關聯式與非關聯式資料庫系統簡介

作者：湯士豐

一、簡介關聯式/非關聯式資料庫

在介紹關聯式資料庫與非關聯式資料庫之前，我們先看一下 DB-Engines 對資料庫流行程度所做的排名調查。

358 systems in ranking, September 2020

Rank Sep 2020	Rank Aug 2020	Rank Sep 2019	DBMS	Database Model	Score Sep 2020	Score Aug 2020	Score Sep 2019
1.	1.	1.	Oracle	Relational, Multi-model	1369.36	+14.21	+22.71
2.	2.	2.	MySQL	Relational, Multi-model	1264.25	+2.67	-14.83
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1062.76	-13.12	-22.30
4.	4.	4.	PostgreSQL	Relational, Multi-model	542.29	+5.52	+60.04
5.	5.	5.	MongoDB	Document, Multi-model	446.48	+2.92	+36.42
6.	6.	6.	IBM Db2	Relational, Multi-model	161.24	-1.21	-10.32
7.	7.	↑ 8.	Redis	Key-value, Multi-model	151.86	-1.02	+9.95
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model	150.50	-1.82	+1.23
9.	9.	↑ 11.	SQLite	Relational	126.68	-0.14	+3.31
10.	↑ 11.	10.	Cassandra	Wide column	119.18	-0.66	-4.22
11.	↓ 10.	↓ 9.	Microsoft Access	Relational	118.45	-1.41	-14.26
12.	12.	↑ 13.	MariaDB	Relational, Multi-model	91.61	+0.69	+5.54
13.	13.	↓ 12.	Splunk	Search engine	87.90	-2.01	+0.89
14.	14.	↑ 15.	Teradata	Relational, Multi-model	76.39	-0.39	-0.57
15.	15.	↓ 14.	Hive	Relational	71.17	-4.12	-11.93
16.	16.	↑ 18.	Amazon DynamoDB	Multi-model	66.18	+1.43	+8.36
17.	17.	↑ 25.	Microsoft Azure SQL Database	Relational, Multi-model	60.45	+3.60	+32.91
18.	18.	↑ 19.	SAP Adaptive Server	Relational	54.01	+0.05	-2.09
19.	19.	↑ 21.	SAP HANA	Relational, Multi-model	52.86	-0.26	-2.53
20.	20.	↓ 16.	Solr	Search engine	51.62	-0.08	-7.35

2020 年 9 月資料庫流行程度排行　　資料來源：DB-Engines 網站

排名前 10 的資料庫系統中，關聯式資料庫占了 6 位，非關聯式資料庫占了 4 位。其中，第 5 名的分數是 446，第 6 名的分數卻掉到 161。由此可知，除了 MongoDB 之外，其他非關聯性資料庫的流行程度並不高。若是將前 10 名 DBMS 所支援的資料庫模型列出來，整理成表格如下：

Ranking	DBMS	關聯式	非關聯式						
		Relational	Document	Key-value	Wide Column	Graph	Search Engine	RDF	Time Series
1	Oracle	◎	○			○		○	
2	MySQL	◎	○						
3	MS SQL Server	◎	○			○			
4	PostgreSQL	◎	○						
5	MongoDB		◎				○		
6	IBM DB2	◎	○					○	
7	Redis		○	◎			○		○
8	Elasticsearch		○				◎		
9	SQLite	◎							
10	Cassandra				◎				

前 10 名 DBMS 所支援的資料庫模型　資料來源：自行整理

關聯式資料庫架構是 1970 年由埃德加·科德 (Edgar Frank Codd) 所提出。在那個電腦運算能力不足、記憶體昂貴、儲存空間速度緩慢的年代，關聯式資料庫的推展是緩慢的；幸運的是，1970 年代 Smalltalk 語言驚艷了軟體/系統開發產業，奠定了物件導向語言的基礎；1976 年陳品山博士(Peter P.S Chen)提出實體關係模型(E-R Model)，讓資料庫設計更加標準化與系統化。

這時候，當時間走向 1980~90 年代，硬體能力開始快速提升，圖形化作業系統開始普及，軟體/系統開產業發勢不可擋、一飛沖天，關聯式資料庫更在這一波潮流中，開始獨霸市場。

就在形勢一片大好之際，全世界正開心歡慶千禧年之餘，網通產業狠狠地跌了一跤，網路泡沫化重擊了軟體/系統開發產業，.com 顯得中看不中用、Broadband 死傷慘重、Datacenter 紛紛倒閉或縮編，說好的願景，通通不見了。在差不多同一時間，微軟在 2002 年推廣 .NET 架構，這個改變讓許多軟體工程師被迫重新開始學習，也就在這個關鍵時刻，很多企業/軟體工程師轉而擁抱 Linux 系統。

在微軟閉關重練、緩慢前進、並推展 .NET 以及 Windows Vista 的那幾年，全世界被金融危機、次貸危機搞得痛不欲生的 2000 年代中後期，臉書讓人們打開心房、蘋果讓大家看到未來、3G 讓大家隨時上網、Youtube 讓生活充滿歡樂。

就在那樣一個交錯的年代，我們對於資料的想法變了，對資料庫的需求也變了。關聯式資料庫以「交易(Transaction)正確性」至上、「節省硬體資源利用」為出發點的理念，突然顯得不重要了。沒有人會在乎臉書按讚的次數有沒有即時更新或是有沒有錯誤，沒有人會在乎 Youtube 影片很吃網路流量與資源，也沒有人會在乎一大堆高清照片沒有儲存空間。世界變了，真的變了，變成了西元 2000 年那個時候，人們幻想中的樣子。

雖然晚了快 10 年，不過，遲到總比不到好。時間進到了一切都活過來的 2010 年代，新的資料庫概念興起，NOSQL 被大聲喊了出來，大數據也說已經準備好了，IOT 變得好棒棒，AI 變成顯學。

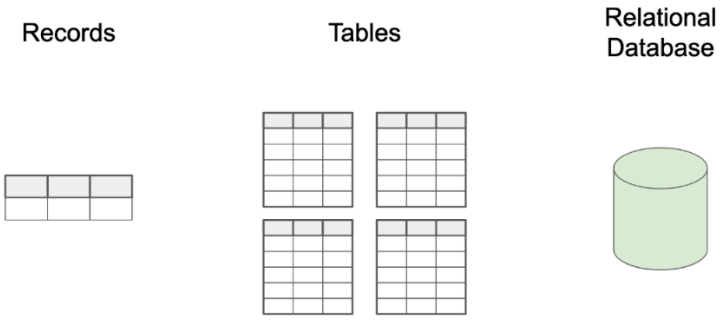
在邁入 2020 年代的當下，新冠肺炎再次改變人類的生活型態，宅經濟、遠距協同作業成為必然性的發展。不過，這樣的改變並不會影響關聯性資料庫的未來。從 DB-Engines 的排名來看，關聯式資料庫大廠的策略相當清楚，支援非關聯式資料庫的資料模型(特別是文件資料庫)。而非關聯性資料庫卻因為設計架

構與開源(open source)的情節，不太可能支援關聯性資料庫(好比說：NOSQL 的架構就很難實現 ACID 的要求)。

因此，在可以預見的未來十年，關聯式資料庫與非關聯式資料將會並存在這個世界上。因為市場區隔的關係，我認為並不會有太劇烈的競爭，只不過隨著時間的發展，那些能夠支援兩種資料庫的廠商，將會是這場競賽的贏家。

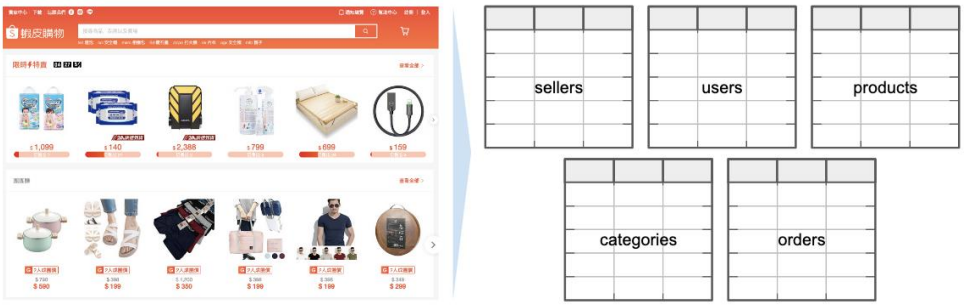
接下來，很快地介紹一下關聯式資料庫與非關聯式資料庫。關聯式資料庫有三個特質：

1. 資料存放在資料表 (table)中：
   關聯式資料庫系統，每一筆資料都是儲存在 table 中的一筆 record，再把不同的 table 建立關聯，並儲存在一個資料庫中。



(圖片來源: Alpha Camp : SQL/NoSQL 是什麼？認識資料庫管理系統 DBMS)

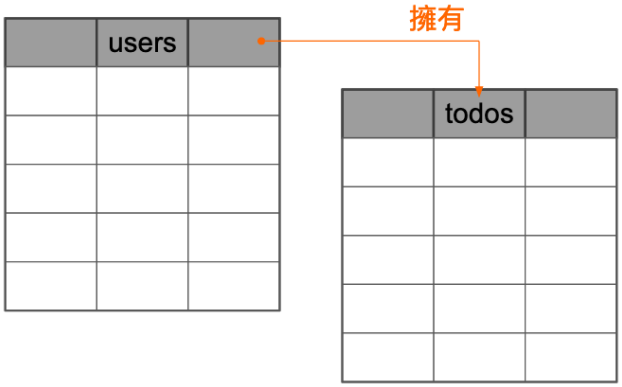
例如：在一個電商網站中，會有賣家、使用者、商品、分類和交易紀錄等資料表(Table)，分別負責儲存不同的資料(唯一資料)。



(圖片來源: Alpha Camp : SQL/NoSQL 是什麼？認識資料庫管理系統 DBMS)

2. 資料之間有明確的關聯：
   以電影清單專案為例，在建立 To-do List 認證系統的時候，我們設計

todos 跟 users 這兩種屬性，透過正規化過程，這兩種資料將會存放在兩個資料表中，並設定「uesers 擁有 todos」這個關聯。



(圖片來源: Alpha Camp : SQL/NoSQL 是什麼？認識資料庫管理系統 DBMS)

3. 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 語法，我們能對關聯式資料庫新增、查詢、更新和刪除資料。

例如：SELECT * FROM [TABLE_NAME] WHERE [COND];

這句話的意思，是「從 [TABLE_NAME] 的資料表中取出滿足 [COND] 條件的資料」。

接下來，很快介紹一下非關聯式資料庫。

非關聯式資料庫會儲存非結構化或半結構化的資料，而不是關聯性資料庫正規化過後的資料表，通常使用 XML 或是 JSON 檔的方式儲存。

從資料模型的角度來看，非關聯式資料庫最常見到的類型有四個：

1. 文件導向資料庫 Document Store Database：

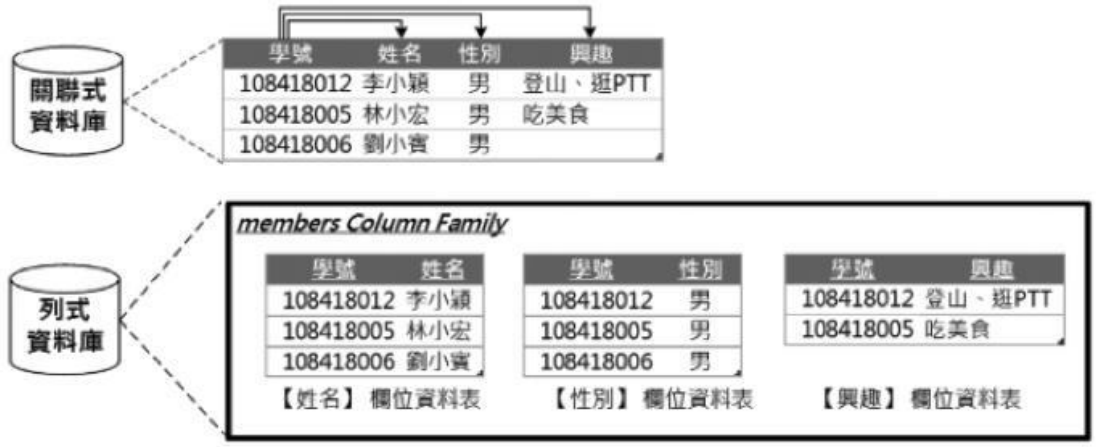
資料不受欄位限制，透過 JSON 檔將資料存放在一個文件(Document)中。而眾多文件收集起來，便成為一個集合(Collection)。

(圖片來源: 7 天學會大數據處理資料 NoSQL-MongoDB 入門與活用)

2. 鍵值導向資料庫 Key-value Store Database：
   每筆資料都會描述兩個參數，一個是 Key，一個是 Value。透過 Key 可以快速取得想要的 Value，而所有資料，將彙整到一個桶子(Bucket)中。只不過從下面這張圖片來看，完成一個交易可能需要很多 Key 才能完成。
   。

(圖片來源: 7 天學會大數據處理資料 NoSQL-MongoDB 入門與活用)

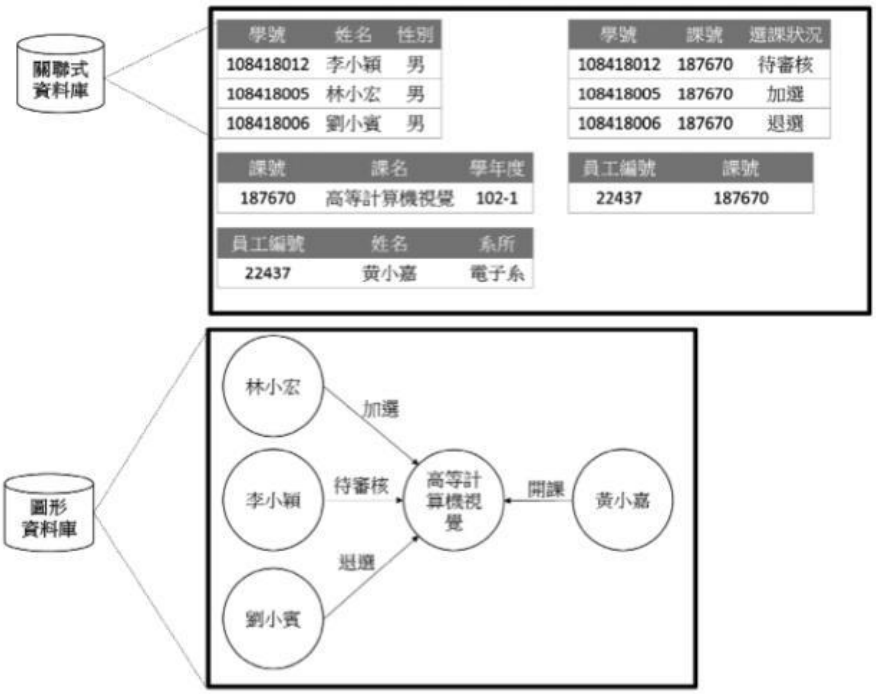
3. 列式導向資料庫 Wide-Column Store Database ：
   傳統關聯式資料庫的一筆資料將被會被拆分，並存在很多個 Table 中，其中，拆出來的 Table 只會有兩個欄位，一個是共同值，一個屬性資料。
   而這些 Table 將會被存在一個 Column Family 中。



(圖片來源: 7 天學會大數據處理資料 NoSQL-MongoDB 入門與活用)

4. 圖形導向資料庫 Graph Store Database ：
   運用圖結構的概念來儲存資料，並運用圖形結構相關演算法提高性能，例

如：用樹狀結構來描述從屬關係，或是用網狀結構來描述朋友關係。看起來就像是還沒有轉成 Schema 的 E-R model 圖。



(圖片來源: 7 天學會大數據處理資料 NoSQL-MongoDB 入門與活用)

二、詳細介紹任一非關聯式資料庫

我們可以從 DB-Engines 的系統功能中，取得一個比較表。從流行程度以及各大 DBMS 支援程度來看，Document store 的資料庫模型算是非關聯性資料庫的顯學，這個章節，我將會針對 MongoDB 進行較詳細的介紹。

Editorial information provided by DB-Engines				
Name	**Cassandra** X	**Elasticsearch** X	**MongoDB** X	**Redis** X
Description	Wide-column store based on ideas of BigTable and DynamoDB	A distributed, RESTful modern search and analytics engine based on Apache Lucene	One of the most popular document stores available both as a fully managed cloud service and for deployment on self-managed infrastructure	In-memory data structure store, used as database, cache and message broker
Primary database model	Wide column store	Search engine	Document store	Key-value store
Secondary database models		Document store	Search engine	Document store Graph DBMS Search engine Time Series DBMS
DB-Engines Ranking Trend Chart	Score 119.18 Rank #10 Overall #1 Wide column stores	Score 150.50 Rank #8 Overall #1 Search engines	Score 446.48 Rank #5 Overall #1 Document stores	Score 151.86 Rank #7 Overall #1 Key-value stores
Website	cassandra.apache.org	www.elastic.co/elasticsearch	www.mongodb.com	redis.io
Technical documentation	cassandra.apache.org/doc/latest	www.elastic.co/guide/en/-elasticsearch/reference/current/-index.html	docs.mongodb.com/manual	redis.io/documentation
Developer	Apache Software Foundation	Elastic	MongoDB, Inc	Salvatore Sanfilippo

非關聯式 DBMS 比較表　　資料來源：DB-Engines 網站

MongoDB 能儲存 JSON 及 Schema-free 的資料，是一種基於文檔(Document store) 的分散式資料庫，相較於傳統關聯式資料庫，MongoDB 對於巨量資料、高併發以及高可靠性有強大的支援能力；和其他 NoSQL 資料庫相比，MongoDB 基於文檔的資料模型及其動態建模的特性使得它更加自由靈活；分片(Sharding)的資料分散處理架構，使 MongoDB 得以透過水平擴充儲存海量資料。

使用彈性

MongoDB 是一種針對半結構化資料而設計的資料庫，允許「資料格式不一致」的彈性以及硬體擴充的大量儲存能力，因此非常適合作為資料倉儲。

開發容易

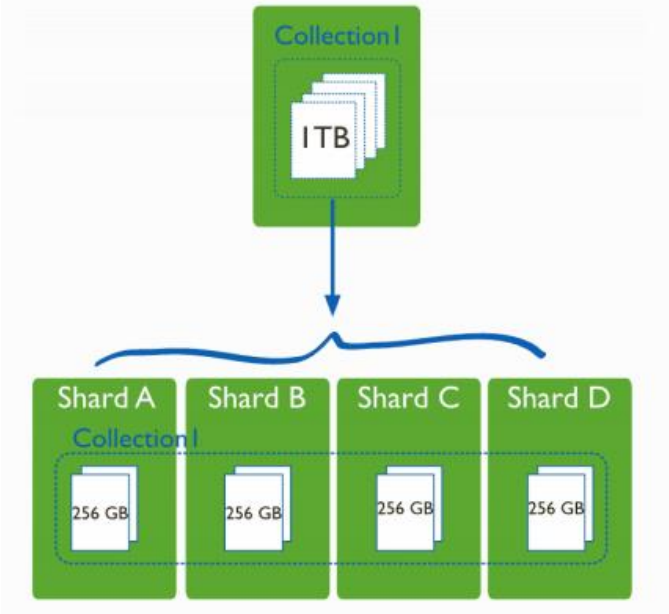
MongoDB 的文檔資料模型使開發人員易於學習和使用，由於半結構化資料不強制規定資料欄位，開發者可以依照業務需求，快速地開發出軟體系統。另外，MongoDB 開發者社群活動狀況算是活耀的(在 DB-Engines 上，是非關聯式資料庫的第一名)，而 MongoDB 官方也提供了線上課程、以及建議的學習路徑。

MongoDB Sharded Cluster

分片(sharding)是一種在多台主機之間分配資料的方法。MongoDB 使用分片來實現具有非常大的數據和高吞吐量操作的部署。

MongoDB 可以透過水平擴充增加更多節點，以補足垂直擴充架構的不足，來達到更大的儲存容量與更高的效能。這些節點可以分擔工作量，因此與單台高速大容量主機相比，可以提供更高的效率。擴充時可以根據需求增加主機，比起購置單台高昂硬體更能降低總體成本。隨著節點數增加，容量和資料吞吐量都會一併增加，隨著資源的投入，容量和性能均可以線性關係增長，可達到 PB 規模的資料量。

下圖為 MongoDB Sharded Cluster 的基本架構圖，Sharding Cluster 使得數據可以平均分散到多個 Shard 儲存，使 MongoDB 具備了橫向擴展的能力。

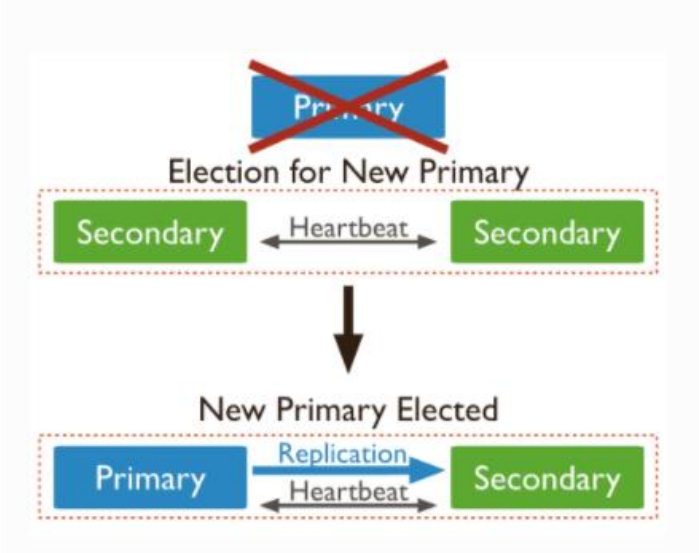


(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

MongoDB Replica Set

MongoDB 內建副本集架構(Replica Set)，副本集是所有生產部署的基礎，並提供冗餘(redundancy)和高可用性(high availability)。

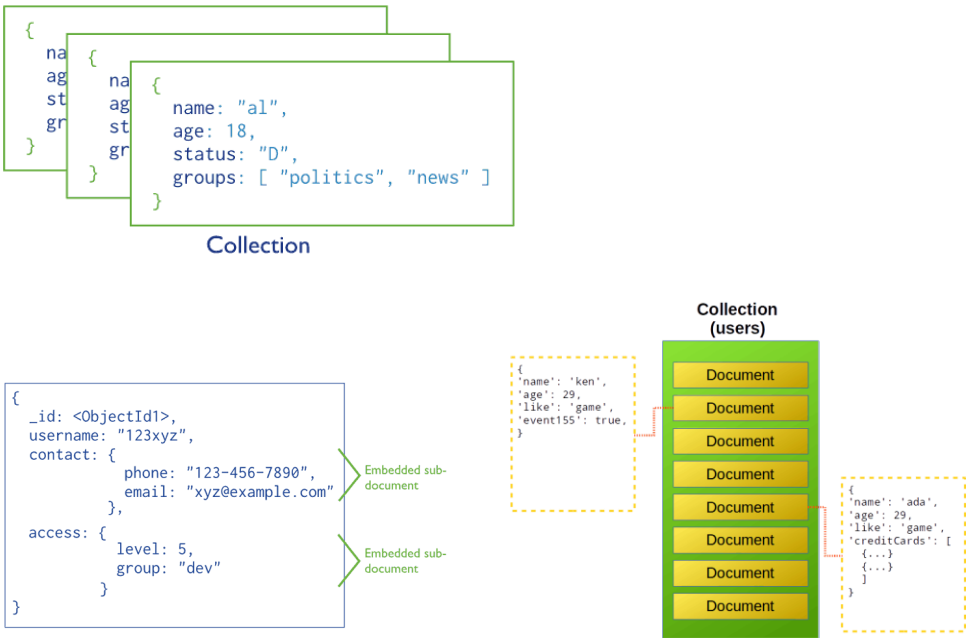
每個副本集內部包含多台主機，其中一台為 Primary，其他台為 Secondary，多台機器之間的狀態及資料會自動同步。當 Primary 因故不可用時，系統會自動在數秒內從 Secondary 中選出一台 Primary Rollback 原先 Primary 的寫入操作並恢復工作，從而實現高可用性。整個流程完全不須維運人員介入，就算系統在半夜當機也不用擔心。



(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

Document Base 基於文檔的資料庫

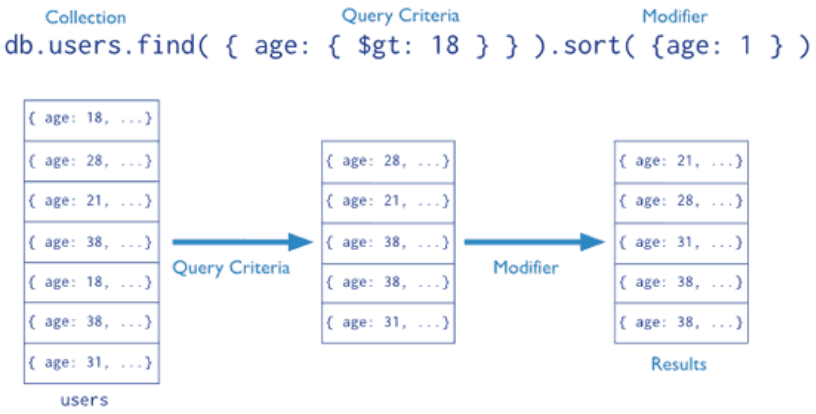
MongoDB 採用 Document Base 的資料模型，每筆資料(對應到傳統資料庫的 Row)被稱為一個文檔，以類似 JSON 的文檔(Document)構建及儲存。基於 JSON 的規格特性，MongoDB 可以在欄位中儲存其他文檔，自然形成目錄般的多層結構，稱為次級結構；MongoDB 內建多值支援，可以在欄位中儲存多值陣列，無需另開表格。陣列欄位可被用戶充份掌握，不但可以在上面建立索引也可以針對陣列中的個別元素進行搜尋與更新等操作，也可動態追加或刪除其中的任意元素。



(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

強大的查詢語言

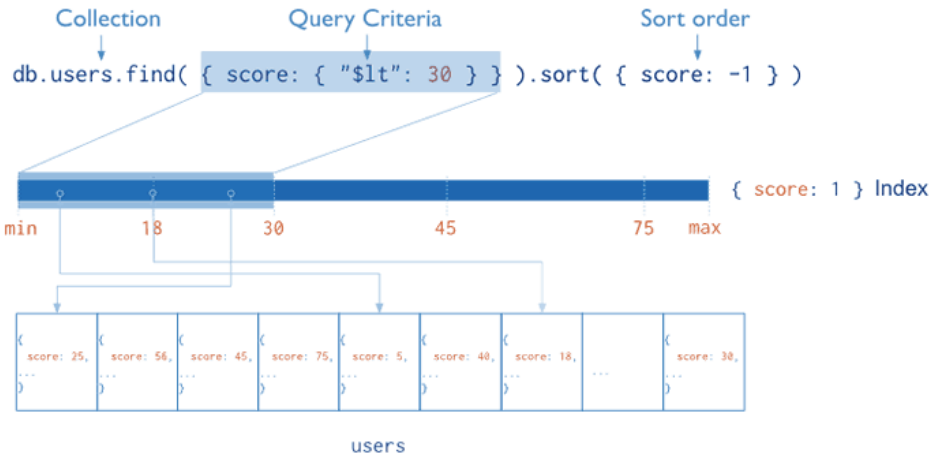
MongoDB 的查詢語言本身就是 JSON 格式，不需要特別串接成 SQL 查詢字串，豐富而清楚的查詢語言可以按任意字段進行過濾和排序，儘管文檔中有多個層級也能查詢。MongoDB 的查詢語言也支持聚合(aggregation)和近代的查詢需求，例如地理搜索、圖形搜索和文本搜索。



(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

豐富的索引

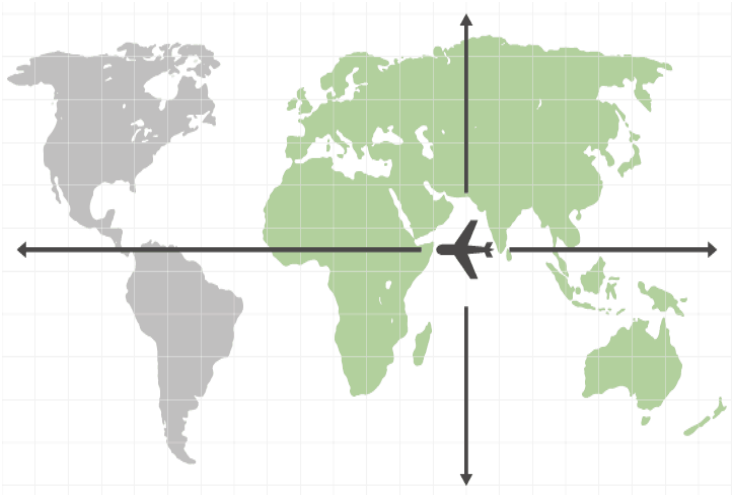
MongoDB 使用基於 Btree 的索引(index)系統，並透過索引執行高效率的查詢。MongoDB 提供了許多不同的索引類型，用戶可以自行依照資料多樣性，建立不同索引，查詢特定類型的數據。MongoDB 允許用戶針對半結構資料中的任何欄位；也允許建立多層級索引，也可以執行「先以出生地排序，再以出生日期排序」的基礎操作。



(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

地理索引

MongoDB 對地理空間資訊有一系列的支援，用戶可以對地理標記進行索引，讓應用程式能執行空間查詢。例如：快速篩檢出最近的地標、找出半徑 200 公尺內的地標、或是依照距離遠近進行排序等。



(圖片來源: 歐立威科技 MongoDB – 為現代應用而生)

三、請比較關聯式/非關聯式資料庫優缺點

我在 AWS 介紹 DynamoDB 的官網找到下面這張比較表。讀起來雖然文謅謅的，卻也點出了兩種資料庫的比較方向與基準。

比較項目	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	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： ・單元性要求交易完整執行或完全不執行。 ・一致性要求進行交易時資料就必須符合資料庫結構描述。 ・隔離性要求並行的交易必須分開執行。 ・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。	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。
API	存放和擷取資料的請求是透過符合結構式查詢語言 (SQL) 的查詢進行通訊。這些查詢是由關聯式資料庫剖析和執行。	以物件為基礎的 API 讓應用程式開發人員可輕鬆存放和擷取資料結構。應用程式可透過分區索引鍵查詢鍵值組、欄集，或包含序列化應用程式物件與屬性的半結構化文件。

關聯式資料庫與 NOSQL 資料庫比較表　資料來源：什麼是 NoSQL 資料庫？ AWS

大企業家大業大，只能含蓄地用「比較表」來說明，我們只好自己想辦法，將「比較表」轉譯成「優缺點」。

在深呼吸一口之後，我發現比較這兩種資料庫系統就好像在比較三國時代的兩大軍師，「既生瑜，何生亮」。非關聯式資料庫的崛起，就是為了要解決關

聯式資料庫無法解決的問題，什麼問題呢？不外乎就是兩大轉變：資料類型變多、以及資料儲存變巨量。

資料類型變多起因於資料應用的多元化，例如：搜尋引擎、網頁技術、社群崛起、影音串流等，在關聯式資料庫想盡辦法滿足這些新資料型態的時候，卻遇到了先天性的問題，如何設計出最佳的關聯性 Table？如何設計資料格式不固定的資料庫。可以做到，但會有點複雜，就在這個時候，非關聯性資料庫百花齊放，針對不同的應用領域，發展出不同的資料庫模型(文件、鍵值、欄位、圖形等)。並透過這些新的資料庫模型，達到 Schema-free 的完美境界。這時候，關聯式資料庫的缺點，就是非關聯式資料庫的優點。

資料儲存變巨量起因於網路風行以及儲存設備又大又便宜。面對單一檔案超級大(像影片)以及每天資料超級多的情境，關聯式資料庫雖然可以透過資料庫叢集技術來解決這個問題，但是，即使在不斷擴充硬體設備的情況下，分散式架構的限制仍然無法達到預期的儲存效率。這時候，非關聯性資料庫又跳出來了。非關聯性資料庫在架構上可以相對簡單的水平擴充，解決龐大資料的 CRUD 的效能問題，在巨量資料的處理上擁有較佳的效率、相對低的成本。 這時候，關聯式資料庫的缺點，就是非關聯式資料庫的優點。

當然，那些在關聯性資料庫原本沒有就沒甚麼大問題，且運行幾十年也沒有太大爭議的優點，非關聯性資料也沒有特別去考慮(當然，恐怕也沒辦法全部兼容)，像 ACID 這種對交易(Transaction)高度要求的 DBMS，非關聯性資料庫恐怕就永遠做不到了。這時候，關聯式資料庫的優點，還是關聯式資料庫的優點。

寫了這麼多，直接與間接地解譯了 AWS 的觀點，是不是有一種「既生瑜，何生亮」的感覺！當然，誠如我前面提到的，隨著時間的發展，那些能夠支援兩種資料庫的廠商，將會是這場競賽的最後贏家。

參考資料：

1. Knowledge Base of Relational and NoSQL Database Management Systems https://db-engines.com/en/ranking
2. NoSQL 入門介紹及主要類型資料庫說明 鄭欣如 Cathy Cheng 2020/09/03 https://www.tpisoftware.com/tpu/articleDetails/2016
3. SQL/NoSQL 是什麼？認識資料庫管理系統 DBMS Posted on 2020-03-09 by ALPHA Camp https://tw.alphacamp.co/blog/sql-nosql-database-dbms-introduction
4. System Properties Comparison Cassandra vs. Elasticsearch vs. MongoDB vs. Redis https://db-engines.com/en/system/Cassandra%3BElasticsearch%3BMongoDB%3BRedis
5. MongoDB – 為現代應用而生 歐立威科技 http://www.omniwaresoft.com.tw/mongodb/
6. 什麼是 NoSQL 資料庫？ https://aws.amazon.com/tw/nosql/
7. SQL vs NoSQL: The Differences https://www.kshuang.xyz/doku.php/database:sql_vs_nosql