# Tổng quan về NoSQL:

NoSQL (Non – only SQL) có thể hiểu một cách đơn giản đó là không chỉ có SQL mà có thể có nhiều cơ chế lưu trữ khác được sử dụng dựa trên yêu cầu phần mềm.

NoSQL Database ra đời, giải quyết được những khuyết điểm của RDBMS:

* Dữ liệu trong NoSQL DB được lưu dưới dạng document, object. Truy vấn dễ dàng và nhanh hơn RDBMS nhiều.
* NoSQL có thể làm việc với dữ liệu dạng không có cấu trúc.
* Việc đổi cấu trúc dữ liệu (Thêm, xóa trường hoặc bảng) rất dễ dàng và nhanh gọn trong NoSQL.
* Vì không đặt nặng tính ACID của transactions và tính nhất quán của dữ liệu, NoSQL DB có thể mở rộng, chạy trên nhiều máy một cách dễ dàng.

Ta có thể chia NoSQL thành 4 loại:

1. Key-Value Database:

Dữ liệu được lưu dưới dạng key-value. Để truy vấn dữ liệu, ta dựa vào key để lấy ra value.

1. Document Database:

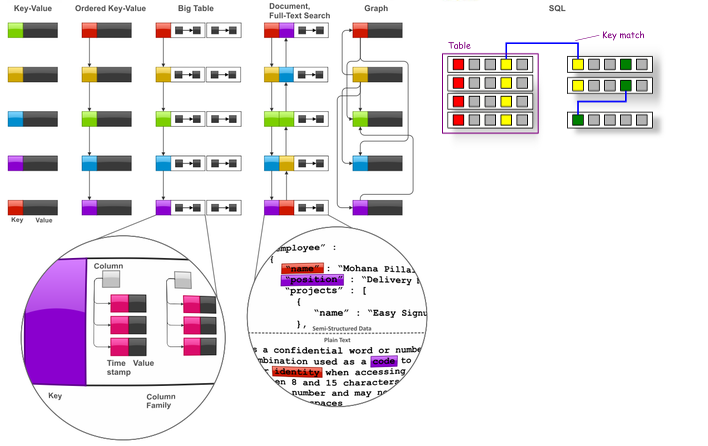
Mỗi đối tượng được lưu trữ trong database dưới dạng document (BJON/ JSON/XML)

1. Column-Family Database:

Dữ liệu được lưu dưới dạng các cột. Mỗi hàng sẽ có một key/id riêng. Số lượng các cột của hàng trong một bảng có thể khác nhau

1. Graph Database

Dữ liệu được lưu dưới dạng các node. Mỗi node sẽ có một nhãn (label) và một số thuộc tính ( properties). Các node sẽ được nối với nhau bằng các relationship.



Cassandra thuộc loại Colunm-Family Database.

# Cassandra

1. **Cassandra là gì? (Giới thiệu tổng quát)**

* ***Cassandra***: Một quản trị hệ cơ sở dữ liệu phân tán mã nguồn mở, được thiết kế để xử lý một khối lượng lớn dữ liệu giàn trải trên nhiều node mà vẫn đảm bảo tính sẵn sàng cao (Highly Availability), khả năng mở rộng hay thu giảm số node linh hoạt (Elastic Scalability) và chấp nhận một số lỗi (Fault Tolerant).
* ***Thiết kế của Cassandra***: Thiết kế phân tán trên hàng ngàn máy chủ mà không có bất cứ điểm chết tập trung nào. Cassandra có thiết kế dựa trên kiến trúc mạng ngang hàng (Peer - to - Peer), tất cả các nút máy chủ trong hệ thống đều có vai trò như nhau và không có nút máy chủ nào đóng vai trò là máy chủ trung tâm.
* ***Các nút máy chủ***:
* Các nút máy chủ của Cassandra là độc lập và tham gia vào kết nối với các nút máy chủ khác trong hệ thống.
* Mỗi nút đều có thể xử lý các thao tác ghi và đọc dữ liệu, không phân biệt là dữ liệu được lưu trữ một cách vật lý trên máy chủ nào trong hệ thống.
* Khi một nút trong hệ thống bị hỏng và dừng hoạt động, các thao tác đọc ghi dữ liệu có thể được xử lý bởi các nút khác trong hệ thống => Quá trình này giúp ứng dụng không bị gián đoạn khi đang chạy và cho phép ẩn đi sự hư hỏng của nút trong hệ thống.

1. **Ưu điểm:**

* Chấp nhận sai sót ( fault tolerant): dữ liệu tự động được sao chép thành nhiều bản trên các server. Nếu một server bị hỏng, dữ liệu vẫn có thể được truy xuất từ server khác.
* Tính co giãn ( Elastic): khả năng đọc và viết tăng tuyến tính theo số lượng máy được thêm vào cụm máy của bạn, mà không có sự gián đoạn đến ứng dụng bạn đang chạy.
* Tính hướng cột ( Column – Oriented): có thể thêm tùy ý số cột mà không cần khai báo trước.
* Có thể điều chỉnh sự nhất quán: có thể kiểm soát số lượng bản sao để chặn với tất cả các bản cập nhật bằng cách thiết lập:
  + *Replication factor:* quyết định chi phí mà bạn muốn bỏ ra với hiệu năng để tăng thêm tính trong suốt
  + *ConsitencyLevel:* là thiết lập mà client có thể chỉ rõ ở tất cả các giao dịch và nó cho phép quyết định số lượng bản sao trong cluster.
* Kiến trúc ngang hàng: Tất cả các node trong cassandra cluster có vai trò tương tự nhau, đều đảm nhận việc đọc và ghi dữ liệu, Cassandra được đánh giá nổi bật là dựa trên yếu tố này, xây dựng kiến trúc ngang hàng sẽ làm giảm nguy cơ bị thắt cổ chai (bottlenect)
* Tính sẵn sàng cao: Khi thực hiện tác vụ đọc/ghi, Cassandra có thể thực hiện trên bản sao gần nhất hoặc trên tất cả các bản sao. Điều này phụ thuộc vào thông số ConsitencyLevel do bạn thiết lập.
* Tính hiệu năng cao (Highly Performance): Cassandra đã được chứng mình là hoạt động đặc biệt tốt với yêu cầu tải nặng. Nó luôn có thể hiện thị rất nhanh chóng để ghi mỗi giây trên một máy trạm cơ bản. Khi bạn bổ sung thêm các máy chủ, bạn có thể duy trì tất cả các tính chất mong muốn mà hiệu suất không hề bị giảm sút.

1. **Những trường hợp nên và không nên sử dụng Cassandra**

* Bởi vì Cassandra có khả năng xử lý tốt một khối lượng lớn dữ liệu nên nó được sử dụng trong rất nhiều các ứng dụng khác nhau: những ứng dụng phù hợp với xử lý về mặt tốc độ, việc kết hợp với các công nghệ khác trong việc xử lý thời gian thực của BigData, phù hợp cho các ứng dụng đảm bảo về không mất mát dữ liệu, thậm chí khi toàn bộ trung tâm dữ liệu bị sụp.
* Cassandra không nên được dùng cho những dữ liều cần xử lý, tính toán nhiều trên storage, nó không hỗ trợ các hàm Sum, Group, Join, Max, Min và bất kỳ hàm nào khác developer muốn sử dụng để tính toán dữ liệu khi query. Cassandra cũng không hỗ trợ build-in full text search, B-tree index hoặc các hàm trợ giúp tính toán.