Thế nào là NoSQL, sự khác nhau giữa NoSQL và MySQL, các dạng NoSQL, chi tiết từng loại, lấy ví dụ từng loại.

1. NoSQL là gì?

NoSQL(Not Only SQL) là một loại hệ thống quản trị cơ sở dữ liệu khác với các hệ quản trị cơ sở dữ liệu quan hệ truyền thống như MySQL. NoSQL được thiết kế để lưu trữ và quản lý dữ liệu có cấu trúc không cố định, dữ liệu phi cấu trúc, hoặc dữ liệu bán cấu trúc.

* Dữ liệu không cố định: là bất kỳ dữ liệu nào không tuân theo một mô hình dữ liệu hoặc cấu trúc cố định  
  Ví dụ: Email, tài liệu văn bản(Word, PDF), hình ảnh, video, âm thanh, bài đăng trên MXH
* Dữ liệu phi cấu trúc: là dữ liệu không có cấu trúc hoặc tổ chức rõ ràng  
  Ví dụ: Văn bản thô (như nội dung email hoặc bài đăng trên MXH mà không có siêu dữ liệu), hình ảnh, video, âm thanh.
* Dữ liệu bán cấu trúc: là dữ liệu không tuân theo một mô hình hay cấu trúc cứng nhắc nhưng vẫn có một số cấu trúc tổ chức để dễ dàng phân tích và hiểu( có thể chứ thẻ hoặc nhãn để xác định phần tử)  
  Ví dụ: JSON, XML, Email

1. Các dạng NoSQL
   1. Cơ sở dữ liệu tài liệu (Document database):
      1. Đặc điểm:
         1. Lưu trữ dữ liệu dưới dạng tài liệu, thường được sử dụng định dạng JSON, BSON, XML, hoặc YAML.
         2. Mỗi tài liệu là một đơn vị độc lập và có thể chứa dữ liệu có cấu trúc phức tạp, bao gồm các mảng và các tài liệu lồng nhau.
      2. Ưu điểm:
         1. Linh hoạt: không yêu cầu một lược đồ cố định, giúp dẽ dàng thay đổi và mở rộng cấu trúc dữ liệu
         2. Dễ dàng mở rộng: hỗ trợ mở rộng theo chiều ngang, có thể thêm nhiều máy chủ để tăng dung lượng lưu trữ và khả năng xử lý
         3. Truy vấn mạnh mẽ: hỗ trợ truy vấn phức tạp và tìm kiếm toàn văn
      3. Ví dụ: MongoDB, CouchDB
   2. Cơ sở dữ liệu khóa-giá trị (Key-Value store):
      1. Đặc điểm:
         1. Lưu trữ dữ liệu dưới dạng cặp khóa-giá trị. Mỗi khóa là duy nhất và liên kết với một giá trị
         2. Không có cấu trúc cố định cho giá trị, có thể là văn bản, số hoặc dữ liệu phức tạp
      2. Ưu điểm:
         1. Đơn giản cấu trúc giúp truy cập và thao tác dữ liệu nhanh chóng
         2. Tối ưu cho các tác vụ truy cập nhanh, đặc biệt là khi truy cập dữ liệu theo khóa
         3. Hỗ trợ mở rộng theo chiều ngang, thích hợp cho các ứng dụng phân tán
      3. Ví dụ: Redis, Amazon DynamoDB
   3. Cơ sở dữ liệu Cột (Column-Family database):
      1. Đặc điểm:
         1. Lưu trữ dữ liệu dưới dạng các cột và bảng. Các cột có thể được nhóm lại thành các gia đình cột
         2. Dữ liệu được phân phối trên nhiều máy chủ, tói ưu hóa cho việc xử lý khối lượng lớn dữ liệu
      2. Ưu điểm:
         1. Hiệu suất tốt khi xử lý và truy vấn khối lượng lớn dữ liệu
         2. Hỗ trợ mở rộng theo chiều ngang, dữ liệu được phân phối trên nhiều máy chủ
         3. Thiết kế để chống lại các sự cố phần cứng và phần mềm, đảm bảo tính sẵn sàng cao
      3. Ví dụ: Apache Cassandra, Hbase
   4. Cơ sở dữ liệu đồ thị (Graph database):
      1. Đặc điểm:
         1. Lưu trữ dữ liệu dưới dạng các đỉnh (nodes) và các cạnh (edges). Các định đại diện cho các thực thể, và các cạnh đại diện cho các mối quan hệ giữa các thực thể
         2. Thích hợp cho các ứng dụng yêu cầu quản lý các mối quan hệ phức tạp
      2. Ưu điểm:
         1. Tối ưu cho các truy vấn liên quan đến mối quan hệ và kết nối
         2. Dễ dàng biểu diễn và truy vấn các quan hệ phức tạp
         3. Phù hợp cho các ứng dụng như mạng xã hội, hệ thống đề xuất, và quản lý chuỗi cung ứng
      3. Ví dụ: Neo4j, OrientDB
   5. Mở rộng dữ liệu (data scaling):

-Có hai cách để mở rộng quy mô cơ sở dữ liệu: mở rộng theo chiều ngang (horizontal scaling) và mở rộng theo chiều dọc (vertical scaling)

* + 1. Vertical scaling:  
       -Vertical scaling hay còn biết đến như là scaling up, là cách mở rộng bằng cách tăng thêm năng lượng (CPU, Ram, DISK, etc. ) cho một bộ máy đang có. Ưu điểm lớn nhất của Vertical Scaling là giảm thiểu chi phí phần mềm, cách thực hiện dễ dàng, phí cấp phép thấp, chi phí tiện ích (điện, làm mát…) thấp hơn nhiều so với Horizontal Scaling.  
       - Tuy nhiên Vertical scaling có khá nhiều hạn chế:  
       \* Vertical có giới hạn dữ liệu nhất định vì không thể thêm vô hạn CPU và bọ nhớ vào một đơn máy chủ. Chỉ có thể thêm tài nguyên tối đa bằng kích thước của máy chủ.

\* Vertical scaling không có bộ chuyển đổi dự phòng, nếu một máy chủ sập thì website hay app đều sẽ sập theo.

* + 1. Horizontal scaling:  
       - Horizontal scaling hay còn gọi là scaling out, kỹ thuật này chia cơ sở dữ liệu lớn trở nên nhỏ hơn, quản lý các bộ phận (shards) dễ dàng hơn. Phương pháp này có nhiều ưu điểm như: các công cụ có thể mở rộng dễ dàng, khả năng chịu lỗi cao, dễ dàng nâng cấp, tuy nhiên có một số nhược điểm như thiết kế về cấu trúc rất phức tạp, phí cấp phép cao, chi phí tiện ích cao.

1. Sự khác nhau giữa NoSQL và SQL:

|  |  |
| --- | --- |
| SQL | NoSQL |
| Cấu trúc dữ liệu:  - Dữ liệu được tổ chức thành dạng bảng (tables), hàng (rows), cột (columns) nhất định và phải tuân theo một cấu trúc nhất định.  \*Ngôn ngữ truy vấn:  - Sử dụng ngôn ngữ truy vấn có cấu trúc SQL để thao tác.  \*Tính toàn vẹn dữ liệu:  - Hỗ trợ các ràng buộc dữ liệu như khóa chính, khóa ngoại, ràng buộc duy nhất, và kiểm tra.  \*Tính nhất quán:  - Tuân theo các thuộc tính ACID (Atomic, Consistency, Isolation, Durability) để đảm bảo tính nhất quán và tin cậy của dữ liệu. | \*Cấu trúc dữ liệu:  - Dữ liệu tổ chức theo nhiều cách khác nhau bao gồm tài liệu (document), khóa- giá trị (key – value), cột (column-family), và đồ thị (graph), cấu trúc linh hoạt và không tuân theo yêu cầu cố định.  \*Ngôn ngữ truy vấn:  Không có ngôn ngữ truy vấn tiêu chuẩn. Các hệ thống NoSQL có thể sử dụng các NN truy vấn riêng.  \*Tính toàn vẹn dữ liệu:  - Ít hỗ trợ các ràng buộc dữ liệu mạnh mẽ như SQL, thay vào đó, tập trung vào tính linh hoạt và hiệu suất.  \*Tính nhất quán:  - Tuân theo các thuộc tính BASE (Basically Available, Soft state, Eventually consistent). Dữ liệu đạt tính nhất quán ở thời điểm cuối cùng. |

* Tính nhất quán (consistency): bất kỳ dữ liệu nào được ghi vào cơ sở dữ liệu cũng đều phải tuân theo tất cả các quy tắc định sẵn.
* Thuộc tính ACID: viết tắt của Atomic, Consistency, Isolation, Durability, đây là bốn thuộc tính quan trọng của một hệ cơ sở dữ liệu khi xử lý bất kỳ giao dịch nào.
  + Tính nguyên tử (Atomicity): một giao dịch có nhiều thao tác khác nhau thì hoặc là toàn bộ các thao tác hoàn thành hoặc là không một thao tác nào được hoàn thành.
  + Tính nhất quán (Consistency): một giao dịch hoặc là sẽ tạo ra một trạng thái mới và hợp lệ cho dữ liệu, hoặc trong trường hợp có lỗi sẽ chuyển toàn bộ dữ liệu về trạng thái trước khi thực thi giao dịch
  + Tính cô lập (Isolation): một giao dịch đang thực thi và chưa được xác nhận phải bảo đảm tách biệt khỏi các giao dịch khác
  + Tính bền vững (Durability): dữ liệu được xác nhận sẽ được hệ thống lưu lại sao cho ngay cả trong trường hợp hỏng hóc hoặc có lỗi hệ thống, dữ liệu vẫn đảm bảo trong trạng thái chuẩn xác.
* Thuộc tính BASE:
  + Tính sẵn sàng ở mức cơ bản (Basically available): người dùng luôn có khả năng truy cập đồng thời cơ sở dữ liệu. Người dùng không cần phải đợi người khác hoàn thành giao dịch rồi mới cập nhật bản ghi
  + Trạng thái mềm (Soft state): trạng thái của hệ thống có thể thay đổi theo thời gian, và không phải lúc nào cũng nhất quán ngay lập tức. Trạng thái này mô tả trạng thái chuyển tiếp của bản ghi khi một số ứng dụng cập nhật bản ghi đồng thời. Giá trị của bản ghi chỉ được hoàn tất toàn bộ sau khi hoàn thành tất cả giao dịch
  + Nhất quán toàn bộ (Eventual consistency): nghĩa là bản ghi sẽ đạt được tính nhất quán khi tất cả các bản cập nhật đồng thời đã được hoàn thành. Tại thời điểm này ứng dụng truy vấn bản ghi sẽ thấy cùng một giá trị.
* SQL đảm bảo các thuộc tính ACID để cung cấp tính toàn vẹn, nhất quán và an toàn của dữ liệu, phù hợp với các ứng dụng yêu cầu dữ liệu chính xác và nhất quán.
* NoSQL tuân thủ các nguyên tắc BASE để tối ưu hóa tính sẵn sàng và khả năng mở rộng, phù hợp với các ứng dụng yêu cầu hiệu suất cao và khả năng chịu lỗi.

1. Các kiểu nhất quán:
   1. Nhất quán mạnh (Strong Consistency):

Là một mô hình bộ nhớ bảo thủ không cho phép bất kỳ lệnh nào sắ xếp lại trên mỗi lòi.điều này

* 1. Nhất quán tuần tự (Sequential Consistency):

Các hoạt động của mọi tiến trình được nhìn thấy bởi tất cả các tiến trình theo cùng thứ tự,mặc dù không nhất thiết là theo thứ tự thời gian.

* 1. Nhất quán nhân quả (Causal Consistency):

Nếu một thao tác ảnh hưởng đến thao tác khác, các thao tác đó phải được nhìn thấy theo thứ tự nhân quả.

* 1. Nhất quán cuối cùng(Eventual consistency):
* SQL đảm bảo tính nhất quán mạnh (ACID)
* NoSQL: Không phải tất cả đều đạt nhất quán cuối cùng. Nhiều hệ thống NoSQL sử dụng nhất quán cuối cùng, nhưng cũng có các hệ thống hoặc cấu hình cho phép đạt được nhất quán mạnh hoặc nhất quán có điều kiện.