

資料庫系統簡介

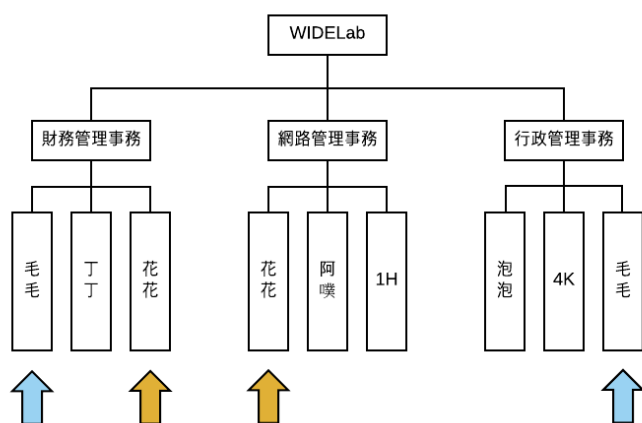
報告者：吳岱螢

資料庫系統

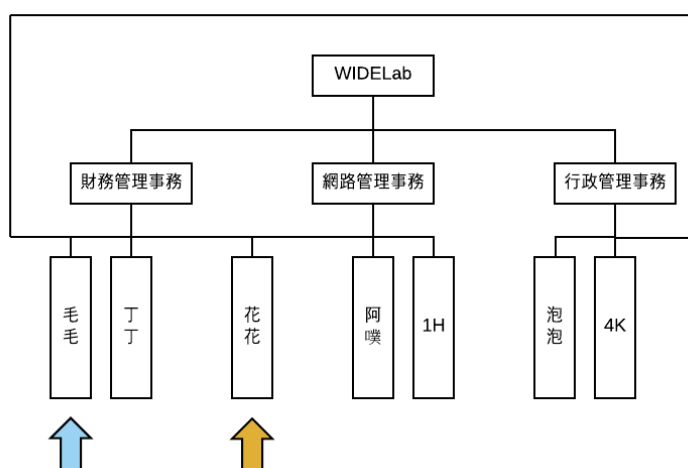
演化過程

根據不同的資料模型（即資料表現的形式），資料庫系統可以分成：

- （1）層次類型/階層式資料庫系統：如下圖，此類型的資料庫系統為最早期所採用的資料庫系統架構，其資料模式的基本架構為樹(Tree) 般的結構，它代表實體間一對多的關係(父—子關係)。一個父(Parent Node) 節點可以有多個子節點(Child Node)，而每個子節點只能有一個父節點。此種關係以圖形表示時，看起來像棵樹。此類結構的行為是父節點擁有多個子節點的紀錄，但是子節點卻只有唯一的父紀錄，因此將會造成資料重複的問題。
- 階層式資料庫很適合描述一對多的關係，但是它卻無法直接表達多對多的關係，因此後來有人提出可以表示多對多關係的網路式資料模式。



- (2) 網路類型/網狀資料庫系統：此類型的資料庫系統便是為了改善上述階層式資料庫系統所帶來的資料重複問題，如下圖，由於階層式資料無法表示多對多關係，因此提出新的網路式資料模式。主要與階層式不同點在於一筆子記錄節點可以有零個至多個父記錄節點。使子節點能夠同時擁有多個父節點的紀錄，這樣可以解決資料重複的問題，不過，當資料變得更複雜，其所帶出的複雜的網路關係會使得資料結構的更新變得更加困難。



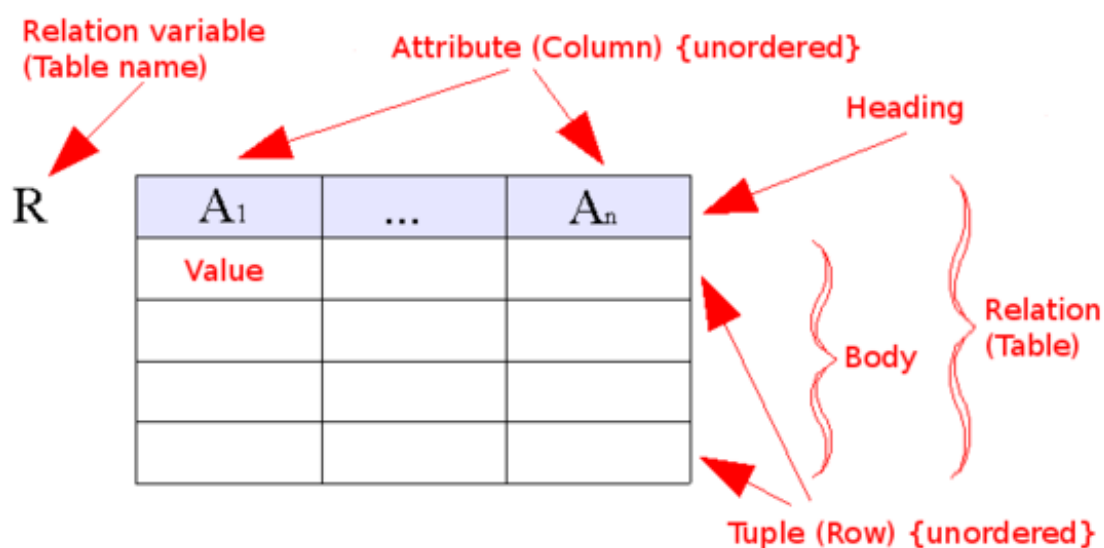
- (3) 關聯類型/關聯式資料庫：關聯式資料庫系統將所有的資料都透過行與列的二元表現形式表示出來，資料實際儲存自然並不是以表格方式儲存，但在使用者眼中知道它是一群表格即可，由於樹狀結構及資料與資料結構之間的依賴程度過高的關係，在網狀資料庫系統與階層式資料庫系統中的資料結構變更困難，而關聯式資料庫以資料值做為資料之間的關聯，使得各個資料獨立存在，消除了資料與資料結構之間

的依賴性，所以在關聯式資料庫系統中，資料結構的更動變得簡單許多。

關聯式資料庫

定義

是建立在關聯式資料模型(註 1)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。現實世界中的各種實體以及實體之間的各種聯絡均用關聯模型來表示，如下圖一



關聯式資料庫運用了一些專有名詞，以免在描述資料時造成混淆，檔案表格即稱為關聯（關聯 **Relation**）是以行（值組 **Tuple**）和列（屬性 **Attribute**）的形式組織起來的資料的集合。一個資料庫包括一個或多個表（關聯 **Relation**）。

例如，可能有一個有關作者資訊的名為 `authors` 的表（關聯 Relation）。每

列（屬性 **Attribute**）都包含特定類型的資訊，如作者的姓氏。每行（值組 **Tuple**）都包含有關特定作者的所有資訊：姓、名、住址等等。

許多關聯表的集合即形成「關聯式資料庫」，關聯式資料庫之所以稱為是關連，是因為資料其具有相互關聯的特性。不同的資料表間，可以透過特定的欄位將資料串起來。因為有這樣的特性，講求一個原則將資料做最有效率的儲存：「盡量避免重複的資料存在不同的資料表中」。

註 1：關聯式資料模型是由埃德加·科德(**Edgar F. Codd**)在 1970 年所提出，將資料間的關係用平面表格的方式來組合。

關聯式資料庫以 SQL 語言操作

SQL (Structured Query Language 結構化查詢語言) 是一種專門用來管理與查詢關聯式資料庫的程式語言。透過 SQL 能在關聯式資料庫裡新增、查詢、更新和刪除資料，同時也能建立和修改資料庫模式。它的語法簡單直接，一切都是以資料為主角去思考，範例如下：

```
SELECT * FROM customer;
```

表示從 **customer** 資料表取出所有資料。

資料庫系統組成

資料庫系統(DBS)=資料庫(DB)+資料庫管理系統(DBMS)

資料庫(Database)：一群相關的檔案所組成的資料集合

資料庫管理系統(Database Management System；DBMS)：一個套裝軟體，管理資料庫中所儲存的資料。

常見的關聯式資料庫管理系統(DBMS)

Microsoft SQL Server：Microsoft SQL Server(微軟結構化查詢語言伺服器)是由美國微軟公司所推出的關聯式資料庫解決方案，最新的版本是 SQL Server 2019，已在美國時間 2019 年 11 月 3 日發布。

資料庫的內建語言原本是採用美國標準局和國際標準組織所定義的 SQL 語言，但是微軟公司對它進行了部分擴充而成為作業用 SQL。

Oracle 資料庫：又名 Oracle RDBMS，或簡稱 Oracle。是甲骨文公司的一款關聯式資料庫管理系統。

MySQL：原本是一個開放原始碼的關聯式資料庫管理系統，原開發者為瑞典的 MySQL AB 公司，該公司於 2008 年被昇陽微系統（Sun Microsystems）收購。2009 年，甲骨文公司（Oracle）收購昇陽微系統公司，MySQL 成為 Oracle 旗下產品。MySQL 在過去由於效能高、成本低、可靠性好，已經成

為最流行的開源資料庫，因此被廣泛地應用在 Internet 上的中小型網站中。隨著 MySQL 的不斷成熟，它也逐漸用於更多大規模網站和應用，比如維基百科、Google 和 Facebook 等網站。

NoSQL 非關聯式資料庫

發展崛起

隨著電腦、行動裝置、與互聯網的普及，網路應用程式的流量大幅地增長，同時 互聯網也進入「使用者生產內容 (user generated content)」為主流的時代。對於 Youtube、Facebook 這些社交網站來說，每分每秒需要處理的資料量是過去一般網站的非常多倍。

而從使用者的角度來看，他們在這些平台上對於資料的需求也跟過去不太一樣。資料庫的主要功能，從過去的「能夠無錯誤地同步處理結構清楚的資料」，到現在慢慢有新需求誕生：「處理高速且大量產生的資料，但不需要即時同步，也不需絕對地無錯誤。」

定義

NoSQL 的意思是 "Not Only SQL"，就是不限定為「關聯式資料庫」的資料庫管理系統的統稱，允許部分資料使用 SQL 系統儲存，而其他資料允許使用 NOSQL 系統儲存。其數據儲存可以不需要固定的表格模式以及中介資料

(metadata)。

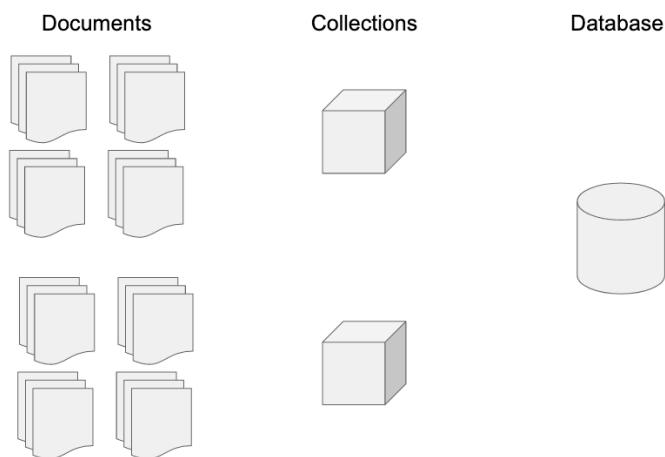
NoSQL 將聚集後的資料，作為儲存的最小單位，透過縝密豐富的資料結構，有利於將資料分散到多個節點；比起資料的關聯，NoSQL 更關注資料所代表的人（例如使用者）與物（例如一篇分享在社交平台上的文章）的「狀態」變動，例如文章被分享、按讚等。

種類

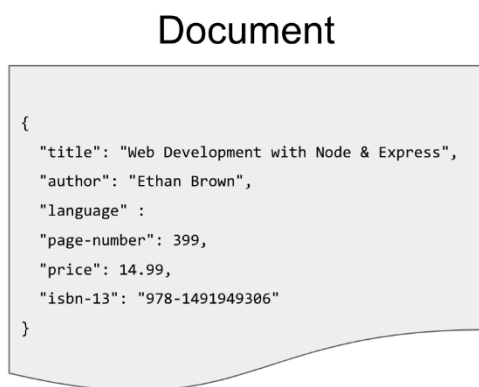
早在 NoSQL 資料庫這個名詞流行之前，就已經出現了很多種非關聯式資料庫，這些資料庫各有不同的特徵，很難像關聯式資料庫那樣，用一套共同的觀念就可以全部了解。只能逐一了解每個 NoSQL 資料庫的特性和用途。四種比較受到關注的 NoSQL 資料庫，分別是 Key-Value 資料庫，記憶體資料庫（In-memory Database）、圖學資料庫（Graph Database）以及文件資料庫（Document Database）。

以文件資料庫 (document database)舉例如下：

文件資料庫是 NoSQL 資料庫的一種，顧名思義，是把資料存放成「文件 (documents)」，這些文件會組成為「集合 (collection)」並放在一起。



在文件資料庫裡，文件會存成 **JSON** 格式，而資料物件會由「屬性-值」(attribute-value pair) 或陣列組成。



市面常見非關聯式資料庫(以 MongoDB 為例)

MongoDB 是一種文件導向的資料庫管理系統，用 C++等語言撰寫而成，以解決應用程式開發社群中的大量現實問題。MongoDB 由 MongoDB Inc.

（當時是 10gen 團隊）於 2007 年 10 月開發，2009 年 2 月首度推出，現以伺服器端公共許可（SSPL）分發。

Logo 是以一片葉子為主題，代表樹狀結構。



MongoDB 屬於文件資料庫（Document Database），以文本方式儲存，本身沒有 Schema，所以在架構上很好調整。

特色

- (1) 在 Relational Database 中的表格(table)類似 MongoDB 的 Collection(集合)，而資料列(Row)類似 MongoDB 的 document(文件)
- (2) MongoDB 提供了 Map-Reduce 的聚合工具來進行資料的統計。
- (3) 為無綱要(Schemaless)設計(->像是 char(30)先給予限制)，不需要事先定義資料庫的綱要，比傳統關聯式資料庫更加有彈性。
- (4) 可以儲存非文件的大型物件，MongoDB 除了可以儲存文件(最大 16MB)，也可以儲存非文件的物件(圖片或影片)，但需使用 MongoDB 的 GridFS(為 MongoDB 所定義的一種規範)
- (5) MongoDB 一種強大，靈活、且易於擴展的文件導向式(document-

oriented)資料庫，文件是採取 JSON 的格式作為儲存資料的方式。

- (6) 應用在網站登出入、擴充快取記憶體、大型檔案儲存有較優的表現，相反的在需要高交易的系統(像是銀行/會計系統)就不太適合，因為可能有資料不一致的現象。

關聯式與非關聯式資料庫比較

	RDBMS	NoSQL	較優
資料完整性	需就事情定義	不一定，通常不一樣	RDBMS
資料庫型態更改	一旦更改要耗大量成本，困難	相對要容易，彈性較佳。	NoSQL
查詢能力	強，有 join 的功能	對於複雜的查詢較弱	RDBMS
重複性	易整理，通過正規化來減少負擔	事前若沒規定會愈積愈亂	RDBMS
擴充性	可以加大 CPU，RAM 等硬體，加強處理能力	加設伺服器處理大流量	視需求而定
節點分布	隨著資料寫入多個節點而受限	減少了關聯，能分佈在多個節點上。	NoSQL
資料庫的增長	很難實施分片或分區操作，技術困難	容易擴建，成本較低	NoSQL
效能測試和分析的報告工具	有各種工具可用	較缺乏	RDBMS
語言標準化	SQL	傾向於使用自己的語法，不相容 SQL，版本升級易有風險。	RDBMS
使用者群的成熟度	老手較多，使用者經驗多。	相對較新	RDBMS

參考資料

<https://ithelp.ithome.com.tw/articles/10185512>

<https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93>

<https://dotblogs.com.tw/killysss/2013/02/27/93295>

<https://lsyucode.wordpress.com/2016/09/24/%E9%97%9C%E8%81%AF%E5%BC%8F%E8%B3%87%E6%96%99%E5%BA%AB-v-s-%E9%9D%9E%E9%97%9C%E8%81%AF%E5%BC%8F%E8%B3%87%E6%96%99%E5%BA%AB-%E4%B8%8A/>

<http://faculty.stust.edu.tw/~jehuang/oracle/ch1/1-3.htm>

<https://tw.alphacamp.co/blog/sql-nosql-database-dbms-introduction>

<https://www.fooish.com/sql/syntax.html>

<http://database.klab.tw/teach/t1.php>

<https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93>

<https://www.ithome.com.tw/news/92506>

<https://tw.alphacamp.co/blog/sql-nosql-database-dbms-introduction>

<https://zh.wikipedia.org/wiki/NoSQL>

<https://www.ithome.com.tw/news/92507>

<https://dotblogs.com.tw/killysss/2013/02/27/93295>

<https://dotblogs.com.tw/exploooosion/2018/01/21/040728>

<https://b0212066.pixnet.net/blog/post/216898056-mongodb-%E5%AD%B8%E7%BF%92%E7%AD%86%E8%A8%98>