

資料庫系統筆記 Ch.1

資料庫

資料庫類型

- 傳統應用類型
 - 數字與文字資料庫（Numeric and Textual）
- 近期發展類型
 - 多媒體資料庫
 - 地理資訊系統（Geographic Information System, GIS）
 - 生物與基因資料庫
 - 資料倉儲（Data Warehouses）
 - 行動資料庫
 - 即時資料庫（Real-time and Active）

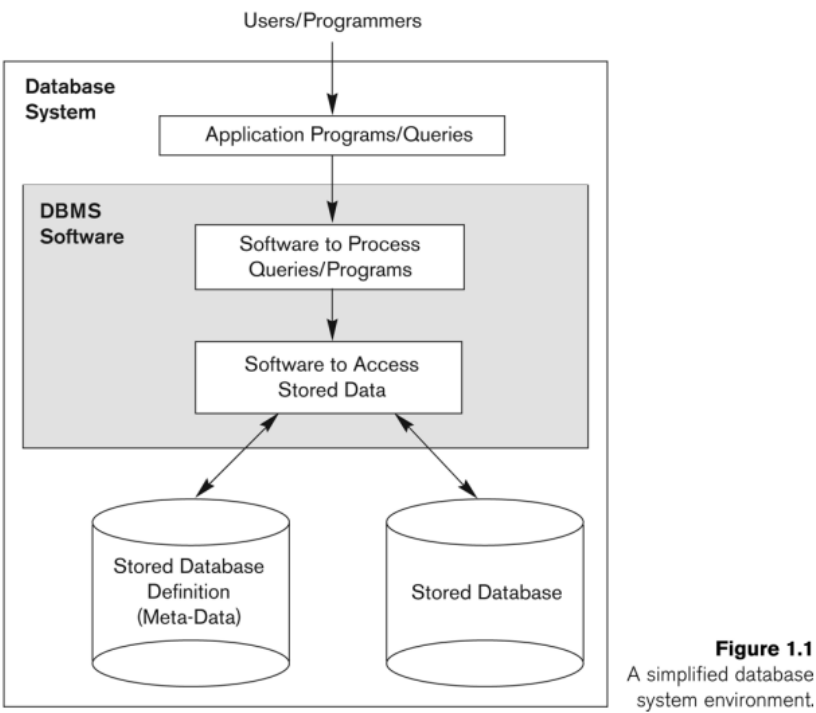
社交網路（Social Network）大量的交流資訊、搜尋引擎蒐集網頁資訊等.....大量資料管理，促進各種新技術發展：

- 大數據儲存系統（Big Data storage system）
 - 使用大量伺服器組成大型資料庫叢集。
- NOSQL Systems（Not Only SQL）

資料庫名詞的基本定義

- 資料庫：一群有關聯的資料的集合。
- 資料：可以被記錄，具有隱含意義的事實。
- 迷你世界（論域，Universe of discourse，UoD）：真實世界的一部分儲存在資料庫中。
- 資料庫管理系統（Database management system, DBMS）：可以管理資料庫的軟體，例如：MySQL、MariaDB、MS SQL ...
- 資料庫系統：由 DBMS 與資料組成，有時應用程式也包括在內。

資料庫的簡化圖



資料庫管理系統功能

以典型的資料庫來舉例：

- 根據資料類型（Types）、結構（Structures）、限制（Constraints），定義特定資料庫。
- 在次要的儲存裝置，建構（Construct）或載入（Load）初始資料庫內容。
- 操作資料庫，例如：查詢、生成報告、插入、刪除、更新，能夠透過網路應用程式來存取。
- 由一組使用者和應用程式同時進行資料的處理與共享，但仍然保持資料的有效性與一致性。

資料庫管理系統額外功能

- 保護（Protection）功能，防止軟硬體故障或當機（Crashes）。
- 安全性（Security）功能，防止未經授權（Unauthorized）或惡意（Malicious）存取。
- 資料呈現（Presentation）與視覺化（Visualization）。
- 在資料庫應用程式的生命週期內，維護資料庫及相關程式。
- 呼叫資料庫、軟體及系統維護。

資料庫應用程式功能

- 查詢（Queries），查詢不同部分的資料，訂定請求的結果。
- 交易（Transactions），讀取一些資料並更新某些值，或是生成新資料並儲存於資料庫中。
- 不允許未經驗證的使用者存取資料。
- 必須滿足使用者對資料庫的需求變化。

資料庫的特性

Source：[Characteristics of the Database Approach - GeekForGeek 2022/10/05](#)

- 自我描述性（Self-describing）
 - 現今的資料庫不僅儲存了資料本身，也儲存了資料庫的詮釋資料（Meta-data）。
 - 詮釋資料（Meta-data）會用於定義資料的結構，讓資料呈現某一種排序、儲存格式等等。
 - 現今的資料庫並不為某一應用程式所撰寫，所以需要藉由詮釋資料，來讓資料庫理解該怎麼格式出結構化的資料。
 - 在傳統，資料的格式定義通常是被寫死在檔案內，所以傳統應用程式通常只能使用特定唯一的資料庫。

當我們將格式定義從應用程式移至資料庫後，能夠讓應用程式較不受資料格式的限制來使用資料庫。

讓應用程式能夠介接第二、第三個甚至更多的資料庫。
- 程式與資料間獨立
 - 在傳統的應用程式中，資料格式通常都是直接被寫死在程式內，所以若我們要將當前的資料庫進行結構變更，則勢必得要從更改寫死在程式內的資料格式，到幾乎更改整個程式。
 - 在具有詮釋資料的 DBMS 中，說明資料格式的詮釋資料移至 DBMS 而不是寫死在程式中，因此我們不需要因為更改資料格式，導致程式需要做出大幅度的更改。
 - 藉由以上兩點，現今的 DBMS 讓程式與資料間做出了獨立性。
- 資料抽象化（Data Abstraction）
 - 若資料庫具有程式資料獨立與程式操作獨立，則 DBMS 具有資料抽象化的特性。
 - DBMS 隱藏了儲存方面的細節，並且提供了 DBMS 概念上的檢視。
- 支援資料 Multiple Views
 - DBMS 支援了多個使用者會看到資料庫不同的視角，讓每個使用者只會看到自己對於資料庫內感興趣的資料。
 - 對於一些需要不同視角的應用程式，這樣的特性會變成一種好處。
- 共享資料和多使用者交易處理
 - 資料庫支援多個使用者進行操作，但同時要維護多個使用者同時更動一筆資料時所產生的問題。
 - 所以資料庫出現了交易的概念，透過控制程序來控制多個使用者來更新同一筆資料時能夠確保資料是正確的。

資料庫的使用者

主要分為使用資料庫的人與操作資料庫的人。

使用資料庫的人

Reference: [Different types of Database Users](#)

在資料庫的管理上，主要可以分成四類人：

- 資料庫管理員（Database Administrators）：負責管理資料庫的人。
 - 負責在系統上運行資料庫、備份資料、規劃安全策略、保持資料庫的完整性。
- 資料庫設計者（Database Designer）：負責設計資料庫的人。
- 終端使用者（End User）：負責使用資料庫的人，可以細分成：
 - 非正式用戶（Casual End User）：只有需要使用資料庫的時候才會出現。
 - 初級用戶（Naive User）：不知道資料庫的運作原理，但是會很頻繁使用資料庫的人，例如：某個商店的店員。
 - 高級用戶（Sophisticated User）：知道資料庫運作原理，但通常不會撰寫程式，而是直接下指令來操作資料庫的人。
 - 獨行用戶（Standalone Users）：主要維護自己架的資料庫，使用已經成熟的應用程式來操作資料庫，例如：稅務計算人員。
- 開發或分析人員：例如系統分析師、應用程式開發人員、商業分析師。

製造資料庫的人

通常不是主要使用資料庫的人，屬於供給使用者使用的人，例如：

- DBMS 設計師與開發人員
- 工具開發者
- 操作人員與維護人員：維護硬體設備。

使用資料庫的好處

主要分成十點：

- 除去冗餘的操作：
 - 傳統的資料庫會因為不同的群組擁有不同的資料庫，但擁有可能相同的學生資料，導致校正困難。
 - 例如：學務處有 A 同學的資料、總務處有 A 同學的資料、教務處有 A 同學的資料，但學務處、教務處的 A 同學資料是錯誤的，導致如果要統一 A 同學的資料，就要去三個處室更改所有同學的資料。
 - 現今的資料庫需要經過一系列的規劃，在[資料庫的特性](#)中提到了資料庫會提供每個使用者感興趣的視角，如此一來處室就不需要各自維護自己的資料庫，而是共用一個資料庫，並且加上嚴謹的資料庫規劃讓資料冗餘的問題除去，就能夠有效除去冗餘的操作。
- 能夠分配使用者存取權限：
 - 現今的 DBMS 能夠提供分配使用者存取某些資料的權限，讓使用者沒有辦法存取越級的資料，讓資料的安全與機密性提升。
- 持久化資料儲存：
 - 對於程式來說，物件都是暫存的，也就是程式關閉之後資料即會消失。
 - 使用 DBMS 的好處就是提供了持久化的資料儲存，資料存在資料庫即使資料庫被關閉也不會消失。
- 提供資料結構化與高效查詢。
- 提供資料備份與資料還原功能。
- 提供多使用者存取介面。
- 呈現資料的複雜關係。
- 強制執行資料完整性約束（Integrity Constraints），限定資料的長度、型態、允許為 NULL 等等。
- 提供推理規則來找到資料。
- 一些其他的優點，例如：能夠建立資料的標準、減少開發者開發應用程式的時間等等。