

## CH6 商業智慧的基礎：資料庫和資訊管理

---

### 1. 傳統檔案系統的問題？

列出並描述資料層次結構中的每個部分

定義並說明實體和屬性的意義

列出並描述傳統檔案的問題 – 資料不一致與冗餘、資料依賴性、缺乏彈性、安全性差、缺乏資料分享與可用性

### 2. 資料庫管理系統 (DBMS) 功能？關聯式 DBMS？

定義 DB 和 DBMS

簡要說明 DBMS 的功能 – DML、DCL、DDL、TCL、Data Dictionary

定義關聯式 DBMS 並說明其如何組織資料 – 集合代數

解釋資料操作語言的重要性 – select、join、project

解釋為什麼 NoSQL 有用 – 雲端運算、大數據、工作負載、新類型資料儲存需求

定義和說明正規化和參照完整性並說明為何能以此設計出良好的關聯式資料庫

定義和說明實體關係圖，並解釋其在資料庫設計中的作用

### 3. DB 如何改善企業績效和決策，工具有哪些

定義大數據(Big Data)並描述用於管理和分析的技術 – volume、variety、velocity、varacity

列出並描述現代商業智慧基礎架構元件 – 資料倉儲、資料市集、OLAP、Text Mining、Data Mining、Web Mining、Database and WEB

描述線上分析處理 (OLAP) 的功能 – 切片、切丁、上轉、下拉、旋轉

描述 Hadoop 並說明與關聯式 DBMS 的不同之處

解釋文字探勘(Text Mining)和 Web 探勘(Web Mining)與一般資料探勘(Data Mining)有何不同 – sentiment analysis、content mining、structure mining、usage mining

描述用戶如何藉由網路訪問公司內部資料庫中的資訊

### 4. 為什麼資訊政策、資料管理和資料品質保證對於管理公司的資料

#### 資源至關重要？

描述資訊政策和資料管理在資訊管理中的作用

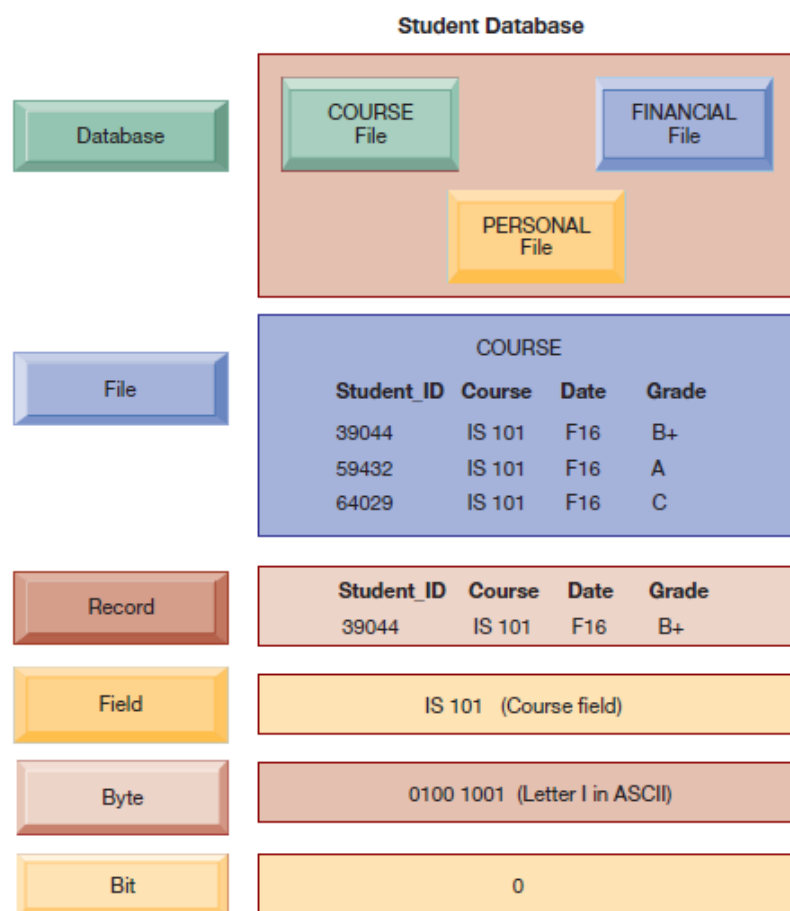
說明為什麼資料品質審查和資料清理重要

---

列出並描述資料層次結構中的每個部分

- (1) Bit(位元): 電腦可處理的最小單元
- (2) Byte(位元組): 8 個 bits 的組合, 一個 byte 可為字元(char)可表示一個字母、符號、數字或是符號, char 可組合成一個自組(word)
- (3) Field (欄位): 一組字組或是完整的數字
- (4) Record(紀錄): 一組相關的欄位集合
- (5) File(檔案): 一組相關的 Record 組合
- (6) Database(資料庫): 一組相關的 File 所組成

**FIGURE 6.1 THE DATA HIERARCHY**



A computer system organizes data in a hierarchy that starts with the bit, which represents either a 0 or a 1. Bits can be grouped to form a byte to represent one character, number, or symbol. Bytes can be grouped to form a field, and related fields can be grouped to form a record. Related records can be collected to form a file, and related files can be organized into a database.

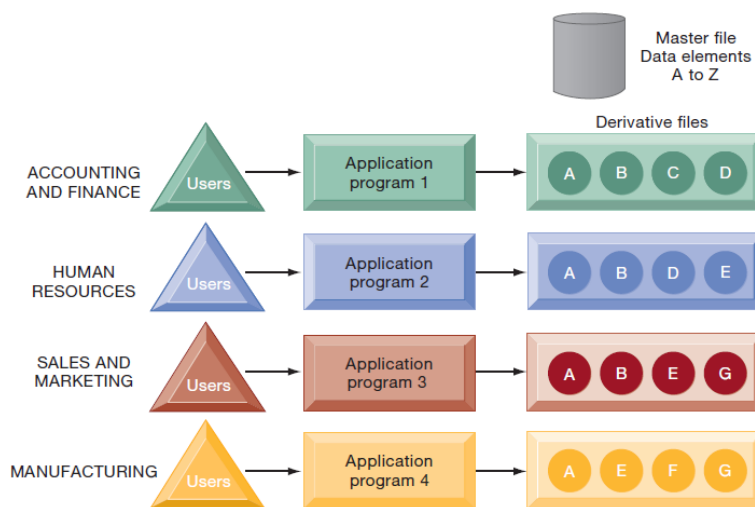
## 定義並說明實體和屬性的意義

- (1) **實體(Entity)**: 用來描述實際存在的事物, 也可以是邏輯抽象的概念, 能夠清楚分辨出兩個不同的實體, 大都是以「名詞」的型式來命名
- (2) **屬性(Attribute)**: 用來描述實體的性質(Property)

## 列出並描述傳統檔案系統(File System)的問題

- (1) **資料冗餘和不一致(data redundancy and Inconsistency)**：多個資料檔案中存在重複資料導致不一致
- (2) **資料依賴性(data dependence)**：資料與程式關係緊密，資料結構更改的話，程式也需修改
- (3) **缺乏彈性(lack of flexibility)**：無法應付特殊或臨時查詢，只能提供固定查詢
- (4) **安全性差(poor security)**：缺乏對資料的控制
- (5) **缺乏資料分享與可用性(lack of data sharing and availability)**：無法分享，各部門各自使用自己的檔案系統

**FIGURE 6.2** TRADITIONAL FILE PROCESSING

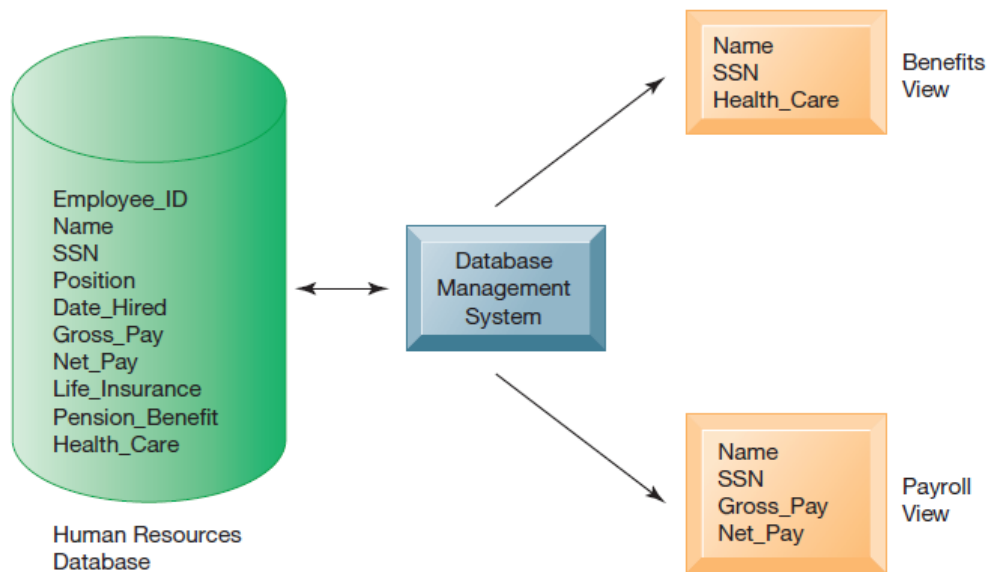


The use of a traditional approach to file processing encourages each functional area in a corporation to develop specialized applications. Each application requires a unique data file that is likely to be a subset of the master file. These subsets of the master file lead to data redundancy and inconsistency, processing inflexibility, and wasted storage resources.

## 定義 DB 和 DBMS

- (1) **Database**：一組資料，這些具有冗餘度小的資料藉由管理與儲存於同一個位置以有效服務多個應用程式並與應用程式彼此獨立
- (2) **DBMS**：一種特殊的軟體，它允許組織集中資料，有效地管理資料以及藉由應用程式存取；可降低冗餘、使得資料獨立、實現臨時查詢(ad-hoc)、減少開發和維護成本、可分享、並集中管理增強安全性

**FIGURE 6.3 HUMAN RESOURCES DATABASE WITH MULTIPLE VIEWS**



A single human resources database provides many different views of data, depending on the information requirements of the user. Illustrated here are two possible views, one of interest to a benefits specialist and one of interest to a member of the company's payroll department.

## 簡要說明 DBMS 的功能

DBMS 包括用於組織，管理和訪問資料庫中資料的功能和工具

- (1) **資料定義語言(data definition language, DDL)**：提供可產生、變更、刪除資料庫或資料表之命令
- (2) **資料操作語言(data manipulation language, DML)**：提供可以處理資料表之命令
- (3) **資料控制語言(data control language, DCL)**：提供管理資料庫或資料表授權使用之命令
- (4) **交易控制語言(Transaction Control Language, TCL)**：多人使用環境的交易控制命令
- (5) **資料字典(Data Dictionary)**：是一個自動的或手動的存儲數據元的定義和屬性的文檔

## 定義關聯式 DBMS 並說明其如何組織資料

建立在關聯模型基礎上的資料庫，藉助於**集合代數**等數學概念和方法來處理資料庫中的資料，資料其具有相互關聯的特性；是由資料表（Table）、紀錄（Record）、欄位（Field）以及資料（Data）所構成的。資料表示資料存在資料庫的形式，一筆一筆橫向的資料的是記錄，直向表示一筆資料的不同屬性，稱為是欄位。交易必須要滿足 ACID 的特性

ACID:

- (1) Atomicity
- (2) Consistency
- (3) Isolation
- (4) Durability

## 解釋資料操作語言的重要性

**Select**：用來篩選在關聯中，滿足某些選擇條件(selection condition) 的值組子集合。SELECT 運算可被視為一個過濾裝置 (filter)，它只留下滿足某個限制條件的值組

**Join**：用來把兩個關聯中的相關值組結合成單一的值組

**Project**：從表格中選擇某些欄位，而丟棄其他欄位，可視為關聯的垂直分割動作

## 解釋為什麼 NoSQL 有用

增長的主要原因有四個：

- (1) 雲端計算
- (2) 大數據
- (3) web service 的大量工作負載
- (4) 存儲新類型的資料需求

使用更靈活的資料模型，並設計用於管理分散式計算網路的大型資料集，可以根據計算需求輕鬆擴充和縮小；可以處理從網站、社群媒體、圖片中捕獲的結構化和非結構化資料，如果以後添加新一筆，則無需預先定義正式的資料庫結構或更改該定義，以加速對大量結構化和非結構化數據的簡單查詢，如：

Amazon SimpleDB、MongoDB、HBase

	關聯式資料庫	NoSQL 資料庫
最佳工作負載	關聯式資料庫專門用於交易性以及高度一致性的線上交易處理 (OLTP) 應用程式，並且非常適合於線上分析處理 (OLAP) 使用。	NoSQL 資料庫專門用於包含低延遲應用程式的多樣資料存取模式。NoSQL 搜尋資料庫專門用於進行半結構資料的分析。
資料模型	關聯式模型將資料標準化，成為由列和欄組成的表格。結構描述嚴格定義表格、列、欄、索引、表格之間的關係，以及其他資料庫元素。此類資料庫強化資料庫表格間的參考完整性。	NoSQL 資料庫提供鍵值、文件和圖形等多種資料模型，具有最佳化的效能與規模。
ACID 屬性	<p>關聯式資料庫則提供單元性、一致性、隔離性和耐用性 (ACID) 的屬性：</p> <ul style="list-style-type: none"> <li>單元性要求交易完整執行或完全不執行。</li> <li>一致性要求進行交易時資料就必須符合資料庫結構描述。</li> <li>隔離性要求並行的交易必須分開執行。</li> <li>耐用性要求從意外的系統故障或停電狀況還原成上個已知狀態的能力。</li> </ul>	NoSQL 資料庫通常透過鬆綁部分關聯式資料庫的 ACID 屬性來取捨，以達到能夠橫向擴展的更彈性化資料模型。這使得 NoSQL 資料庫成為橫向擴展超過單執行個體上限制的高吞吐量、低延遲使用案例的最佳選擇。
效能	一般而言，效能取決於磁碟子系統。若要達到頂級效能，通常必須針對查詢、索引及表格結構進行優化。	效能通常會受到基礎硬體叢集大小、網路延遲，以及呼叫應用程式的影響。
擴展	關聯式資料庫通常透過增加硬體運算能力向上擴展，或以新增唯讀工作負載複本的方式向外擴展。	NoSQL 資料庫通常可分割，因為存取模式可透過使用分散式架構來向外擴展，以近乎無限規模的方式提供一致效能來增加資料吞吐量。

比較標準	RDBMS	NoSQL	備註
資料庫原理	完全支持	部分支援	<ul style="list-style-type: none"> <li>RDBMS有關聯代數理論作為基礎</li> <li>NoSQL沒有統一的理論基礎</li> </ul>
資料規模	大	超大	<ul style="list-style-type: none"> <li>RDBMS很難實現<b>橫向擴展</b>，縱向擴展的空間也比較有限，性能會隨著資料規模的增大而降低</li> <li>NoSQL可以很容易透過添加更多設備來支援更大規模的資料</li> </ul>
資料庫模式	固定	靈活	<ul style="list-style-type: none"> <li>RDBMS需要定義資料庫模式，嚴格遵守<b>資料定義</b>和<b>相關限制條件</b></li> <li>NoSQL不存在資料庫模式，可以自由靈活定義並儲存各種不同類型的資料</li> </ul>
查詢效率	快	可以實現高效的簡單查詢，但是不具備高度結構化查詢等特性，複雜查詢的性能不盡人意	<ul style="list-style-type: none"> <li>RDBMS借助於<b>索引機制</b>可以實現快速查詢（包括記錄查詢和範圍查詢）</li> <li>很多NoSQL資料庫沒有以複雜查詢為主的索引，雖然NoSQL可以使用MapReduce來加速查詢，但是，在複雜查詢方面的性能仍然不如RDBMS</li> </ul>



比較標準	RDBMS	NoSQL	備註
一致性	強一致性	弱一致性	<ul style="list-style-type: none"> <li>RDBMS嚴格遵守交易ACID模型，可以保證交易<b>強一致性</b></li> <li>很多NoSQL資料庫放鬆了對交易ACID四特性的要求，而是遵守BASE模型，只能保證<b>最終一致性</b></li> </ul>
資料完整性	容易實現	很難實現	<ul style="list-style-type: none"> <li>任何一個RDBMS都可以很容易實現完整性限制，比如透過主鍵或者非空限制來實現<b>個體完整性</b>，透過主鍵、外來鍵來實現<b>參考完整性</b>，透過限制或者觸發器來實現<b>用戶自訂完整性</b></li> <li>但是，在NoSQL資料庫卻無法實現</li> </ul>
擴展性	一般	好	<ul style="list-style-type: none"> <li>RDBMS很難實現<b>橫向擴展</b>，縱向擴展的空間也比較有限</li> <li>NoSQL在設計之初就充分考慮了橫向擴展的需求，可以很容易透過添加廉價設備實現擴展</li> </ul>
可用性	好	很好	<ul style="list-style-type: none"> <li>RDBMS在任何時候都以保證資料一致性為優先目標，其次才是最佳化系統性能。隨著資料規模的增大，RDBMS為了保證嚴格的一致性，只能提供<b>相對較弱的可用性</b></li> <li>大多數NoSQL都能提供較高的可用性</li> </ul>

比較標準	RDBMS	NoSQL	備註
標準化	是	否	<ul style="list-style-type: none"> <li>RDBMS已經標準化（有ANSI SQL）</li> <li>NoSQL還沒有業界標準，不同的NoSQL資料庫都有自己的查詢語言，很難規範應用程式介面</li> <li>Stone Braker認為：NoSQL缺乏統一查詢語言，將會拖慢NoSQL發展</li> </ul>
技術支援	高	低	<ul style="list-style-type: none"> <li>RDBMS經過幾十年的發展，已經非常成熟，Oracle等大型廠商都可以提供很好的技術支援</li> <li>NoSQL在技術支持方面仍然處於起步階段，還不成熟，缺乏有力的技術支援</li> </ul>
可維護性	複雜	複雜	<ul style="list-style-type: none"> <li>RDBMS需要專門的資料庫管理員(DBA)維護</li> <li>NoSQL資料庫雖然沒有DBMS複雜，卻也難以維護</li> </ul>

[補充]

<https://aws.amazon.com/tw/nosql/>

定義和說明正規化和參考完整性並說明為何能以此設計出良好的關

## 聯式資料庫

**正規化**：是資料庫設計的一系列原理和技術，以**減少和精簡資料庫中資料冗餘**，**增進資料的一致性**，不過**資料表會變多**

**參考完整性(Referential Integrity)**：是當關聯表存在外來鍵時，外來鍵的值一定是來自參考主鍵，或為空值；也就是說，外來鍵的屬性值集合即是參考主鍵的屬性值集合

(1) **restricted**：不可更動參考的主鍵

(2) cascade：隨著參考的主鍵變動

(3) set to null

(4) set to default

[補充]

實體完整性：primary key 不可為 null，且需具備唯一性

→良好的關聯式資料庫依據需求進行設計，並且採用正規化的形式。未正規化的資料庫將在插入，刪除和修改時會遇到問題

## 定義和說明實體關係圖，並解釋其在資料庫設計中的作用

**實體關係圖**：實體關係圖以圖形方式描述了實體(關係)之間的關係

實體關係圖有助於製定企業的資料模型，有助於確保資料準確、完整且易於檢索；在概念結構設計階段用來描述資訊需求和／或要儲存在資料庫中的資訊的類型且設計良好的關係數據庫將不具有多對多關係

## 定義大數據(Big Data)並描述用於管理和分析的技術

Big data：不適用典型 DBMS 能夠儲存、分析的大量資料，其可為半結構化或非結構 → 大、快、雜、疑

(1) Volume 資料量

(2) Velocity 資料輸入輸出速度

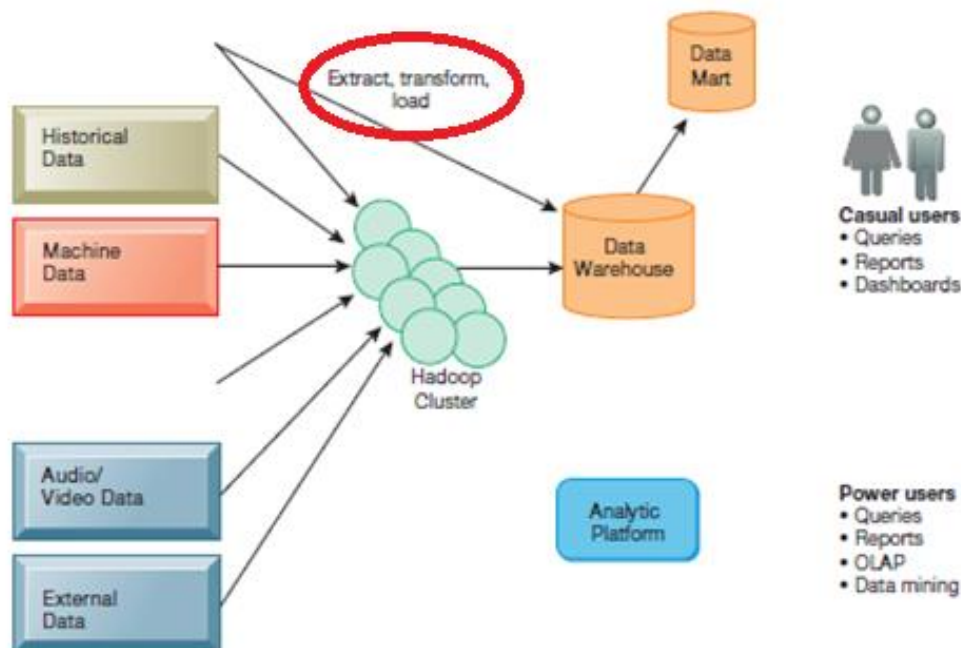
(3) Variety 資料類型

(4) Veracity 真實性

**Hadoop**：一個能夠儲存並管理大量資料的雲端平台，儲存超過一個伺服器所能容納的超大檔案，還能同時儲存、處理、分析幾千幾萬份這種超大檔案

## 列出並描述現代商業智慧基礎架構元件





- (1) **Data Warehouses(資料倉儲)**：儲存目前與過去來自交易處理系統、操作資料庫與其他企業應用軟體的所有營運資料，將多個主題領域整合在一起，可改善管理階層決策制定的資料庫 → 整合性、主題性、不可變動性、時間變動性
- (2) **Data Marts(資料市集)**：資料倉儲的子集，其目的是專門支援某些特定部門、特定地區，因此具有特定主題領域
- (3) **Hadoop**：關聯式資料庫及資料倉儲並不適合組織與分析巨量或非行列形式的資料，而 Hadoop 可以用來處理任何類型的大型資料
  - I. HDFS (hadoop distributed file system)：將 hadoop 群集中的眾多節點上的檔案系統連結成一個大型的檔案系統
  - II. MapReduce：用以拆解龐大資料集的處理程序，並分配工作給各節點執行 → spark
- (4) **analytical platform**
  - I. **Online Analytical Processing (OLAP)**：以多維度的模型讓使用者檢視所需的資料面向

	MOLAP	ROLAP	HOLAP
特色	<ul style="list-style-type: none"> <li>• Cube的內部資料與彙總資料儲存於多維度資料庫(MDBS)</li> <li>• 事先做彙總運算並寫入Cube內</li> </ul>	<ul style="list-style-type: none"> <li>• Cube的內部資料儲存於關聯式資料庫(RDBS)</li> <li>• 不事先做彙總運算</li> </ul>	<ul style="list-style-type: none"> <li>• Cube的內部資料儲存於關聯式資料庫(RDBS)</li> <li>• 彙總資料則儲存於多維度資料庫(MDBS)</li> <li>• 事先做彙總運算並寫入Cube內</li> </ul>
優點	<ul style="list-style-type: none"> <li>• 查詢效率最佳</li> <li>• 可處理複雜運算功能</li> </ul>	<ul style="list-style-type: none"> <li>• 建置成本低</li> <li>• 允許大範圍隨意查詢</li> </ul>	<ul style="list-style-type: none"> <li>• 建置成本與查詢效能介於MOLAP與ROLAP之間</li> </ul>
缺點	<ul style="list-style-type: none"> <li>• 耗費大量儲存容量</li> <li>• 建置成本高</li> </ul>	<ul style="list-style-type: none"> <li>• 查詢效能較差</li> <li>• 維護成本高</li> </ul>	<ul style="list-style-type: none"> <li>• 系統複雜度較高</li> <li>• 有User數限制</li> </ul>
代表產品	<ul style="list-style-type: none"> <li>• Cognos</li> <li>• Hyperion Essbase</li> <li>• Oracle Express</li> </ul>	<ul style="list-style-type: none"> <li>• Business Objects</li> <li>• Micro Strategy</li> </ul>	<ul style="list-style-type: none"> <li>• Microsoft Analyzer</li> </ul>

- II. **data mining** : 尋找無法由 OLAP 得到的資料隱藏類型和關係，利用統計分析、人工智慧(AI)或其他的分析技術分析資料，找出資料間的隱藏關係與規則，主要為存在資料庫或檔案中結構化的資料 → 關聯(Associations)、次序(Sequences)、預測(Forecasting)、分類(Classification)、群集(Clustering)、回歸分析(Regression)
- III. **text mining** : 可從大量非結構化的文字檔案中找出關鍵要素，如：以文字探勘分析客戶服務中心的手抄紀錄以找出主要的服務與維修議題 → sentiment analysis (情感分析)
- IV. **web mining** : 發掘與分析來自全球資訊網有用的類型與資訊 → 內容(content)挖掘、結構(structure)挖掘、使用度(usage)挖掘
- (5) **Databases and the Web** : 公司利用網路來讓夥伴或客戶接觸內部資料 → Web server、Application server—middleware, CGI scripts(在 server 上實現動態頁面提供了一種通用的協定)、Database server

[補充]

資料探勘

- (1) **關聯(Associations)** : 消費者所購買的產品間是否具有特定的關聯。例如，一個超市可能會收集顧客購買習慣的資料。運用關聯規則學習，超市可以確定哪些產品經常一起買，並利用這些資訊幫助行銷。這有時被稱為市場購物籃分析
- (2) **次序(Sequences)** : 與關聯分析相似，只是順序型態分析中的數據中具有次序及時間的關係
- (3) **預測(Forecasting)** : 和迴歸分析的概念相同，也是藉由已知的數據來預測未來數據的可能值，分析模型中的數據中必須含有時間關聯性
- (4) **分類(Classification)** : 透過研究資料庫中的特徵，將已知資料做出分類，並

根據已知的特徵預測未經分類的新資料

- (5) **群集(Clustering)**：是在未知資料的結構下，發現資料的類別與結構；資料集中的資料點，加以分群成數個群集，使得每個的資料點間相似程度高於與其它群集中資料點的相似程度

資料前置處理(Processing)

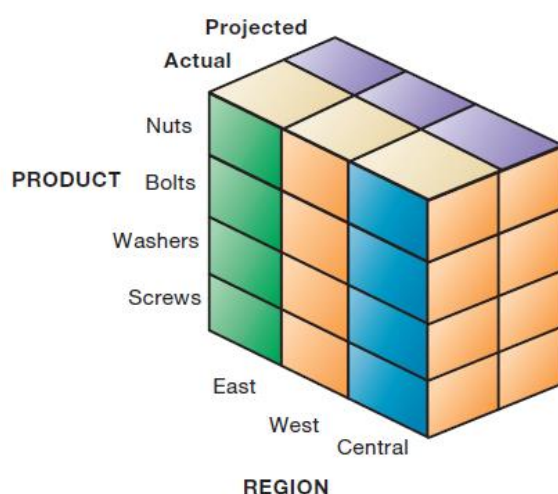
- (1) 萃取(Extraction)
- (2) 合併(Consolidation)
- (3) 過濾(Filtering)
- (4) 淨化(Cleansing)
- (5) 轉換(Conversion)
- (6) 彙總(Aggregation)

## 描述線上分析處理（OLAP）的功能

→ 一種可以協助使用者瀏覽、分析多維資訊及預測類比的工具，提供多維度、多角度的資訊，讓使用者可以藉其分析洞察營運狀況或指定決策

- (1) **切片(Slice)**：用戶可以從 OLAP 多維資料集中取出（切片）一組特定的資料
- (2) **切丁(Dice)**：從不同的角度檢視（切丁）
- (3) **下拉(Drill down)**：由宏觀到微觀的角度觀察資料，例如從年度銷售額往下拉到季，再往下拉到周，再往下拉到日
- (4) **上轉(Roll up)**：與下拉相反，例如從日的資料往上整合轉到週，再由週往上轉到季
- (5) **旋轉(Rotation)**：旋轉之後會改變維度的角度

**FIGURE 6.13 MULTIDIMENSIONAL DATA MODEL**



This view shows product versus region. If you rotate the cube 90 degrees, the face that will show is product versus actual and projected sales. If you rotate the cube 90 degrees again, you will see region versus actual and projected sales. Other views are possible.

[補充]

**線上交易處理(OLTP, Online transaction processing)**：透過資訊系統、電腦網路及資料庫，以線上交易的方式處理一般即時性的作業資料，和更早期傳統資料庫系統大量批次的作業方式並不相同，通常被運用於自動化的資料處理工作

	OLTP	OLAP
特色	作業性處理	資訊處理分析
導向	交易導向	分析導向
使用者	業務執行者、基層管理者、資料庫人員	知識工作者(如：高階管理者、執行長、決策人員)
功能	日常例行作業，交易活動	長期資訊分析，決策支援
資料庫設計	個體關係導向	主題、多維度導向
資料存取	由作業性資料庫存取資料，降低作業效率	不影響作業效率
資料來源	由交易產生	大多來自於資料倉儲
資料性質	即時性資料，交易明細	歷史性、彙總性資料
資料外觀	細節性、平面的關聯	彙整性、多維度
資料操作	讀寫	大多為讀取
異動頻率	資料隨時可新增、刪除與修改	歷史性彙整資料，異動機會少

**描述 Hadoop 並說明與關聯式 DBMS 的不同之處**

支援在廉價的計算機上以分散式平行處理大數據、In-memory 計算，將大數據問題分為子問題，將其分配到多達數千個廉價的計算機處理節點中，然後將結



果組合為較小的數據集處理後傳回，資料量比 DBMS 大很多，多樣性比較豐富，無自己的資料庫，藉由資料市集或倉庫存取，可處理非結構化資訊，可擴充性更高

## 解釋文字探勘(Text Mining)和 Web 探勘(Web Mining)與一般資料探勘

### (Data Mining)有何不同

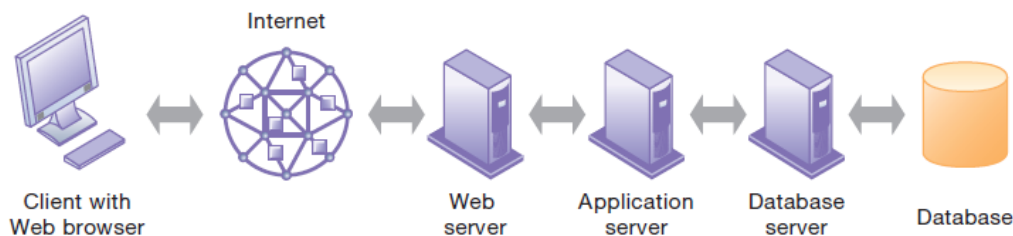
文字探勘(Text Mining)：可從大量非結構化的文字檔案中找出關鍵要素，如：以文字挖掘分析客戶服務中心的手抄紀錄以找出主要的服務與維修議題  
→sentiment analysis (情感分析)

網頁分析(web mining)：發掘與分析來自全球資訊網有用的類型與資訊

- (1) 內容(content)探勘：從網頁內容中提取知識的過程，其中可能包括文字，圖片，音樂和影片資料
- (2) 結構(structure)探勘：分析網頁連結
- (3) 使用度(usage)探勘：從網也伺服器分析使用者與網頁的互動

### 描述用戶如何通過網絡訪問公司內部資料庫中的資訊

FIGURE 6.14 LINKING INTERNAL DATABASES TO THE WEB



Users access an organization's internal database through the web using their desktop PC browsers or mobile apps.

Web 瀏覽器(客戶端 PC)：用於通過 Internet 訪問公司網站，並使用 html 與網頁伺服器溝通，再將這些對數據的請求傳遞給特殊的中介軟體，然後該中介軟體將 HTML 命令轉換為 SQL，再傳給資料庫伺服器做處理並結果返回

### 描述資訊政策(policy)和資料管理(administration)

資訊政策：針對資料管理的政策，規定了分享、傳播、獲得、規範、分類和儲存資料的規則，有關用戶和公司單位有權分享資料的程序和責任以及更新和維護資料的人員 → 規則、責任、維護人員

**資料管理**：負責將資料作為組織資源進行管理的特定策略和過程，包括制定資訊策略、規劃資料、監督資料庫設計和資料字典的開發以及監視資訊系統專家和最終用戶如何使用資料 → 資料為資源、特定策略、過程

[補充]

**data governance**：管理資料的妥善性、可用性、整體性及安全性的政策與程序，特別注重在隱私、安全、資料品質與符合政府法規

**database administration**：定義與組織資料庫的結構與內容，建立實體資料庫單元間的邏輯關係及存取的規則與安全程序

## 說明為什麼資料品質審查和資料清理重要

數據不正確，不完整或不一致會給企業造成嚴重的運營和財務問題，因為它們可能會導致產品不正確價格，客戶帳戶和庫存資料，並導致有關以下方面的決策不正確公司應該採取的行動。

**資料品質審查(Data Quality Audit)**：對資訊系統中資料的準確性和完整性進行結構化調查，調查整個資料檔案，調查資料檔案的樣本或調查用戶對資料品質的看法

**資料清理(Data Cleansing)**：用於檢測和糾正不正確、不完整、格式不正確或冗餘的資料，增強來自獨立資訊系統的不同資料集之間的一致性