

Bài đánh giá về Big Data và NoSQL Kiến trúc cơ sở dữ liệu

Moko, Anasuodei

Khoa Khoa học máy tính và Thông tin

Đại học liên bang Otuke,

Bang Bayelsa, Nigeria

Asagba, Prince Oghenekaro

Khoa Khoa học máy tính

Đại học Port Harcourt,

Bang Rivers, Nigeria

Tóm tắt

Dữ liệu trong thời gian gần đây ngày càng lớn hơn, khiến cho dữ liệu phi cấu trúc trở nên phức tạp khi sử lý và quản lý, bởi vì những người có kinh nghiệm về dữ liệu có cấu trúc gặp khó khăn tiếp xúc với khối dữ liệu lớn của dữ liệu phi cấu trúc, từ đó “Big Data” – “Dữ liệu lớn” được định nghĩa, nhiều cấu trúc và phương thức được thêm vào. Đây là bài viết đánh giá về Dữ liệu lớn và tính năng của nó, cùng với các sự bàn luận về bốn tính chất của cơ sở dữ liệu phi cấu trúc là: Dữ liệu lớn, phân loại, thể mạnh và hạn chế.

Các từ khóa: Big Data, Unstructured, NoSQL, Database, Structured

I. Giới thiệu

Thế giới hiện nay, có một trở ngại lớn trong quản lý và lưu trữ dữ liệu, bởi vì chỉ trong vòng vài nano giây hàng triệu được tiếp nhận và tạo ra. Vì thế việc xử lý khối lượng lớn dữ liệu là một thử thách lớn cùng với vấn đề gia tăng dân số, ta cần một giải pháp cho vấn đề quản lý và thu thập dữ liệu.

(Sultana, et al, 2017) nhấn mạnh rằng, trong thời gian gần đây nhu cầu xử lý và tạo ra dữ liệu đã dẫn đến thực trạng là hơn 2,6 nghìn tỷ dữ liệu đang được tạo ra hàng ngày. Các nhà học giả đó dự đoán thêm rằng trong tương lai sẽ có sự gia tăng theo cấp số nhân của việc sử dụng và tạo dữ liệu trong tương lai, lưu ý đến cách nó đang được sử dụng ngày nay. Nói rõ rằng việc quản lý và lưu trữ dữ liệu trong thời đại 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 trong việc đầu tư, thu thập và lưu trữ dữ liệu để sử dụng trong tương lai, do đó khiến các tổ chức khác nhau đặt trong 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 nhưng nó có thể được tạo từ nhiều nguồn khác nhau, chẳng hạn như bài đăng trên phương tiện truyền thông xã hội, nội dung đa phương tiện với kho lưu trữ tự động. Email, công cụ tìm kiếm, kho lưu trữ quản lý tài liệu, dữ liệu cảm biến thuộc các loại khác nhau, trao đổi chứng khoán, ảnh vệ tinh, hệ thống giám sát, ứng dụng sức khỏe điện tử, ...

Đôi khi nó được hiểu là 'Không chỉ SQL', để có thể diễn đạt 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 sẵn sàng 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. (Franscis 2019).

III. Cấu trúc dữ liệu truy vấn

SQL (Structured Query language) mặt khác theo truyền thống thì đây là cơ sở dữ liệu nổi tiếng từ đầu những năm 1970. Một ví dụ về Relational database (được truy cập bằng cách sử dụng SQL – e.g. MySQL) nơi mà các dữ liệu được lưu trên những bảng có hàng và cột. Những nhà phát triển trở lại sau đó chủ yếu thực hiện các thiết kế của họ theo mô hình hình phát triển phần mềm thác nước. Điều này có nghĩa là trong mọi giai đoạn của việc phát triển phần mềm đã được lên kế hoạch trước khi quá trình phát triển bắt đầu sử dụng triệt để tổ hợp **entity-relationship** đảm bảo cho tất những gì cần thiết cho cơ sở dữ liệu đã được suy nghĩ cẩn thận và cung cấp đầy đủ. Mặc dù relational database rất hữu ích, đã có những thử thách dành cho những người phát triển phần mềm như là nếu như có một sự cải tiến nhẹ trong chu kỳ phát triển phần mềm, thì các nhà phát triển sẽ cần phải đấu tranh để giữ cho dự án không vượt quá mức ngân sách quy định và phần mềm có khả năng sẽ không đáp ứng được yêu cầu của người dùng. **Another reason is that there has been a spike in the amount of data produced**, đó là do có một loại cơ sở dữ liệu mới tên là NoSQL.

Đã có một phương pháp quản lý mới được coi là cần thiết để có thể hỗ trợ cho các phần mềm ứng dụng xem lại các tệp nhật ký theo thời gian thực, các giao dịch thương mại điện tử và dữ liệu đã được đăng một lượng lớn trên các truyền thông xã hội. Một cách tiếp cận thay thế cần phải đưa vào thực tế để có thể xử lý vấn đề này sự gia tăng phi tuyến tính về dữ liệu được tạo ra và để vượt qua tất cả những thử thách nêu trên. Cơ sở dữ liệu đã được giới thiệu, đó là cơ sở dữ liệu NoSQL **notwithstanding do possess some degree of** không hiệu quả do sản xuất một lượng lớn dữ liệu và thiếu sự hỗ trợ của **ACID properties**.

IV. Big data and Database

Big data có thể lưu trữ bằng cả Structured databases (MySQL a relational database) và Unstructured database (MongoDB a non-relational database). Xem xét về thời gian phản ứng của các loại dữ liệu, các thuật toán khác cần phải được phân tích để có thể nâng cao hiệu suất trong hệ thống giám sát thời gian thực với mối quan hệ của cả hai SQL và NoSQL cập nhật và chèn của big data.

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

- Model: Mô hình dữ liệu cơ bản không phải là quan hệ.
- At least 3 vs: Một lượng lớn dữ liệu (Khối lượng), linh hoạt các cấu trúc dữ liệu (Đa dạng), xử lý thời gian thực được đưa vào hệ thống cơ sở dữ liệu (Tốc độ).
- Schema: Một lược đồ cơ sở dữ liệu không bị ràng buộc bởi hệ thống quản lý cơ sở dữ liệu.

-Architecture: Kiến trúc dữ liệu được hỗ trợ mở rộng theo chiều ngang và đầy đủ **distributed web applications**.

-Replication: Hệ thống của quản lý dữ liệu hỗ trợ cho việc sao chép các dữ liệu.

-Consistency assurance: Tính nhất quán có thể đảm bảo với độ trễ ưu tiên cho tính sẵn sàng cao và **tolerance of partitions**.

V. Big Data NoSQL Architecture

So với “Not SQL”, NoSQL nghĩa là “Not Only SQL” như nhiều người xem nó là một loại của 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 sự đơn giản và phạm vi dịch vụ rộng .Mô hình kiến trúc của nó được cung cấp một cách logic để dữ liệu có thể được xử lý trên cơ sở dữ liệu.

Tính năng của NoSQL:

-Lướt đồ miễn phí

-Tính nhất quán cuối cùng

-Sao chép kho lưu trữ dữ liệu để loại bỏ những điểm lỗi

-Có khả năng xử lý dữ liệu trong một phạm vi lớn.

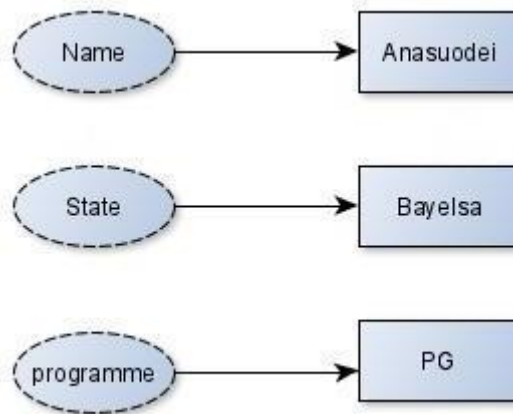
VI. Mẫu kiến trúc của NoSQL

Dữ liệu được lưu trữ trong NoSQL theo một trong bốn kiểu mẫu dữ liệu kiến trúc sau đây.

- Chìa khoá – giá trị thư viện cơ sở dữ liệu
- Cột thư viện cơ sở dữ liệu
- Tài liệu cơ sở dữ liệu
- Đồ thị cơ sở dữ liệu

A. Chìa khoá – giá trị thư viện cơ sở dữ liệu

Một trong những mô hình cơ sở dữ liệu cơ bản nhất trong NoSQL là mô hình này. Dữ liệu được thu thập theo cặp mẫu chìa khoá - giá trị, như tên của nó. Một chuỗi các chuỗi, số nguyên hoặc ký tự thường là khóa, nhưng nó cũng có thể là một dạng dữ liệu nâng cao hơn. Thông thường, giá trị được kết nối hoặc cùng liên quan đến khóa. Cơ sở dữ liệu cho thư viện cặp khóa - giá trị thường lưu trữ thông tin dưới dạng bảng chia nhỏ trong đó mỗi khóa là độc nhất. Giá trị có thể ở bất kỳ dạng nào (Ký hiệu đối tượng JavaScript (JSON), Đối Tượng Nhị Phân Không Lờ (BLOB), chuỗi, v.v.). Kiểu kiến trúc này thường được sử dụng trong các trang web mua sắm hoặc những ứng dụng thương mại điện tử và tài sản quan trọng của nó là khả năng quản lý lượng lớn khối lượng dữ liệu, tải nặng và dễ dàng sử dụng các phím để truy xuất dữ liệu.



Hình 3. Một ví dụ về Chìa khoá – Giá trị

Các ràng buộc liên quan đến thư viện cơ sở dữ liệu chìa khóa - giá trị là sự phức tạp của nó trong việc xử lý các truy vấn sẽ cố gắng bao gồm nhiều cặp chìa khóa - giá trị có thể làm chậm đầu ra và có thể khiến dữ liệu xung đột với càng ngày càng nhiều các mối quan hệ.

Ví dụ ở đây là:

- DynamoDB (do Amazon phát triển)
- Berkeley DB (được phát triển bởi Oracle)

- **REDIS:** Một thư viện chìa khoá – giá trị có nguồn mở tiên tiến, còn được gọi là máy chủ cấu trúc dữ liệu bởi vì các khóa có thể bao gồm chuỗi, chia nhỏ, danh sách, tập hợp và tập hợp được sắp xếp. Sản phẩm này, được viết bằng C/C++, cực kỳ nhanh, khiến nó trở nên hoàn hảo cho việc thu thập dữ liệu trong thời gian thực.
- **Riak:** Một nguồn mở mạnh mẽ, cơ sở dữ liệu phân tán, có thể dự đoán khả năng thay đổi quy mô và đơn giản hóa việc tạo bằng cách tạo nguyên 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 lỗi minh bạch, một API toàn diện và linh hoạt, hoàn hảo cho các hệ thống kiểm soát nhà máy và điểm bán hàng.
- **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, được tự gọi là NewSQL. Công nghệ này dựa vào phân đoạn và sao chép để đạt được ảnh chụp nhanh dữ liệu có tính sẵn sàng cao và ghi nhật ký lệnh lâu dài bằng cách sử dụng các quy trình được lưu trữ trên Java (để khôi phục sự cố), làm cho nó trở nên lý tưởng cho thị trường vốn, mạng kỹ thuật số, dịch vụ mạng và chơi game trực tuyến.

B. Cột thư viện cơ sở dữ liệu

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 dữ liệu 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 kỳ từ hàng này sang hàng khác. Mỗi cột được xử lý khác nhau, tuy nhiên, 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 ở dạng chế độ lưu trữ này. Dữ liệu có sẵn và có thể thực hiện các truy vấn như SỐ, TRUNG BÌNH, ĐẾM trên các cột một cách dễ dàng.

Table 1			
	Column 1	Column 2	Column 3
Row A			

Table 2			
	Column 1	Column 2	Column 3
Row B			

Hình 4. Một ví dụ về cột thư viện

Những trở ngại đối với hệ thống này bao gồm: các giao dịch nên tránh hoặc không được hỗ trợ, các truy vấn có thể làm giảm hiệu suất cao khi nối bảng, cập nhật và xóa bản ghi làm giảm hiệu quả lưu trữ và có thể khó thiết kế các lược đồ phân vùng/lập chỉ mục hiệu quả.

Ví dụ ở đây là:

- HBase: HBase là Kho lưu trữ dữ liệu lớn, di động, phân tán được mô phỏng theo công nghệ BigTable của Google, cơ sở dữ liệu Hadoop.
- Bảng lớn của Google
- Cassandra: Một hệ thống quản lý cơ sở dữ liệu phân tán 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 dịch vụ có khả năng truy cập cao. Được viết bằng Java, sản phẩm này phù hợp nhất để phân tích dữ liệu thời gian thực phi giao dịch với khả năng mở rộng tuyến tính và khả năng chịu lỗi đã được chứng minh kết hợp với chỉ mục cột.

C. Cơ sở dữ liệu tài liệu

Ở dạng 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 mở rộng) hoặc bất kỳ định dạng nào khác. Việc sử dụng các tài liệu lồng nhau là vô cùng phổ biến. Nó có hiệu quả cao vì hầu hết 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 phù hợp với dữ liệu bán cấu trúc, đồng thời việc truy xuất và xử lý tài liệu từ bộ lưu trữ trở nên đơn giản. Những hạn chế liên quan đến hệ thống này bao gồm yếu tố thách thức khi xử lý nhiều tài liệu và hoạt động tổng hợp không chính xác.

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

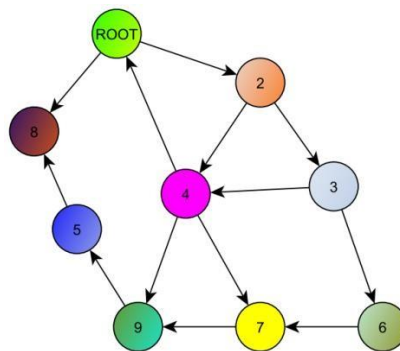
- MongoDB: Cơ sở dữ liệu NoSQL mã nguồn mở, hiệu năng cao, có thể mở rộng này có tính năng lưu trữ hướng tài liệu (giống JSON), hỗ trợ chỉ mục đầy đủ, sao

chép và cập nhật tại chỗ nhanh chóng từ "humongous". Sản phẩm này phù hợp với các truy vấn động, cấu trúc dữ liệu động, được viết bằng C/C++ và nếu bạn thích các chỉ mục của mình hơn Map/Reduce.

- 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 các định nghĩa riêng về lược đồ. cuối cùng tính nhất quán được thực thi bởi ngữ nghĩa ACID để ngăn khóa các tệp cơ sở dữ liệu trong khi ghi. Sản phẩm này, được viết bằng Java, thích hợp cho các ứng dụng dựa trên web quản lý lượng lớn dữ liệu được tổ chức lỏng lẻo.

D. Cơ sở dữ liệu đồ thị

Mẫu kiến trúc này giải quyết rõ ràng việc lưu trữ và quản lý thông tin trong biểu đồ. Đồ thị về cơ bản là các cấu trúc biểu thị 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 các cạnh. Có một mã định danh duy nhất trên mỗi cạnh. Đối với biểu đồ, mỗi nút đóng vai trò là một điểm tiếp xúc. Trong các mạng xã hội có nhiều và số lượng lớn các thực thể, mẫu 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 mẫu cơ sở dữ liệu quan hệ, trong khi các biểu đồ thường mạnh và cứng nhắc về bản chất, có tốc độ duyệt nhanh hơn do các kết nối và cho phép xử lý dữ liệu không gian dễ dàng, 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: Cơ sở dữ liệu đồ thị gốc và nền tảng đồ thị hàng đầu là Neo4J: Neo4j. Đối với mức độ bảo mật doanh nghiệp và hiệu suất cao và độ tin cậy bằng cách phân cụm, nó có sẵn 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 nguồn mở mới được phát hành, "Cypher trên Apache

Spark (CAPS) và Cypher cho Gremlin để hoạt động trên các sản phẩm dựa trên Neo4j, Apache Spark và Gremlin."

- FlockDB (Được sử dụng bởi Twitter): 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 chẳng hạn như các 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 bản địa đang phát triển nhanh chóng, với hơn một triệu lượt tải xuống.
- OrientDB: OrientDB là 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 đồ thị trải rộng trên một cụm nhiều máy bao gồm hàng trăm tỷ đỉnh và cạnh. Titan là một cơ sở dữ liệu giao dịch có thể hỗ trợ việc thực hiện theo thời gian thực các giao dịch đồ thị phức tạp bởi hàng nghìn người dùng đồng thời.
- DataStax: Trong một môi trường thay đổi nhanh chóng, nơi có nhiều khát vọng, DataStax giúp các doanh nghiệp phát triển mạnh và các công nghệ mới xuất hiện hàng ngày.
- Amazon Neptune: Amazon Neptune với bộ dữ liệu được kết nối cao để tạo và chạy ứng dụng an toàn, nhanh chóng và có dịch vụ cơ sở dữ liệu đồ thị được quản lý hoàn toàn dễ dàng. (Niharika, 2020).

VII. Điểm mạnh và điểm yếu của NOSQL

S/N	Điểm mạnh	Điểm yếu
i.	Phù hợp, có thể mạnh lưu trữ và tra cứu Big Data.	Yêu cầu một cơ sở hạ tầng đắt tiền
ii.	Is Application focused	Quá phức tạp
iii.	Hỗ trợ dung lượng dữ liệu KHÔNG LỖ	Nhân tài kỹ thuật vẫn khó tìm
iv.	Nhập dữ liệu nhanh (tải)	Nói chung, không có giao diện SQL
v.	Tốc độ tra cứu nhanh (trên các cụm)	Giao diện lập trình hạn chế
vi.	Cho phép truyền dữ liệu và máy chủ dữ liệu hiệu suất cao không cần thiết	Không đủ cho Truy vấn phân tích (tổng hợp, số liệu, BI)

VIII. Kết luận

Công trình này đã xem xét và nghiên cứu dữ liệu lớn trong thời gian gần đây đã và đang có những cách xử lý ngày càng tăng đột biến về lượng dữ liệu. 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 Big Data, kiến trúc và phương pháp mới là cần thiết, công việc này cũng đã kiểm tra Kiến trúc Cơ sở dữ liệu khác nhau Big Data NoQuery, các loại được liên kết với chúng, tầm quan trọng và cách sử dụng.

Tham khảo

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian Journal of Research in Computer Science*, 4(2), 1–10. <https://doi.org/10.9734/ajrcos/2019/v4i230108>

Andreas Meier, & Michael Kaufmann. (2019). SQL & NoSQL Databases Models, Languages, Consistency Options and Architectures for Big Data Management. Springer Vieweg. <https://doi.org/10.1007/978-3-658-24549-8> *Best Graph Databases in 2020*. (2020, May 27). G2. <https://www.g2.com/categories/graph-databases>

Big Data and NoSQL Technologies / DB Best Chronicles. (2012, June 15). DB BEST. <https://www.dbbest.com/blog/big-data-nosql-technologies/>

Column-Oriented Database Technologies / DB Best Chronicles. (2012, July 24). DB BEST. <https://www.dbbest.com/blog/column-oriented-database-technologies/>

Francis, K. K. (2019). NoSQL Databases for Big Data Management: Review and Application in Mobile Commerce. *ResearchGate*, 9. <https://doi.org/10.13140/RG.2.2.10239.87204>

Garrett, Alley. (2019, March 4). *What Is Big Data Architecture? - DZone Big Data*. Dzone.Com. <https://dzone.com/articles/what-is-big-data-architecture>

Guy Harrison. (2015). Next Generation Databases NoSQL, NewSQL, and Big Data. APRESS.

Ian. (2016, June 16). *What is a Graph Database?* / *Database.Guide*.
<https://database.guide/what-is-a-graph-database/>

Kalid, S., Syed, A., Mohammad, A., & Halgamuge, M. N. (2017). Big-data NoSQL databases: A comparison and analysis of -Big-Tablel, -DynamoDBl, and -Cassandral. *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*(, 89–93.
<https://doi.org/10.1109/ICBDA.2017.8078782>

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, 1–5.
<https://doi.org/10.1109/ASEEZone1.2014.6820689>

Le, J. (2019, November 20). *An Introduction to Big Data: NoSQL*. Medium.
<https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-nosql-96b882f35e50>

Liyakathunisa Syed, Saima Jabeen, S. Manimala, & Hoda Ahmed Galal Elsayed. (2019, January). (5) *Data Science Algorithms and Techniques for Smart Healthcare Using IoT and Big Data Analytics: Towards Smarter Algorithms*. ResearchGate.
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). A literature review on NoSQL database for big data processing. *International Journal of Engineering & Technology*, 7(2), 902–906. <https://doi.org/10.14419/ijet.v7i2.12113>

Niharika, P. (2020, January 2). NoSQL Data Architecture Patterns. *GeeksforGeeks*.
<https://www.geeksforgeeks.org/nosql-data-architecture-patterns/>

Raouf, D & Ali, N. (2019). Improving the performance of big data databases. *Kurdistan Journal of Applied Research*. 4. 206-220. 10.24017/science.2019.2.20. S chaefer, L. (2015). What is NoSQL? Available at <https://www.mongodb.com/nosql-explained>

The basics of NoSQL databases—And why we need them. (2019, January 31).

FreeCodeCamp.Org.<https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/>

Vahid Rahmati. (2016). Big Data: Now and Then. *International Journal of Emerging Computing Methods in Engineering*, 1(2), 1–6.

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