

BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM
KHOA HỆ THỐNG THÔNG TIN & VIỄN THÁM



BÁO CÁO ĐỒ ÁN MÔN HQT CSDL

Đề tài: A Comparative Study of NoSQL and Relation Database

Thành viên của nhóm:

1. Huỳnh Long Vũ (0950080119)
2. Trần Quốc Huy (0950080124)
3. Nguyễn Vũ Nguyên (0950080155)

Lớp : 09_ĐHCNTT4

Thành phố Hồ Chí Minh, tháng 12, năm 2022

BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP. HCM
KHOA HỆ THỐNG THÔNG TIN & VIỄN THÁM



BÁO CÁO ĐỒ ÁN MÔN HQT CSDL

Đề tài: A Comparative Study of NoSQL and Relation Database

Thành viên của nhóm:

1. Huỳnh Long Vũ (0950080119)
2. Trần Quốc Huy (0950080124)
3. Nguyễn Vũ Nguyên (0950080155)

Lớp : 09_ĐHCNTT4

Thành phố Hồ Chí Minh, tháng 12, năm 2022

LỜI MỞ ĐẦU

Ngày nay với sự phát triển không ngừng của ngành công nghệ thông tin, ngày càng được ứng dụng cho nhiều thứ trong cuộc sống hằng ngày. Nói đến công nghệ thông tin không thể không nhắc đến hệ quản trị cơ sở dữ liệu. Là phần mềm hay hệ thống được thiết kế để quản lý cơ sở dữ liệu, hỗ trợ khả năng lưu trữ, sửa, xóa, tìm kiếm thông tin trong cơ sở dữ liệu.

Và có nhiều loại cơ sở dữ liệu khác nhau: từ những phần mềm nhỏ chạy trên máy tính cá nhân đến hệ quản trị phức tạp chạy trên một hoặc nhiều siêu máy tính.

Với sự gợi ý và giúp đỡ của giảng viên Phạm Trọng Huỳnh chúng em đã tìm hiểu về hai loại mô hình cơ sở dữ liệu là Relational Database và NoSQL. Và làm nên báo cáo theo bài báo đã tìm hiểu về sự khác nhau giữa chúng, rồi kết luận xem cái nào tốt hơn và thường được sử dụng hơn với đa số người.

Đề tài báo cáo đồ án Relational Database và NoSQL bao gồm các nội dung:

Chương 1: Giới thiệu về Relational Database và NoSQL

Chương 2: Relational Database và các loại NoSQL

Chương 3: Tính năng của Relational Database và mô hình cơ sở dữ liệu NoSQL

Chương 4: Những thách thức của NoSQL

Chương 5: Chọn mô hình cơ sở dữ liệu tốt nhất

Chương 6: NoSQL có thể được coi như một sự thay thế của Relational Database

Chương 7: Kết luận và các hoạt động trong tương lai

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin gửi lời cảm ơn đến thầy Phạm Trọng Huỳnh đã hỗ trợ nhóm em trong quá trình thực hiện đồ án này. Trong thời gian qua đã được thầy những kiến thức về hệ quản trị cơ sở dữ liệu giúp chúng em làm tốt đồ án này. Cũng xin gửi lời cảm ơn đến cô Nguyễn Thị Thảo Nguyên đã dạy cho chúng em những kiến thức về MongoDB. Do thời gian hạn hẹp cùng những vấn đề khác thì đồ án này có thể có nhiều sai sót mong được thầy góp ý thêm.

Kính chúc thầy và cô hạnh phúc và thành công hơn trong sự nghiệp của mình. Chúc thầy cô luôn mạnh khỏe!

Sinh Viên Thực Hiện

Nhóm 8

Tóm tắt:

Relational Database (Cơ sở dữ liệu quan hệ) và NoSQL (Cơ sở dữ liệu phi quan hệ) là các loại mô hình cơ sở dữ liệu đang cạnh tranh với nhau. Relational Database tồn tại từ năm 1979 và NoSQL ra đời từ năm 2000. Bởi nhu cầu của các ứng dụng hiện đại, đặc biệt là các trang Web 2.0, 3.0, nhờ nguồn dữ liệu lớn đã biến NoSQL trở thành databases phổ biến được lựa chọn. Việc chọn một mô hình databases phù hợp để sử dụng là một quyết định quan trọng mà những developer phải làm dựa trên các tính năng của một mô hình databases. Bài báo này so sánh các tính năng của Relational Database và NoSQL nhằm củng cố databases nào hỗ trợ nhu cầu các ứng dụng hiện đại tốt hơn. Bài báo cũng đưa ra những thách thức của NoSQL. Cuối cùng, bài báo kết luận bằng cách xác định liệu Relational Database có thể hoàn toàn bị thay thế bởi mô hình databases NoSQL hay không. Những phát hiện cho thấy Relational Database dựa trên mô hình ACID, mà mô hình này nhấn mạnh tính nhất quán, bảo mật và cung cấp ngôn ngữ truy vấn tiêu chuẩn tốt hơn. Tuy nhiên, Relational Databases có khả năng mở rộng kém, hiệu năng yếu, tốn nhiều chi phí hơn, gặp bất lợi khi hỗ trợ số lượng lớn người dùng và xử lý lượng dữ liệu lớn một cách khó khăn. Ngược lại, NoSQL dựa trên mô hình BASE, mô hình này chú trọng khả năng mở rộng và cung cấp lược đồ linh hoạt, cung cấp hiệu suất tốt hơn, chủ yếu mà mã nguồn mở, giá rẻ nhưng thiếu ngôn ngữ truy vấn tiêu chuẩn và không cung cấp đầy đủ các cơ chế bảo mật. Cả hai databases sẽ tiếp tục tồn tại song song với nhau mà không có cái nào tối ưu hơn. Việc lựa chọn databases để sử dụng sẽ phụ thuộc vào đặc tính của ứng dụng đang phát triển. Mỗi loại databases đều có những ưu và nhược điểm riêng, với relational database thì thiếu hỗ trợ cho dữ liệu phi cấu trúc trong khi NoSQL thiếu tiêu chuẩn hóa và bảo mật kém. Những ứng dụng hiện đại trong web 2.0, 3.0 và dữ liệu lớn rất thích hợp để sử dụng NoSQL nhưng vẫn nhiều các ứng dụng dựa vào Relational Databases.

Mục lục

LỜI MỞ ĐẦU.....	1
LỜI CẢM ƠN.....	2
Tóm tắt:	3
Mục lục.....	4
Chương 1. Giới thiệu.....	5
Chương 2. Relational Database và Các Loại NoSQL.....	7
Chương 3. Tính Năng Của Relational Database Và Mô Hình Cơ Sở Dữ Liệu NoSQL.....	8
3.1. Mã nguồn đóng và mã nguồn mở.....	8
3.2. Khả năng mở rộng.....	8
3.3. Chi phí.....	9
3.4. Khối lượng và sự đa dạng của dữ liệu.....	9
3.5. Khả dụng.....	9
3.6. Hiệu suất.....	10
3.7. Tính phức tạp.....	10
3.8. Ngôn ngữ truy vấn.....	11
3.9. Tính nhất quán.....	12
3.10. Bảo mật.....	12
Chương 4. Những Thách Thức Của NoSQL:.....	15
Chương 5. Chọn Mô Hình Cơ Sở Dữ Liệu Tốt Nhất.....	16
Chương 6. NoSQL Có Thể Được Coi Như Một Sự Thay Thế Cho Relational Database	17
Chương 7. Kết Luận Và Các Hoạt Động Trong Tương Lai.....	18
Demo	19
Tài liệu tham khảo.....	25

Chương 1. Giới thiệu

Cơ sở dữ liệu đã thay thế các flat file (tệp phẳng) làm một kho lưu trữ các nhóm dữ liệu lớn. Từ khi cơ sở dữ liệu ra đời, Relation Databases đã làm chủ hơn 30 năm cho đến năm 2000 khi mà NoSQL dần thay thế chúng trong một số ứng dụng. Bản chất cần thiết của ứng dụng là thay đổi theo thời gian và dẫn đến các ứng dụng có tương tác cao, xử lý được khối lượng dữ liệu lớn, chẳng hạn như các trang thương mại điện tử và phương tiện truyền thông xã hội. Việc cung cấp các tính năng tương tác với một cơ sở dữ liệu là một yêu cầu quan trọng đối với những cơ sở dữ liệu xây dựng nên ứng dụng web 2.0 và 3.0. Sự thay đổi trong nhu cầu ứng dụng đã khiến .

Relational Databases không ứng được nhu cầu của nhà phát triển và người dùng. Các công ty như Google, Facebook và Yahoo đã chuyển sang dùng NoSQL để giảm thiểu những thiếu sót của Relational Databases. Bất chấp xu hướng trên, nhiều công ty vẫn sử dụng Relational Databases. Các công ty này có đặc điểm là khối lượng dữ liệu hạn chế đồng thời đề cao tính chắc chắn. Relational Databases dựa trên i.e (Internet Explorer) mô hình ACID. Atomicity (Tính nguyên tử), Consistency (Tính nhất quán), Isolation (Tính độc lập), và Durability (Tính bền vững). Atomicity đảm bảo tính hoàn chỉnh của giao dịch. Consistency mang lại sự ổn định của dữ liệu trong cơ sở dữ liệu, Isolation đảm bảo được sự độc lập của các giao dịch được thực hiện cùng lúc và Durability đảm bảo các giao dịch được lưu trữ dù có xảy ra sự cố. ACID cung cấp sự chắc chắn và tiện lợi như những đặc tính mạnh mẽ giúp Relational Databases trở nên phổ biến. Mặt khác, NoSQL dựa trên mô hình BASE (Basically Available, Soft State and Eventually Consistent). Bản chất phân tán của NoSQL mang đến phần nào có ích của dữ liệu khi một số phần của cơ sở dữ liệu phân tán không hoạt động hoặc không thể truy cập vì thế, thuật ngữ Basically Available (Tính sẵn sàng ở mức cơ bản). Soft State (Trạng thái mềm) cho phép dữ liệu thay đổi theo thời gian hoặc không cần đầu vào. Cuối cùng, Consistent (Tính nhất quán) sẽ đảm bảo dữ liệu nhất quán trong tương lai chứ

không phải sau một thao tác. BASE cung cấp cho NoSQL khả năng mở rộng quy mô dễ dàng, mang lại hiệu suất tốt hơn và mang lại sự tiện lợi cao hơn cho người dùng.

Bài viết này dựa trên những đánh giá của những tài liệu trước đây và bắt đầu bằng việc mô tả về Relational Databases và mô hình cơ sở dữ liệu NoSQL. Sau đó, cuộc tranh luận chuyển sang so sánh cách tính năng của Relational Databases và NoSQL, sau đó là những thách thức của NoSQL. Dựa trên những tính năng, các tác giả đã cố gắng xác định xem NoSQL có tốt hơn Relational Databases trong việc hỗ trợ nhu cầu của các ứng dụng cơ sở dữ liệu hiện đại và liệu NoSQL có thể thay thế hoàn toàn Relationases hay không.

Chương 2. Relational Database và Các Loại NoSQL

Chỉ có một dạng của Relation Database được dựa trên mô hình quan hệ. Nhiều tổ chức, đã thay đổi các yêu cầu ứng dụng của họ để phù hợp với sự nghiêm ngặt của lược đồ đã được cố định từ trước ở Relational Databases. Lược đồ nghiêm ngặt yêu cầu ứng dụng phải tuân theo yêu cầu của cơ sở dữ liệu thay vì ngược lại. Những ví dụ về Relational Databases là MySQL, Microsoft SQL Server và Postgres.

Cơ sở dữ liệu NoSQL có khá nhiều loại và từ những tài liệu đã được đánh giá, có bốn loại nổi bật. Chúng bao gồm Key-Value (Khóa – Giá trị), Document oriented (hướng tài liệu), Column databases (cơ sở dữ liệu dạng cột) and Graph databases (cơ sở dữ liệu đồ thị). Trong Key-Value, dữ liệu được lưu trữ dưới dạng tập hợp các khóa và giá trị, trong đó khóa là 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. Key – Value dễ sử dụng nhưng nó không hỗ trợ xử lý mối quan hệ giữa các mục dữ liệu. Ví dụ về cơ sở dữ liệu key value bao gồm Memcached và Redis. Cơ sở dữ liệu Document oriented sử dụng khóa và tài liệu làm thuộc tính trong đó khóa đề cập đến toàn bộ tài liệu. Ví dụ về cơ sở dữ liệu Document oriented là MongoDB và CouchDB, rất phù hợp để xử lý các cấu trúc dữ liệu phức tạp những vẫn thiếu khả năng xử lý các mối quan hệ giữa các mục dữ liệu. Column databases chứa các hàng và cột tương tự như Relational Databases nhưng mỗi cột được lưu trữ trong một tệp riêng biệt. Một khóa trong cơ sở dữ liệu column oriented ám chỉ đến một cột. Các giá trị thuộc tính khác được lưu trữ bao gồm giá trị và thời gian đánh dấu. Bigtable và Cassandra là những ví dụ và theo thiết kế, chúng kém linh hoạt hơn nhưng cung cấp năng suất lớn hơn. Graph databases biểu thị dữ liệu dưới dạng đồ thị được kết nối và dựa trên lý thuyết đồ thị. Graph databases ít có khả năng mở rộng hơn nhưng hỗ trợ kết nối tốt hơn. Các ví dụ bao gồm GraphDB và OrientDB.

Chương 3. Tính Năng Của Relational Database Và Mô Hình Cơ Sở Dữ Liệu NoSQL

3.1. Mã nguồn đóng và mã nguồn mở

Relational Databases bao gồm các mã nguồn mở và độc quyền. Những loại Relational Databases độc quyền như Oracle có khả năng mở rộng quy mô hơn so với các đối tác mã nguồn mở như MySQL. Tuy nhiên, rất nhiều mô hình cơ sở dữ liệu NoSQL là mã nguồn mở ví dụ như MongoDB, CouchDB và Cassandra. Bản chất mã nguồn mở của NoSQL mang lại nhiều cơ hội hơn cho các nhà nghiên cứu trong việc tìm hiểu các tính năng của cơ sở dữ liệu và cung cấp dung lượng lưu trữ rẻ hơn cho những người dùng không có khả năng mua các mô hình cơ sở dữ liệu độc quyền.

3.2. Khả năng mở rộng

Relational Databases, thường mở rộng quy mô, phải nâng cấp phần cứng máy tính cho máy chủ để cho nó chạy hiệu quả hơn. Điều này khiến cho các quản trị viên cần phải nỗ lực hơn trong việc nâng cấp Relational Databases. Phương pháp nâng cấp này đối mặt với khó khăn 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 của phần cứng có giá trị không đổi do nhà sản xuất quy định. Nghĩa là Relational Databases có khả năng mở rộng nhưng sẽ có giới hạn về mức độ mở rộng do nó được xác định bởi phần cứng. Để cung cấp khả năng mở rộng, NoSQL yêu cầu sử dụng máy chủ hàng hóa, mở rộng theo chiều ngang. Việc mở rộng quy mô theo chiều ngang không bị ảnh hưởng bởi các giới hạn phần cứng vì các máy chủ nhỏ hơn, rẻ hơn và yếu hơn có thể được kết hợp để mang lại khả năng mở rộng quy mô to hơn thay vì có một máy chủ đắt tiền. Điều đó giúp việc thực hiện trở nên dễ dàng vì các máy ảo có thể được sử dụng làm máy chủ hàng hóa trong tình huống không có phần cứng thực tế. Máy ảo có thể được thêm và xóa mà không gây ảnh hưởng đến cơ sở dữ liệu. Các ứng dụng internet hiện đại như phương

tiện truyền thông yêu cầu mở rộng quy mô to mà Relational Databases không đáp ứng được nhưng được cung cấp trong NoSQL.

3.3. Chi phí

Relational Databases tốt nhất là độc quyền và do đó, đòi hỏi số tiền đầu tư đáng kể 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 khác. Điều này làm cho Relational Databases trở thành một phương pháp lưu trữ dữ liệu đắt tiền. NoSQL chủ yếu là mã nguồn mở, làm cho nó trở thành một giải pháp thay thế rẻ hơn cho Relational Databases. 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 chi phí thấp hấp dẫn cho các tổ chức.

3.4. Khối lượng và sự đa dạng của dữ liệu

Các ứng dụng Internet làm tăng lượng dữ liệu mà cơ sở dữ liệu phải xử lý. Internet đã chứng kiến sự xuất hiện của web 2.0 và 3.0 đã làm tăng số lượng và sự đa dạng của dữ liệu phải được lưu trữ. Sự xuất hiện của dữ liệu lớn cũng đã làm tăng số lượng và sự đa dạng của dữ liệu. Relational Databases không thể xử lý khối lượng lớn dữ liệu từ các nguồn này. NoSQL vượt trội trong việc xử lý lượng dữ liệu lớn làm cho nó phù hợp với các ứng dụng internet sử dụng nhiều dữ liệu. Điều này có thể thấy ở các công ty như Google, Facebook và Yahoo đã chuyển sang NoSQL.

3.5. Khả dụng

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, Relational Databases thường bị lỗi ngay cả đối với các máy chủ rất mạnh. Tính khả dụng bị hạn chế hơn nữa do mở rộng Relational Databases.

Một điểm thất bại duy nhất là không phù hợp với các ứng dụng internet hiện đại ngày nay mà người dùng đặt niềm tin lớn vào để hỗ trợ họ trong cuộc sống hàng ngày. Do đó, bản chất phân tán của NoSQL làm cho nó trở thành lựa chọn tốt hơn để luôn cung cấp cho người dùng tính khả dụng ngay cả trong trường hợp lỗi phần cứng. Các nguyên tắc cơ bản tích hợp sẵn của NoSQL giúp truy cập các phần của cơ sở dữ liệu trong trường hợp có lỗi. Bất kể lỗi hệ thống, người dùng được đảm bảo tiếp tục truy cập vào cơ sở dữ liệu.

3.6. Hiệu suất

Relational Databases cần nhiều thời gian hơn để xử lý thông tin, khiến chúng chậm hơn so với NoSQL xử lý nhanh. Hiệu suất của NoSQL được cải thiện hơn nữa vì nó truy xuất dữ liệu từ bộ nhớ khả biến, không giống như Relational Databases truy xuất dữ liệu từ bộ nhớ không khả biến. Theo thiết kế, bộ nhớ khả biến nhanh hơn bộ nhớ không khả biến. Trong các ứng dụng tìm kiếm trên Internet, NoSQL đã vượt trội so với cơ sở dữ liệu quan hệ trong việc truy xuất thông tin. Các thử nghiệm đã được tiến hành để kiểm tra hiệu suất của NoSQL và Relational Databases. So sánh cơ sở dữ liệu quan hệ với MongoDB cho thấy MongoDB có hiệu suất tốt hơn khi đọc, cập nhật và truy vấn cơ bản, trong khi SQL chỉ hoạt động tốt khi cập nhật các thuộc tính không phải khóa .

3.7. Tính phức tạp

Relational Databases tạo ra dữ liệu phức tạp trong đó dữ liệu do người dùng lưu trữ khó chuyển đổi thành bảng. Việc nhấn mạnh vào việc lưu trữ dữ liệu có cấu trúc trong Relational Databases mang lại sự phức tạp này. Khi một số thao tác đọc hoặc ghi đơn giản có thể đủ, chẳng hạn như trong phương tiện truyền thông xã hội, các truy vấn và giao dịch phức tạp của Relational Databases có thể không cần thiết. NoSQL có thể lưu trữ dữ liệu bán cấu trúc và phi cấu trúc.

Khả năng lưu trữ dữ liệu bán cấu trúc và phi cấu trúc của NoSQL mang lại sự linh hoạt cần thiết để hỗ trợ các loại dữ liệu khác nhau ở trạng thái ban đầu mà không làm mất thông tin. Ví dụ: chuyển đổi bản ghi âm khiếu nại của khách hàng thành văn bản để lưu trữ trong Relational Databases dẫn đến mất thông tin về cảm nhận của khách hàng. Những thông tin như vậy có thể được lưu giữ trong NoSQL vì các bản ghi có thể được lưu trữ ở trạng thái của chúng mà không cần chuyển đổi.

3.8. Ngôn ngữ truy vấn

Relational Databases có nền tảng vững chắc và được ghi chép đầy đủ trên SQL. SQL là ngôn ngữ thao tác dữ liệu duy nhất được sử dụng bởi tất cả các Relational Databases. Tuy nhiên, việc triển khai SQL hơi khác nhau đối với các cơ sở dữ liệu quan hệ khác nhau được sử dụng. Nền tảng vững chắc do SQL cung cấp đã làm cho Relational Databases trở nên phổ biến đối với các nhà phát triển vì quá trình học tập để triển khai bất kỳ Relational Databases là tương đối ngắn. NoSQL vẫn thiếu nền tảng này vì nó dựa vào API hướng đối tượng để thao tác dữ liệu. Mỗi triển khai của NoSQL có ngôn ngữ thao tác dữ liệu riêng, đòi hỏi các nhà phát triển phải dành thời gian học hỏi khi phát triển một loại mô hình NoSQL khác với những gì họ quen thuộc. Có nhiều cách để truy vấn NoSQL, giới hạn số lượng truy vấn được hỗ trợ do mỗi lần triển khai phải cung cấp các truy vấn duy nhất của riêng nó. Các yêu cầu của web 2.0 và 3.0 yêu cầu các phương pháp phát triển nhanh, mà NoSQL có thể không đáp ứng được do thời gian phát triển tăng lên vì các nhà phát triển cần học ngôn ngữ triển khai.

3.9. Tính nhất quán

Relational Databases cung cấp tính nhất quán mạnh mẽ hơn thông qua các lược đồ nghiêm ngặt [8]. Tính năng này làm cho Relational Databases hy sinh khả năng sử dụng vì không miễn phí. Tính nhất quán cao rất tốt để cung cấp chế độ xem dữ liệu nhất quán ngay sau khi một hành động được thực hiện. Tuy nhiên, trong các ứng dụng như phương tiện truyền thông xã hội, tính linh hoạt quan trọng hơn tính nhất quán. NoSQL cung cấp tính khả dụng cao hơn nhưng tính nhất quán kém hơn. Vì vậy, đối với phương tiện truyền thông xã hội, NoSQL là tùy chọn lưu trữ tốt hơn so với Relational Databases.

3.10. Bảo mật

Relational Databases phải đối mặt với nhiều thách thức về bảo mật, chẳng hạn như SQL injection(tác động đến cơ chế có sẵn và lợi dụng nó) và cross-site scripting(tấn công bằng mã độc rồi mạo danh người dùng và ăn cắp dữ liệu). Bất chấp những thách thức này, SQL có các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu, bao gồm xác thực, ủy quyền, mã hóa, toàn vẹn và kiểm tra . 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 . Điều này khiến cơ sở dữ liệu dễ bị tấn công. Ngoài ra, các cơ chế bảo mật được triển khai trong phần mềm trung gian phải được triển khai theo cách không ảnh hưởng đến khả năng mở rộng và hiệu suất.

BẢNG 1
SO SÁNH RELATIONAL DATABASES VÀ NoSQL

Tiêu chí	Relational Database	NoSQL
Sự đa dạng	Nền tảng mã nguồn đóng và mã nguồn mở[5]	NoSQL chủ yếu là mã nguồn mở [9]
Khả năng mở rộng quy mô	Mở rộng quy mô bằng cách nâng cấp phần cứng của máy chủ [1].	Mở rộng quy mô với máy chủ hàng hóa [8]
Chi phí	Cách tốn kém để lưu trữ dữ liệu [7]	Rẻ hơn vì nó là mã nguồn mở và rẻ để nâng cấp [9]
Khối lượng dữ liệu	Xử lý dữ liệu hạn chế [11].	Xử lý dữ liệu khối lượng lớn, đặc biệt là dữ liệu lớn [12].
Khả dụng	Là một điểm thất bại [5]	Bản chất phân tán luôn cung cấp cho người dùng tính khả dụng trong trường hợp lỗi phần cứng [10]
Hiệu suất	Cần thêm thời gian để xử lý thông tin làm chúng chậm lại [13]	Cho ra hiệu suất truy vấn lớn hơn[16]
Tính phức tạp	Tạo dữ liệu phức tạp trong trường hợp người dùng lưu trữ dữ liệu khó chuyển thành bảng [1]	Lưu trữ dữ liệu bán cấu trúc và phi cấu trúc ít phức tạp hơn [16]

Ngôn ngữ truy vấn	SQL là ngôn ngữ thao tác dữ liệu duy nhất được sử dụng bởi tất cả cơ sở dữ liệu quan hệ, với cách triển khai hơi khác nhau [5]	Mỗi lần thực hiện NoSQL có ngôn ngữ thao tác dữ liệu riêng [19]
Tính nhất quán	Có tính nhất quán cao với lược đồ nghiêm ngặt [8]	Ít nhất quán hơn với các phương pháp không có lược đồ [7]
Bảo mật	Có các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu [1]	Việc bảo mật được xử lý bởi phần mềm trung gian và không phải là một phần của cơ sở dữ liệu [7]

Chương 4. Những Thách Thức Của NoSQL:

Một trong những thách thức của NoSQL là nó thiếu ngôn ngữ truy vấn chuẩn. Có hơn 50 cách triển khai NoSQL, với mỗi cách cung cấp ngôn ngữ và giao diện riêng. Điều này đã cản trở sự chấp nhận rộng rãi của NoSQL vì các nhà phát triển khó có thể thành thạo tất cả các ngôn ngữ thao tác để triển khai NoSQL. Do đó, NoSQL có ít người dùng hơn Cơ sở dữ liệu quan hệ(Relational Databases).

Một thách thức khác của NoSQL là bản thân nó có tính bảo mật kém vì nó vẫn là một công nghệ chưa trưởng thành hoàn toàn. Theo thiết kế được đưa ra , NoSQL cung cấp khả năng bảo mật hạn chế vì bản thân nó tập trung vào xử lý dữ liệu. Cơ sở dữ liệu NoSQL có thể bị tấn công bằng cách quét các số cổng đã biết và dữ liệu ở trạng thái nghỉ không được mã hóa. Đối với dữ liệu NoSQL đang chuyển tiếp, truyền tải SSL có thể được sử dụng nhưng nó không được bắt theo mặc định như trường hợp của MongoDB. NoSQL không có khả năng ghi nhật ký và điều đó khiến nó dễ bị tấn công bởi các cuộc tấn công từ bên trong mà không thể dễ dàng truy soát được.

Chương 5. Chọn Mô Hình Cơ Sở Dữ Liệu Tốt Nhất

Relational Databases (Cơ sở dữ liệu quan hệ) dễ triển khai, mạnh mẽ, nhất quán và an toàn nhưng lại quá cứng nhắc. NoSQL lại hoạt động tốt trong việc xử lý khối lượng dữ liệu khổng lồ, hỗ trợ dữ liệu phi cấu trúc nhưng bản thân nó lại kém nhất quán và không an toàn. Chúng ta không thể kết luận rằng cơ sở dữ liệu này tốt hơn cơ sở dữ liệu kia bởi vì mỗi mô hình cơ sở dữ liệu đều có thể được chọn tùy thuộc vào ứng dụng được phát triển. Đối với các ứng dụng nhỏ yêu cầu tính nhất quán cao, các nhà phát triển có thể chọn Relational Databases và đối với cơ sở dữ liệu động lớn, nhà phát triển có thể chọn NoSQL. Trong các ứng dụng web 2.0, 3.0 và dữ liệu lớn, NoSQL là lựa chọn tốt hơn so với Relational Databases (Cơ sở dữ liệu quan hệ).

Chương 6. NoSQL Có Thể Được Coi Như Một Sự Thay Thế Cho Relational Database

NoSQL có thể đã trở nên phổ biến, nhưng nó sẽ không thay thế hoàn toàn Relational Databases (Cơ sở dữ liệu quan hệ) . Đối với các dữ liệu lớn, mạng xã hội, internet ... , NoSQL sẽ tiếp tục giữ lấy vị trí đứng đầu nhưng vẫn có rất nhiều ứng dụng vẫn sẽ tiếp tục dựa vào Relational Databases (Cơ sở dữ liệu quan hệ). NoSQL và Relational Database vẫn sẽ tiếp tục tồn tại song song với nhau để có thể bổ sung cho những thiếu sót của nhau.

Chương 7. Kết Luận Và Các Hoạt Động Trong Tương Lai

Trong bài báo cáo này, chúng em đã trình bày so sánh giữa NoSQL và Cơ sở dữ liệu quan hệ dựa trên các tài liệu hiện có. Nghiên cứu cho thấy rằng các tính năng 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 rằng các tính năng của NoSQL được thiết kế cho khả năng mở rộng và hiệu suất, với lớp bảo mật mỏng trên ngôn ngữ không chuẩn Query. Các hoạt động phát triển trong tương lai có thể được tiến hành để xác định khả năng cung cấp ngôn ngữ truy vấn tiêu chuẩn cho NoSQL...Sau khi đã lên ý tưởng chúng em tiếp tục làm rõ các vấn đề liên quan đến ý tưởng đã đề ra.

Xây dựng giao diện:

- Mục đích là xây dựng được một phần mềm có giao diện dễ nhìn, phù hợp với môi trường công tác, làm việc của người quản lý thư viện.
- Yêu cầu là thiết kế được một giao diện dễ nhìn dễ hiểu và dễ sử dụng.
- Phần thiết kế giao diện được chia làm hai phần: giao diện đăng nhập và giao diện sau khi đăng nhập (giao diện làm việc).
- Giao diện đăng nhập gồm đầy đủ phần điền tên đăng nhập và mật khẩu, nút nhấn đăng nhập, phía trên là tên phần mềm.
- Giao diện làm việc gồm phần chọn sách, nhà xuất bản, phần nội dung nơi hiển thị sách, thông tin người đọc, còn có những nút nhấn tính năng của phần mềm.





Demo

➤ So sánh ngôn ngữ giữa Relational Database và NoSQL. Cụ thể SQL Server và MongoDB

- Sự tương quan giữa các thuật ngữ:

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (mặc định là _id)

- Sự tương quan trong cấu trúc câu lệnh
 - Các thành phần trong thao tác tìm kiếm của MongoDB :

<code>db.users.find(</code>	 <code>collection</code>
<code> { age: { \$gt: 18 } },</code>	 <code>query criteria</code>
<code> { name: 1, address: 1 }</code>	 <code>projection</code>
<code>).limit(5)</code>	 <code>cursor modifier</code>

- Sơ đồ tiếp theo hiển thị cùng một truy vấn trong SQL:

SELECT	_id, name, address	←	projection
FROM	users	←	table
WHERE	age > 18	←	select criteria
LIMIT	5	←	cursor modifier

1. Insert

- SQL Server:

```
insert into teacher(teacher_id, teacher_name, dept_name, salary, status_teacher)
values ('T01', 'Mr.Hải', 'Technology', '7000', 'N');
```

- MongoDB:

```
db.demoo.insertOne(
{
  "teacher_id": "T01",
  "teacher_name": "Mr.Hải",
  "dept_name": "Technology",
  "salary": "7000",
  "status_teacher": "N"
})
```

2. Truy vấn

- Hiện thị tất cả dữ liệu của bảng teacher
 - SQL Server

```
SELECT * FROM teacher;
```

- MongoDB

```
db.demoo.find({})
```

- Cho biết tất cả các cột của teacher có id = T01

- SQL Server

```
SELECT * FROM teacher
WHERE teacher_id = 'T01'
```

- MongoDB

```
db.demoo.find( {teacher_id: "T01"} )
```

3. Cập nhập

- Thay đổi giá trị trường dept_name của teacher có id = T02 thành Technology

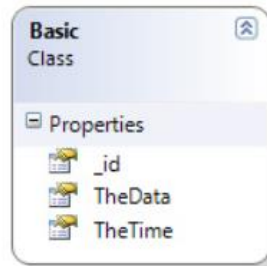
- SQL Server

```
UPDATE teacher
SET
dept_name = 'Technology'
where teacher_id = 'T02'
```

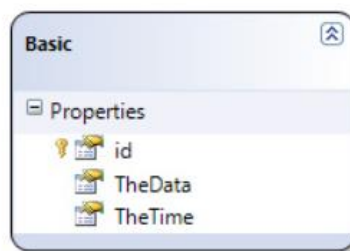
- MongoDB

```
db.demoo.update(
  { 'teacher_id' : 'T02' },
  { $set: { 'dept_name': 'Technology' } } )
```

- So sánh hiệu suất hoạt động của Relational Database và NoSQL:
- So sánh chức năng Insert:
 - Quá trình này được thực hiện bằng cách chèn 50.000 đối tượng độc lập với mô hình mẫu như sau:

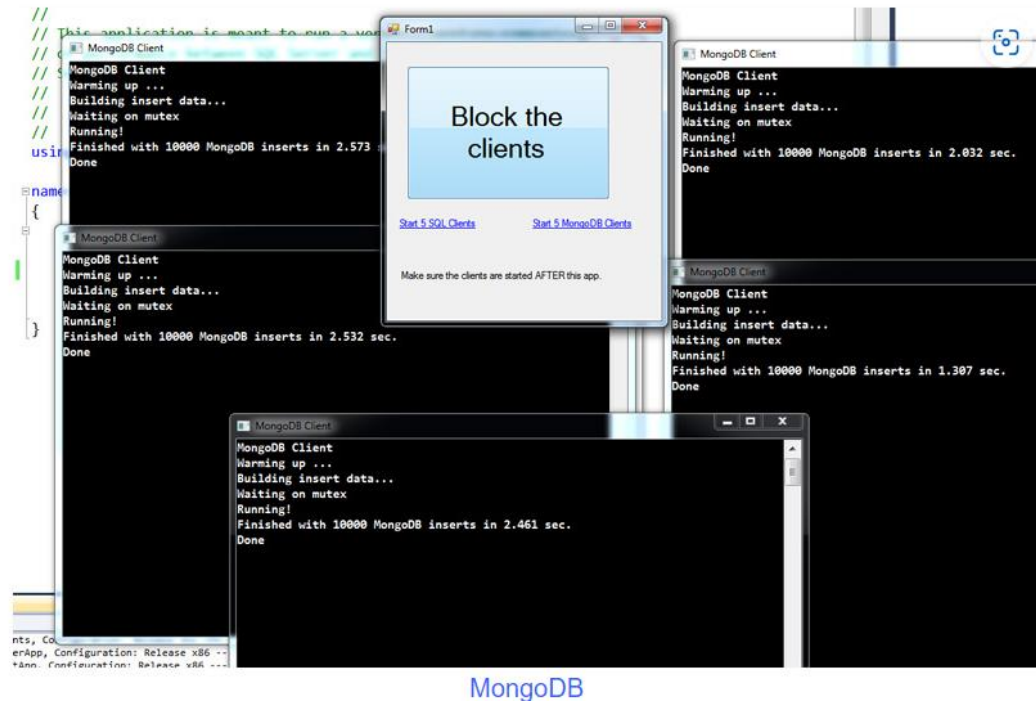


MongoDB

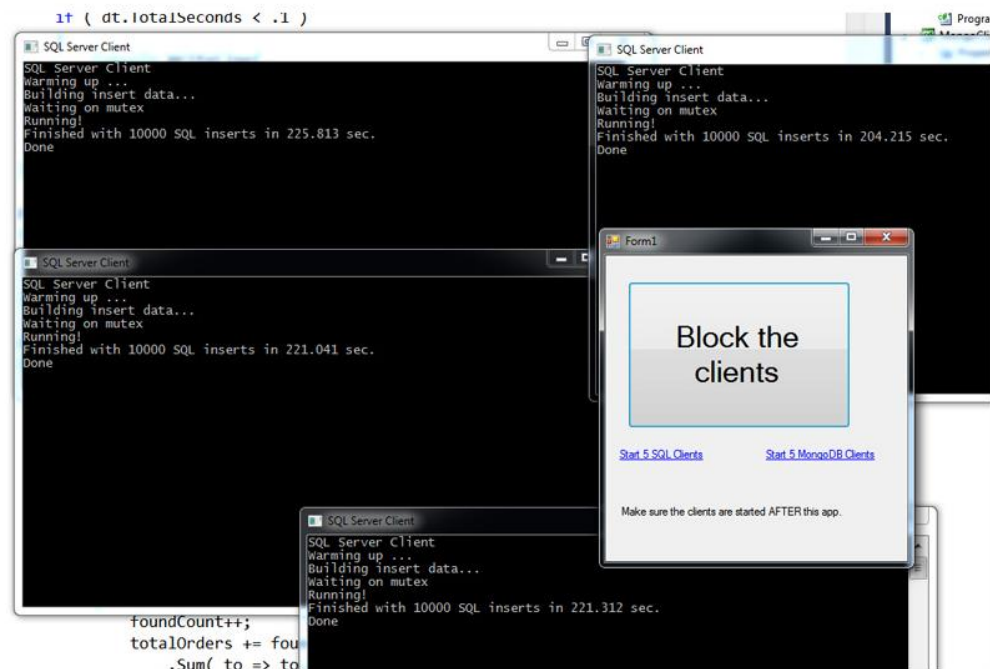


SQL Server

- Sau đó, chạy 5 ứng dụng client vào cơ sở dữ liệu với chức năng insert:

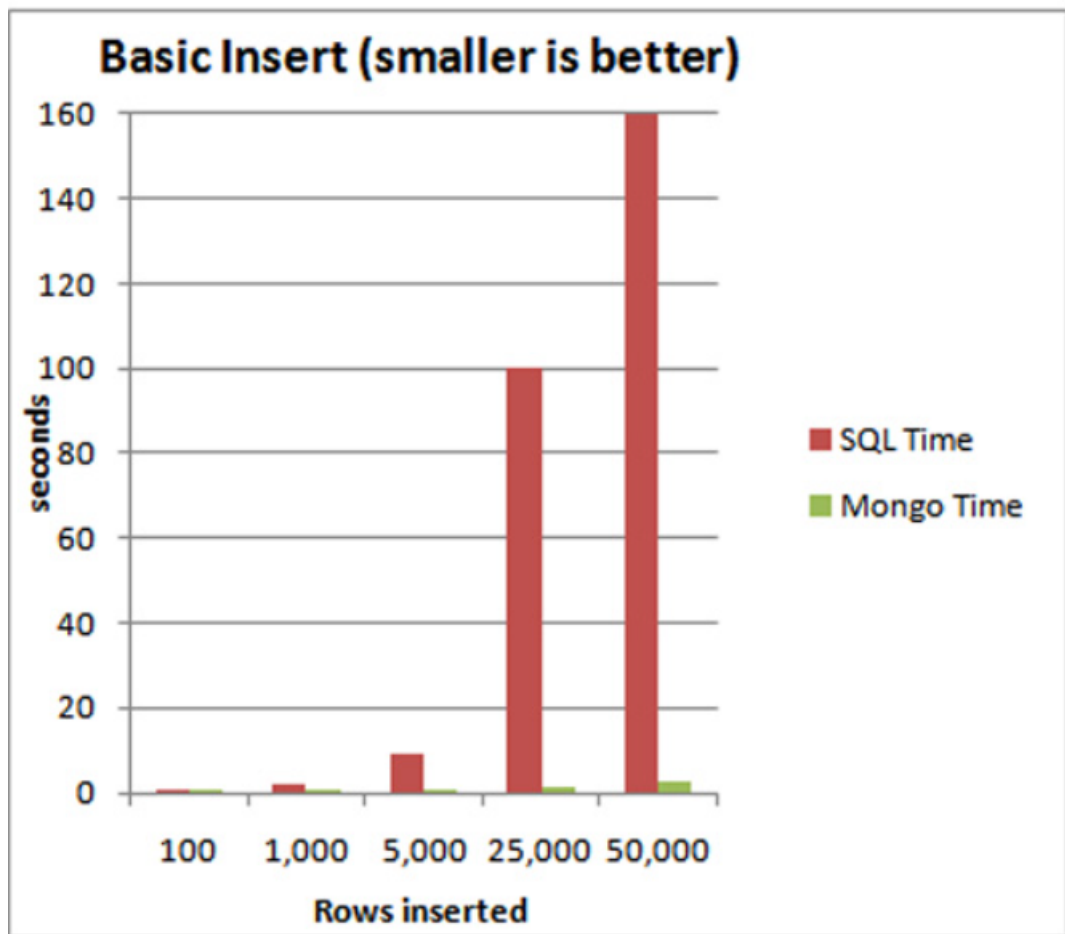


MongoDB



SQL Server

- Có thể thấy sự khác biệt giữa 2 giây đối với MongoDB và 205 giây đối với SQL Server



Biểu đồ so sánh tốc độ Insert giữa MongoDB và SQL Server

Tài liệu tham khảo

Sách:

- [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.
- [4] H. L. Zhen, H. Beda, M. Doug, L. Ying and J. C. Hui, "Closing the functional and Performance Gap between SQL and NoSQL," 2016.
- [5] M. A. Mohamed, "Relational vs. NoSQL Databases: A Survey," International Journal of Computer and Information Technology, vol. 3, no. 03, 2014.
- [6] S. Priyanka and AmitPal, "A Review of NoSQL Databases, Types and Comparison with Relational Database," International Journal of Engineering Science and Computing,, vol. 6, no. 5, pp. 4963-4966, 2016.
- [7] A. K. Zaki, "NoSQL Databases: New Milleneum Database For Big Data, Big Users, Cloud Computing and Its Security Challenges," International Journal of Research in Engineering and Technology, vol.3, no. 3, 2014.
- [8] A. Singh, "NoSQL : A New Horizon in Big Data," International Journal of Scientific Research in Science, Engineering and Technology, vol. 2, no. 2, 2016.

- [9] W. Kim, "Web data stores (aka NoSQL databases): a data model and data management perspective," *Int. J. Web and Grid Services*, Vol. 10, No. 1, 2014, vol. 10, no. 1, pp. 100-110, 2014.
- [10] S. Sharma, R. Shandilya, S. Patnaik and A. Mahapatra, "Leading NoSQL models for handling Big Data:," *Int. J. Business Information Systems*, vol. 22, no. 1, 2016.
- [11] A. B. Moniruzzaman and S. A. Hossain, "NoSQL Database: New Era of Databases for Big data Analytics -," *International Journal of Database Theory and Application*, vol. 06, no. 4, 2013.
- [12] A. Nayak, A. Poriya and D. Poojary, "Types of NOSQL Databases and its Comparison with Relational Databases," *International Journal of Applied Information Systems*, vol. 5, no. 4, 2013.
- [13] L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes and J. Abramov, "Security Issues in NoSQL Databases," in *International Joint Conference of IEEE TrustCom, IEEE ICSS-11*, 2011.
- [14] Z. Parker, S. Poe and S. V. Vrbsky, "Comparing NoSQL MongoDB to an SQL DB," vol. *ACMSE'13*, 2013.
- [15] S. Srinivas and A. Nair, "Security Maturity in NoSQL Databases – Are they Secure Enough to Haul the Modern IT Applications?," in *International Conference on Advances in Computing, Communications and Informatics IEEE*, 2015.
- [16] J. Kepner, D. Hutchison, H. Jonathan, T. Mattison, S. Samsi and A. Ruether , "Associative Array Model of SQL, NoSQL, and NewSQL Databases," 2016.
- [17] E. Barbierato, M. Gribaudo and M. Iacono, "Performance evaluation of NoSQL big-data applications using," *Future Generation Computer Systems*, no. 37, pp. 345-353, Elsevier, 2014.

- [18] L. Rocha, F. Vale, E. Cirilo, D. Barbosa and M. Fernando, "A Framework for Migrating Relational," *Procedia Computer Science*, vol.51, pp. 2593-2602, Elsevier, 2015.
- [19] K. K.-Y. Lee and W.-C. Tang, "Alternatives to relational database:Comparison of NoSQL," *Computer Methods and Programs in Biomedicine*, no. 110, pp. 99-109, Elsevier, 2013.
- [20] A. Makrisa, K. Tserpesa, V. Andronikoub and D. Anagnostopoulo, "A classification of NoSQL data stores based on key design," *Procedia Computer Science*, no. 97, pp. 94-103, Elsevier, 2016.
- [21] P. Atzeni, F. Bugiotti and L. Rossi, "Uniform access to No SQL systems," *Information Systems*, no. 43, pp. 117-133, Elsevier, 2014.

Web:

- [1] <https://www.mongodb.com>
- [2] <https://www.mysql.com/>

Links:

- [1] <https://vn.bitdegree.org/huong-dan/relational-database-la-gi/>
- [2] <https://topdev.vn/blog/mongodb-la-gi/>
- [3] <https://blog.vinahost.vn/nosql-la-gi>
- [4] <https://www.tma.vn/Hoi-dap/Cam-nang-nghe-nghiep/Nhung-diem-khac-biet-giua-SQL-va-NoSQL/26803>
- [5] <https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data>
- [6] <https://quantrimang.com/cong-nghe/so-sanh-hieu-suat-hoat-dong-cua-mongodb-va-sql-server-2008-84584>

[7] <https://github.com/dotnet/docs/blob/main/docs/architecture/cloud-native/relational-vs-nosql-data.md>