**Kiến trúc cơ sở dữ liệu lớn và NoSQL: Đánh giá**

**Moko, Anasuodei**

Khoa Khoa học Máy tính và Tin học,

Đại học Liên bang Otuoke,

Bang Bayelsa, Nigeria.

**Asagba, Hoàng tử Oghenekaro**

Khoa Khoa học Máy tính,

Đại học Port Harcourt,

Bang Rivers, Nigeria.

# Khái Niệm

*Dữ liệu trong thời gian gần đây ngày càng lớn hơn để xử lý ngày càng tăng đặc biệt về khối lượng, khiến dữ liệu phi cấu trúc cụ thể trở nên khó xử lý và quản lý, bởi vì cơ sở dữ liệu có cấu trúc gặp khó khăn khi xử lý dữ liệu phi cấu trúc do kích thước của nó, được gọi là Dữ liệu lớn. Do đó, để hiểu được*  Dữ liệu lớn, cần có kiến thức và  *phương pháp mới, bài báo này giới thiệu về Dữ liệu lớn và các tính năng của nó, bốn cấu trúc NoSQL có sẵn để xử lý Dữ liệu lớn, điểm mạnh, điểm yếu và các loại có sẵn sẽ được thảo luận.*

***Từ khóa:*** *Dữ liệu lớn*, Không có cấu trúc, *NoSQL, Cơ sở dữ liệu, Có cấu trúc*

## Giới thiệu

Một thách thức công nghệ lớn mà thế giới đang phải đối mặt là quản lý và lưu trữ dữ liệu, trong đó hàng triệu dữ liệu trong thời gian gần đây đang được tạo ra với thời gian gián đoạn dưới nano giây. Do đó, việc xử lý một lượng lớn dữ liệu là một thách thức đáng kể và do đó với sự gia tăng tăng trưởng dân số, cần có công nghệ thu thập và quản lý dữ liệu hiện đại.

(Sultana, et al, 2017) nhấn mạnh rằng nhu cầu xử lý và tạo dữ liệu nhanh chóng trong thời gian gần đây đã dẫn đến thực tế là hơn 2,6 nghìn tỷ dữ liệu đang được sản xuất hàng ngày. Họ dự đoán thêm rằng trong tương lai sẽ có sự gia tăng theo cấp số nhân hơn về việc sử dụng và tạo dữ liệu nữa trong tương lai, quan tâm nhiều hơn đến cách nó đang được sử dụng ngày nay. Chính xác rằng việc quản lý và lưu trữ dữ liệu trong kỷ nguyên dữ liệu lớn, nơi phần mềm thông thường đang bị đẩy đến giới hạn bởi lượng dữ liệu khổng lồ, tạo ra nhu cầu thay đổi đáng kể đầu tư, thu thập và lưu trữ dữ liệu để sử dụng trong tương lai, do đó khiến các tổ chức khác nhau nỗ lực có ý thức để bảo mật và giữ mọi phần dữ liệu tiềm năng. Mặc dù dữ liệu thường không có cấu trúc, nó có thể được tạo ra từ nhiều nguồn khác nhau, chẳng hạn như đăng bài trên phương tiện truyền thông xã hội, nội dung đa phương tiện với kho lưu trữ tự động. Email, truy vấn công cụ tìm kiếm, kho tài liệu quản lý nội dung, dữ liệu cảm biến các loại khác nhau, sàn giao dịch chứng khoán, hình ảnh vệ tinh, hệ thống giám sát và ứng dụng e-health.etc.

Nó đôi khi được hiểu là 'Không chỉ SQL', để nó có thể thể hiện thực tế rằng các công nghệ khác được sử dụng trong các ứng dụng web phân tán hàng loạt, bên cạnh các công nghệ dữ liệu quan hệ. Quan trọng nhất, các công nghệ NoSQL được yêu cầu nếu dịch vụ web yêu cầu tính khả dụng cao. Nó là một kiến trúc lưu trữ phân tán rộng rãi, có cấu trúc hệ thống quản lý cơ sở dữ liệu hữu ích cơ bản. Dữ liệu thực tế của các giá trị chính được lưu trữ theo cặp, cột hoặc họ cột, tài liệu và đồ thị. Do đó, các khái niệm dự phòng khác nhau được hỗ trợ để đảm bảo tránh các lỗi và tăng tính khả dụng của các hệ thống cơ sở dữ liệu NoSQL. (Phanxicô, 2019)

Đối với tất cả các phương pháp quản lý dữ liệu phi quan hệ, thuật ngữ NoSQL được sử dụng để đáp ứng các tiêu chí được liệt kê dưới đây:

**i.**  Thông tin không được lưu trong bảng

**ii.** Ngôn ngữ SQL không phải là ngôn ngữ cơ sở dữ liệu.

Tuy nhiên, công nghệ cơ sở dữ liệu tương đối với công nghệ NoSQL cần được mở rộng để tạo ra quyền truy cập toàn cầu không đổi với các dịch vụ mà nó cung cấp cho các ứng dụng Web sâu rộng hoặc các ứng dụng xử lý cơ sở dữ liệu lớn. (Ali, 2019)

Hiểu công cụ "Dữ liệu lớn" nằm trong khái niệm tạo dựng dữ liệu rất hữu ích, vì nó giúp lưu trữ và quản lý lượng thông tin vô hạn được tạo ra mỗi giây, mỗi ngày. Mặc dù đã có hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) để lưu trữ dữ liệu cố định trong một số năm, khả năng mở rộng, tính nhất quán, hiệu quả hệ thống, tích hợp thu thập dữ liệu và trích xuất dữ liệu vẫn chưa được giải quyết.

Công nghệ Dữ liệu lớn là những công cụ cho phép chúng tôi có được nhiều ý nghĩa hơn từ máy học dữ liệu và những công cụ cho phép chúng tôi duy trì khối lượng dữ liệu cao hơn với độ chi tiết cao hơn bao giờ hết. Những công nghệ Dữ liệu lớn này được Google tiên phong và chúng đã tìm thấy con đường của mình dưới dạng Hadoop vào cộng đồng CNTT rộng lớn hơn.

NoSQL hoạt động để giúp giải quyết các yêu cầu về khối lượng, sự đa dạng và vận tốc dữ liệu lớn (Andreas & Michael, 2019) giải thích rằng Dữ liệu lớn vẫn chưa có định nghĩa ràng buộc. Tuy nhiên, nhiều chuyên gia dữ liệu sẽ đồng ý về ba chữ V (3V): V olume cho khối lượng dữ liệu mở rộng, ***V***ariety nhiều định dạng, dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc, và ***V***elocity để xử lý dữ liệu tốc độ cao và thời gian thực.

(Nzar & Dashne 2019) đã phân loại các tính năng của Dữ liệu lớn thành ―5Vs‖: khối lượng, vận tốc, sự đa dạng, tính xác thực và giá trị như được mô tả dưới đây:

Dữ liệu lớn

Khối Lượng

Giá trị

Tính đa dạng

Tính xác thực

thuc

Tốc Độ

**Hình 1:** Dữ liệu lớn 5 V (Nguồn: Nzar & Dashne, 2019)

1. Khối lượng: Hiển thị lượng dữ liệu khổng lồ, chẳng hạn như dữ liệu cho thiết bị di động, được sử dụng cho các chức năng khác nhau.
2. Vận tốc: chỉ định tốc độ hoặc tần suất tạo, cập nhật, xử lý và truy cập dữ liệu.
3. Đa dạng: dữ liệu được truy cập thông qua nhiều loại thiết bị khác nhau, chẳng hạn như video, ảnh, v.v.
4. Giá trị: giải thích cách rút ra kiến thức hữu ích từ các tập dữ liệu khổng lồ. Khía cạnh quan trọng nhất của bất kỳ công cụ dữ liệu lớn nào là giá trị, vì nó cho phép tạo ra kiến thức có giá trị.
5. Tính xác thực: Đề cập đến độ chính xác và giá trị thông tin rất lớn.

(Khan, et al, 2017) phân loại 1 - 6 v dữ liệu lớn, phát triển thành giá trị dữ liệu, làm cho nó trở thành 7 V của dữ liệu lớn. Bảy chữ V là - Khối lượng, Vận tốc, Sự đa dạng, Tính biến đổi, Tính xác thực, Trực quan hóa và Giá trị.

Dữ liệu chỉ có ý nghĩa khi liên tục thay đổi

Tốc độ mà dữ liệu được tạo ra

Dữ liệu lớn sẽ là vô ích trước khi chúng ta biến nó thành giá trị

**Hình 2:** Bảy chữ V của Dữ liệu lớn (Nguồn: Syed và cộng sự, 2019)

## II. Đánh giá các công việc liên quan

Kalid và cộng sự (2017) Bài báo nhấn mạnh tài liệu giải quyết các vấn đề và tìm ra giải pháp thông qua việc sử dụng các phương pháp Big Data tương phản để quản lý cơ sở dữ liệu No SQL; BigTable, DybanoDB và Cassandra. Công trình cũng chỉ ra rằng các công ty lớn không có hệ thống lưu trữ chính xác không cần lưu trữ và quản lý dữ liệu lớn. Nghiên cứu cho thấy BigTable từ Google và DynamoDB từ Amazon rất quan trọng và mang lại lợi nhuận cho chính họ và Casandra là sự kết hợp của cả hai hệ thống.

Md. Razu và cộng sự (2018) đã nghiên cứu cơ sở dữ liệu NoSQL để xử lý Dữ liệu lớn, bao gồm các vấn đề về giao dịch và cấu trúc của nó. Công trình cũng nêu bật các hướng nghiên cứu và thách thức liên quan đến việc xử lý Dữ liệu lớn mà nghiên cứu tin rằng thông tin đáng kinh ngạc để xem xét tài liệu về Cơ sở dữ liệu lớn NoSQL, bao gồm dữ liệu cấu trúc, các vấn đề và phương pháp thu thập thông tin hữu ích được đo trong thời gian thực.

Vahid (2016) trong Dữ liệu lớn: Bây giờ và sau đó, cho rằng sự tăng trưởng của dữ liệu và kiến thức phụ thuộc vào sự sẵn có của nó trong tay người tiêu dùng. Điều này rất đơn giản, vì có thể dễ dàng truy cập điện thoại di động, máy tính xách tay, PC, v.v. Trong trường hợp này, sự phát triển kiến thức đang nhanh chóng tràn ngập — vì nội dung mới do chính người dùng tạo ra để cần phải đáp ứng nhu cầu quản lý, sử dụng, phân loại và bảo mật dữ liệu đó. Bài báo cố gắng tổ chức và phân tích các quy trình, vấn đề tiềm ẩn, nghiên cứu và sáng kiến Dữ liệu lớn và hiển thị hiện tượng gia tăng mạnh mẽ này từ một chân trời tiềm năng.

Ali et al (2019) đã đệ trình rằng NoSQL được thay đổi bởi hầu hết các cơ sở dữ liệu quan hệ được sử dụng để lưu trữ dữ liệu, nhưng nó không hoàn toàn thay thế SQL, như tên gọi của nó. Bài báo thảo luận về cơ sở dữ liệu SQL và NoSQL, sự tương phản của SQL thông thường với cơ sở dữ liệu NoSQL phân tích dữ liệu lớn, mẫu dữ liệu NoSQL, biểu mẫu lưu trữ dữ liệu NoSQL, các tính năng và tính năng của từng bộ lưu trữ dữ liệu, ưu điểm và nhược điểm của NoSQL và RDBMS.

## III. Ngôn ngữ truy vấn có cấu trúc (SQL)

SQL (Ngôn ngữ truy vấn có cấu trúc) mặt khác theo truyền thống là cơ sở dữ liệu nổi tiếng nhất từ đầu những năm 1970. Một ví dụ là Cơ sở dữ liệu quan hệ (được truy cập bằng cách sử dụng SQL

- ví dụ: (MySQL) nơi dữ liệu được lưu trữ trong một bảng có hàng và cột. Các nhà phát triển hồi đó chủ yếu triển khai các thiết kế của họ theo mô hình phát triển phần mềm thác nước. Điều này có nghĩa là mọi giai đoạn phát triển phần mềm đều được lên kế hoạch tốt trước khi quá trình phát triển bắt đầu bằng cách sử dụng mối quan hệ thực thể phức tạp kỹ lưỡng, đảm bảo rằng tất cả những gì cần thiết trong cơ sở dữ liệu đã được suy nghĩ và cung cấp cẩn thận (Schaefer, 2015). Mặc dù cơ sở dữ liệu quan hệ rất hữu ích, nhưng vẫn có những thách thức sẵn sàng đối với các nhà phát triển phần mềm, chẳng hạn như nếu muốn có một chút cải tiến trong chu kỳ phát triển của phần mềm, các nhà phát triển sẽ phải vật lộn để giữ dự án dưới mức ngân sách quy định và phần mềm có thể không đáp ứng nhu cầu của người dùng. Một lý do khác là đã có sự gia tăng đột biến về lượng dữ liệu được tạo ra, đó là do một loại cơ sở dữ liệu NoSQL mới.

Một cách tiếp cận quản lý mới được coi là cần thiết để hỗ trợ các ứng dụng như xem xét thời gian thực các tệp nhật ký, giao dịch thương mại điện tử và dữ liệu được đăng lên phương tiện truyền thông xã hội có khối lượng khổng lồ. Một cách tiếp cận thay thế phải được đưa vào thực tế để xử lý sự gia tăng phi thường này trong dữ liệu được tạo ra và để vượt qua tất cả những thách thức nêu trên. Các cơ sở dữ liệu được giới thiệu, đó là cơ sở dữ liệu NoSQL, mặc dù có một số mức độ không hiệu quả do khối lượng sản xuất dữ liệu lớn và thiếu hỗ trợ cho các thuộc tính ACID.

## IV. Dữ liệu lớn và cơ sở dữ liệu

Dữ liệu lớn có thể được lưu trữ bằng cách sử dụng cả cơ sở dữ liệu có cấu trúc (MySQL là cơ sở dữ liệu quan hệ) và cơ sở dữ liệu phi cấu trúc (MongoDB là cơ sở dữ liệu phi quan hệ). Có tính đến sự thay đổi về thời gian phản hồi của từng loại cơ sở dữ liệu, các thuật toán khác nhau cần được phân tích để nâng cao hiệu suất trong việc giám sát hệ thống theo thời gian thực liên quan đến cả cập nhật SQL và NoSQL và chèn dữ liệu lớn.

Hệ thống cơ sở dữ liệu NoSQL được gọi là hệ thống lưu trữ dựa trên web miễn là chúng đáp ứng các yêu cầu sau:

* Mô hình: Mô hình cơ sở dữ liệu cơ bản không phải là quan hệ.
* Ít nhất 3 Vs: Một lượng lớn dữ liệu (khối lượng), cấu trúc dữ liệu linh hoạt (đa dạng) và xử lý thời gian thực được đưa vào hệ thống cơ sở dữ liệu (vận tốc).
* Lược đồ: Lược đồ cơ sở dữ liệu tập hợp không bị ràng buộc bởi hệ thống quản lý cơ sở dữ liệu.
* Kiến trúc: Kiến trúc cơ sở dữ liệu hỗ trợ mở rộng quy mô ngang và các ứng dụng web được phân phối đầy đủ.
* Nhân rộng: Hệ thống quản lý cơ sở dữ liệu hỗ trợ sao chép dữ liệu.
* Đảm bảo tính nhất quán: tính nhất quán có thể được đảm bảo với độ trễ để ưu tiên tính khả dụng cao và khả năng chịu đựng của các phân vùng. (2019 Andreas & Michael)

## V. Kiến trúc Big Data NoSQL

So với 'Không phải SQL', NoSQL có nghĩa là 'Không chỉ SQL' vì nhiều người coi nó là một loại cơ sở dữ liệu giúp thực hiện các hoạt động dữ liệu lớn và lưu trữ chúng ở định dạng hợp lệ. Nó thường được sử dụng vì tính đơn giản và nhiều loại dịch vụ. Mô hình kiến trúc của nó cung cấp một cách hợp lý để dữ liệu được xử lý trên cơ sở dữ liệu. (Nzar &Dashne, 2019) Các tính năng của NoSQL:

* Lược đồ miễn phí
* Cuối cùng nhất quán (như trong thuộc tính BASE)
* Sao chép các kho dữ liệu để loại bỏ một điểm thất bại duy nhất. Có khả năng xử lý một loạt dữ liệu và khối lượng lớn dữ liệu.

## VI. Các mẫu kiến trúc của NoSQL

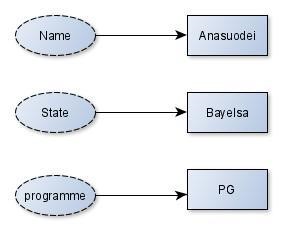
Dữ liệu được lưu trữ trong NoSQL sẽ tuân theo một trong 4 mẫu kiến trúc dữ liệu.

1. Cơ sở dữ liệu lưu trữ khóa-giá trị
2. Cơ sở dữ liệu cửa hàng cột
3. Cơ sở dữ liệu tài liệu
4. Cơ sở dữ liệu đồ thị

# A. Cơ sở dữ liệu lưu trữ khóa - Giá trị

Một trong những mô hình cơ sở dữ liệu NoSQL cơ bản nhất là mô hình này. Dữ liệu được thu thập theo mô hình của các Cặp khóa - Giá trị. Một loạt các chuỗi, số nguyên hoặc ký tự thường là khóa, nhưng nó cũng có thể là một dạng dữ liệu nâng cao. Thông thường, giá trị được kết nối hoặc liên quan đến khóa. Cơ sở dữ liệu lưu trữ cặp khóa - Giá trị thường lưu trữ thông tin dưới dạng JSON hoặc XML trong đó mỗi khóa là duy nhất. Giá trị có thể ở bất kỳ dạng nào (Ký hiệu đối tượng JavaScript (JSON), Đối tượng nhị phân (BLOB), chuỗi, v.v.). Phong cách kiến trúc này thường được sử dụng trong các trang web mua sắm hoặc ứng dụng thương mại điện tử và vai trò quan trọng của nó chính là khả năng quản lý khối lượng lớn dữ liệu và dễ dàng sử dụng khóa để truy xuất dữ liệu.

Khoá Gía Trị



**Hình 3.**   ***Ví dụ về Khoá – Gía Trị***

Các ràng buộc liên quan đến cơ sở dữ liệu lưu trữ khóa-giá trị là sự chi tiết của nó trong việc cố gắng xử lý các truy vấn bao gồm nhiều cặp khóa-giá trị có thể dừng đầu ra và có thể khiến dữ liệu xung đột với các mối quan hệ nhiều - nhiều.

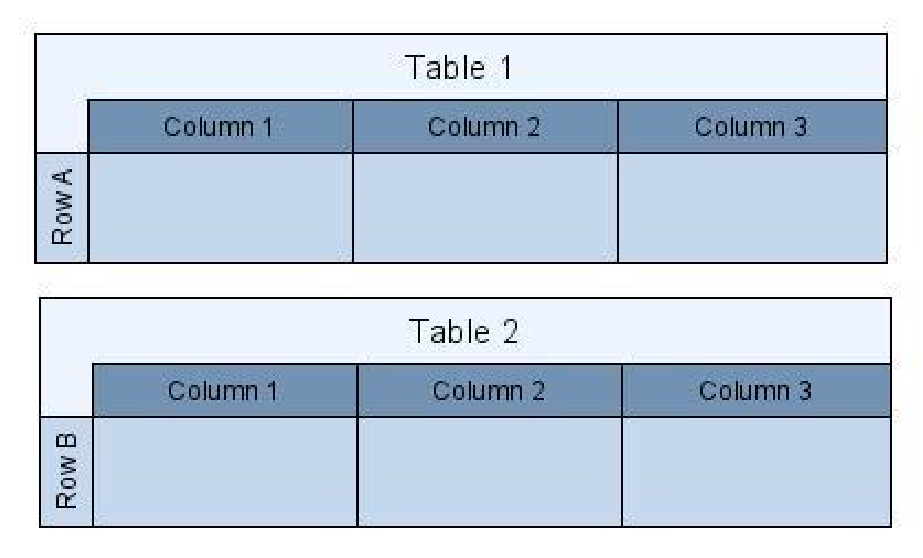
***Ví dụ ở đây là:***

* **DynamoDB** (*do Amazon phát triển*)
* **Berkeley DB** (*phát triển bởi Oracle*)
* **REDIS:** Một kho lưu trữ khóa - giá trị mã nguồn mở nâng cao, còn được gọi là máy chủ cấu trúc dữ liệu vì các khóa có thể bao gồm chuỗi, hàm băm, danh sách, tập hợp và tập hợp được sắp xếp. Sản phẩm này, được viết bằng C / C ++, rất nhanh chóng, điều này làm cho nó trở nên hoàn hảo để thu thập dữ liệu trong thời gian thực.
* **Riak:** Một nguồn mở mạnh mẽ, cơ sở dữ liệu rộng rãi, có thể dự đoán khả năng mở rộng quy mô và đơn giản hóa việc tạo bằng cách tạo mẫu, phát triển và triển khai các ứng dụng một cách nhanh chóng. Được viết bằng Erlang và C, công nghệ này cung cấp chức năng chịu lỗi / chuyển đổi dự phòng minh bạch, một API toàn diện và linh hoạt, hoàn hảo cho các hệ thống kiểm soát tại điểm bán hàng và nhà máy.
* **VoltDB:** cơ sở dữ liệu có thể mở rộng trong bộ nhớ cung cấp tính nhất quán ACID giao dịch hoàn chỉnh và thông lượng cực cao, tự gọi là NewSQL. Công nghệ này dựa vào phân đoạn và sao chép để đạt được ảnh chụp nhanh dữ liệu có độ sẵn sàng cao và ghi nhật ký lệnh bền bỉ bằng cách sử dụng các quy trình được lưu trữ Java (để khôi phục sự cố), lý tưởng cho thị trường vốn, mạng kỹ thuật số, dịch vụ mạng và chơi game trực tuyến.

# B. Cơ sở dữ liệu cửa hàng cột

Mẫu này sử dụng lưu trữ dữ liệu trong các ô riêng lẻ được chia thành các cột, thay vì lưu trữ dữ liệu trong các bộ quan hệ. Cơ sở dữ liệu hướng cột chỉ hoạt động trên các cột. Chúng cùng nhau lưu trữ một lượng lớn dữ liệu trong các cột. Định dạng cột và tiêu đề sẽ phân từ hàng này sang hàng khác. Mỗi cột được xử lý khác nhau, nhưng vẫn giống như cơ sở dữ liệu thông thường, mỗi cột riêng lẻ sẽ chứa một số cột khác. *(Niharika, 2020)*

Về cơ bản, các cột nằm trong loại chế độ lưu trữ này. Dữ liệu có sẵn và có thể thực hiện các truy vấn như ***Number, AVERAGE, COUNT*** trên các cột một cách dễ dàng.



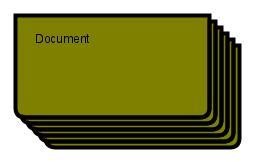
**Hình 4.  *Ví dụ về Column Store***

Những trở ngại cho hệ thống này bao gồm: các giao dịch cần tránh hoặc không được hỗ trợ, các truy vấn có thể làm giảm hiệu suất với các liên kết bảng, cập nhật bản ghi và xóa làm giảm hiệu quả lưu trữ và có thể khó thiết kế các sơ đồ phân vùng / lập chỉ mục hiệu quả. Ví dụ ở đây là:

* **HBase**: HBase là một Kho dữ liệu lớn phân tán, di động, được mô phỏng theo công nghệ BigTable của Google, cơ sở dữ liệu Hadoop.
* **BigTable** của **Google**
* **Cassandra:** Một hệ thống quản lý cơ sở dữ liệu phân tán mã nguồn mở được xây dựng để quản lý khối lượng dữ liệu rất lớn nằm rải rác trên một số máy chủ mà không có một điểm lỗi nào trong khi cung cấp một dịch vụ có khả năng truy cập cao. Được viết bằng Java, chính là một sản phẩm tốt nhất để phân tích dữ liệu thời gian thực không giao dịch với khả năng mở rộng tuyến tính và khả năng chịu lỗi đã được chứng minh kết hợp với các chỉ mục cột.

## C. Cơ sở dữ liệu tài liệu

Ở dạng các cặp khóa-giá trị, cơ sở dữ liệu bản ghi tìm nạp và tích lũy thông tin, nhưng ở đây các giá trị được gọi là tài liệu. Một cấu trúc dữ liệu phức tạp có thể được biểu diễn dưới dạng văn bản. Tài liệu có thể ở dạng văn bản, mảng, chuỗi, JSON (Ký hiệu đối tượng JavaScript), XML (Ngôn ngữ đánh dấu có thể mở rộng) hoặc bất kỳ định dạng nào khác. Việc sử dụng các tài liệu lồng ghép vào nhau là vô cùng phổ biến. Nó sẽcó hiệu quả cao vì hầu hết các thông tin được tạo ra thường ở dạng JSON và không có cấu trúc.



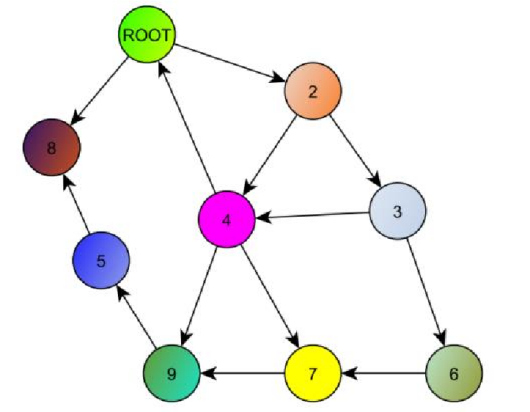
**Hình 5.  *Một ví dụ về tài liệu***

Định dạng này cực kỳ hữu ích và thích hợp cho dữ liệu bán cấu trúc và rất đơn giản để truy xuất và xử lý tài liệu từ bộ nhớ. Những hạn chế liên quan đến hệ thống này bao gồm yếu tố thách thức của việc xử lý nhiều tài liệu và hoạt động không chính xác của các hoạt động tổng hợp. Ví dụ về các cơ sở dữ liệu như vậy là:

* **MongoDB**: Cơ sở dữ liệu NoSQL mã nguồn mở, hiệu suất cao, có thể mở rộng này có tính năng lưu trữ theo định hướng tài liệu *(giống như JSON*), hỗ trợ chỉ mục đầy đủ, sao chép và cập nhật nhanh tại chỗ từ "khiêm tốn". Sản phẩm này phù hợp với các truy vấn động, cấu trúc dữ liệu động, được viết bằng C / C ++ và nếu bạn thích chỉ mục hơn Bản đồ / Thu nhỏ.
* **CouchDB**: Cũng là một cơ sở dữ liệu mã nguồn mở tập trung vào việc dễ dàng lưu trữ dữ liệu trong một loạt các tài liệu JSON, mỗi tài liệu có định nghĩa riêng về lược đồ. Tính nhất quán cuối cùng được thực thi bởi ngữ nghĩa ACID ngăn chặn việc khóa các tệp cơ sở dữ liệu trong quá trình ghi. Sản phẩm này, được viết bằng Java, phù hợp với các ứng dụng dựa trên web quản lý số lượng lớn dữ liệu được tổ chức lỏng lẻo.

## D. Cơ sở dữ liệu đồ thị

Mô hình kiến trúc này đề cập rõ ràng đến việc lưu trữ và quản lý thông tin trong biểu đồ. Đồ thị về cơ bản là các cấu trúc đại diện cho mối quan hệ giữa hai hoặc nhiều đối tượng trong một số dữ liệu. Các đối tượng hoặc thực thể được gọi là các nút và được kết nối với các mối quan hệ được gọi là cạnh. Có một mã định danh duy nhất trên mỗi cạnh. Đối với biểu đồ, mỗi nút đóng vai trò là một điểm chạm. Trong các mạng xã hội nơi có nhiều và số lượng lớn các thực thể, mô hình này được sử dụng rất rộng rãi và mỗi thực thể có một hoặc nhiều đặc điểm được liên kết bởi các cạnh.



**Hình 6.  *Một ví dụ về đồ thị***

Có các bảng được kết nối lỏng lẻo trong mô hình cơ sở dữ liệu quan hệ, trong khi đồ thị thường mạnh mẽ và cứng rắn về bản chất, có tốc độ truyền nhanh hơn do các kết nối và cho phép dễ dàng xử lý dữ liệu không gian, nhưng các kết nối không chính xác có thể dẫn đến các vòng lặp vô hạn. (Ian, 2016)

**Ví dụ về các cơ sở dữ liệu như vậy là:**

* **Neo4J**: Nền tảng đồ thị và cơ sở dữ liệu đồ thị gốc hàng đầu là Neo4J: Neo4j. Đối với các cấp độ bảo mật doanh nghiệp và hiệu suất và độ tin cậy cao bằng cách phân cụm, nó có sẵn cả dưới dạng mã nguồn mở và thông qua giấy phép thương mại. Cypher, ngôn ngữ truy vấn đồ thị của Neo4j, rất đơn giản để học và có thể sử dụng các bộ công cụ mã nguồn mở mới phát hành, "Cypher trên Apache Spark (CApS) và Cypher cho Gremlin hoạt động trên các sản phẩm dựa trên Neo4j, Apache Spark và Gremlin."
* **FlockDB***(Được Twitter sử dụng)*: FlockDB dễ dàng hơn các cơ sở dữ liệu đồ thị khác vì nó cố gắng giải quyết ít vấn đề hơn. Nó phù hợp theo chiều ngang và được tối ưu hóa cho các môi trường trực tuyến, độ trễ thấp, thông lượng cao như trang web.
* **ArangoDB**: loại cơ sở dữ liệu đồ thị này yêu cầu một cơ sở dữ liệu, một ngôn ngữ truy vấn, ba mô hình cho dữ liệu. Khả năng vô hạn. ArangoDB là một cơ sở dữ liệu NoSQL đa mô hình gốc đang phát triển nhanh chóng, với hơn một triệu lượt tải xuống.
* **OrientDB:** OrientDB là mô hình đa mô hình DBMS phân tán đầu tiên có Trình điều khiển đồ thị thực. Đa mô hình có nghĩa là NoSQL thế hệ thứ 2 có khả năng quản lý các miền phức tạp với hiệu quả đáng kinh ngạc.
* **Titan:** Titan là một cơ sở dữ liệu đồ thị có thể mở rộng được thiết kế để lưu trữ và truy vấn các biểu đồ trải rộng trên một cụm đa máy bao gồm hàng trăm tỷ đỉnh và cạnh. Titan là một cơ sở dữ liệu giao dịch có thể tạo điều kiện thuận lợi cho việc thực hiện thời gian thực các giao diện đồ thị phức tạp bởi hàng nghìn người dùng đồng thời.
* **DataStax**: Trong một môi trường thay đổi nhanh chóng, nơi khát vọng mạnh mẽ, DataStax giúp các doanh nghiệp phát triển mạnh mẽ và các công nghệ mới diễn ra hàng ngày.
* **Amazon Neptune:** Amazon Neptune với bộ dữ liệu có độ kết nối cao để tạo và chạy ứng dụng rất bảo mật, nhanh chóng và có dịch vụ cơ sở dữ liệu đồ thị được quản lý toàn phần, dễ dàng. (Niharika, 2020).

## VII. Điểm mạnh và điểm yếu của NOSQL

|  |  |  |
| --- | --- | --- |
| **S / N** | **Sức mạnh** | **Nhược điểm** |
| **tôi.** | Phù hợp, có thế mạnh để lưu trữ và tra cứu Big Data. | Yêu cầu một hình dung cơ sở hạ tầng đắt đỏ |
| **Ii.** | Là ứng dụng tập trung | Quá phức tạp |
| **Iii.** | Hỗ trợ dữ liệu HUGE khả năng | Tài năng kỹ thuật vẫn khó tìm |
| **Iv.** | Thu nạp dữ liệu nhanh (tải) | Nói chung, không có SQL  giao diện |
| **v.** | Tốc độ tra cứu nhanh (trên các cụm) | Lập trình hạn chế giao diện |
| **Vi.** | Cho phép truyền dữ liệu và máy chủ dữ liệu hiệu suất cao không cần thiết | Không đủ cho các truy vấn phân tích (tổng hợp,  số liệu, BI) |

## VIII. Kết luận

Công việc này đã xem xét và nghiên cứu dữ liệu lớn trong thời gian gần đây có và cách xử lý dữ liệu khối lượng ngày càng tăng. Cơ sở dữ liệu có cấu trúc gặp khó khăn khi xử lý dữ liệu phi cấu trúc do kích thước của nó. Để hiểu được Dữ liệu lớn, kiến trúc và phương pháp mới là cần thiết, công việc này cũng đã kiểm tra các Kiến trúc cơ sở dữ liệu NoSQL dữ liệu lớn khác nhau, các loại liên quan đến chúng, tầm quan trọng và cách sử dụng.

# Tham khảo

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). So sánh giữa SQL và

Cơ sở dữ liệu NoSQL và mối quan hệ của chúng với Phân tích dữ liệu lớn. *Tạp chí Châu Á của Nghiên cứu về Khoa học* *Máy tính*, *4* (2), 1–10. <https://doi.org/10.9734/ajrcos/2019/v4i230108>

Andreas Meier và Michael Kaufmann. (2019) Mô hình cơ sở dữ liệu .SQL & NoSQL, ngôn ngữ, tùy chọn nhất quán và kiến trúc để quản lý dữ liệu lớn. Springer Vieweg. https://doi.org/10.1007/978-3-658-24549-8

*Cơ sở dữ liệu đồ thị tốt nhất năm 2020*. (2020, ngày 27 tháng XNUMX). G2. [https://www.g2.com/categories/graphcơ sở dữ liệu](https://www.g2.com/categories/graph-databases)

*Công nghệ Dữ liệu lớn và NoSQL | DB Biên niên sử hay nhất*. (2012, ngày 15 tháng XNUMX). DB TỐT NHẤT. https://www.dbbest.com/blog/big-data-nosql-công [nghệ/](https://www.dbbest.com/blog/big-data-nosql-technologies/)

*Công nghệ cơ sở dữ liệu hướng cột | DB Biên niên sử hay nhất*. (2012, ngày 24 tháng XNUMX). DB TỐT NHẤT. [https://www.dbbest.com/blog/column](https://www.dbbest.com/blog/column-oriented-database-technologies/) [định hướng cơ](https://www.dbbest.com/blog/column-oriented-database-technologies/) [sở dữ](https://www.dbbest.com/blog/column-oriented-database-technologies/) [liệu-công](https://www.dbbest.com/blog/column-oriented-database-technologies/) [nghệ/](https://www.dbbest.com/blog/column-oriented-database-technologies/)

Đức Phanxicô, K. K. (2019). Cơ sở dữ liệu NoSQL để quản lý dữ liệu lớn: Đánh giá và ứng dụng trong thương mại di động. *Cổng nghiên cứu*, 9. <https://doi.org/10.13140/RG.2.2.10239.87204>

Garrett, Ngõ hẻm. (2019, ngày 4 tháng XNUMX). *Kiến trúc dữ liệu lớn là gì? - Dữ liệu lớn DZone*. Dzone.Com. [Kiến trúc](https://dzone.com/articles/what-is-big-data-architecture) https://dzone.com/articles/what-là-lớn-dữ [liệu](https://dzone.com/articles/what-is-big-data-architecture)

Chàng trai Harrison. (2015). Cơ sở dữ liệu thế hệ tiếp theo NoSQL, NewSQL và Big Data. ẤN TƯỢNG.

Ian. (2016, ngày 16 tháng XNUMX). *Cơ sở dữ liệu đồ thị là gì*?*| Cơ sở dữ liệu.Hướng dẫn.* https://database.guide/what-is-a-graph-database[/](https://database.guide/what-is-a-graph-database/)

Kalid, S., Syed, A., Mohammad, A., & Halgamuge, M. N. (2017). Cơ sở dữ liệu NoSQL dữ liệu lớn: So sánh và phân tích ―Big-Table‖, ―DynamoDB‖ và ―Cassandra‖. *2017 IEEE 2nd Hội nghị quốc tế về phân tích dữ liệu lớn (ICBDA) (,* 89–93. <https://doi.org/10.1109/ICBDA.2017.8078782>

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Bảy chữ V của Big Data hiểu biết lớn

Dữ liệu để trích xuất giá trị. *Kỷ yếu của Hội nghị Khu vực 1 năm 2014 của Mỹ*

*Hiệp hội Giáo dục* *Kỹ thuật*, 1–5. <https://doi.org/10.1109/ASEEZone1.2014.6820689>

Le, J. (2019, ngày 20 tháng XNUMX). *Giới thiệu về dữ liệu lớn: NoSQL*. Đau vừa. [https://medium.com/cracking-the-data-science-interview/](https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-nosql-96b882f35e50)an-introduction-to-big-data nosql-96b882f35e50

Liyakathunisa Syed, Saima Jabeen, S. Manimala và Hoda Ahmed Galal Elsayed. (2019, tháng Giêng). *(5) Các thuật toán và kỹ thuật khoa học dữ liệu cho chăm sóc sức khỏe thông minh sử dụng IoT và phân tích dữ liệu lớn: Hướng tới các thuật toán thông minh hơn*. Cổng nghiên cứu. [https://www.researchgate.net/publication/330723399\_Data\_Science\_Algorithms\_and\_Te](https://www.researchgate.net/publication/330723399_Data_Science_Algorithms_and_Techniques_for_Smart_Healthcare_Using_IoT_and_Big_Data_Analytics_Towards_Smarter_Algorithms" \l "fullTextFileContent) [chniques\_for\_Smart\_Healthcare\_Using\_IoT\_and\_Big\_Data\_Analytics\_Tow ards\_Smarte r\_Algorithms#fullTextFileContent](https://www.researchgate.net/publication/330723399_Data_Science_Algorithms_and_Techniques_for_Smart_Healthcare_Using_IoT_and_Big_Data_Analytics_Towards_Smarter_Algorithms#fullTextFileContent)

Md. Razu, A., Mst. Arifa, K., Md. Asraf, A., & Kenneth, S. (2018). Một đánh giá tài liệu về cơ sở dữ liệu NoSQL để xử lý dữ liệu lớn. *Tạp chí Kỹ thuật Quốc tế &*

*Công nghệ*, *7*(2), 902–906. <https://doi.org/10.14419/ijet.v7i2.12113>

Niharika, P. (2020, ngày 2 tháng XNUMX). Mẫu kiến trúc dữ liệu NoSQL. *GeeksforGeeks*. https://www.geeksforgeeks.org/nosql-data-architecture-patterns[/](https://www.geeksforgeeks.org/nosql-data-architecture-patterns/)

Raouf, D & Ali, N. (2019). Cải thiện hiệu suất của cơ sở dữ liệu dữ liệu lớn. Tạp chí Nghiên cứu Ứng dụng Kurdistan. 4. 206-220. 10.24017/khoa học.2019.2.20.

Schaefer, L. (2015). NoSQL là gì? Có sẵn tại [https://www.mongodb.com/nosql giải thích](https://www.mongodb.com/nosql-explained)

*Khái niệm cơ bản về cơ sở dữ liệu NoSQL — Và tại sao chúng ta cần chúng*. (2019, ngày 31 tháng XNUMX).

FreeCodeCamp.Org. [Cơ](https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/) [sở dữ liệu](https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/) [https://www.freecodecamp.org/news/nosql](https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/) [5f6639ed9574/](https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/)

Vahid Rahmati. (2016). Dữ liệu lớn: Bây giờ và sau đó. *Tạp chí quốc tế về sự nổi lên*

*Phương pháp*  tính toán *trong kỹ thuật*, 1(2), 1–6.

<https://www.researchgate.net/publication/309458088_Big_Data_Now_and_Then>

|  |  |
| --- | --- |
| IIARD - Viện Nghiên cứu và Phát triển Học thuật Quốc tế | Trang  **10** |