1.簡介關聯式/非關聯式資料庫(NoSQL、NonSQL)

關聯式資料庫是一組資料項目，項目之間具有預先定義的關係。這些項目會整理成由直欄和橫列構成的一組表格。表格會儲存資料庫中所要表示的物件的相關資訊。表格的每一直欄儲存特定類型的資料，而每個欄位儲存某個屬性的實際數值。表格中的橫列代表一個物件或實體的一組相關數值。表格的每一橫列可以用稱為主索引鍵的唯一識別符加以標記，而多個表格之間的橫列可使用外部索引鍵建立關聯。您不需要重新整理資料庫表格，即可用許多不同方法存取這些資料。

關聯式資料庫的重要面向

SQL關聯式資料庫使用 SQL 或結構式查詢語言做為主要的通訊界面。美國國家標準協會 (ANSI) 在 1986 年將 SQL 納入標準。所有常見的關聯式資料庫引擎都支援標準 ANSI SQL，而且有些引擎還提供 ANSI SQL 延伸功能，以支援該引擎的專屬功能。您可以利用 SQL 新增、更新、刪除多個橫列的資料、擷取資料子集供交易處理和分析應用程式使用，以及管理資料庫的所有面向。

資料完整性

資料完整性是指資料整體完備無缺、準確而且一致。關聯式資料庫利用一套限制條件讓資料庫達成資料完整性。其中包括主索引鍵、外部索引鍵、「非空白值」限制條件、「唯一」限制條件、「預設」限制條件、「檢查」限制條件。這些完整性限制條件對表格中的資料實施商業規則，確保資料的準確性和可靠性。除此之外，大部分關聯式資料庫也允許在觸發中嵌入自訂程式碼，當資料庫發生某個動作時便會執行。

交易處理資料庫交易是指透過一連串的操作執行的一或多個 SQL 陳述式，形成單一邏輯工作單位。交易的原則是「全有或全無」，表示整個交易必須做為單一單位完成並寫入資料庫，否則交易的任何個別元件都不應該通過。在關聯式資料庫的術語中，交易的結果包括「認可」(COMMIT) 或「轉返」(ROLLBACK)。處理每一個交易皆採用一致可靠的方式，和其他交易互不相干。

ACID 合規所有資料庫交易必須符合 ACID，即不可分割性 (Atomic)、一致性 (Consistent)、獨立性 (Isolated) 和耐用性 (Durable)，以確保資料完整性。

不可分割性要求交易必須整體成功執行，若是交易有一部分操作失敗，整個交易都會失效。一致性要求做為交易的一部分寫入資料庫的資料，必須遵守所有明定規則以及約束，包括限制條件、級聯、觸發。獨立性是達成並行控制的重要關鍵，可以確保每一個交易都是獨立的。持久性要求在一個交易成功完成後，對資料庫所做的變更都是永久性的。

非關聯式資料庫最初指非SQL或非關係的NoSQL是一種數據庫，它提供了一種用於存儲和檢索數據的機制。該數據是通過關係數據庫中使用的表格關係以外的方式建模的。這樣的數據庫在1960年代末期出現，但直到二十一世紀初迅速普及後才獲得NoSQL的稱號。NoSQL數據庫用於實時Web應用程序和大數據中，並且其使用隨著時間的推移而增長。NoSQL系統有時也稱為NoSQL，以強調它們可能支持類似SQL的查詢語言這一事實。NoSQL數據庫包括設計的簡單性，對機器集群的更簡單的水平擴展和對可用性的更好控制。NoSQL數據庫使用的數據結構與關係數據庫中默認使用的數據結構不同，這使得NoSQL中的某些操作更快。給定NoSQL數據庫的適用性取決於應解決的問題。NoSQL數據庫使用的數據結構有時也被認為比關係數據庫表更靈活。許多NoSQL存儲都犧牲了一致性，以提高可用性，速度和分區容忍度。NoSQL存儲被更多采用的障礙包括使用低級查詢語言，缺乏標準化的接口以及以前在現有關係數據庫上的大量投資。大多數NoSQL商店都缺乏真正的ACID（原子性，一致性，隔離性，持久性）事務，但是一些數據庫（例如MarkLogic，Aerospike，FairCom c-treeACE，Google Spanner（雖然從技術上來說是NewSQL數據庫），Symas LMDB和OrientDB已經使它們成為現實他們設計的核心。大多數NoSQL數據庫提供最終一致性的概念，其中數據庫更改會傳播到所有節點，因此對數據的查詢可能不會立即返回更新的數據，或者可能導致讀取的數據不准確，這就是一個過時的讀取問題。同樣，某些NoSQL系統可能會出現丟失的寫入和其他形式的數據丟失。某些NoSQL系統提供諸如預寫日誌記錄之類的概念，以避免數據丟失。對於跨多個數據庫的分佈式事務處理，數據一致性是一個更大的挑戰。對於NoSQL和關係數據庫而言，這都是困難的。甚至當前的關係數據庫也不允許參照完整性約束跨越數據庫。很少有系統同時維護X / Open XA標準和ACID事務以進行分佈式事務處理。

2. 詳細介紹任一非關聯式資料庫

Google Cloud Bigtable 的介紹與應用

Cloud Bigtable 完全代管的 NoSQL 資料庫服務



Cloud Bigtable 是什麼?

Cloud Bigtable 是一個可以擴展數千列和數十億行的稀疏表 (sparsely populated table)。讓顧客可以儲存數 TB 甚至數 PB 的資料。每行中只有一個值 (value) 會被編入索引 (index) ; 這個值被稱為行鍵 (row key)。Cloud Bigtable 非常適合以極低的延遲存儲大量單鍵數據。它可以在低延遲的情況下支持高讀寫吞吐量，是 MapReduce 操作的理想數據源。

Cloud Bigtable 支援多種不同的客戶端函式庫 (client library)， 其支援包括對 [Apache HBase 1.x Java library](https://hbase.apache.org/) 的擴展。因此，Bigtable 是與現有 Apache 生態系統的開源 Big Data 軟體相整合。

Cloud Bigtable 強大的後端伺服器與自行管理安裝的 HBase 相比，提供了幾個關鍵優勢：

⬩ 超高擴展性：  
Cloud Bigtable 的效能與您的叢集 (cluster) 中的機器數量成正比。自我控管的 HBase 安裝有一個設計瓶頸，在達到某程度的 QPS 後效能便無法再提升。Cloud Bigtable 則沒有這個瓶頸。所以客戶可以透過增加機器數量來擴展你的叢集以處理更多的查詢。  
⬩ 易於管理：  
Cloud Bigtable 可以透明化地處理升級和重新啟動，並自動保持數據的高持久性。不需再管理叢集的主結點 (masters)、地區 (regions)，集群 (clusters) 或節點 (nodes); 你只需要設計自己的表格模式，其餘的問題Cloud Bigtable 會為你處理。  
⬩ 叢集調整不需停機：  
您可以擴大 Cloud Bigtable 數幾個小時以處理龐大的負載量，然後再次縮小叢集的大小 —- 所有這些舉動都不需停機。更改叢集大小後，Cloud Bigtable 在負載下通常只需要幾分鐘就可以平衡叢集中所有節點的性能。

Cloud Bigtable 的優勢

Cloud Bigtable 非常適合需要對非結構化鍵/值數據有高吞吐量和高擴展性的應用。其中，每個值 (key/value data) 通常不大於 10 MB。Cloud Bigtable 也是批量的 MapReduce 操作、串流處理/分析和機器學習應用優良的儲存引擎。

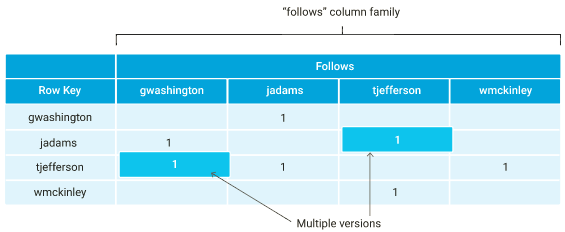
您可以使用 Cloud Bigtable 來儲存和查詢以下所有類型的數據：  
⬩ 行銷數據：如採購歷史和顧客偏好。  
⬩ 財務數據：如交易歷史、股票價格和匯率。  
⬩ IoT數據：如電錶和家用電器的使用情況報告。  
⬩ 時間序列數據：例如在多個伺服器的 CPU 和 memory 使用狀況。

Cloud Bigtable 儲存模式

Cloud Bigtable 將數據儲存在大規模可擴展的表中，每個表都是一個排序過的鍵/值對應表(key/value map)。表是由列 row（每個行通常描述單個實體）和行 column（包含每行的單個值）組成。每列由單個 row key 索引，而彼此有關的行通常歸為一個行族 (column family)。每行的識別由 column family 和 column qualifier 來標識。

每個行/列交叉點可以在不同時間點涵蓋多個單元格 (cells) ，提供存儲數據隨著時間的變化的記錄。Cloud Bigtable 表格很稀疏 ; 如果一個單元格裡沒有任何數據，它就不佔用任何儲存空間。

例如，假設您正在為美國總統建立一個社交網絡 – 暫且稱它為 Prezzy。每個總統都可以關注其他總統的發文動態。下圖顯示的 Cloud Bigtable 表格可以看出在 Prezzy 上總統間各自追隨的跡象。



在這個例子中可以注意到幾點：  
⬩ 這個表格包含一個 column family –  follows family。 Column family包含多個column qualifiers。  
⬩ Column qualifiers 被用作數據。這種設計利用了 Cloud Bigtable 表格的稀疏性和可以即時添加新行限定符的特性。  
⬩ 用戶名用作列鍵 (Row key)。假設用戶名在字母表中均勻分佈，訪問的數據會在整個表中平均分布。有關 Bigtable 如何處理不均勻負載，可以參考 “ [負載平衡](https://cloud.google.com/bigtable/docs/overview#load-balancing) ”獲得更多資訊。也可以參考 “ [選擇列鍵](https://cloud.google.com/bigtable/docs/schema-design#row-keys) ” 學到更多挑選列鍵的小撇步。

3. 請比較關聯式/非關聯式資料庫優缺點

(1) 關聯式資料庫

優勢：以完善的關聯代數理論作為基礎，有嚴格的標準，支援交易

ACID四特性，借助索引機制可以實現高效的查詢，技術成熟

，有專業公司的技術支援

劣勢：可擴展性較差(尤其是橫向擴展性)，無法較好地支援海量資料

儲存，資料模型過於死板、無法較好地支援Web2.0應用，交

易機制影響了系統的整體性能等

(2) NoSQL資料庫

優勢：可以支援超大規模資料儲存，靈活的資料模型可以很好地支

援Web2.0應用，具有強大的橫向擴展能力等

劣勢：缺乏數學理論基礎，複雜查詢性能不高，大都不能實現交易

強一致性，很難實現資料完整性，企業關鍵任務不能採用。

技術尚不成熟，缺乏專業團隊的技術支援，維護較困難等

