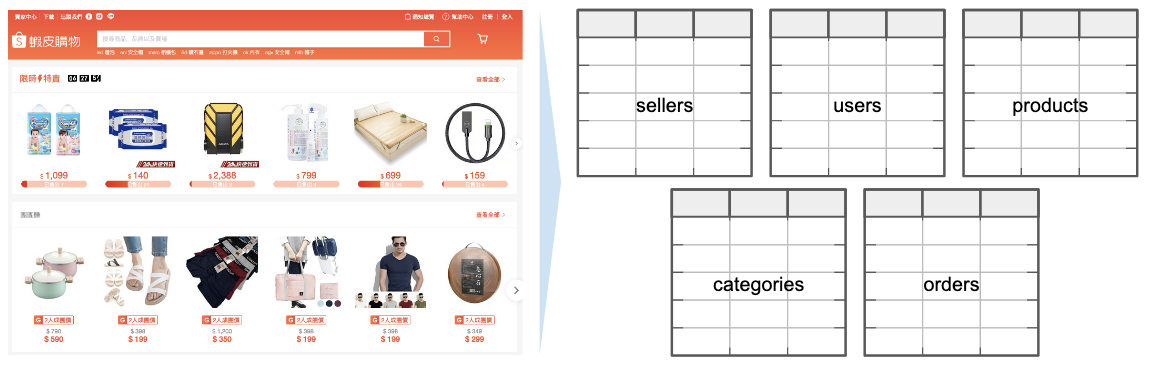
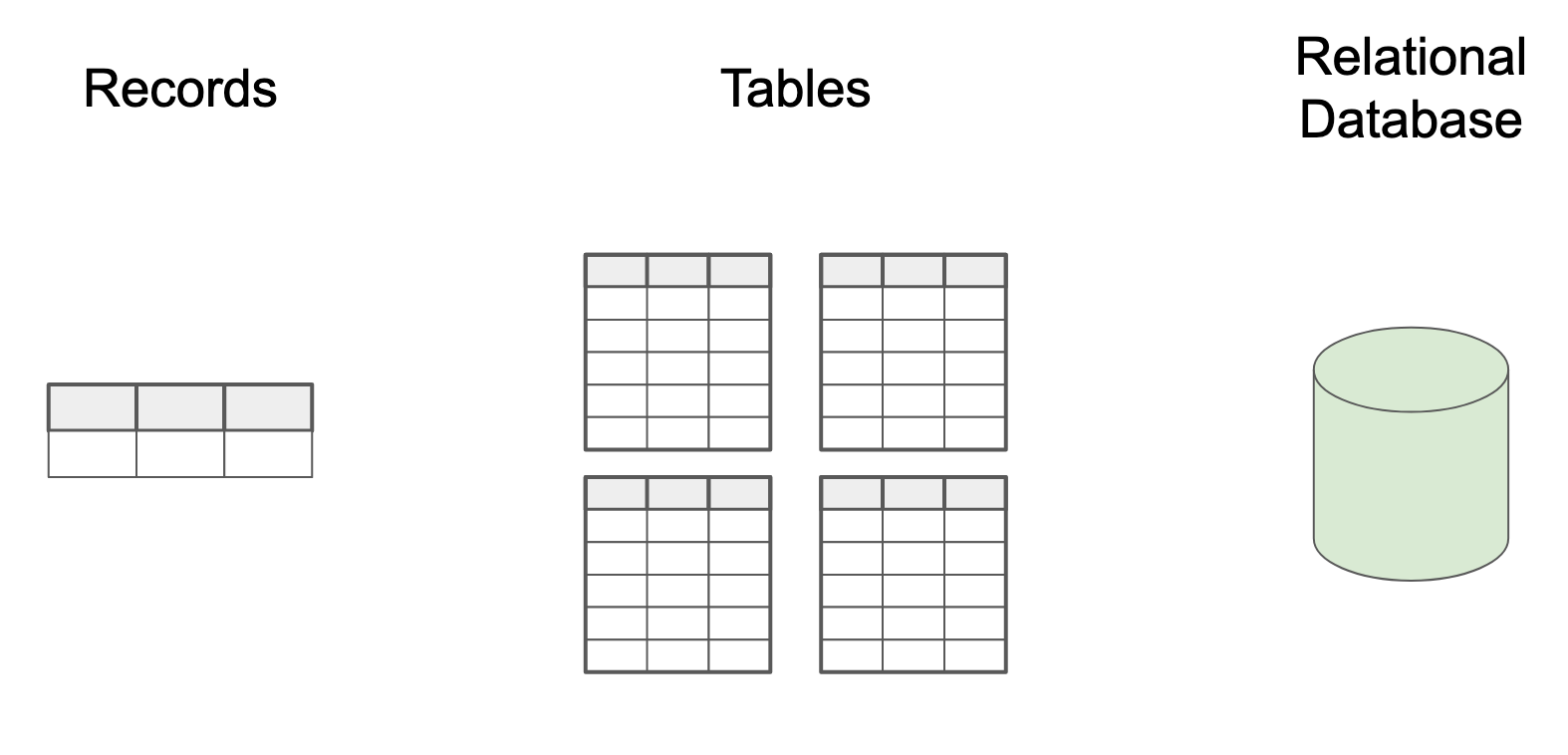
#### 1.關聯式與非關聯式資料庫(SQL、NoSQL)

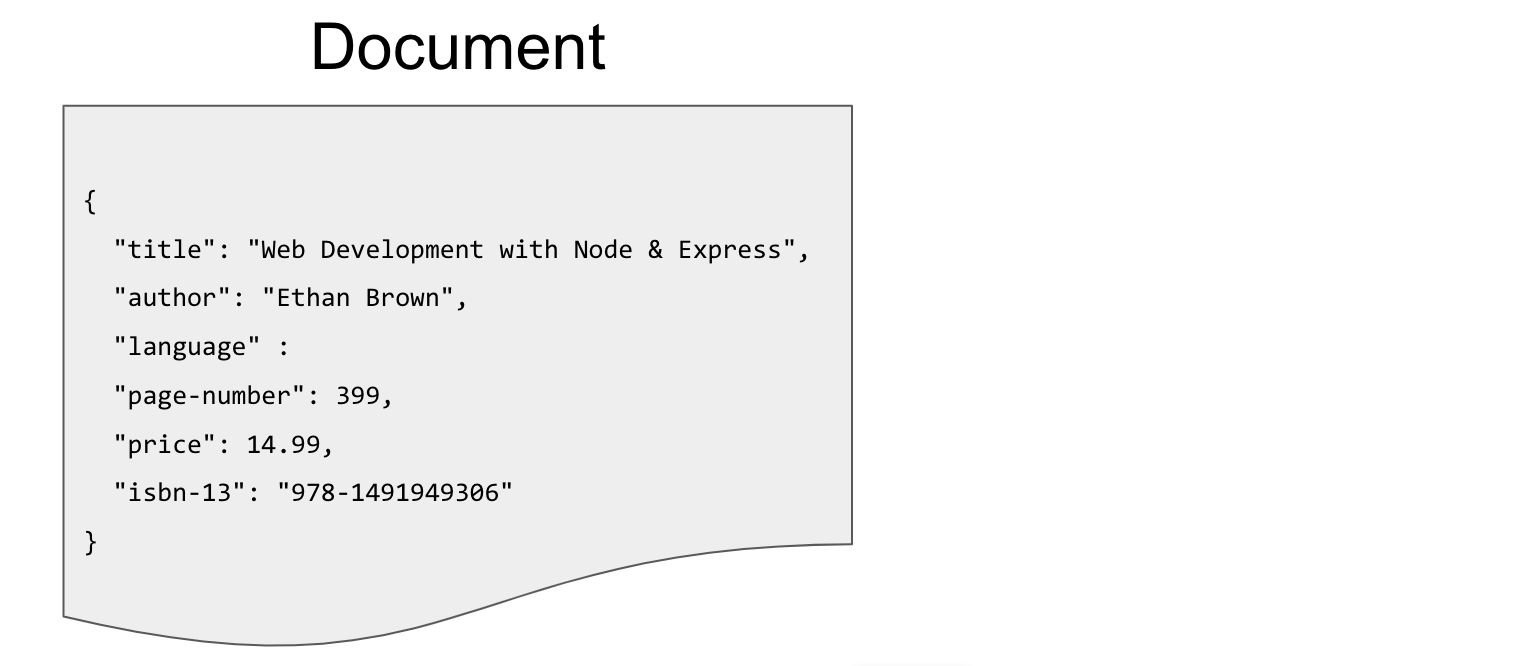
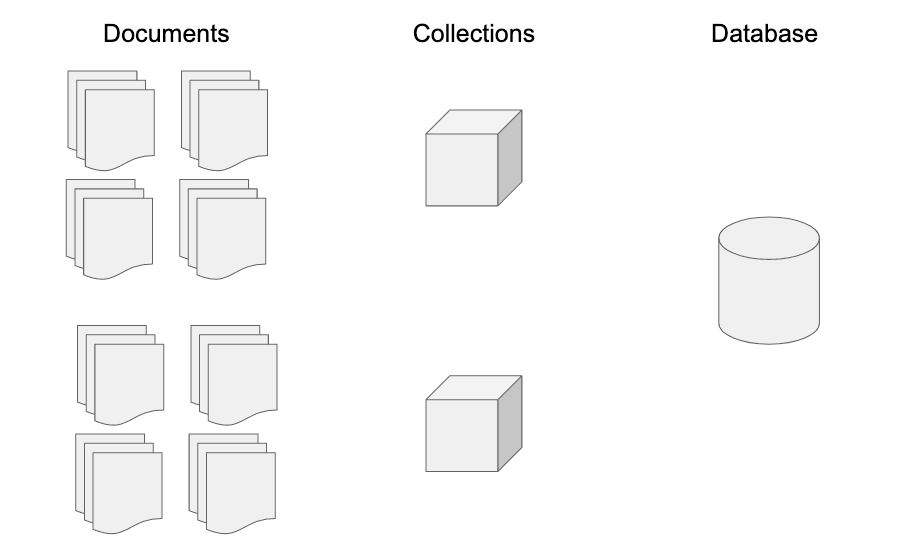
關聯式資料庫就是在**設計好**的框架中，將每一筆獨立的紀錄，找出（建立）可識別的標記後，移除無意義的資料(字串)，拆解資料並整理至設定**好各類(不重疊不遺漏)的屬性欄位**，所以存儲方式會以一至多個表格的方式進行存放．SQL (Structured Query Language 結構化查詢語言) 是一種專門用來管理與查詢關聯式資料庫（Relational Database Menagement System, RDBMS）的程式語言．

舉例來說，在關聯式資料庫裡，每一筆資料都是在 table 中的一個 record，然後再把不同的 table 集合起來，就成為一個關聯式資料庫。所以使用關聯式資料庫的網站，背後都有多個 table，負責紀錄不同的資料。而各個表格間又有明確的關係，譬如消費表格中有賣家,買家與產品資訊．



Not Only SQL (NoSQL)也就是不限定為「關聯式資料庫」的資料庫管理系統的統稱，在操作上，NoSQL 並不支持 SQL 語法與 邏輯。所以，NoSQL 並不一定需要預先設計好的框架來進行存儲整理資料．

舉例來說，文件資料庫是NoSQL資料庫的一種，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(圖1) ，在文件資料庫裡，而資料物件會由「屬性-值」 (attribute-vaule pair) (圖2)，就每一本書，將項目、ISBN、書名、版本編號、作者名稱和 AuthorID 存成單一文件中的屬性，資料針對可以進行橫向擴充進行優化。



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

#### 2.請比較關聯式/非關聯式資料庫優缺點

論起兩種資料庫的優缺點，只能以「文無第一」來形容，就像七言絕句和五言律詩一樣無從比擬哪一種韻體更美，但我們可以效法古人點評，在此時此刻此環境下哪一首作品的意境更為「貼切」來決定使用何種資料庫型態存儲所需要的資料．

|  |  |
| --- | --- |
| SQL | NoSQL |
| 1.資料須有實時一致性(ATM餘額=網銀餘額;購物交易)  2. ACID為必要原則(轉帳、帳號鎖定與併行控管)  3. 資料與資料之間有關聯性  4. 需要複雜的查詢 | 1. ACID為非必要性原則  2. 資料量和格式多元  3. 資料間較無關聯也無關聯之必要  4. 資料格式可擴展性  5.接受各資料間同步需要有更新時間 |

Amazon資料庫的比較表（硬體軟體都AMAZON的，對於兩種資料庫比較相對客觀）



圖片來自 <https://aws.amazon.com/tw/nosql/>

#### 3.詳細介紹任一非關聯式資料庫

Facebook，Twitter，Cisco，eBay，Twitter，Netflix都在用的 **Cassandra**以能夠處理大量資料的分散式架構、完全去中心化為亮點，在處理多個節點之間的巨量資料工作負載，且無需擔心單點故障，且具有高擴展性，新增與刪除節點容易並支援跨區域的叢集架構，每個區域儲存一份完整資料，來提供存取局部性、容錯與備援．

**Cassandra的主要組成**

* **節點(Node)**：Cassandra節點是儲存資料的地方。
* **資料中心(Data center)**：資料中心是相關節點的集合。
* **叢集(Cluster)**：叢集是包含一個或多個資料中心的元件。
* **提交紀錄檔(Commit log)**：在Cassandra中，提交紀錄檔是一個崩潰恢復機制。 每個寫入操作都將寫入提交紀錄檔。
* **儲存表(Mem-table)**：記憶體表是記憶體駐留的資料結構。 提交紀錄檔後，資料將被寫入記憶體表。 有時，對於單列系列，將有多個內容表。
* **SSTable**：當內容達到閾值時，它是從記憶體表重新整理資料的磁碟檔案。
* **布魯姆過濾器(Bloom filter)**：這些只是快速，非確定性的，用於測試元素是否是集合成員的演算法。 它是一種特殊的快取。 每次查詢後都會存取Bloom過濾器。

### **Cassandra的資料模型**

### **群集:** Cassandra資料庫分佈在執行的幾(多)台機器上。 最外層的容器被稱為包含不同節點的群集。 每個節點都包含一個副本，如果發生故障，副本將負責頂上。 Cassandra將節點以環形格式排列在群集中，並為其分配資料。

### **鍵空間:** Keyspace是Cassandra中最外層的數容器，一個Keyspace包含多個欄位 Column families ，一個欄位 中包含多個列值 ，一個 列值中包含多個 column ， column 是Cassandra最基本的資料結構，一個 column 中包含三個值：鍵、值和時間戳。

### **列家族：**列家族置於鍵空間之下。 鍵空間是一個或多個列族的列表的容器，而列族是一組行的容器。 每行包含已排序的列。列家族表示資料的結構。每個鍵空間至少一個並且常有很多列家族。

**SSTables儲存格式優點**

傳統的資料庫的寫入（包括 INSERT、UPDATE、Delete），是一個讀後寫的過程。而 Cassandra 的寫入，是沒有先讀這個動作的，這也是它快的根本原因。Mem-table中的資料每次重新整理都會建立一個新的SSTables，並且對同一份資料的修改也是通過增加一份資料，**而不是修改原有資料來實現的**。 所以一份資料可能存在多個SSTables中，並且需要其他工具來輔助讀操作（這也是為什麼cassandra適用於寫多讀少的場景）。 Cassandra會定期合併SSTables並刪除舊的資料，這種操作叫做壓縮：首先收集同一個row key所有版本的資料，然後對比column資料中的版本號**（時間戳）**Timestamp 的記錄為準，使用最新的版本號合併為一個最新的資料集， 這樣可以增加讀效能，避免在讀操作時掃描所有SSTable表。這樣就不需要去對記錄資源上鎖，以及把需要上鎖的記錄找出來的開銷都省了，快就快在這個地方。

|  |  |
| --- | --- |
| Cassandra | 關聯式資料庫 |
| 在Cassandra中，表是「巢狀鍵值對」列表  (行x列鍵x列值)。 | 在RDBMS中，表是陣列的陣列  (一行x列) |
| 在Cassandra中，keyspace是包含與應用對應的資料的最外層的容器。 | 在RDBMS中，資料庫是包含與應用程式對應的資料的最外層的容器。 |
| 在Cassandra中，表或列族是鍵空間的實體。 | 在RDBMS中，表是資料庫的實體。 |
| 在Cassandra中，行是一個複製單元。 | 在RDBMS中，行是單個/條記錄。 |
| 在Cassandra中，列是一個儲存單元。 | 在RDBMS中，列是表示關係的屬性。 |
| 在Cassandra中，使用集合來表示關係。 | 在RDBMS中，有外來鍵，連線等的概念。 |

資料來源

<https://www.itread01.com/lilyi.html>

<https://www.tw511.com/18/135/3912.html>

<https://docs.google.com/document/d/18vWwuuBywNqGtfnc0q0HNWpUqr2nirm4a2u3wsm8zxo/edit#>