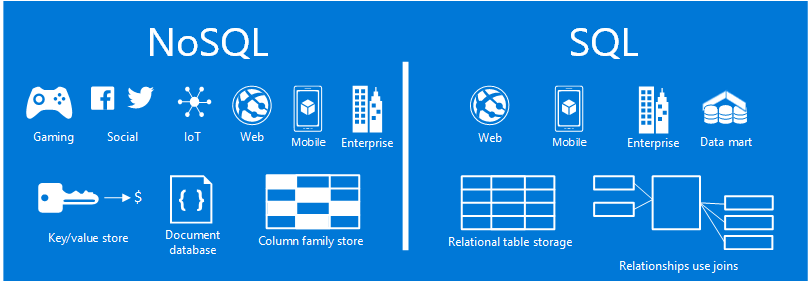
# Giới thiệu về NoSQL database

Cơ Sở dữ liệu quan hệ được thiết kế cho những mô hình cần đảm bảo tính chặt chẽ và dữ liệu không quá lớn, trong khi các dịch vụ mạng xã hội lại có một lượng lớn dữ liệu và được cập nhật liên tục do số lượng người dùng quá nhiều. Do đó cơ sở dữ liệu NOSQL sinh ra để giải quyết các vấn đề mà RDBMS (A relational database management system) đã bộc lộ những yếu kém như: tốc độ thực thi, khả năng lưu trữ, các nghiệp vụ phức tạp (như phân trang, đánh chỉ mục …) Nhờ vậy giải pháp sử dụng cơ sở dữ liệu NOSQL sẽ mang lại một chi phí thấp hơn nếu so sánh với RDBMS truyền thống.



## Định Nghĩa

NoSQL là một xu hướng cơ sở dữ liệu mà không dùng dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm. NOSQL có nghĩa là Non-Relational – không ràng buộc. Tuy nhên, thuật ngữ đó ít phổ biến hơn và ngày nay người ta thường dịch thành Not Only SQL – không chỉ SQL.

NOSQL được xem như thế hệ database kế tiếp của RDBMS, là một thế hệ cơ sở dữ liệu Non-relational (không ràng buộc), distributed (phân tán), open source (mã nguồn mở), horizontal scalable (khả năng mở rộng theo chiều ngang), có độ chịu tải, lỗi cao.

## Một Số Thuật Ngữ Trong NoSQL

* Tính ràng buộc (Relational): Thuật ngữ để mô tả tính ràng buộc giữa các bảng trong cơ sở dữ liệu quan hệ (MySQL, SqlSever, PostgreSQL…).
* Không ràng buộc (Non-relational): Dữ liệu sẽ không có các ràng buộc giữa các bảng nữa, mà dữ liệu sẽ ở dạng Json hoặc Bson.
* Khả năng mở rộng (High Scalability): Khi hệ thống lớn lên, ta có thể bổ sung thêm các Nodes mới, các Sever mới để chia tải hoặc chia dữ liệu. Để hệ thống không bị quá tải.
* Khả năng mở rộng theo chiều dọc (Vertical Scalable / Scale Up): Là việc nâng cấp phần cứng hệ thống bằng việc nâng cấp RAM, hay bộ nhớ.
* Khả năng mở rộng theo chiều ngang (Horizontal Scalable / Scale-Out): Là việc bổ sung phần cứng tránh sự hoạt động quá tải của hệ thống.
* Phân tán dữ liệu (Distributed Data): Là việc mô tả dữ liệu được phân tán ở các địa điểm khác nhau.
* Triển khai linh hoạt (Deployment Flexibilitty): Dễ dàng mở rộng thêm các nodes (Severs) mà không ảnh hưởng đến hoạt động của hệ thống.
* Tính sẵn sàng (High Availability): Hệ thống sẽ không bị ảnh hưởng khi một node bị trục trặc.
* Nhất quán cuối (Eventual Consistency): Khi ta đưa một dữ liệu mới vào một node của hệ thống, dữ lệu sẽ được lan truyền sang các node khác của hệ thống và cuối cùng tất cả node sẽ được đồng bộ.
* Lưu trữ tốt (Durability).

**Khi làm việc với Nosql ta sẽ gặp một số khái niệm sau:**

* FIELDS: Tương đương với khái niệm Columns trong SQL
* Documents: Thay thế khái niệm Rows trong SQL. Đây cũng chính là khái niệm làm nên sự khác biệt giữa NOSQL và SQL, 1 document chưa số cột (fields) không cố định trong khi 1 row thì số cột (columns) là định sẵn trước.
* Collection: Tương đương với khái niệm table trong SQL. Một Collection là tập hợp các documents.
* Key-Value: Cặp khóa - giá trị được dùng để lưu trữ dữ liệu trong NOSQL
* Cursor: Tạm dịch là con trỏ, sử dụng cursor để lấy dữ liệu từ database.

Trong các hệ cơ sở dữ liệu quan hệ, các cột được định nghĩa theo bảng, còn với hệ cơ sở dữ liệu không ràng buộc, các cột được định nghĩa ở mỗi document. Bởi thế, các document quản lý gần như tất cả, các collection không cần quản lý chặt chẽ những gì đang xảy ra trong nó nữa.

|  |  |
| --- | --- |
| RDBMS | NOSQL |
| Columns  Row  Table  Schema | Fields  Documents  Collection  Free Schema |

## Đặc Điểm

- NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin.

- Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau.

- Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu.

- Có hiệu suất cao (High performance) và tính sẵn sàng cao (High availability).

## So Sánh Giữa SQL và NoSQL

|  |  |  |
| --- | --- | --- |
| Tính Năng | SQL | NoSQL |
| Hiệu xuất | Kém hơn NoSQL vì khi truy vấn nó phải tính toán, kiểm tra và xử lý mối quan hệ các bảng | Tốt hơn SQL do nó bỏ qua xử lý các ràng buộc giữa các bảng |
| Mở rộng theo chiều ngang | Có thể thực hiện được nhưng sẽ rất phức tạp nếu đã tồn tại dữ liệu trong database | Mở rộng dễ dàng mà không ảnh hưởng đến hệ thống |
| Tốc độ Đọc/Ghi | Kém hơn NoSQL do phải đảm bảo tính ràng buộc giữa các bảng | Tốc độ nhanh hơn SQL nhiều vì bỏ qua cơ chế ràng buộc giữa các bảng  Vì dữ liệu được lưu trong RAM ,sau đó đẩy xuống ổ cứng và cuối cùng là nhất quán cuối |
| Phần cứng | Đòi hỏi phần cứng cao | Không đòi hỏi cao về phần cứng |
| Thay đổi số node trong hệ thống | Vì tính nhất quán về dữ liệu nên khi thêm hoặc xóa một node thì hệ thống cần phải shutdown trong một khoảng thời gian | Vì tính nhất quán cuối nên không cần phải shutdown hệ thống |
| Truy vấn và Báo cáo | Dễ dàng, sử dụng ngôn ngữ SQL Query để truy vấn trực tiếp dữ liệu từ Database hoặc dùng công cụ hỗ trợ để lấy báo cáo | Việc lấy báo cáo dữ liệu trực tiếp từ NoSQL chưa được hỗ trợ tốt, thực hiện chủ yếu thông qua giao diện ứng dụng |
| Mở rộng dữ liệu | Khi muốn bổ sung thêm cột cho một bảng, cần phải khai báo trước | Không cần khai báo trước khi muốn bổ sung dữ liệu |
| Ứng Dụng | Sử dụng để xây dựng những hệ thống có quan hệ chặt chẽ và cần tính đồng nhất về dữ liệu như: tài chính, ngân hàng, chứng khoán… | Sử dụng xây dựng những hệ thống lưu trữ thông tin lớn, không quá quan trọng trong vấn đề đồng nhất dữ liệu trong 1 thời gian nhất định. |
| Ví dụ | **Facebook**: Sử dụng Hive(một dạng biến thể của SQL)  **Twitter:** sử dụng MySQL  **Stack Overflow**: sử dụng SQL Sever  **Youtube**: sử dụng MySQL  **Wikipedia:** sử dụng MySQL | **Facebook**: Sử dụng Cassandra(cơ sở dữ liệu đa chiều, lưu trữ phân tán)  **Twitter:** sử dụng cassandra |

## Ưu Và Nhược Điểm Của Cơ Sở Dữ Liệu NoSQL

**Hiệu suất hoạt động cao:** NoSQL có hiệu suất hoạt động cao, lưu trữ lượng lớn dữ liệu để đáp ứng nhu cầu lưu trữ ngày càng tăng hiện nay. Tuy nhiên để đạt được này, cần phải loại bỏ đi một số thứ như: các ràng buộc giữa các bảng, tính nhất quán dữ liệu, ngôn ngữ truy vấn SQL. Đồng thời NoSQL có một số cải tiến mới như sử dụng tốt index, khả năng phân tán dễ dàng đã giúp NoSQL có một hiệu suất hoạt động cao.

**Khả năng phân trang:** phân trang trong cơ sở dữ liệu khá khó khăn khi không có một phương pháp chính thống nào để phục vụ cho việc này. Người lập trình phải dùng các phương pháp khác nhau để có thể lấy đúng số item cần lấy.

**NoSQL là nguồn mở:** các sản phẩm nguồn mở đưa ra cho những người phát triển với nhiều lợi ích to lớn, ví dụ như việc sử dụng miễn phí. Ngoài ra, phần mềm mã nguồn mở có xu hướng tin cậy hơn, an ninh hơn và nhanh hơn để triển khai so với các lựa chọn thay thế sở hữu độc quyền. Ví dụ như các hệ quản trị cơ sở dữ liệu NoSQL như: Cassandra, CouchDB, Hbase, MongoDB, RavenDB và Redis.

**Việc mở rộng phạm vi là mềm dẻo:** Thay vì bổ sung thêm các máy chủ lớn hơn để điều khiển nhiều dữ liệu hơn, thì NoSQL cho phép một công ty phân tán dữ liệu qua nhiều máy chủ khi mà tải gia tăng.

**Các Cơ Sở Sở Dữ Liệu (CSDL) NoSQL đáp ứng những mục đích khác nhau:**

* MongoDB và Redis là những lựa chọn tốt cho việc lưu trữ các dữ liệu thống kê ít được đọc mà lại được viết thường xuyên

Apache Cassandra là một trong các hệ thống cơ sở dữ liệu mã nguồn mở phổ biến nhất. Nó được thiết kế với mục đích xử lý lượng lớn dữ liệu được lưu trữ trong nhiều máy chủ phân bố khắp nơi song song đó cung cấp khả năng mở rộng cao và sẵn có mà không gặp một lỗi nào. Các hệ thống Cassandra có thể mở rộng nhiều trung tâm dữ liệu, cho phép độ trễ thấp cho tất cả các máy kết nối.