.透過不同位元組合方式可代表數字、英文字母、符號等，又稱為字元(character) 3.一個中文字元是由兩個位元組所組成	10100100
欄位(Field)	1.由數個字元所組成 2.一個資料欄位可能由中文字元、英文字元、數字或符號字元組合而成	學號
資料錄(Record)	1.描述一個實體(Entity)相關欄位的集合 2.數個欄位組合形成一筆記錄	學籍資料
資料表(Table)	由相同格式定義之紀錄所組成	全班學籍資料
資料庫(Database)	由多個相關資料表所組成	校務行政資料庫，包括：成績資料表、學籍資料表、選課資料表…等
資料倉儲 (Data Warehouse)	1.整合性的資料儲存體 2.內含各種與主題相關的大量資料來源 3.可提供企業決策性資訊	教育部的全國校務行政資料倉儲，可進行彙整分析提供決策資訊

1.5 資料庫系統 ANSI/SPARC 架構

資料庫管理系統的主要目的是提供使用者一個有效率和方便的工作環境去操作和查詢資料。為了達到此目的，**美國國家標準協會綜合規畫委員會**(ANSI/SPARC)的資料庫管理小組在1970年訂定了一個資料庫系統的組織架構，此架構被稱之為ANSI/SPARC架構。制定此架構最主要的目的除了是**將使用者的應用程式與資料庫的實體分開之外**，同時將資料庫中一些**複雜的資料結構隱藏起來，以方便資料庫系統的使用者使用**。其ANSI/SPARC架構如圖1-7所示：

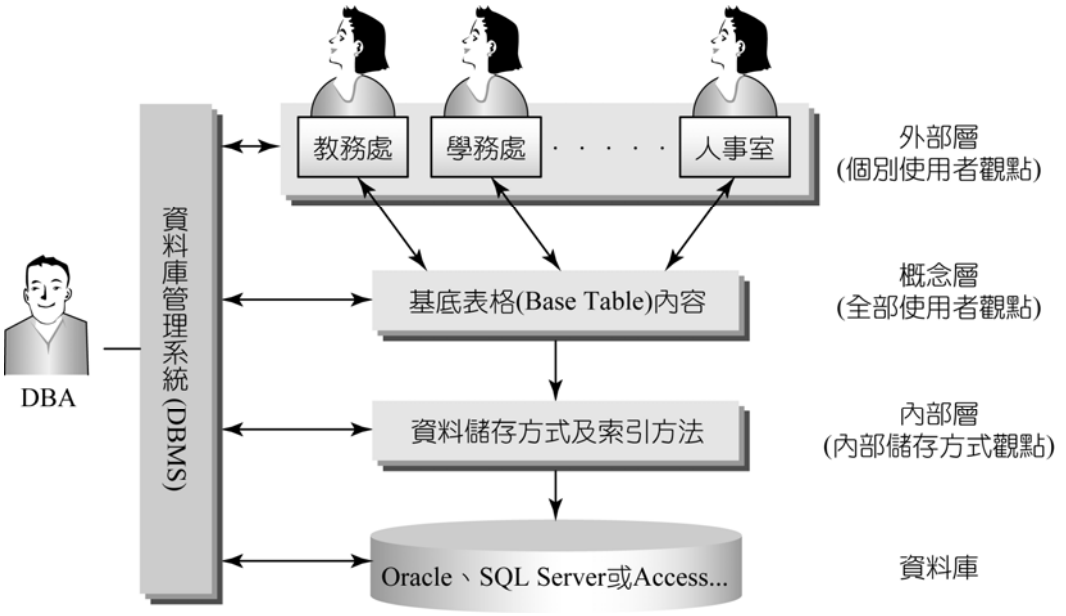

圖1-7 ANSI/SPARC架構

在圖1-7中，整個ANSI/SPARC架構可分為三大層次：

1. 外部層(External Level)
   (1) **個別使用者觀點**，即使用者只能看到概念層的一部份，其他部份通常被遮掉而不能存取。
   (2) **使用者大多以「查詢」動作為主**。

(3) 舉例：學校中的教務處與學務處所查詢的資料
教務處：學號、姓名、電話、地址、學業成績及名次。
學務處：學號、姓名、電話、地址、操性成績及曠課時數。

2. 概念層(Conceptual Level)

(1) 全部使用者觀點。

(2) 表示資料庫中全部的基底表格內容。

(3) 舉例：在基底表格內容只存一份資料表（關聯）。
學生資料表（學號、姓名、電話、地址、學業成績、名次、操性成績及曠課時數）

3. 內部層(Internal Level)

(1) 內部儲存方式觀點。

(2) 資料庫的實體架構。

(3) 舉例：每一個欄位在表格中的位置、學號索引。

在ANSI/SPARC架構中，資料實際上是儲存於資料庫中。當使用者經過其所屬的外部層要求存取資料庫中的資料時，資料庫管理系統須先將其要求經過兩層的轉換才能真正的存取資料庫中的資料，而這兩層轉換分別稱為「外部層」與「概念層」之間的對映和「概念層」與「內部層」之間的對映。但是在執行此對映時，則會浪費一些時間降低了資料庫管理系統的執行效率。但採用ANSI/SPARC架構最大的好處是它可讓資料庫管理系統具有「資料獨立性」。所謂「資料獨立性(Data Independent)」是指資料與應用程式間無關或獨立。也就是說，當使用者對使用界面有不同需求時，去修改外部層的應用程式，並不影響內部層的儲存結構。反之即為「資料相依(Data Dependent)」。

重要觀念：資料獨立性

★邏輯資料獨立(Logical data independence)

當概念層次結構改變時，不會影響外部層次。

例如：我們可以改變概念層綱要來擴充資料庫（增加一個欄位或改變資料型態），而外部層的應用程式不受影響。

★實體資料獨立(Physical data independence)

當內部層次結構改變時，不會影響外部層次。

例如：我們可以增加內部層綱要的新儲存媒體或新儲存路徑，而不會影響到概念層綱要及外部層綱要。

1.6 資料處理模式的演進

Databases in Theory and Practice

資料處理模式所使用的方式可以分成以下幾個演進階段：

1. **第一階段**—「人工作業」方式
2. **第二階段**—以電腦化「循序檔」系統方式
3. **第三階段**—以電腦化「直接檔」系統方式
4. **第四階段**—以「記錄」為處理單元的「資料庫管理系統」方式

   此種資料處理模式還可以依照「記錄結構」的不同區分為以下三種：

   (1) 階層式資料庫管理系統(Hierarchical Database Management Systems)

   (2) 網路式資料庫管理系統(Network Database Management Systems)

   (3) 關聯式資料庫管理系統(Relational Database Management Systems)
5. **第五階段**－以「物件」為處理單元的資料庫管理系統方式

   此種資料處理模式還可分成以下兩種：

   (1) 物件導向式資料庫管理系統(Object-Oriented Database Management Systems)

   (2)物件關聯式資料庫管理系統(Object-Relational Database Management Systems)

6. 第六階段－資料倉儲與資料探勘
   (1) 資料倉儲(Data warehouse)
   (2) 資料探勘(Data Mining; DM)

1.6.1 第一階段－「人工作業」方式

這是最早期的資料處理方式，主要是透過人工記錄在紙張方式。如：戶政、病歷、圖書館藏書資料……等等。

1.6.2 第二階段－以電腦化「循序檔」系統方式

因電腦的發明，所以可以透過紙帶、卡片、磁帶等媒體來記錄資料，透過電腦讀出及寫入的處理方式。例如：錄音帶及唱帶。

1.6.3 第三階段－以電腦化「直接檔」系統方式

在此階段中，磁碟漸漸地取代了磁帶，使得電腦得以直接存取檔案，成為直接存取式檔案系統(Direct Access File System)，但仍是以「檔案」為處理對象，與現今應用上常常存取的對像是以「記錄」或「欄位」來說，仍有一些處理上的差異性。例如：磁碟片與光碟片。

1.6.4 以「記錄」為主(Record-based)的資料模式(Data Model)

資料庫的種類，雖然有許多種，但以「記錄」為主的資料模式有下列三種模式：階層式資料模式(Hierarchical Data Model)、網路式資料模式(Network Data Model)及關聯式資料模式 (Relational Data Model)。

一、階層式資料模式(Hierarchical Data Model)

（一）定　義

階層式資料模式是一種「由上而下」(Top-down)的結構，而資料相互之間是一種樹狀的關係，所以又稱為樹狀結構(Tree)。它要存取資料時是從樹狀的根部(Root)開始沿著子節點逐一的去找。它適合大量資料記錄和固定查詢的應用系統。由於各平行階層之間，彼此並無關，因此要處理資料庫中的資料記錄時，通

常都要從最上一層的資料錄開始，沿著整個結構中的層次，一層層地往下走，直到找到所需要的資料為止。所有的節點都只有一個較高層次節點和其相連接，也就是父節到子節都是一對多(1：M)或一對一(1：1)的關係，且沒有一個節點可以有一個以上的父節，所以這種資料庫模式適合於**一對多(1：M)或一對一(1：1)**的資料結構應用。

（二）優　點

階層式資料庫的設計原理相當簡單容易，適用於已經**預先設定好數個固定層級的應用**，其處理的速度十分快速，適合資料量相當大的應用，如航空訂位系統、銀行的自動提款機系統等，這些系統的查詢多半是固定的方式。其優點如下：

1. 存取快速、有效率。
2. 適於**處理大量資料記錄的應用系統。**

（三）缺　點

階層式資料庫的資料間的關係已經設定，一旦要變動時就會相當麻煩，而且因為一個節點只能有一個父節，所以有時會造成結構上的重覆，或只為了要連結其他的節點而建立一些空的節點，形成浪費。其缺點如下：

1. 資料重覆儲存，浪費空間。
2. **無法表示多對多之關係（只能描述一對一及一對多的關係）。**
3. 無法適用於需要因應突然資料需求的DSS（因為資料的關係須事先設定好）。

資料庫系統中資料表與資料表之間是相互關係且呈階層式的結構。如圖1-8所示：

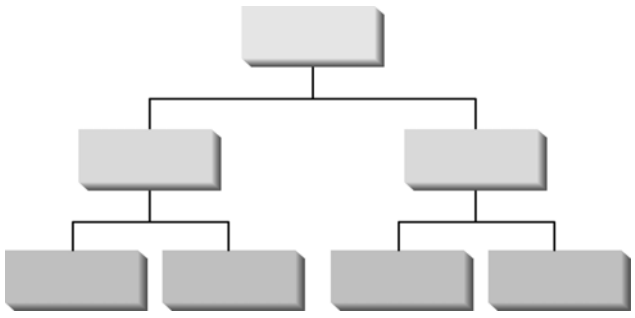

圖1-8　階層式結構示意圖

（四）實　例

例如查詢校務行政系統的資料庫，校務檔是根(Root)，而要查詢學生智育成績的資料時就必須由此點開始，沿著鏈結向下找。如圖1-9所示：

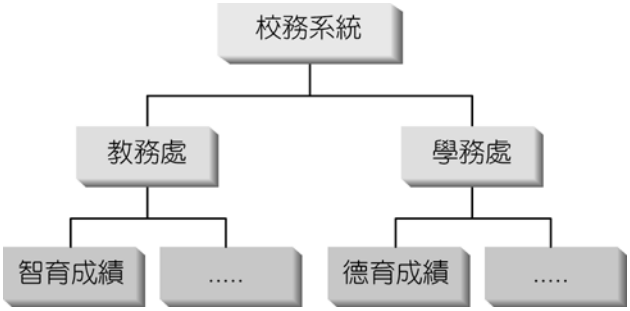

圖1-9　校務系統階層圖

二、網路式資料模式(Network Data Model)

（一）定　義

網狀式資料庫的組成結構和階層式資料庫類似，但**提供多對多(M：N)的關係**，就像一張網子一樣，一個子節點可以有多個父節點相連結，如此就**可以消除階層式模式的資料重覆問題**。因為它允許一個子節點有多個父節點，所以若要查詢資料時必先找到通往該資料的存取路徑，才能順利找到。

（二）優　點

網狀式資料庫不限定一子節點只能有一父節點的連結，所以易於描繪多種複雜的關係，且避免了大量的重覆，其優點如下：

1. **符合現實世界中的多對多關係。**
2. 存取有效率。
3. 提供實體資料獨立。