Nghiên cứu so sánh về NoSQL và cơ sở dữ liệu quan hệ

Douglas Kunda Hazael Phiri

School of Science, Engineering and Technology School of Science, Engineering and Technology

Mulungushi University Mulungushi University

Box 80415, Kabwe, Zambia Box 80415, Kabwe, Zambia

[dkunda@mu.ac.zm](mailto:dkunda@mu.ac.zm)

Tóm tắt - Cơ sở dữ liệu quan hệ và NoSQL là các loại mô hình cơ sở dữ liệu cạnh tranh nhau. - Cơ sở dữ liệu quan hệ đã tồn tại từ năm 1979 và NoSQLlà từ năm 2000. Nhu cầu của các ứng dụng hiện đại, đặc biệt là trong web 2.0, 3.0 và dữ liệu lớn đã khiến NoSQL trở thành cơ sở dữ liệu phổ biến được lựa chọn. Chọn một mô hình cơ sở dữ liệu thích hợp để sử dụng là một quyết định quan trọng mà các nhà phát triển phải đưa ra dựa trên các tính năng của một mô hình cơ sở dữ liệu nhất định. Bài viết này so sánh các tính năng của Cơ sở dữ liệu quan hệ và NoSQL để thiết lập cơ sở dữ liệu nào tốt hơn trong việc hỗ trợ nhu cầu của các ứng dụng hiện đại. Bài báo cũng đưa ra những thách thức của NoSQL. Cuối cùng, bài viết kết luận bằng cách xác định liệuCơ sở dữ liệu quan hệ sẽ được thay thế hoàn toàn bằng mô hình cơ sở dữ liệu NoSQL . Các phát hiện cho thấy rằng, cơ sở dữ liệu quan hệ dựa trên mô hình ACID, mô hình này nhấn mạnh tính nhất quán, bảo mật tốt hơn và cung cấp ngôn ngữ truy vấn tiêu chuẩn. Tuy nhiên, cơ sở dữ liệu quan hệ có khả năng mở rộng kém, hiệu suất yếu, chi phí cao hơn, đối mặt với những thách thức về tính khả dụng khi hỗ trợ số lượng lớn người dùng và xử lý khối lượng dữ liệu hạn chế. Mặt khác , NoSQL dựa trên mô hình BASE, mô hình này nhấn mạnh khả năng mở rộng lớn hơn và cung cấp lược đồ linh hoạt, mang lại hiệu suất tốt hơn, chủ yếu là nguồn mở, giá rẻ nhưng thiếu ngôn ngữ truy vấn tiêu chuẩn và không cung cấp bảo mật đầy đủcơ chế. Cả hai cơ sở dữ liệu sẽ tiếp tục tồn tại song song với nhau mà không có cơ sở dữ liệu nào tốt hơn cơ sở dữ liệu nào. Việc lựa chọn cơ sở dữ liệu để sử dụng sẽ phụ thuộc vào bản chất của ứng dụng đang được phát triển. Mỗi loại cơ sở dữ liệu đều có những thách thức và thế mạnh riêng, trong đó cơ sở dữ liệu quan hệ thiếu hỗ trợ cho dữ liệu phi cấu trúc trong khi NoSQL thiếu tiêu chuẩn hóa và bảo mật kém. Các ứng dụng hiện đại trong web 2.0, 3.0 và dữ liệu lớn rất phù hợp để sử dụng NoSQL, nhưng vẫn còn nhiều ứng dụng dựa trên cơ sở dữ liệu quan hệ. Từ khóa - Cơ sở dữ liệu quan hệ, NoSQL, ngôn ngữ truy vấn, bảo mật

1. **GIỚI THIỆU**

Cơ sở dữ liệu đã thay thế các tệp phẳng làm kho lưu trữ các nhóm dữ liệu lớn. Kể từ khi ra đời cơ sở dữ liệu, Cơ sở dữ liệu quan hệ đã thống trị hơn 30 năm cho đến năm 2000 khi cơ sở dữ liệu NoQuery bắt đầu thay thế chúng trong một số ứng dụng [1]. Bản chất của nhu cầu ứng dụng không tĩnh theo thời gian và dẫn đến các ứng dụng có tính tương tác cao và xử lý khối lượng dữ liệu lớn, chẳng hạn như thương mại điện tử và phương tiện truyền thông xã hội. Cung cấp các tính năng tương tác trong cơ sở dữ liệu là một yêu cầu chính đối với cơ sở dữ liệu phục vụ các ứng dụng web 2.0 và 3.0 [2]. Sự thay đổi trong nhu cầu ứng dụng đã khiến Cơ sở dữ liệu quan hệ không đáp ứng được nhu cầu của nhà phát triển và người dùng. Các công ty như Google, Facebook và Yahoo đã chuyển sang NoSQL để giảm thiểu những thiếu sót của Cơ sở dữ liệu quan hệ [3]. Bất chấp những xu hướng này, có nhiều công ty vẫn sử dụng Cơ sở dữ liệu quan hệ. Các công ty này được đặc trưng bởi khối lượng dữ liệu hạn chế đòi hỏi mức độ nhất quán cao. Cơ sở dữ liệu quan hệ dựa trên mô hình ACID tức là Tính nguyên tử, Tính nhất quán, Cách ly và Độ bền [4]. Tính nguyên tử đảm bảo tính hoàn chỉnh của giao dịch, Tính nhất quán đảm bảo tính ổn định của dữ liệu trong cơ sở dữ liệu, Tính cách ly đảm bảo tính độc lập của nhiều giao dịch được thực hiện cùng lúc và Độ bền đảm bảo rằng các giao dịch được lưu trữ không thay đổi trạng thái ngay cả khi có sự hiện diện củathất bại. ACID cung cấp tính nhất quán và tính sẵn sàng dưới dạng các thuộc tính mạnh giúp Cơ sở dữ liệu quan hệ trở nên phổ biến. Mặt khác , NoSQL dựa trên mô hình BASE (Có sẵn về cơ bản, Trạng thái mềm và Nhất quán cuối cùng) [3]. Bản chất phân tán của NoSQL mang lại khả năng dữ liệu có sẵn một phần khi một số phần của cơ sở dữ liệu phân tán không hoạt động hoặc không thể truy cập được do đó, thuật ngữ Cơ bản có sẵn. Trạng thái mềm cho phép dữ liệu thay đổi theo thời gian có hoặc không có đầu vào. Cuối cùng Tính nhất quán đảm bảo rằng dữ liệu sẽ trở nên nhất quán trong tương lai chứ không phải ngay sau một thao tác. BASE cung cấp cho NoSQL khả năng mở rộng quy mô dễ dàng, cung cấphiệu suất tốt hơn và mức độ sẵn có cao hơn cho người dùng của nó.Bài viết này dựa trên việc xem xét các tài liệu trước đây và bắt đầu bằng một mô tả về Cơ sở dữ liệu quan hệ và các mô hình cơ sở dữ liệu NoSQL . Sau đó, cuộc thảo luận sẽ chuyển sang so sánh các tính năng của Cơ sở dữ liệu quan hệ và NoSQL, sau đó là các thách thức của NoSQL. Dựa trên các tính năng, các tác giả cố gắng xác định xem NoSQL có tốt hơn Cơ sở dữ liệu quan hệ trong việc hỗ trợ các nhu cầu ứng dụng cơ sở dữ liệu hiện đại hay không và liệu NoSQL có thay thế hoàn toàn Cơ sở dữ liệu quan hệ hay không.

1. **CƠ SỞ DỮ LIỆU QUAN HỆ VÀ CÁC LOẠI NOSQL**

Chỉ có một dạng Cơ sở dữ liệu quan hệ dựa trên mô hình quan hệ [5]. Nhiều tổ chức, đã điều chỉnh các yêu cầu ứng dụng của họ để phù hợp với các lược đồ nghiêm ngặt được cố định trước trong Cơ sở dữ liệu quan hệ. Lược đồ nghiêm ngặt yêu cầu ứng dụng tuân theo nhu cầu của cơ sở dữ liệu thay vì cơ sở dữ liệu tuân theo nhu cầu của ứng dụng. Ví dụ về Cơ sở dữ liệu quan hệ là MySQL, Microsoft SQL Server và Postgres.

Có nhiều loại cơ sở dữ liệu NoSQL và từ các tài liệu được xem xét, có bốn loại nổi bật. Chúng bao gồm Cơ sở dữ liệu khóa- giá trị, hướng tài liệu, cơ sở dữ liệu cột và cơ sở dữ liệu đồ thị [6]. Trong giá trị khóa, dữ liệu được lưu trữ dưới dạng tập hợp các cặp khóa và giá trị, trong đó khóa là một thành phần duy nhất trong cơ sở dữ liệu được xác định bởi thuộc tính của nó và giá trị là giá trị của thuộc tính [7]. Giá trị khóa dễ sử dụng nhưng không hỗ trợ xử lý mối quan hệ giữa các mục dữ liệu. Ví dụ về cơ sở dữ liệu giá trị khóa bao gồm Memcached và Redis [3]. Cơ sở dữ liệu hướng tài liệu sử dụng khóa và tài liệu làm thuộc tính trong đó khóa đề cập đến toàn bộ tài liệu [8]. Ví dụ về cơ sở dữ liệu hướng tài liệu là MongoDB và CouchDB,rất phù hợp để xử lý các cấu trúc dữ liệu phức tạp nhưng vẫn thiếu khả năng xử lý các mối quan hệ giữa các mục dữ liệu. Cơ sở dữ liệu cột chứa các hàng/cột tương tự như Cơ sở dữ liệu quan hệ nhưng mỗi cột được lưu trữ trong một tệp riêng biệt. Một khóa trong cơ sở dữ liệu hướng cột đề cập đến một cột. Các thuộc tính khác được lưu trữ bao gồm giá trị và dấu thời gian. Bigtable và Cassandra là những ví dụ và theo thiết kế, chúng kém linh hoạt hơn nhưng cung cấp thông lượng lớn hơn. Cơ sở dữ liệu đồ thị biểu thị dữ liệu dưới dạng đồ thị được kết nối và dựa trên lý thuyết đồ thị [8]. Cơ sở dữ liệu đồ thị ít có khả năng mở rộng hơn nhưng hỗ trợ kết nối tốt hơn. Các ví dụ bao gồm GraphDB và OrientDB.

1. **TÍNH NĂNG CỦA CƠ SỞ DỮ LIỆU QUAN HỆ VÀ MÔ HÌNH CƠ SỞ DỮ LIỆU NOSQL**

***A. Mã nguồn đóng và mã nguồn mở***

Cơ sở dữ liệu quan hệ bao gồm cả nền tảng nguồn mở và độc quyền [5]. Các loại Cơ sở dữ liệu quan hệ độc quyền như Oracle thường có khả năng mở rộng quy mô tốt hơn so với các đối tác nguồn mở như MySQL. Tuy nhiên, nhiều mô hình cơ sở dữ liệu NoSQL là mã nguồn mở như MongoDB, CouchDB và Cassandra [9]. Bản chất nguồn mở của NoSQL mang lại nhiều cơ hội hơn cho các nhà nghiên cứu trong việc điều tra các tính năng của cơ sở dữ liệu và cung cấp dung lượng lưu trữ rẻ hơn cho những người dùng không đủ khả năng mua các mô hình cơ sở dữ liệu độc quyền.

1. ***Khả năng mở rộng***

Cơ sở dữ liệu quan hệ, thường mở rộng quy mô, trong đó nâng cấp phần cứng phải được thực hiện cho một máy chủ để làm cho nó hiệu quả hơn. Điều này làm tăng số lượng nỗ lực cần thiết từ các quản trị viên trong việc nâng cấp Cơ sở dữ liệu quan hệ [1]. Phương pháp nâng cấp này cũng phải đối mặt với những thách thức về giới hạn phần cứng được thiết kế cố định và không thể thay đổi. Ví dụ: dung lượng RAM tối đa hoặc dung lượng lưu trữ thứ cấp được phần cứng hỗ trợ có giá trị cố định do nhà sản xuất phần cứng xác định. Điều này có nghĩa là Cơ sở dữ liệu quan hệ có khả năng mở rộng nhưng sẽ luôn có giới hạn về mức độ mở rộng do nó được xác định bởi phần cứng. Để cung cấp khả năng mở rộng, NoSQLyêu cầu sử dụng máy chủ hàng hóa tức là mở rộng theo chiều ngang [8] [10]. Việc mở rộng quy mô theo chiều ngang không bị ảnh hưởng đáng kể bởi các giới hạn phần cứng vì các máy chủ nhỏ hơn, rẻ hơn và kém mạnh hơn có thể được kết hợp để mang lại khả năng mở rộng ở mức cao hơn thay vì có một máy chủ đắt tiền. Khả năng này giúp việc triển khai trở nên dễ dàng vì các máy ảo có thể được sử dụng làm máy chủ hàng hóa trong các tình huống không thể có được phần cứng thực tế. Máy ảo có thể được thêm và xóa mà không làm giảm hiệu suất của cơ sở dữ liệu. Các ứng dụng Internet hiện đại như phương tiện truyền thông xã hội yêu cầu mức độ mở rộng cao mà không được giải quyết thỏa đáng trong Cơ sở dữ liệu quan hệ nhưng được cung cấp hiệu quảtrong NoSQL [16].

***C. Chi phí***

Cơ sở dữ liệu quan hệ tốt hơn là độc quyền và do đó, đòi hỏi số tiền đầu tư lớn từ các tổ chức và cá nhân muốn hưởng lợi từ các tính năng nâng cao của chúng. Phần cứng bổ sung để nâng cấp cũng thêm các chi phí bổ sung khác. Điều này làm cho Cơ sở dữ liệu quan hệ trở thành một cách tiếp cận tốn kém để lưu trữ dữ liệu [7]. NoSQL chủ yếu là mã nguồn mở nên nó trở thành giải pháp thay thế rẻ hơn cho Cơ sở dữ liệu quan hệ [9]. Khả năng sử dụng máy ảo làm máy chủ hàng hóa giúp giảm hơn nữa chi phí duy trì cơ sở dữ liệu NoSQL, khiến NoSQL trở thành kho lưu trữ dữ liệu giá rẻ hấp dẫn cho các tổ chức.

***D. Khối lượng và sự đa dạng của dữ liệu***

Cơ sở dữ liệu quan hệ tốt hơn là độc quyền và do đó, đòi hỏi số tiền đầu tư lớn từ các tổ chức và cá nhân muốn hưởng lợi từ các tính năng nâng cao của chúng. Phần cứng bổ sung để nâng cấp cũng thêm các chi phí bổ sung khác. Điều này làm cho Cơ sở dữ liệu quan hệ trở thành một cách tiếp cận tốn kém để lưu trữ dữ liệu [7]. NoSQL chủ yếu là mã nguồn mở nên nó trở thành giải pháp thay thế rẻ hơn cho Cơ sở dữ liệu quan hệ [9]. Khả năng sử dụng máy ảo làm máy chủ hàng hóa giúp giảm hơn nữa chi phí duy trì cơ sở dữ liệu NoSQL, khiến NoSQL trở thành kho lưu trữ dữ liệu giá rẻ hấp dẫn cho các tổ chức.

***E. Sẵn có***

Số lượng người dùng và thời gian truy cập dữ liệu đã tăng lên, với các ví dụ như phương tiện truyền thông xã hội, thương mại điện tử và lưu trữ đám mây đang dẫn đầu. Theo thiết kế, Cơ sở dữ liệu quan hệ thường bị lỗi một điểm ngay cả đối với các máy chủ rất mạnh [5]. Tính khả dụng bị hạn chế hơn nữa do Cơ sở dữ liệu quan hệ mở rộng quy mô. Các điểm lỗi đơn lẻ không phù hợp với các ứng dụng internet hiện đại ngày nay mà người dùng phụ thuộc rất nhiều vào đó để hỗ trợ họ trong cuộc sống hàng ngày. Do đó, bản chất phân tán của NoSQL đưa ra lựa chọn tốt hơn để cung cấp tính khả dụng cho người dùng mọi lúc ngay cả khi có lỗi phần cứng [10]. Về cơ bản có sẵnbản chất của NoSQL giúp có thể truy cập một phần cơ sở dữ liệu khi có lỗi. Người dùng được đảm bảo tiếp tục truy cập vào cơ sở dữ liệu bất kể lỗi hệ thống.

***F. Hiệu suất***

Cơ sở dữ liệu quan hệ đòi hỏi nhiều thời gian hơn để xử lý thông tin khiến chúng chậm hơn so với NoSQL xử lý nhanh [13]. Hiệu suất của NoSQL được cải thiện hơn nữa khi nó truy xuất dữ liệu từ bộ nhớ khả biến, không giống như Cơ sở dữ liệu quan hệ truy xuất dữ liệu từ bộ nhớ không khả biến . Theo thiết kế, bộ nhớ dễ bay hơi nhanh hơn so với Cơ sở dữ liệu quan hệ không dễ bay hơi vượt trội khi tìm kiếm thông tin [14]. Các thử nghiệm đã được tiến hành để kiểm tra hiệu suất của cả NoSQL và Cơ sở dữ liệu quan hệ. So sánh Cơ sở dữ liệu quan hệ với MongoDB cho thấy MongoDB có hiệu suất tốt hơn đối với các truy vấn đọc, cập nhật và cơ bản trong khi SQL chỉ hoạt động tốt khi cập nhật các thuộc tính không khóa [15].

***G. Độ phức tạp***

Cơ sở dữ liệu quan hệ tạo dữ liệu phức tạp trong trường hợp dữ liệu được lưu trữ bởi người dùng khó chuyển đổi thành bảng [1]. Việc nhấn mạnh vào việc lưu trữ dữ liệu có cấu trúc trong Cơ sở dữ liệu quan hệ mang lại sự phức tạp này. Các truy vấn và giao dịch phức tạp của Cơ sở dữ liệu quan hệ có thể không cần thiết trong một số tình huống mà các thao tác đọc hoặc ghi đơn giản có thể đủ, chẳng hạn như trong phương tiện truyền thông xã hội. NoSQL có thể lưu trữ cả dữ liệu bán cấu trúc và phi cấu trúc [16]. Khả năng lưu trữ cả dữ liệu bán cấu trúc và phi cấu trúc của NoSQL mang lại sự linh hoạt cần thiết để hỗ trợ nhiều loại dữ liệu ở trạng thái thô mà không làm mất thông tin. Ví dụ: chuyển đổi bản ghi âm củakhiếu nại của khách hàng thành văn bản để lưu trữ trong Cơ sở dữ liệu quan hệ, dẫn đến mất thông tin về tâm trạng của khách hàng. Thông tin như vậy có thể được lưu giữ trong NoSQL, vì bản ghi có thể được lưu trữ ở trạng thái của nó mà không cần chuyển đổi.

***H. Ngôn ngữ truy vấn***

Cơ sở dữ liệu quan hệ có nền tảng vững chắc và tài liệu được ghi chép đầy đủ về SQL. SQL là ngôn ngữ thao tác dữ liệu duy nhất mà tất cả các Cơ sở dữ liệu quan hệ sử dụng [5]. Tuy nhiên, có những biến thể nhỏ của việc triển khai SQL cho các Cơ sở dữ liệu quan hệ khác nhau đang được sử dụng. Nền tảng vững chắc do SQL cung cấp, làm cho Cơ sở dữ liệu quan hệ trở nên phổ biến đối với các nhà phát triển vì thời gian học tập ngắn hơn đối với bất kỳ triển khai Cơ sở dữ liệu quan hệ nào. Nền tảng này vẫn còn thiếu trong NoSQL vì nó dựa trên API hướng đối tượng để thao tác dữ liệu [1]. Mỗi triển khai của NoSQL đều có ngôn ngữ thao tác dữ liệu riêng, yêu cầu các nhà phát triển dành thời gian tìm hiểu khi phát triển trênloại mô hình NoSQL khác với mô hình mà họ đã quen thuộc. Có nhiều cách truy vấn NoSQL, giới hạn số lượng truy vấn được hỗ trợ vì mỗi triển khai phải cung cấp các truy vấn duy nhất của riêng nó [17]. Yêu cầu của web 2.0 và 3.0 đòi hỏi các phương pháp phát triển nhanh và NoSQL có thể không đáp ứng được những yêu cầu này, vì thời gian phát triển tăng lên bởi các nhà phát triển cần học ngôn ngữ triển khai.

***I. Tính nhất quán***

Cơ sở dữ liệu quan hệ cung cấp tính nhất quán mạnh mẽ hơn với lược đồ nghiêm ngặt [8]. Tính năng này làm cho Cơ sở dữ liệu quan hệ hy sinh tính khả dụng vì cả hai không miễn phí. Tính nhất quán cao rất tốt cho việc cung cấp chế độ xem dữ liệu thống nhất ngay sau khi các thao tác được thực hiện. Tuy nhiên, có những ứng dụng như phương tiện truyền thông xã hội thì tính linh hoạt quan trọng hơn tính nhất quán [16]. NoSQL cung cấp tính khả dụng cao hơn nhưng có tính nhất quán kém [7]. Vì vậy, đối với phương tiện truyền thông xã hội, NoSQL phù hợp làm tùy chọn lưu trữ hơn Cơ sở dữ liệu quan hệ.

***J. An ninh***

Cơ sở dữ liệu quan hệ cung cấp tính nhất quán mạnh mẽ hơn với lược đồ nghiêm ngặt [8]. Tính năng này làm cho Cơ sở dữ liệu quan hệ hy sinh tính khả dụng vì cả hai đều là Cơ sở dữ liệu quan hệ phải đối mặt với một số thách thức bảo mật như SQL injection và cross site scripting. Bất chấp những thách thức này, SQL có các cơ chế bảo mật mạnh mẽ được sử dụng để bảo vệ dữ liệu bao gồm xác thực, ủy quyền, mã hóa, toàn vẹn và kiểm tra [1]. Các cơ chế bảo mật là một phần của cơ sở dữ liệu. Trong NoSQL, bảo mật không phải là một phần của cơ sở dữ liệu mà được xử lý bởi phần mềm trung gian [7]. Điều này khiến cơ sở dữ liệu dễ bị tấn công. Hơn nữa, các cơ chế bảo mật được triển khai trong phần mềm trung gian nên đượcđược triển khai theo cách không ảnh hưởng đến khả năng mở rộng và hiệu suất. không miễn phí. Tính nhất quán cao rất tốt cho việc cung cấp chế độ xem dữ liệu thống nhất ngay sau khi các thao tác được thực hiện. Tuy nhiên, có những ứng dụng như phương tiện truyền thông xã hội thì tính linh hoạt quan trọng hơn tính nhất quán [16]. NoSQL cung cấp tính khả dụng cao hơn nhưng có tính nhất quán kém [7]. Vì vậy, đối với phương tiện truyền thông xã hội, NoSQL phù hợp làm tùy chọn lưu trữ hơn Cơ sở dữ liệu quan hệ.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tiêu chuẩn | Cơ sở dữ liệu quan hệ | NoSQL |
| 1. | Đa dạng | Cả nền tảng mã nguồn mở và đóng [5] | NoSQL chủ yếu là mã nguồn mở [9] |
| 2. | Khả năng mở rộng | Mở rộng quy mô bằng cách nâng cấp phần cứng của một máy chủ [1]. | Mở rộng quy mô theo chiều ngang bằng cách sử dụng máy chủ hàng hóa [8] |
| 3. | Chi phí | Phương pháp tốn kém để lưu trữ dữ liệu [7] | Rẻ hơn vì nó là mã nguồn mở và nâng cấp không tốn kém [9] |
| 4. | Khối lượng dữ liệu | Xử lý dữ liệu hạn chế [11]. | Xử lý dữ liệu khối lượng lớn, đặc biệt là trong Dữ liệu lớn [12]. |
| 5. | Sẵn có | Bị một điểm lỗi [5] | Bản chất phân tán cung cấp tính khả dụng cho người dùng mọi lúc khi có lỗi phần cứng [10] |
| 6. | Hiệu suất | Đòi hỏi nhiều thời gian hơn để xử lý thông tin khiến chúng chậm lại [13] | Có xu hướng có hiệu suất truy vấn tốt hơn [16] |
| 7. | Độ phức tạp | Tạo dữ liệu phức tạp trong trường hợp dữ liệu được lưu trữ bởi người dùng rất khó chuyển đổi thành bảng [1] | Lưu trữ cả dữ liệu bán cấu trúc và phi cấu trúc ít phức tạp hơn [16] |
| 8. | Ngôn ngữ truy vấn | SQL là ngôn ngữ thao tác dữ liệu duy nhất mà tất cả Cơ sở dữ liệu quan hệ sử dụng với các biến thể nhỏ trong quá trình triển khai [5] | Mỗi triển khai NoSQL có ngôn ngữ thao tác dữ liệu riêng [19] |
| 9. | Tính nhất quán | Có tính nhất quán cao với lược đồ nghiêm ngặt [8] | Có tính nhất quán kém với cách tiếp cận ít lược đồ hơn [7] |
| 10. | Bảo mật | Có cơ chế bảo mật mạnh được sử dụng để bảo vệ dữ liệu [1] | Để bảo mật được xử lý bởi phần mềm trung gian và không phải là một phần của cơ sở dữ liệu [7] |

**IV.Thách thức của NOSQL**

Một trong số những thách thức của NoSQL là việc nó thiếu đi một ngôn ngữ truy vấn tiêu chuẩn[19][20]. Có khoảng hơn 50 phương thức để triển khai NoSQL, với mỗi cách triển khai của riêng ngôn ngữ và giao diện [8]. Điều này đã thành trở ngại trong việc chấp nhận rộng rãi NoSQL vì các nhà phát triển khó có thể thành thạo tất cả các triển khai của ngôn ngữ thao tác NoSQL. Do đó, NoSQL có ít người dùng hơn Cơ sở dữ liệu quan hệ.

Một thách thức khác của NoSQL là tính bảo mật kém vì nó vẫn một công nghệ chưa đươc hoàn thiện [21]. Theo thiết kế, NoSQL cung cấp bảo mật hạn chế vì nhấn mạnh vào việc xử lý dữ liệu. Cơ sở dữ liệu NoSQL có thể bị tấn công bằng cách quét cổng port máy tính con số và dữ liệu ở phần còn lại không được mã hóa [17]. Đối với NoQuery dữ liệu đang được chuyển tiếp, có thể sử dụng truyền tải SSL, nhưng không được bật theo mặc định như trường hợp của MongoDB [21]. NoSQL không đủ khả năng ghi chép khiến nó dễ bị tấn công hơn trong các cuộc tấn công nội bộ mà không thể lần ra được dấu vết một cách dễ dàng.

**V. Lựa chọn mô hình cơ sở dữ liệu tốt hơn**

Cơ sở dữ liệu quan hệ rất dễ thực hiện, mạnh mẽ, phù hợp và an toàn nhưng quá cứng nhắc[18]. NoSQL thực hiện tốt trong việc xử lý một lượng lớn dữ liệu, hỗ trợ dữ liệu phi cấu trúc tuy nhiên ít phù hợp hơn và không an toàn. Không thể kết luận rằng cơ sở dữ liệu này tốt hơn cơ sở dữ liệu kia[8]. Mỗi mô hình cơ sỡ dữ liệu có thể được chọn tùy thuộc vào ứng dụng để phát triển. Đối với các ứng dụng nhỏ đòi hỏi tính nhất quán mạnh mẽ, nhà phát triển có thể chọn quan hệ cơ sở dữ liệu và đối với cơ sở dữ liệu động lớn, nhà phát triển có thể chọn NoQuery. Trong các web 2.0, 3.0 và dữ liệu lớn NoSQL là lựa chọn tốt hơn cơ sở dữ liệu quan hệ.

**VI. NoSQL như một sự thay thế cho cơ sở dữ liệu quan hệ.**

NoSQL có thể đã trở nên phổ biến, nhưng nó sẽ không thay thế hoàn toàn Cơ sở dữ liệu quan hệ [4]. Đối với dữ liệu lớn, mạng xã hội, hàng vạn thứ của internet, No SQL sẽ tiếp tục thống trị tuy nhiên, có rất nhiều ứng dụng vẫn sẽ tiếp tục phụ thuộc vào cơ sở dữ liệu quan hệ. NoSQL và cơ sở dữ liệu quan hệ sẽ còn tiếp tục tồn tại cạnh nhau để bổ sung những thiếu sót cho nhau.

**VII. Kết luận và công việc tương lai.**

Trong bài báo này, chúng tôi đã trình bày sự so sánh của NoSQL và cơ sở dữ liệu quan hệ dựa trên tài liệu hiện có. Nghiên cứu chỉ ra rằng các đặc tính của cơ sở dữ liệu quan hệ rất phù hợp để xử lý khối lượng dữ liệu có cấu trúc hạn chế. Nghiên cứu cũng cho thấy các tính năng NoSQL được thiết kế cho tốc độ và hiệu suất, với một lớp bảo mật mỏng không đạt tiêu chuẩn trên ngôn ngữ truy vấn. Công việc trong tương lai có thể được tiến hành để xác định khả năng cung cấp ngôn ngữ truy vấn tiêu chuẩn cho NoSQL

Người Giới Thiệu

[1] M. Abourezq and A. Idrissi, "Database-as-a-Service for Big Data: An

Overview," International Journal of Advanced Computer Science and

Applications, vol. 7, no. 1, pp. 157-177, 2016.

[2] A. . T. Kabakus and R. Kara, "A performance evaluation of in-memory

databases," Journal of King Saud University – Computer and

Information sciences, 2016.

[3] J. Batra and S. Batra, "MONGODB Versus SQL: A Case Study on

Electricity Data," Emerging Research in Computing, Information,,

2016.

[1] M. Abourezq and A. Idrissi, "Database-as-a-Service for Big Data: An

Overview," International Journal of Advanced Computer Science and

Applications, vol. 7, no. 1, pp. 157-177, 2016.

[2] A. . T. Kabakus and R. Kara, "A performance evaluation of in-memory

databases," Journal of King Saud University – Computer and

Information sciences, 2016.

[3] J. Batra and S. Batra, "MONGODB Versus SQL: A Case Study on

Electricity Data," Emerging Research in Computing, Information,,

2016.

[1] M. Abourezq and A. Idrissi, "Database-as-a-Service for Big Data: An

Overview," International Journal of Advanced Computer Science and

Applications, vol. 7, no. 1, pp. 157-177, 2016.

[2] A. . T. Kabakus and R. Kara, "A performance evaluation of in-memory

databases," Journal of King Saud University – Computer and

Information sciences, 2016.

[3] J. Batra and S. Batra, "MONGODB Versus SQL: A Case Study on

Electricity Data," Emerging Research in Computing, Information,,

2016.