

資料庫導論

認識資料、資料庫及資訊的關係

1. **資料(Data)**：是指**未經處理**的**原始記錄**。

例如：學生考試的**原始成績**。

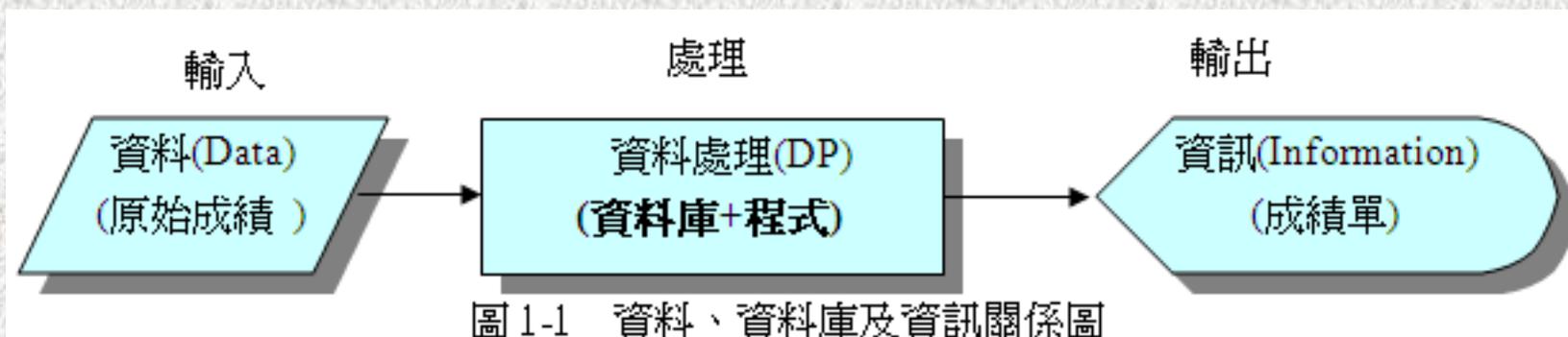
2. **資訊(Information)**：就是有經過「**資料處理**」的結果。

例如：全班同學成績之**排名及分佈圖**。

➤ **資料處理(Data Processing)**則是將「**資料**」轉換成「**資訊**」的一連串

處理過程，而這一連串的處理過程就是先輸入原始資料到「**資料庫**」

中，再透過「**程式**」來處理。例如：**成績處理系統**。如下圖所示：



資料(Data)

- (1)是客觀存在的、具體的、事實的記錄。
- (2)簡單來說，日常生活中所記錄的事實資料(姓名、生日、電話及地址)或學生在期中考的各科原始成績，這些都是未經過資料處理的資料。如表1-1所示。

表 1-1 學生的各科原始成績

科目 學生	國文	英文	數學	計概
李安安	75	55	100	90
王靜靜	66	81	73	60
李雄雄	90	55	65	80

資訊(Information)

(1)經過「資料處理」之後的結果即為資訊。而「資料」與「資訊」的特性比較。

表 1-2 資料與資訊的特性對照表

資料	資訊
潛在的資訊	有用的資料
靜態的	動態的
過去的歷史	未來的預測
由行動產生	輔助決策
儲存只是成本	運用才有效益

(2) 「資料處理」會將原始資料以加整理、計算及分析之後，變成有用的資訊(含總成績、平均及排名次)。

如表1-3 所示。

表 1-3 學生完整成績表

科目 學生	國文	英文	數學	計概	總和	平均	排名
李安安	75	55	100	90	320	80	1
王靜靜	66	81	73	60	280	70	3
李雄雄	90	55	65	100	310	77.5	2

(3) **有用的資訊**是決策者在思考某一個問題時所需用到的資料，它是**主觀**認定的。例如：**班導師(決策者)**在學生考完期中考之後，想依學生考試成績來獎勵。

表 1-3 學生完整成績表

科目 學生	國文	英文	數學	計概	總和	平均	排名
李安安	75	55	100	90	320	80	1
王靜靜	66	81	73	60	280	70	3
李雄雄	90	55	65	100	310	77.5	2

資料庫的意義

隨著資訊科技的進步，資料庫系統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，我們只要透過網路就可以立即查詢到這本書的相關訊息。而這種便利性最主要的幕後工程就是圖書館中有一部功能強大的資料庫。如圖1-1所示。

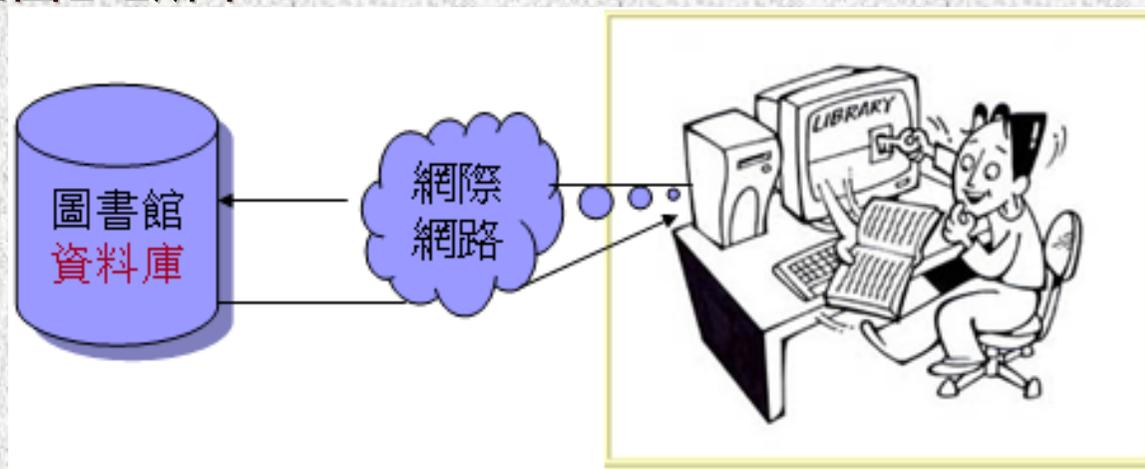


圖1-1 圖書館幕後工程---資料庫

何謂資料庫(Database)?

簡單來說，**資料庫**就是**儲存資料的地方**，這是比較不正式的定義方式。

比較正式的定義：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**方便及有效率**的管理所須要的資訊。

常見的應用如下所示：

例如1：個人通訊錄上的運用

例如2：行動通訊錄的運用

例如3：在校務行政系統的學生「成績處理系統」之運用

.....

資料庫有什麼好處

資料庫除了可以讓我們依照群組來儲存資料，以方便爾後的查詢之外，其最主要的好處非常多，我們可以歸納以下七項：

- 一、降低資料的重複性(Redundancy)
- 二、達成資料的一致性(Consistency)
- 三、達成資料的共享性(Data Sharing)
- 四、達成資料的獨立性(Data Independence)
- 五、達成資料的完整性(Integrity)
- 六、避免紙張與空間浪費(Reduce Paper)
- 七、達成資料的安全性(Security)

資料庫與資料庫管理系統

我們都知道，**資料庫**是儲存資料的地方，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，我們還需要有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**』。

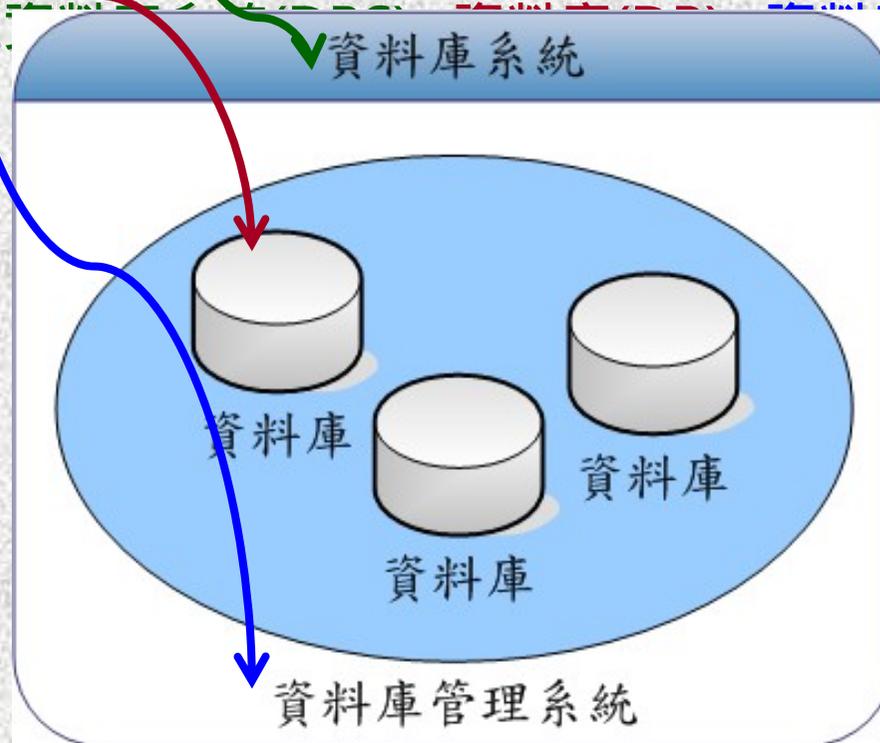
什麼是「**資料庫管理系統**」呢？其實就是一套管理「資料庫」的軟體，並且它可以**同時管理數個資料庫**。因此，**資料庫**加上**資料庫管理系統**，就是一個完整的『**資料庫系統**』了。所以，一個**資料庫系統** (Database System)可分為**資料庫**(Database)與**資料庫管理系統** (Database Management System, DBMS)兩個部份。

【重要觀念】

1. 資料庫(DB)：是由一群相關資料的集合體。

2. 資料庫管理系統(DBMS)：管理這些資料庫檔案的軟體(如：Access)。

3. 資料庫系統(資料庫管理系統(DBMS))。



資料庫系統的組成

嚴格來說，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**：即**資料庫**；它是由許多**相關聯的表格**所組合而成。

2.**硬體**：即**磁碟、硬碟等輔助儲存設備**；或稱一切的**週邊設備**。

3.**軟體**：即**資料庫管理系統(Data Base Management System, DBMS)**

(1)是指用來管理「使用者資料」的軟體

(2)作為「**使用者**」與「**資料庫**」之間的**界面**

(3)目前常見有：**Access**、MS **SQL Server**、Oracle、Sybase、
IBM DB2

4.使用者：一般使用者(End User)、程式設計師及資料庫管理師(DBA)。

(1) 一般使用者(End User)：透過「DBMS」與「資料庫」溝通的使用者

(2) 程式設計師(Programmer)：負責撰寫使用者操作介面的應用程式，

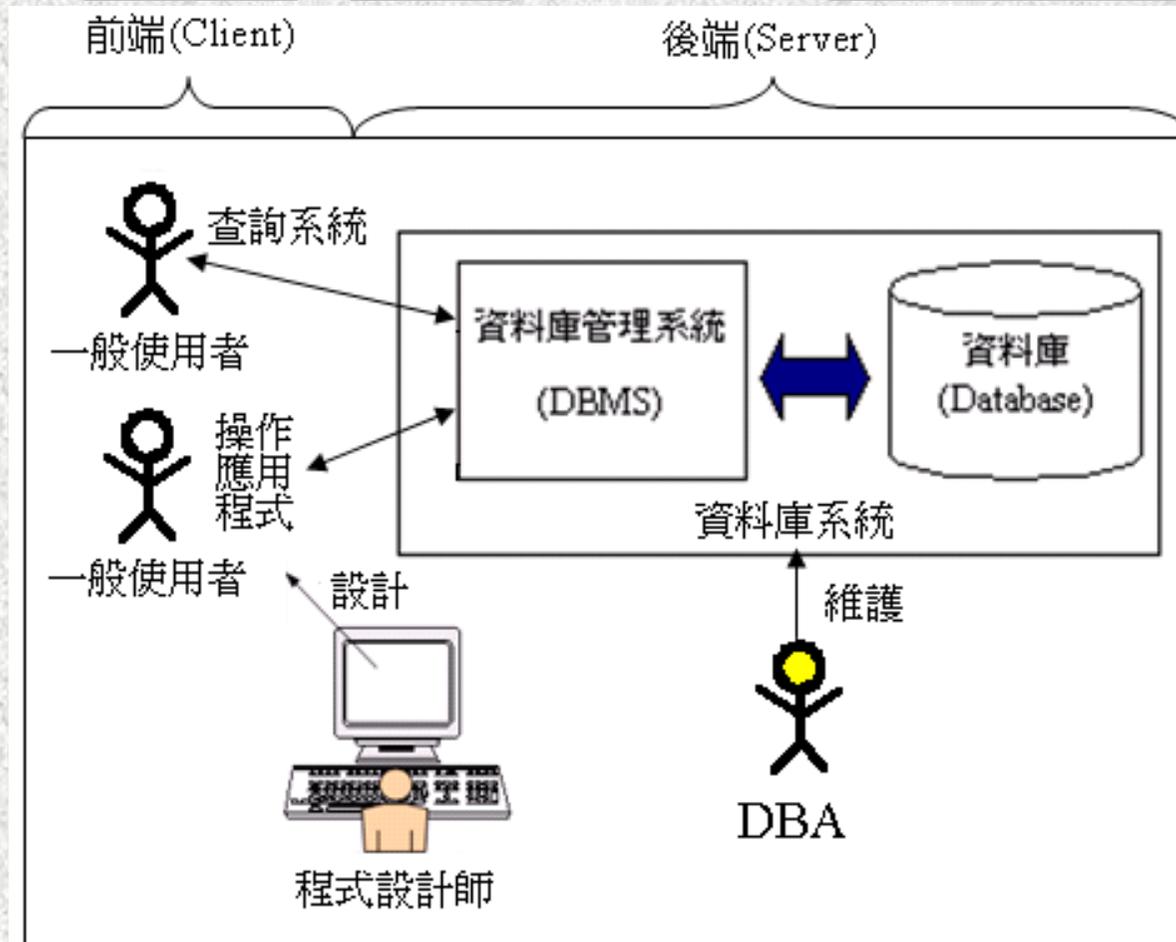
讓使用者能以較方便簡單的介面來使用資料庫

(3) 資料庫管理師(DBA)的主要職責如下：

- ① 定義資料庫的屬於結構及限制條件。
- 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。
- 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。

綜合上述，我們可以從下圖中來說明「資料庫系統」，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)之資料庫管理系統(DBMS)與資料庫(DB)的資料存取及安全性，則必須要有資料庫管理師(DBA)來維護之。



資料庫管理系統的功能

在上面的章節中，我們已經瞭解**資料庫管理系統(DBMS)**是用來**管理「資料庫」**的軟體，以作為「使用者」與「資料庫」之間溝通的界面。因此，在本單元中，將介紹DBMS是透過**那些功能**來**管理「資料庫」**呢？**其主要的功能如下：**

- 1.資料的**定義**(Data Define)
- 2.資料的**操作**(Data Manipulation)
- 3.**重複性**的控制(Redundancy Control)
- 4.表示資料之間的**複雜關係**(Multi-Relationship)
- 5.實施**完整性限制**(Integrity Constraint)
- 6.提供「**備份**」與「**回復**」的能力(Backup and Restore)

資料的定義(Data Define)

【定義】 ---它是建立資料庫的第一個步驟

是指提供DBA建立資料格式及儲存格式的能力。亦即設定資料「欄位名稱」、「資料類型」及相關的「限制條件」。其「資料類型」的種類非常多。

【例如】文字、數字或日期等等，此功能類似在「程式設計」中宣告「變數」的「資料型態」。如下圖所示：



資料的操作(Data Manipulation)

在**定義**完成**資料庫的格式**(亦即**建立資料表**)之後，接下來，就可以讓我們儲存資料，並且必須能夠讓使用者方便的**存取資料**。

【定義】是針對「資料庫執行」四項功能：

1.新增(INSERT)

2.修改(UPDATE)

3.刪除(DELETE)

4.查詢(SELECT)

【實例】

新增「學號」為**S0004**，「姓名」為**李安**同學的記錄到「學生資料表」中

```
SQL 指令
INSERT INTO 學生資料表
VALUES('S0004','李安')
```

學生資料表

	學號	姓名
#1	S0001	張三
#2	S0002	李四
#3	S0003	王五
#4	S0004	李安

重複性的控制(Redundancy Control)

【功能】主要是為了達成「資料的一致性」及「節省儲存空間」。

【作法】設定「主鍵」來控制。如下圖所示：

設定主鍵

The screenshot shows the Microsoft Access interface. The top window is titled "Microsoft Access - [學籍資料表: 資料表]". The table design view shows columns: 欄位名稱 (Field Name), 資料類型 (Data Type), 主索引鍵 (Primary Key), and 描述 (Description). The "學號" (Student ID) field is marked as the primary key. Below, the data view shows a table with columns: 學號 (Student ID), 姓名 (Name), 系碼 (Department Code), and 新增欄位 (New Field). The data rows are: S0001 (張三, D001), S0002 (李四, D001), S0003 (王五, D002), and S0004 (李安, D003). A red box highlights the "學號" column in the data view, and a yellow arrow points from this box to an error message dialog box.

Microsoft Office Access

您要求變更資料表，由於會對索引、主索引鍵或關聯產生重複的值，所以不會成功。變更欄位的資料或變更那些包含著重複資料的欄位、移除索引或重新定義索引以允許重複索引值，然後再試一次。

確定 說明(H)

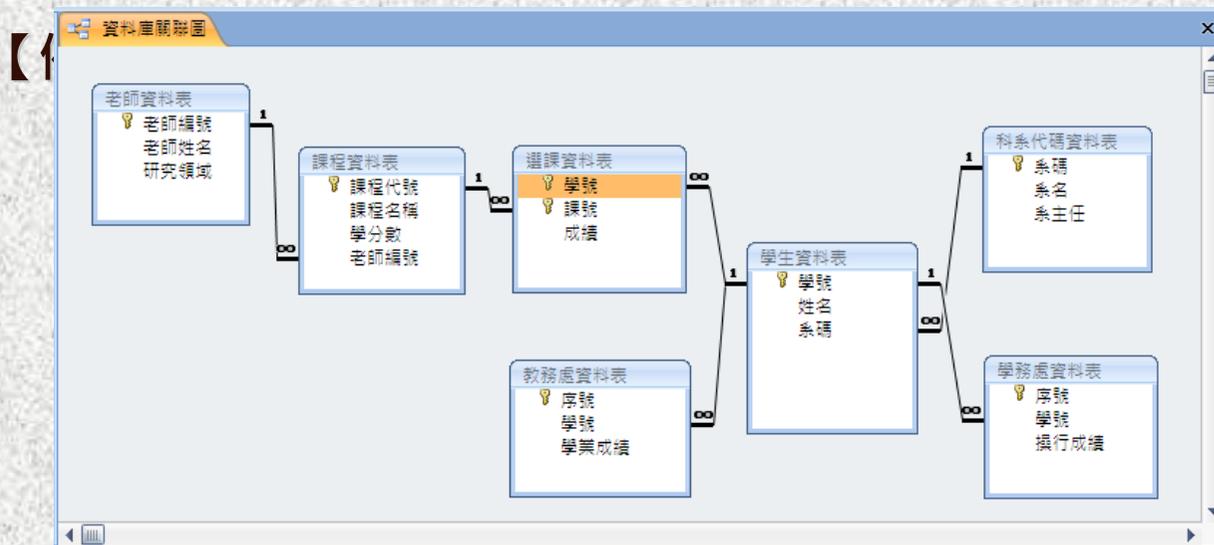
說明：如果「學號」設定為主鍵時，則如果再輸入相同的學號時，就會產生錯誤。

表示資料之間的複雜關係

【定義】是指DBMS必須要有能力來表示資料之間的複雜關係，基本上，

有四種不同的關係，分別為：

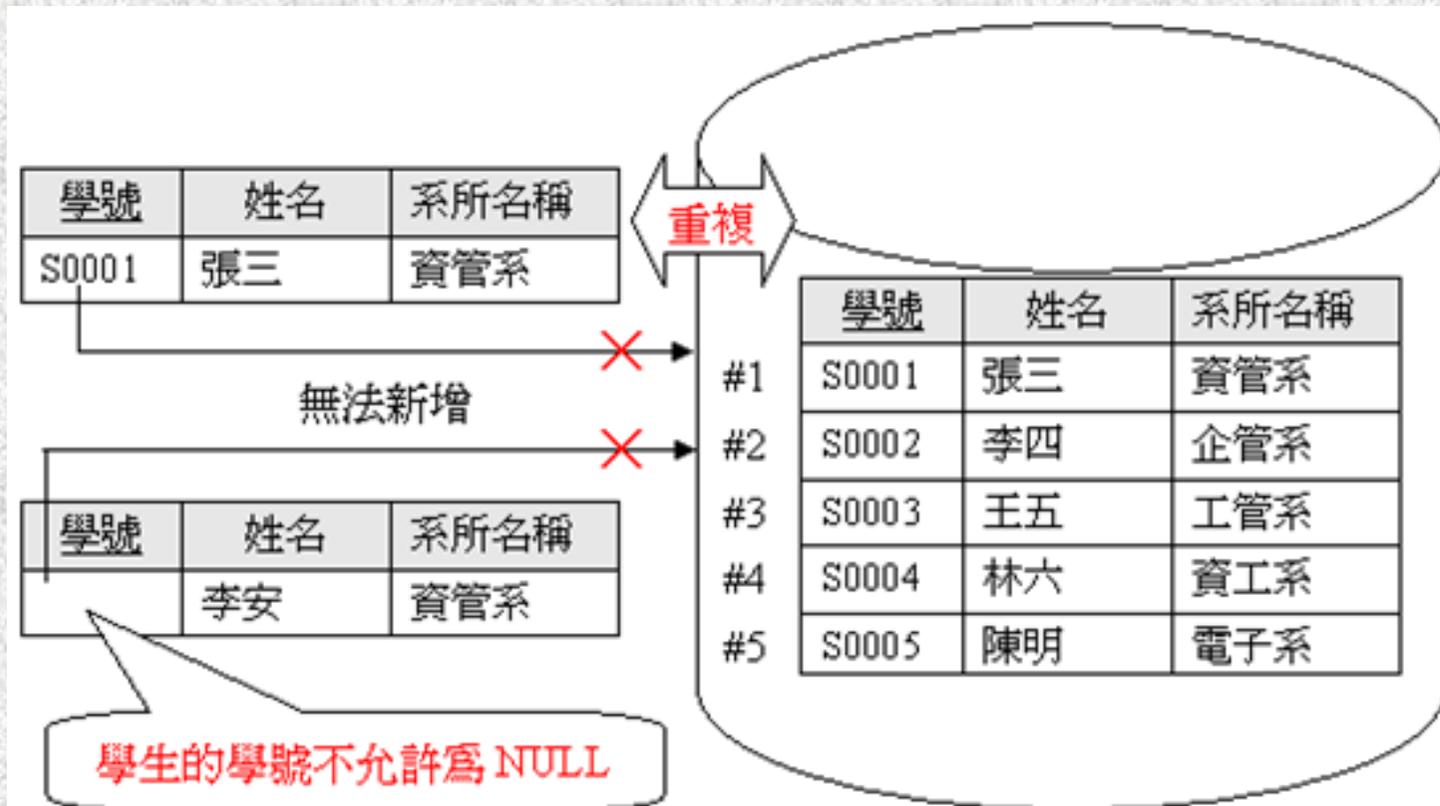
1. 一對一
2. 一對多
3. 多對一
4. 多對多



實施完整性限制

【定義】

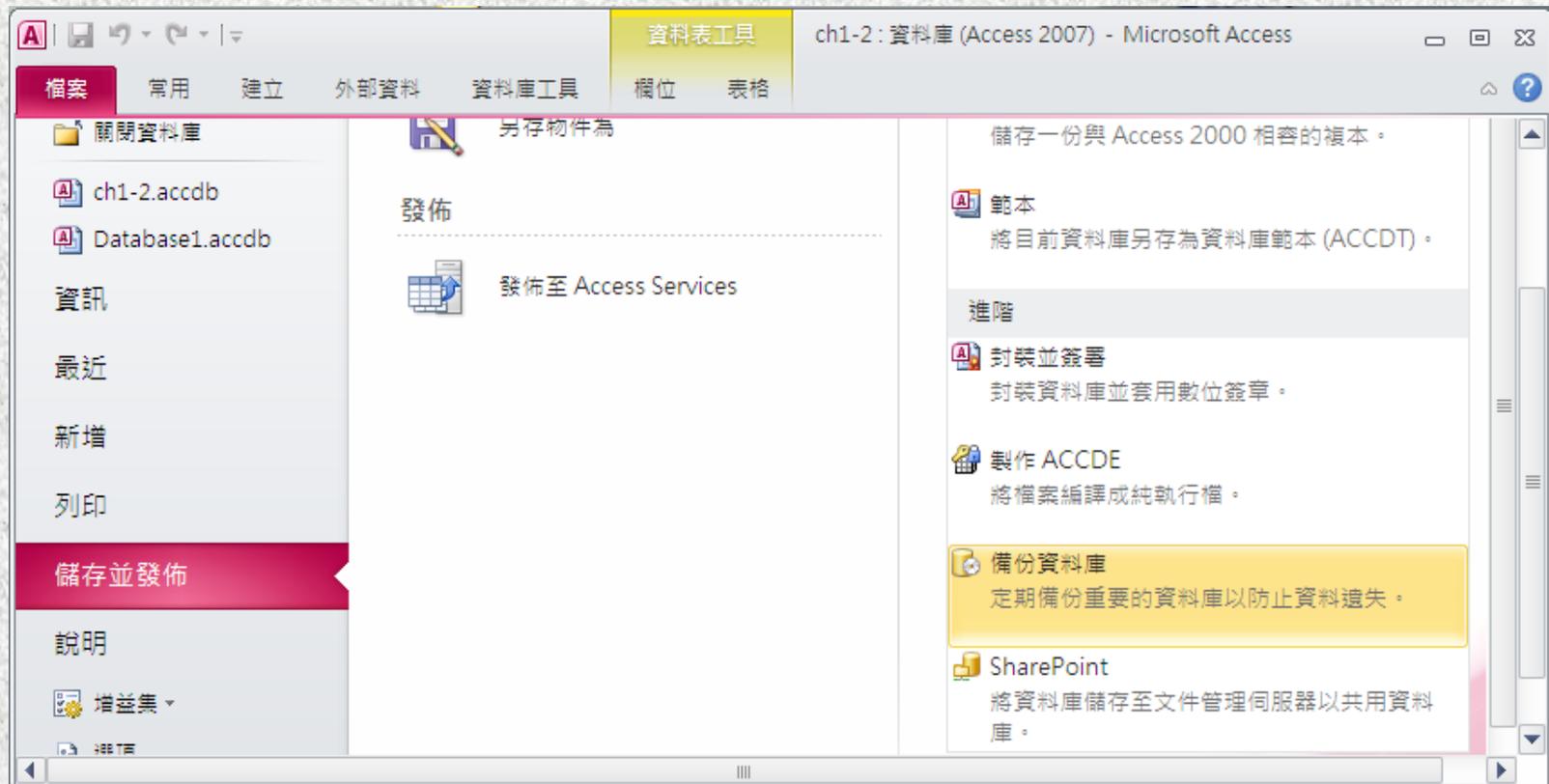
是指用來規範關聯表中的資料在經過新增、修改及刪除之後，將錯誤或不合法的資料值存入「資料庫」中。如下圖所示：



提供「備份」與「回復」的能力

【定義】

是指讓使用者能方便的「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。如下圖所示：



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(RDBMS)**為主。

一、常見的**商業資料庫系統**：

1.**SQL Server(企業使用)**：微軟公司(Microsoft)所開發。

【使用對象】企業的資訊部門。

2.**Access(個人使用)**：微軟公司(Microsoft)所開發。

【使用對象】學校的教學上及個人使用，它屬於微軟Office系列中的一員。

3.**DB2**：是由IBM公司所開發。

4.**Oracle**：是由甲骨文公司 (Oracle Corporation) 所開發。

5.**Sybase**：是由賽貝斯公司所開發。

6.**Informix**：是由Informix公司所開發。

二、常見的免費資料庫系統：

1. MySQL(最常見，常與PHP使用)
2. MySQL MaxDB
3. PostgreSQL

資料庫系統與檔案系統比較

目前有兩種常見「資料處理系統」：

第一種：檔案系統

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。

【適用時機】在「不複雜」的場合使用

【缺點】每一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。

第二種：資料庫系統→解決「檔案系統」的缺點

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，在此種發展模式下，**每一套程式**和**檔案**皆自成一個系統，因此，**同一個子系統**中，「檔案」與「程式」之間的**相依性高**，而**子系統與子系統**之間是**相依性低**(亦即**相互獨立**)。

資料庫系統

由於傳統的檔案系統缺點實在太多(上一個章節中的七個缺點)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。

以「大學校務行政電腦化系統」為例，當我們由傳統的「檔案系統」決定改採用「資料庫系統」來發展一個系統時，我們必須要依據大學校務組織的整體需求做分析考量，將大學各單位所有相關的資料以相同的「資料結構」來建置資料庫，讓不同單位的資訊系統之使用者也可以利用現有的資料庫來發展所需的應用程式。

在此種發展模式下，如果其他單位又有新的需求產生，則只需要將原先資料庫直接提供給所需要的使用者來開發新的系統，而不需要另外再建立新的資料庫。

在資料庫系統中主要強調資料的「集中化」管理，因此，可以讓來自不同處室的多位合法使用者透過「資料庫管理系統」來存取資料庫中的資料。

【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1-5所示。

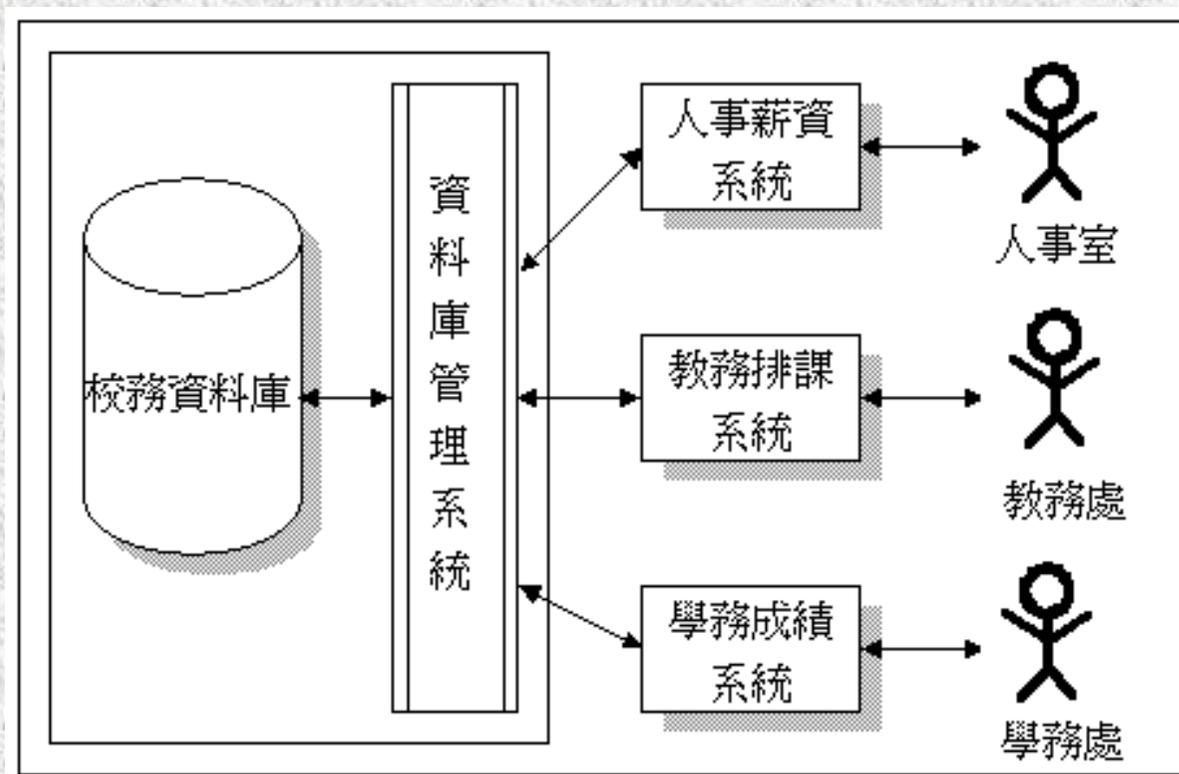


圖1-5 校務行政資料庫示意圖

【資料庫系統的優點】

- 一、降低資料的重複性(Redundancy)
- 二、達成資料的一致性(Consistency)
- 三、達成資料的共享性(Data Sharing)
- 四、達成資料的獨立性(Data Independence)
- 五、達成資料的完整性(Integrity)
- 六、避免紙張與空間浪費(Reduce Paper)
- 七、達成資料的安全性(Security)

【資料庫系統的缺點】

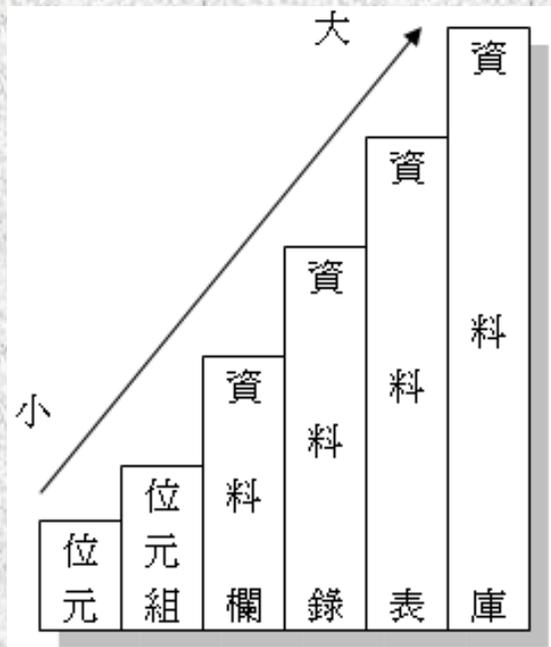
- 一、資料庫管理系統(DBMS)的成本較高
- 二、資料庫管理師(DBA)專業人員較少
- 三、當DBMS發生故障時，比較難復原(集中控制)
- 四、提供安全性、同步控制、復原機制與整合性，比較花費大量資源。

資料庫的階層

資料庫的階層是有循序的關係，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）。

資料依其單位的大小與相互關係分為幾個層次，說明如下：

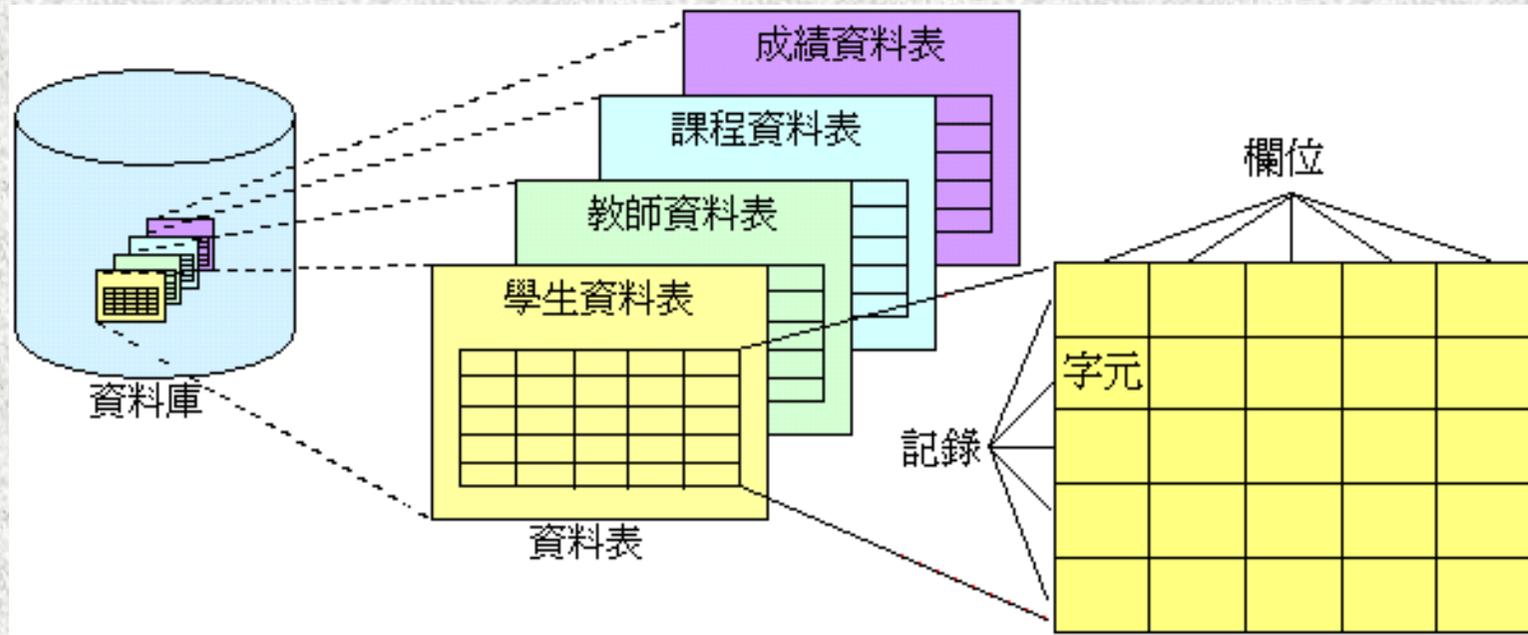
Bit（位元）→Byte（字元）→Field（資料欄）→Record（資料錄）
→Table（資料表）→Data Base（資料庫）。如下圖所示：



資料庫是由許多資料表所組成，每一個資料表則由許多筆記錄所組成，每一筆記錄又以許多欄位組合而成，每一個欄位則存放著一筆資料。

資料庫中的每一個欄位，皆只能存放一筆資料，這些資料必須遵守著一定的結構標準來記錄各種訊息。例如：文字、數字或日期等格式，而在資料表中的欄位值也可能是空值（Null）。

除了從**資料庫階層**的觀點之外，我們可以從**資料庫剖析圖**來詳細說明。
如下圖所示：



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

表1-1 資料庫階層表

資料階層	階層描述	資料範例
位元(Bit)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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)	<ol style="list-style-type: none"> 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 	個人學籍資料
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)	<ol style="list-style-type: none"> 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

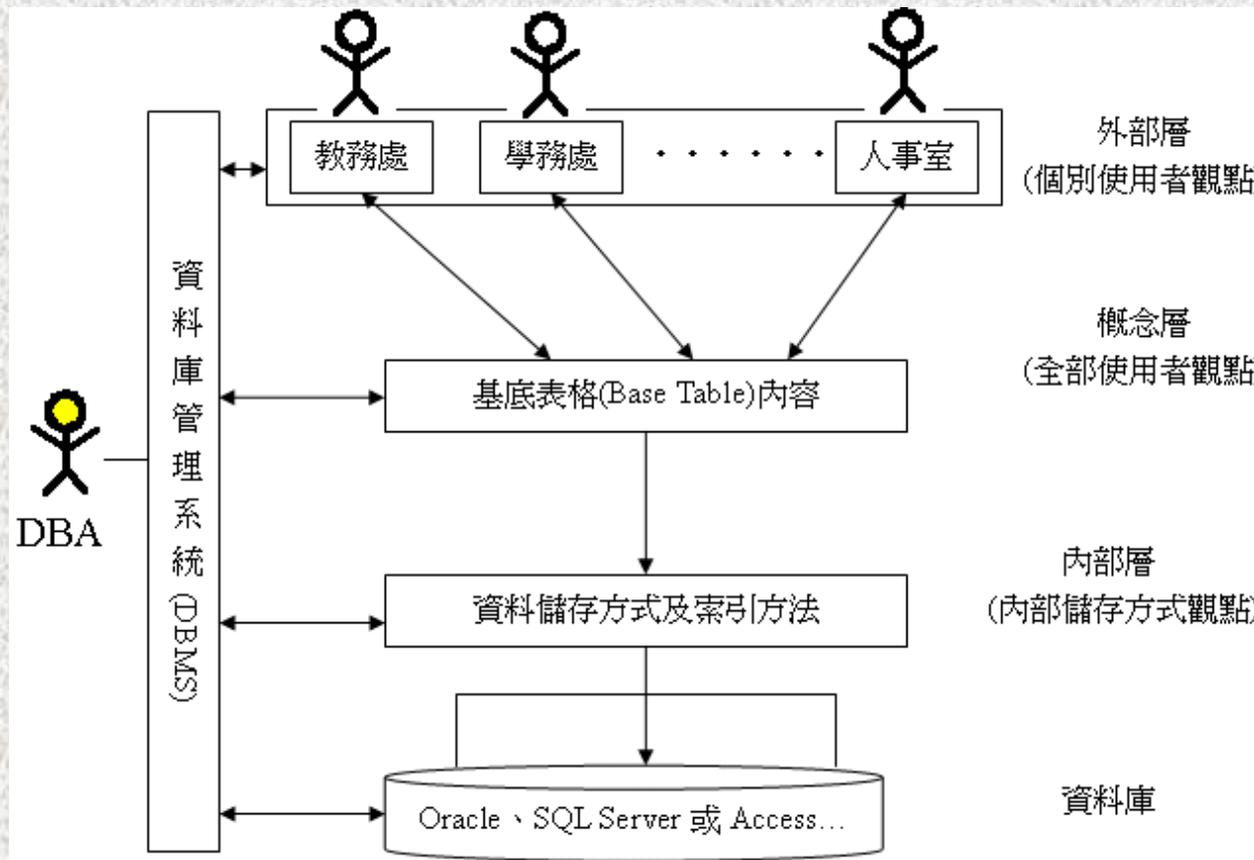
資料庫系統ANSI/SPARC架構

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。

此架構最主要的目的除了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。

ANSI/SPARC架構可分為三大層次：

- 1.外部層(External Level)
- 2.概念層(Conceptual Level)
- 3.內部層(Internal Level)



1.外部層(External Level)

(1) 個別使用者觀點，是指依不同的使用者提供不同的資料庫之資料。

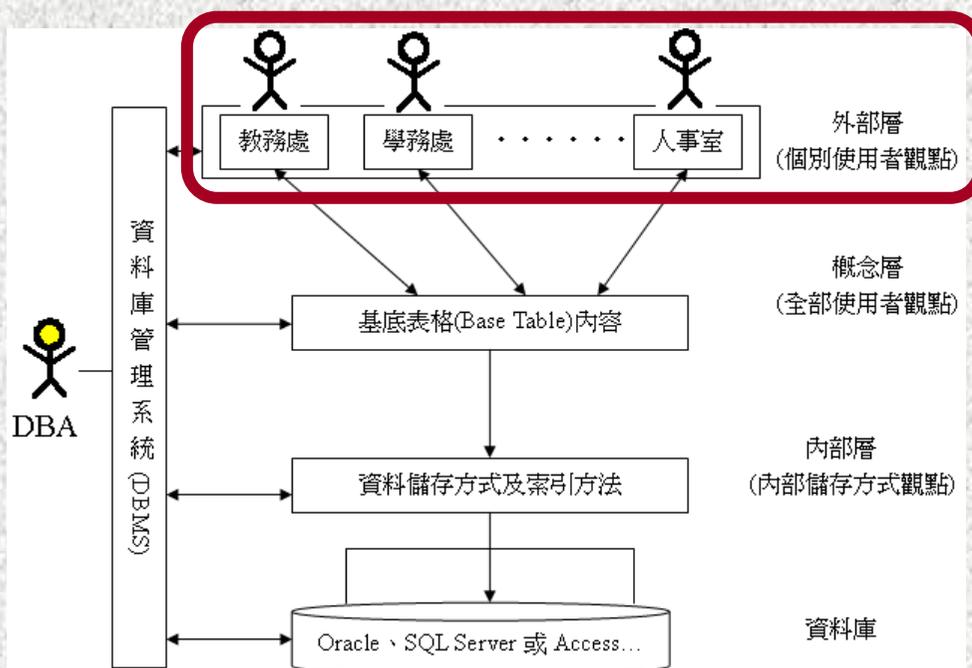
(2) 使用者大多以「查詢」動作為主。

【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」



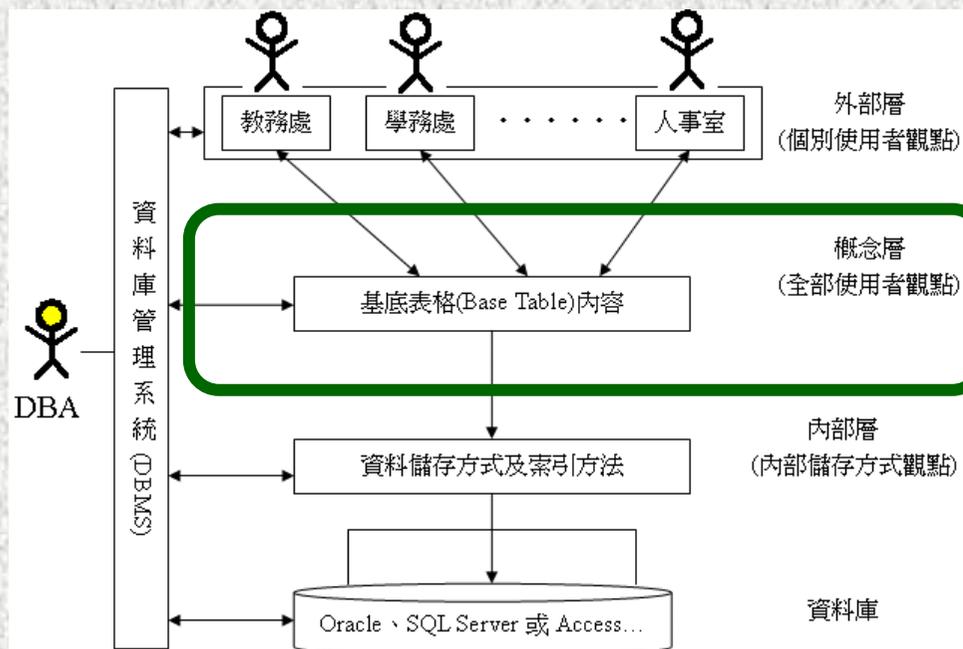
2. 概念層 (Conceptual Level)

(1) 全部使用者觀點。

(2) 表示資料庫中全部的基底表格內容。但不用考量資料實際的儲存結構。

【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)

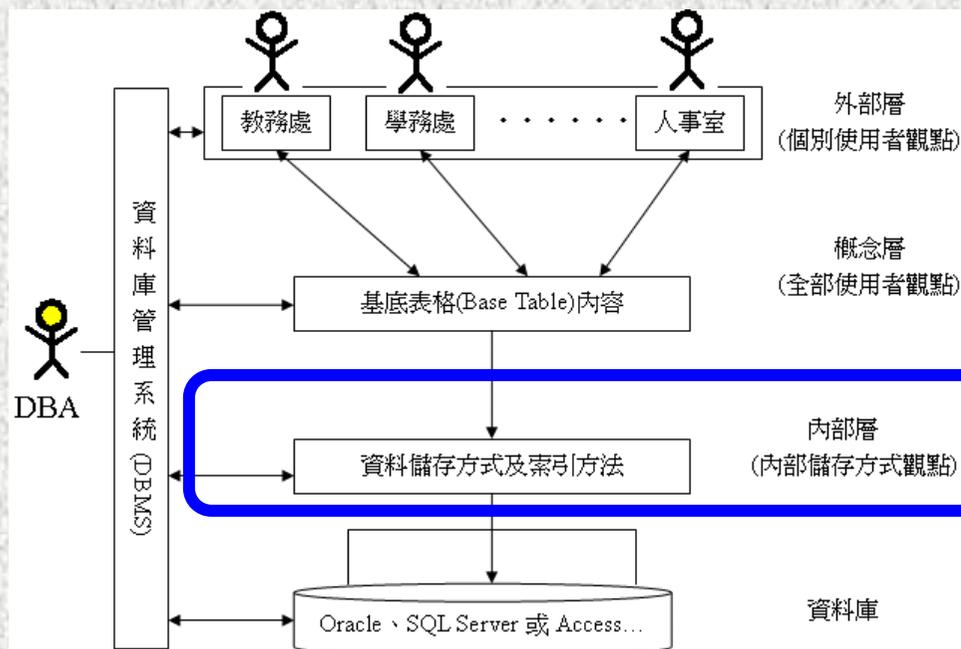


3. 內部層 (Internal Level)

(1) 內部儲存方式觀點，亦即實際儲存在磁碟等儲存裝置的資料。

(2) 資料庫的實體架構。

【舉例】每一個欄位在表格中的位置、學號索引。



以「記錄」為主的資料模式

以「記錄」為主的資料模式有下列三種模式：

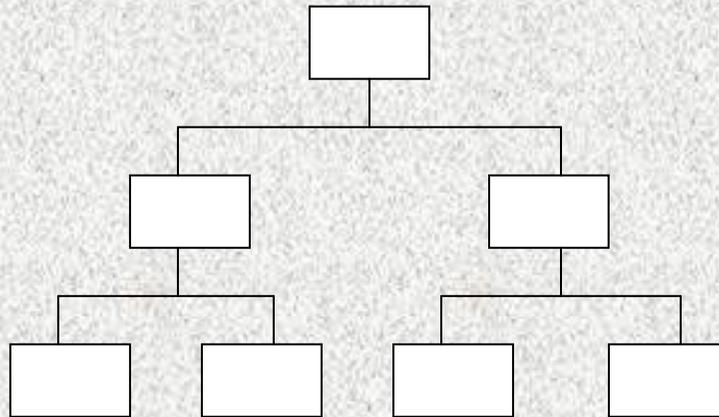
1. 階層式資料模式(Hierarchical Data Model)
2. 網路式資料模式(Network Data Model)
3. 關聯式資料模式 (Relational Data Model)

1.階層式資料模式(Hierarchical Data Model)

【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)。

如下圖所示：



【資料存取方式】是由樹根(Root)開始往下存取資料。

【適用時機】大量資料記錄和固定查詢的應用系統。

【優點】

1. 存取快速、有效率。
2. 適於處理大量資料記錄的應用系統。

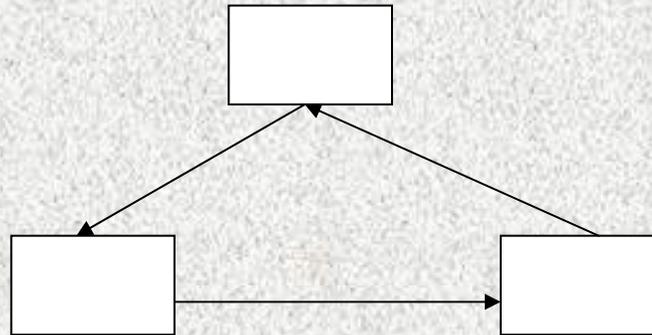
【缺點】

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
(因為資料的關係須事先設定好)

2.網路式資料模式(Network Data Model)

【定義】

網狀式資料庫的組成結構和階層式資料庫類似，其差異點是提供多對多(M:N)的關係，就像一張網子一樣，每一個子節點可以有多个父節點相連結，可以消除階層式模式的資料重覆問題。如下圖所示：



【優點】

- 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**。

【缺點】 **較為複雜**。