

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG
KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM



ĐỒ ÁN

**HYBRID DATABASE SYSTEM FOR BIG DATA STORAGE AND
MANAGEMENT**

Môn học:

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Giảng viên hướng dẫn: Phạm Trọng Huỳnh

Sinh viên thực hiện: Nhóm 5

Lớp: 09_CNTT04

TP.HCM, ngày 7 tháng 12 năm 2022

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG
KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM



ĐỒ ÁN

**HYBRID DATABASE SYSTEM FOR BIG DATA STORAGE AND
MANAGEMENT**

Môn học:

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Giảng viên hướng dẫn: Phạm Trọng Huỳnh

Sinh viên thực hiện: Nhóm 5

1. Nguyễn Lê Thảo Quyên

2. Đỗ Huỳnh Phương Trinh

3. Lý Thị Quế Trân

Lớp: 09_CNTT04

TP.HCM, ngày 7 tháng 12 năm 2022

LỜI MỞ ĐẦU

Cùng với sự phát triển của thời đại ngày nay các ngành công nghệ ngày càng được sử dụng rộng rãi trên hầu hết mọi lĩnh vực. Việc quá nhiều các lĩnh vực và mạng lưới sử dụng đã dẫn đến sự xuất hiện một lĩnh vực dữ liệu đồ sộ có qui mô không thể tưởng tượng nổi gọi là dữ liệu lớn. Nó được coi là khối lượng dữ liệu khổng lồ mà các hệ thống cơ sở dữ liệu thông thường không thể xử lý được. Sự gia tăng khổng lồ về dữ liệu mở ra cơ hội lớn cho những thành tựu khoa học, cải thiện kinh doanh bằng phương pháp mới. Hệ thống cơ sở dữ liệu quan hệ phục vụ đã phục vụ suốt nhiều năm tuy nhiên những năm này, sự ra đời của NoSQL là một giải pháp tốt. Bởi vì cơ sở dữ liệu quan hệ không được thiết kế để lưu trữ hoặc quản lý tốc độ cao, dữ liệu phi cấu trúc hay sự phát triển nhanh. Mà sự ra đời của NoSQL dường như đã giải quyết được những vấn đề đó nên không lạ gì khi các doanh nghiệp dần chuyển sang sử dụng hệ cơ sở dữ liệu này. Tuy nhiên “lợi bất cập hại” dần chúng ta cũng nhận ra chúng có những hạn chế nhất định. Chúng dường như đã bỏ qua sự toàn vẹn, bảo mật mà cơ sở dữ liệu quan hệ chưa thể cung cấp. Và đó là một nguyên nhân dẫn đến sự ra đời của một hệ thống cơ sở dữ liệu lai sử dụng cơ sở dữ liệu MongoDB. Trước khi lưu dữ liệu chúng sẽ được phân loại thành dữ liệu cấu trúc và phi cấu trúc để lưu vào trong hệ cơ sở dữ liệu. Nếu phi cấu trúc sẽ được lưu vào MongoDB. Còn có cấu trúc thì lưu vào MySQL. Hệ thống lai tạo nên hệ thống có thể hoạt động riêng biệt. Phát triển các hướng tiếp cận khác nhằm cung cấp lợi ích của cả hai hệ thống thành một hệ thống cơ sở dữ liệu duy nhất. Hệ thống này dường như đã khắc phục những hạn chế của hai hệ cơ sở dữ liệu MongoDB và MySQL. Đây có thể là một hướng phát triển tốt trong lĩnh vực quản lý dữ liệu. Chúng ta sẽ cùng tìm hiểu hệ thống này.

LỜI CẢM ƠN

Trăm lời câu nói hoa lệ không bằng một câu cảm ơn chân thành. Trước tiên nhóm em xin gửi lời cảm ơn sâu sắc đến thầy Phạm Trọng Huỳnh. Dẫu trong khoảng thời gian gấp rút thầy vẫn luôn ở bên hướng dẫn nhiệt tình và sát sao. Nhờ những lời dạy công tâm và đầy nhiệt huyết ấy đã giúp chúng em có thêm động lực trên con đường đi tìm những tri thức mới. Từ những kiến thức thầy dạy chúng em xin phép trình bày những gì mình đã tìm hiểu được về đề tài đồ án Hybrid Database bao gồm MySQL và MongoDB gửi đến thầy.

Mặc dù nhóm em đã rất cố gắng nhưng chúng em biết kiến thức của mình còn hạn hẹp cũng như thời gian quá gấp rút nên nhiều điều vẫn chưa thể truyền đạt hết. Chính vì vậy nhóm em mong nhận được sự góp ý cũng như khích lệ của thầy để tạo nên một nguồn động lực cho chúng em.

Kính chúc thầy sẽ ngày càng thành công và mỹ mãn trên con đường mình đã chọn. Cảm ơn thầy vì khoảng thời gian gấp gò ít ỏi trong thời gian qua nhưng cách dạy và truyền đạt của thầy sẽ luôn là một kí ức không phai mờ của chúng em. Cảm ơn thầy rất nhiều ạ.

Sinh viên thực hiện

NHÓM 5

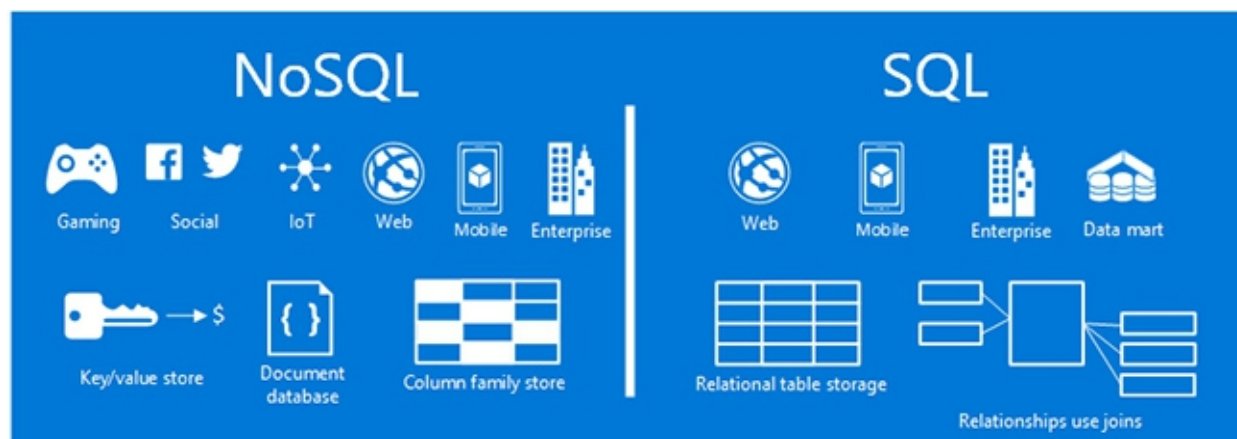
Mục lục

CHƯƠNG 1: Giới thiệu NoSQL và SQL.....	6
1.1. Tổng quan về cơ sở dữ liệu NoSQL và SQL:.....	6
1.2. MongoDB – Cơ sở dữ liệu NoSQL.....	10
1.3. MySQL – Cơ sở dữ liệu hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS).....	12
CHƯƠNG 2: SO SÁNH MONGODB VỚI MYSQL.....	14
CHƯƠNG 3: BIG DATA.....	18
3.1. Khái niệm.....	18
3.2. Tính chất.....	18
3.3. Thách thức.....	19
3.4. Ứng dụng.....	20
CHƯƠNG 4: SỰ RA ĐỜI CỦA HỆ THỐNG KẾT HỢP HYBRID.....	21
4.1. Nguyên nhân:.....	21
4.2. Thiết kế:.....	21
4.3. Thuật toán.....	21
4.4. Thiết lập thử nghiệm và kết quả.....	22
4.5. Kết luận và sự đóng góp.....	22
CHƯƠNG 5: OrientDB.....	23
5.1. Tóm tắt về OrientDB.....	23
5.2. Đặc trưng cơ bản.....	23
5.3. Tính năng đặc trưng.....	23
5.4. Ứng dụng.....	23
5.5. So sánh với các hệ quản trị cùng dòng.....	23
CHƯƠNG 6: Demo.....	25
KẾT LUẬN:.....	28

CHƯƠNG 1: Giới thiệu NoSQL và SQL

1.1. Tổng quan về cơ sở dữ liệu NoSQL và SQL:

NoSQL	SQL
Cơ sở dữ liệu NoSQL thường được gọi là cơ sở dữ liệu phi quan hệ/ DB NoSQL/ NoSQL để làm nổi bật lên việc chúng có thể lưu trữ lượng lớn dữ liệu linh hoạt, phi cấu trúc theo nhiều cách khác nhau so với cơ sở dữ liệu quan hệ (SQL) bằng cột và bảng. Được phát triển vào những năm 2000 khi nhu cầu lưu trữ dữ liệu phi cấu trúc và nhu cầu xử lý nhanh tăng vọt. Mô hình NoSQL sử dụng hệ thống dữ liệu phân tán, nghĩa là một hệ thống có nhiều máy tính giúp xử lý nhanh cơ sở dữ liệu phi cấu trúc và cơ sở dữ liệu linh hoạt.	Được phát triển vào những năm 1970, nó là công cụ được thiết kế nhằm để quản lý dữ liệu sử dụng trong nhiều lĩnh vực , cho phép bạn truy cập và thực hiện các thao tác lấy các hàng hoặc sửa đổi các hàng, trích xuất ,tạo, xóa dữ liệu . SQL chỉ làm việc với những dữ liệu có cấu trúc dạng bảng (table), đối tượng của SQL là các bảng dữ liệu và các bảng này bao gồm nhiều cột và hàng.



Tính năng cốt lõi của NoSQL và SQL:

NoSQL	SQL
<ul style="list-style-type: none">• Lược đồ bất khả tri: Cơ sở dữ liệu lược đồ là mô tả của tất cả dữ liệu và cấu trúc dữ liệu có thể có trong cơ sở dữ liệu quan hệ. Với cơ sở dữ liệu NoSQL, không cần lược đồ, cho phép bạn tự do lưu trữ thông tin mà không cần thiết kế lược đồ từ trước.• Phi quan hệ: Các kết nối relation trong cơ sở dữ liệu được thiết lập giữa các bảng dữ liệu.• Phần cứng thương mại: Một số cơ sở dữ liệu được thiết kế để hoạt động tốt nhất (hoặc chỉ) với phần cứng xử lý và lưu trữ chuyên dụng. Với cơ sở dữ liệu NoSQL, có thể sử dụng các máy chủ bán sẵn với giá rẻ hơn mở rộng quy mô để xử lý nhiều dữ liệu hơn.• Khả năng phân tán cao: Cơ sở dữ liệu phân tán có thể lưu trữ và xử lý một tập hợp thông tin trên nhiều thiết bị. Với NoSQL, chỉ cần duy nhất một cụm máy chủ là có thể dùng để chứa một cơ sở dữ liệu lớn.	<ul style="list-style-type: none">• Chọn lọc một số cột nhất định trong bảng dữ liệu• Lọc các bản ghi theo những tiêu chuẩn khác nhau• Kết hợp hai hay nhiều bảng theo chiều ngang: Trong CSDL, mỗi bảng lưu trữ thông tin về một đối tượng và các bảng liên hệ với nhau qua các trường khóa.• Nối hai hay nhiều bảng theo chiều dọc: khi dữ liệu rất lớn hoặc phân tán ở nhiều nơi ta có thể phải quản lý nhiều bảng theo cùng một mẫu.• Kết nối với dữ liệu trên máy chủ (Server).• Sắp xếp các bản ghi theo những tiêu chuẩn khác nhau: Mỗi loại báo cáo thường có yêu cầu sắp xếp các bản ghi theo những cột khác nhau để tiện cho việc theo dõi.
MongoDB, Cassandra, HBase, Neo4j	MySQL, Postgres, Oracle Database

Ưu nhược điểm của SQL và NoSQL

	NoSQL	SQL
Ưu điểm	<p>High scalability:</p> <p>Cơ sở dữ liệu sử dụng sharding để chia tỷ lệ ngang. Chia tỷ lệ dọc có nghĩa là thêm nhiều tài nguyên hơn vào máy hiện có trong khi chia tỷ lệ ngang có nghĩa thêm nhiều máy hơn để xử lý dữ liệu. => Có thể dễ dàng xử lý dữ liệu không lồ do khả năng mở rộng, khi dữ liệu tự phát triển, NoSQL tự động mở rộng quy mô để xử lý dữ liệu đó một cách hiệu quả.</p> <p>Hight availability:</p> <p>Tự động sao chép vì trong trường hợp có bất kì lỗi nào, dữ liệu sẽ tự sao chép về trạng thái nhất quán trước.</p>	<p>Không cần code:</p> <p>Dễ dàng quản lý không cần viết code nào.</p> <p>Tiêu chuẩn được quy định rõ ràng:</p> <p>Hai tiêu chuẩn ISO và ANSI</p> <p>Tính di động:</p> <p>Sử dụng trong nhiều chương trình : PC, servers, laptops và cả mobile phones</p> <p>Ngôn ngữ tương tác:</p> <p>Giao tiếp với cơ sở dữ liệu và nhận câu hỏi phức tạp trong vài giây</p> <p>Multiple data views:</p> <p>Tạo các hiển thị khác nhau về cấu trúc cơ sở dữ liệu và cơ sở dữ liệu cho những người dùng khác nhau.</p>
Nhược điểm	<p>Quản lý dữ liệu:</p> <p>Phức tạp hơn nhiều so với cơ sở dữ liệu quan hệ. Khó cài đặt và thậm chí là để quản lý hằng ngày cũng tốn thời gian.</p> <p>Sao lưu dữ liệu: Không có cách tiếp cận để sao lưu dữ</p>	<p>Giao diện khó dùng:</p> <p>Giao diện phức tạp.</p> <p>Không được toàn quyền kiểm soát:</p> <p>Không có toàn quyền kiểm soát.</p> <p>Thực thi: Hầu hết đều có</p>

	<p>liệu một cách nhất quán.</p> <p>Thiếu tính nhất quán: Bỏ qua sự nhất quán để ưu tiên tốc độ, hiệu suất hiệu quả lớn.</p> <p>Trọng tâm hẹp: Trọng tâm hẹp vì chủ yếu thiết kế lưu trữ nhưng cung cấp ít chức năng.</p> <p>Mã nguồn mở: Là cơ sở dữ liệu mã nguồn mở và không có tiêu chuẩn đáng tin cậy.</p> <p>Không có lược đồ</p>	<p>phần mở rộng độc quyền riêng của nhà cung cấp bên cạnh các tiêu chuẩn SQL.</p> <p>Giá cả: Chi phí khiến cho một số lập trình viên khó tiếp cận.</p>
--	--	---

1.2. MongoDB – Cơ sở dữ liệu NoSQL

CẤU TRÚC

Được xây dựng dựa trên các sưu tập và tài liệu. Đơn vị dữ liệu cơ bản bao gồm một cặp key – value cho phép tài liệu có các trường và cấu trúc khác nhau. Cơ sở dữ liệu này sử dụng định dạng lưu trữ tài liệu có tên là BSON, đây là kiểu JSON nhị phân. Mô hình dữ liệu mà MongoDB tuân theo có tính co giãn cao cho phép kết hợp và lưu trữ dữ liệu thuộc nhiều loại biến mà không cần thay đổi các tùy chọn chỉ mục, quyền truy cập dữ liệu và quy tắc xác thực. Giúp tiết kiệm thời gian.

NGUYÊN TẮC CẤU TRÚC CỐT LÕI

- Tài liệu và ngôn ngữ truy vấn MongoDB: Đổi mới nhanh chóng.

Lược đồ linh hoạt: Tự động lưu thay đổi.

Phổ biến: Khối lượng nhẹ, không phụ thuộc vào ngôn ngữ và con người có thể đọc được. Documents là một siêu tập hợp tất cả các mô hình dữ liệu khác nhau để có thể thiết kế dữ liệu theo bất kỳ cách nào mà ứng dụng của bạn cần, cặp key – value, tables, dữ liệu chuỗi thời gian – không gian địa lý, nodes và edges của biểu đồ.

Mạnh mẽ: Có thể lưu trữ lượng lớn dữ liệu.

- Cơ sở dữ liệu toàn cầu, nhiều lưu trữ dạng đám mây: Tự do và linh hoạt.

Phạm vi tiếp cận rộng – Clouds: Private, Hybrid, Public: Với mô hình triển khai chung, nó cho phép bạn tận dụng mọi khả năng độc nhất trong mỗi nền tảng mà không cần thay đổi bất kỳ dòng mã ứng dụng nào cũng như không gặp phải trách nhiệm và rủi ro khi di chuyển cơ sở dữ liệu phức tạp.

MongoDB Atlas – Cơ sở dữ liệu hoàn toàn tự động dưới dạng dịch vụ: MongoDB Atlas là dịch vụ cơ sở dữ liệu dạng đám mây toàn cầu dành cho các ứng dụng hiện đại. Có thể triển khai MongoDB trên AWS, Google Clouds hoặc Azure với tính năng tự động hóa tốt nhất và các phương pháp đã được chứng minh đảm bảo tính khả dụng, khả năng mở rộng và tuân thủ các tiêu chuẩn bảo mật.

MongoDB do bạn quản lý, với các công cụ của chúng tôi.

- MongoDB Clouds: Trải nghiệm hợp nhất các công cụ hiện đại từ các nhà phát triển.

MongoDB Atlas Search: Tìm kiếm: Dễ dàng tạo các tìm kiếm toàn văn nhanh, có liên quan dựa trên cơ sở dữ liệu Cloud của bạn và xây dựng trên Apache Lucene, thư viện tiêu chuẩn công nghiệp.

MongoDB Atlas Data Lake: Mang đến một kho dữ liệu không có máy chủ, có thể mở rộng quy mô trên nền tảng Cloud với dịch vụ truy vấn theo yêu cầu cho phép bạn phân tích dữ liệu trong kho lưu trữ dữ liệu đám mây (Amazon s3) bằng cách sử dụng ngôn ngữ truy vấn MongoDB (MQL).

MongoDB Realm for Data at the Edge: Mở rộng nền tảng dữ liệu của bạn ra rìa mạng và được tích hợp hoàn toàn với MongoDB Cloud, Realm là một cơ sở dữ liệu nhẹ được nhúng trực tiếp trên thiết bị khách. Realm giúp giải quyết những thách thức đặt biệt trong việc xây dựng cho thiết bị di động, giúp việc lưu trữ dữ liệu trên thiết bị trở nên đơn giản và cho phép truy cập dữ liệu khi đang ngoại tuyến.

MongoDB đáp ứng nhu cầu của các ứng dụng hiện đại với nền tảng dữ liệu được xây dựng trên 3 nền tảng cấu trúc cốt lõi:

1. Mô hình dữ liệu Documents và Ngôn ngữ truy vấn MongoDB, mang đến cho các nhà phát triển cách đổi mới nhanh chóng trong việc xây dựng các ứng dụng giao dịch, cách thức vận hành và phân tích.

2. Cơ sở dữ liệu toàn cầu, nhiều lưu trữ dạng đám mây cho phép các nhà phát triển tự do chạy ứng dụng của họ bất kỳ đâu mà không phải thay đổi bất kỳ dòng mã nào.
3. MongoDB Cloud, cung cấp cho các nhà phát triển trải nghiệm thống nhất giữa các ứng dụng trên toàn bộ Cloud. Trong cơ sở dữ liệu, tìm kiếm và lưu trữ được hỗ trợ bởi các ứng dụng tích hợp chi tiết và toàn diện.

1.3. MySQL – Cơ sở dữ liệu hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS)

Cũng như các cơ sở dữ liệu quan hệ khác, MySQL lưu trữ dữ liệu trong các bảng được tạo thành từ các hàng và cột. Người dùng có thể xác định, thao tác, kiểm soát và truy vấn dữ liệu bằng ngôn ngữ truy vấn có cấu trúc, thường được gọi là SQL .

Xin lưu ý rằng MySQL và SQL không giống nhau. SQL trong MySQL là viết tắt của ngôn ngữ truy vấn có cấu trúc. Nó là ngôn ngữ tiêu chuẩn được sử dụng để tương tác với cơ sở dữ liệu. Còn MySQL là hệ quản trị cơ sở dữ liệu quan hệ giúp thao tác với cơ sở dữ liệu được lưu trữ trong các bảng khác nhau trong máy tính.

Các loại DBMS, chủ yếu có 4 loại DBMS:

- **DBMS phân cấp:** Loại DBMS này có kiểu quan hệ kiểu tiền nhiệm-kế thừa. Vì vậy, nó có cấu trúc hệ thống quản lý cơ sở dữ liệu dạng phân tầng tương tự cấu trúc của một cái cây, trong đó các nút đại diện cho các bản ghi và các nhánh của cây đại diện cho các trường.
- **DBMS quan hệ (RDBMS):** Loại DBMS này sử dụng cấu trúc cho phép người dùng xác định và truy cập dữ liệu liên quan với nhau trong cùng hệ thống cơ sở dữ liệu.
- **Mạng lưới DBMS:** Loại DBMS này hỗ trợ mối quan hệ nhiều – nhiều (many – to – many relationship), trong đó những bản ghi của người dùng có thể được liên kết với nhau.

- **DBMS hướng đối tượng :** Loại DBMS này sử dụng phần mềm nhỏ riêng lẻ được gọi là các đối tượng. Mỗi đối tượng chứa một phần nhỏ dữ liệu và các hướng dẫn cho các hành động được thực hiện với dữ liệu đó.

Các tính năng của MySQL

Dễ quản lý – Phần mềm rất dễ tải và có chức năng tự động cập nhật các update.

Hỗ trợ giao dịch mạnh mẽ – Giữ thuộc tính ACID (Atomicity (nguyên tử), Consistency (nhất quán), Isolation (độc lập), và Durability (bền vững)).

Phát triển ứng dụng toàn diện – MySQL có các thư viện plugin để ghi cơ sở dữ liệu vào bất kỳ ứng dụng nào. Nó cũng hỗ trợ các thao tác như: lưu trữ, kích hoạt chương trình, chế độ xem và nhiều thứ khác để phát triển ứng dụng. Bạn có thể tham khảo RDS Tutorial , để hiểu về RDBMS của Amazon.

Hiệu suất cao – Tải nhanh các tiện ích với bộ nhớ cache riêng biệt và phân vùng chỉ mục.

Chi phí để sở hữu thấp – Điều này làm giảm chi phí cấp phép và chi phí phần cứng.

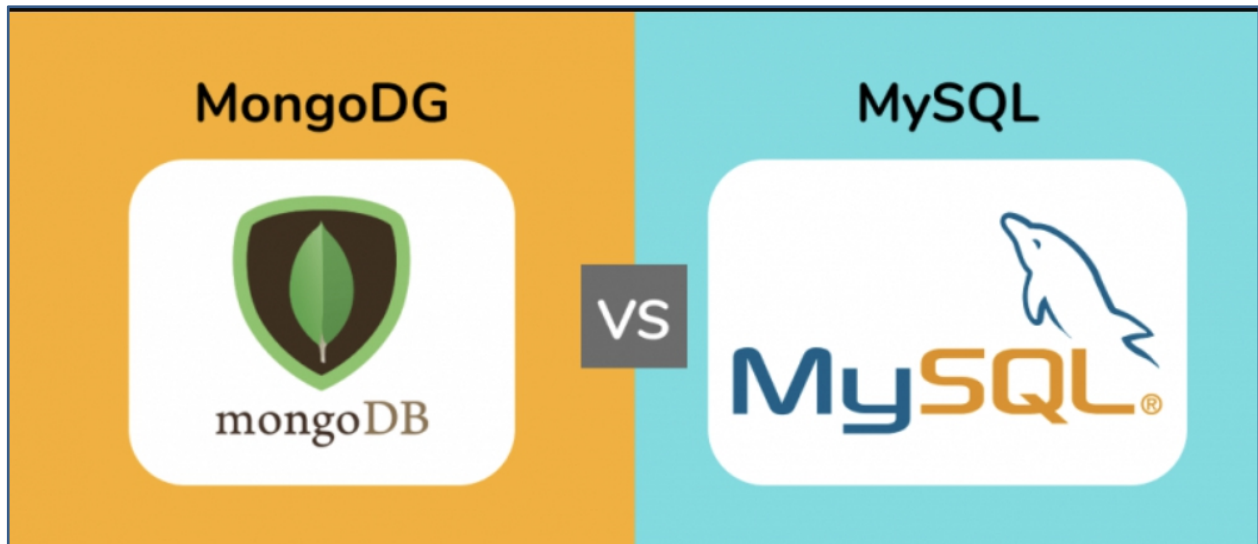
Nguồn mở & được hỗ trợ 24 / 7 – RDBMS này có thể được sử dụng trên mọi nền tảng và cung cấp hỗ trợ 24 / 7 cho phiên bản nguồn mở và doanh nghiệp.

Bảo vệ an toàn dữ liệu – MySQL hỗ trợ các cơ chế mạnh mẽ để đảm bảo rằng chỉ những người dùng được ủy quyền mới có quyền truy cập vào cơ sở dữ liệu.

Tính sẵn sàng cao – MySQL có thể chạy các cấu hình sao chép chính/phụ với tốc độ cao và cung cấp nó các máy chủ của cụm.

Khả năng mở rộng & tính linh hoạt – Với MySQL, bạn có thể chạy các ứng dụng được nhúng sâu và tạo kho dữ liệu chứa một lượng dữ liệu khổng lồ.

CHƯƠNG 2: SO SÁNH MONGODB VỚI MYSQL



	MongoDB	MySQL
Cấu trúc dữ liệu	Dữ liệu được lưu trữ và biểu diễn dưới dạng JSON.	Mỗi bản ghi được lưu trữ dưới dạng cấu trúc ô bảng với các hàng và cột.
Lược đồ	MongoDB lưu trữ dữ liệu trong các bộ sưu tập không có lược đồ bắt buộc. Nói cách khác, dữ liệu (lấy) vào có thể có cấu trúc được xác định trước và nó có thể tuân theo cấu trúc đó, tuy nhiên, các tài liệu khác nhau trong cùng một bộ sưu tập có thể có cấu trúc khác nhau nếu cần.	MySQL yêu cầu một định nghĩa lược đồ cho các bảng trong cơ sở dữ liệu. Do đó, lược đồ không thể thay đổi được. Chỉ các input phù hợp với lược đồ nhất định mới được chấp nhận.
Ngôn ngữ	MongoDB sử dụng MongoDB Query Language (MQL), được thiết kế để các nhà phát triển dễ	MySQL sử dụng ngôn ngữ SQL, là ngôn ngữ có cấu trúc.

	dùng sử dụng.	
Thân thiện với người dùng	MongoDB là một lựa chọn hấp dẫn đối với các nhà phát triển vì dễ sử dụng và dễ hiểu về lưu trữ dữ liệu.	MySQL hơi phức tạp hơn so với MongoDB vì lược đồ của bảng, khóa ngoại, chuẩn hóa, v.v.
Khả năng mở rộng	Cơ sở dữ liệu MongoDB có thể được chia tỷ lệ theo cả chiều dọc và chiều ngang (VD: nâng cấp CPU và RAM).	Cơ sở dữ liệu MySQL có thể được chia tỷ lệ theo chiều dọc (Chia tỷ lệ theo chiều dọc có nghĩa là bổ sung thêm bộ nhớ cho máy hiện có của bạn, ví dụ như nâng cấp CPU và RAM).
Khóa ngoại	MongoDB không hỗ trợ sử dụng khóa ngoại.	MySQL hỗ trợ việc sử dụng khóa ngoại (khóa ngoại được sử dụng để liên kết bảng này với bảng khác).
Hỗ trợ ngôn ngữ	MongoDB hỗ trợ các ngôn ngữ như C và C++.	MySQL hỗ trợ các ngôn ngữ như C, C++ và JavaScript.
Hiệu suất	MongoDB tối ưu hóa chủ yếu cho hiệu suất ghi (nghĩa là hiệu suất tốc độ ghi của cơ sở dữ liệu MongoDB lớn hơn nhiều so với cơ sở dữ liệu MySQL).	MySQL được tối ưu hóa để có hiệu suất kết nối cao trên nhiều bảng.
Tiến trình đồng bộ hóa dữ liệu từ nhiều Server	MongoDB sử dụng sharding và replication (sharding cho phép phân vùng dữ liệu trên nhiều máy chủ bằng cách sử dụng shared key và kỹ thuật đồng bộ hóa (sắp xếp) dữ liệu trên nhiều	MySQL hỗ trợ replication master-slave và replication master-master (tính năng replication cho phép một máy chủ - master gửi tất cả các thay đổi đến máy chủ khác - slave).

	máy chủ để dự phòng được gọi là replication).	
Tính linh hoạt	Bản chất không có lược đồ của MongoDB giúp việc xây dựng và nâng cấp ứng dụng theo thời gian trở nên đơn giản mà không cần các quy trình di chuyển lược đồ khó khăn và tốn kém như bạn làm với cơ sở dữ liệu quan hệ.	Cơ sở dữ liệu MySQL kém linh hoạt hơn so với MongoDB do thiết kế lược đồ của nó.
Tấn công mạng và bảo mật	Trong MongoDB, không yêu cầu định nghĩa lược đồ nên rủi ro bị tấn công ít hơn và an toàn hơn so với MySQL do thiết kế không có lược đồ của nó.	MySQL bảo mật kém hơn do thiết kế lược đồ.
Thao tác JOIN	MongoDB không hỗ trợ thao tác JOIN.	MySQL hỗ trợ thao tác JOIN.
Đóng góp từ cộng đồng	Hiện tại MongoDB có khoảng 200 nghìn kho lưu trữ trên GitHub với khoảng 1 triệu lượt truy cập, điều này cho thấy sẽ khá khó khăn để giải quyết vấn đề trong MongoDB nếu bạn gặp khó khăn ở đâu đó so với MySQL do tài liệu và sự hỗ trợ từ cộng đồng ít hơn).	MySQL có hơn 200 nghìn kho lưu trữ trên GitHub và khoảng 7 triệu lượt truy cập, vì vậy hỗ trợ của cộng đồng MySQL lớn hơn nhiều so với MongoDB.
Xử lý dữ liệu phi cấu trúc	MongoDB là một lựa chọn lý tưởng nếu bạn có dữ liệu phi	MySQL không thể xử lý dữ liệu phi cấu trúc.

	cấu trúc và/hoặc có cấu trúc với tiềm năng phát triển nhanh chóng.	
--	--	--

CHƯƠNG 3: BIG DATA

3.1. Khái niệm

Là nhóm dữ liệu đã trở nên đồ sộ bao gồm nhiều kiểu dữ liệu. Được dùng trong thời gian dài và ngày càng phức tạp. Sự phức tạp này đến từ thực tế dữ liệu được thu thập từ nhiều nguồn và cần phải tích hợp chúng thành 1 đơn vị để phân tích. Độ lớn đến mức các phần mềm xử lý dữ liệu truyền thống không có khả năng thu thập, quản lý và xử lý dữ liệu trong một khoảng thời gian hợp lý. Những tập dữ liệu lớn này có thể bao gồm các dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc, mỗi tập có thể được khai thác để tìm hiểu insights.

3.2. Tính chất

Mặc dù từ “lớn” dùng để chỉ kích thước của một thứ gì đó, nhưng trong dữ liệu lớn, “lớn” không chỉ giới hạn ở khối lượng dữ liệu mà còn là kết hợp các thuộc tính khác như tốc độ và tính đa dạng. Ba thuộc tính này mô tả các thuộc tính chính của dữ liệu lớn được gọi là 3 V của dữ liệu lớn:

- **Khối lượng:** Đôi khi được coi là thuộc tính cuối cùng của dữ liệu lớn. Nó miêu tả lượng dữ liệu rất lớn và tăng dần theo thời gian từ terabyte (10¹² byte) đến yotta-byte có hàng nghìn tỷ gigabyte.
- **Tốc độ:** Tốc độ đề cập đến tính khả dụng của dữ liệu trong thời gian dùng để xử lý. Tỷ lệ thuận với tốc độ dữ liệu được lan truyền trong hệ thống.
- **Tính đa dạng:** Điều này thể hiện tính đa dạng của dữ liệu. Dữ liệu lớn được tạo thành từ dữ liệu gốc từ nhiều nguồn khác nhau như email, máy móc, mạng xã hội, kinh doanh giao dịch, thiết bị di động, v.v. Dữ liệu từ các nguồn khác nhau có các dạng khác nhau như dạng bảng tính, hình ảnh, video, v.v. Tính đa dạng như một thuộc tính của dữ liệu lớn dùng mô tả các dạng dữ liệu khác nhau thu được từ các nguồn khác nhau.

- **Tính xác thực:** Đề cập đến tính chính xác của dữ liệu. Cần quan tâm đến sự thích hợp của dữ liệu (xử lý hoặc phân tích) cho nhiệm vụ hiện tại. Cho thấy sự cần thiết phải tránh tích lũy dữ liệu bản.
- **Độ biến động thị trường:** Giải quyết vấn đề thời gian thích hợp để lưu trữ dữ liệu trong thời gian thực. Nó điều tra tính hợp lệ của dữ liệu được lưu trữ cho phân tích hiện tại.
- **Tính hợp lệ:** Các quyết định cũng có giá trị tương tự như dữ liệu được sử dụng trong các phân tích.

3.3. Thách thức

Để tận dụng nhiều lợi ích của dữ liệu lớn, dữ liệu phải trải qua các quy trình sau:

Nhập dữ liệu: Quy trình thu thập và nhập dữ liệu để sử dụng hoặc lưu trữ. Dữ liệu có thể nhập sau khi được cung cấp bởi nguồn hoặc được nhóm thành các đợt và được nhập trong khoảng thời gian quy định. Quá trình này thường bắt đầu với việc xếp hạng các nguồn dữ liệu, xác thực từng tệp trước khi truyền dữ liệu đến đích chính xác.

- **Lưu trữ:** Lưu trữ rất phức tạp; bao gồm tìm kiếm và truy xuất dữ liệu cũng có thể bao gồm các vấn đề bảo mật và quyền riêng tư phức tạp. Khung để lưu trữ cho dữ liệu lớn nên chứa được khối lượng lớn dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc.
- **Phân tích:** Dữ liệu lớn gần như vô dụng nếu không có các công cụ và quy trình phân tích hiệu quả. Thông qua đó các thông tin hữu ích sẽ được chất lọc từ những thứ dường như là dữ liệu rác. Phân tích trong lĩnh vực dữ liệu lớn cũng bao gồm các tính toán trên dữ liệu lớn. Nó bao gồm các framework và công cụ như MapReduce và Hadoop được sử dụng để rút ra kết luận từ dữ liệu lớn. Kết quả phân tích dữ liệu có thể được sử dụng để cải thiện các dịch vụ dùng để cung cấp cho khách hàng, chiến lược tiếp thị và ra quyết định chung.

- Trực quan hóa: Trình bày dữ liệu để dễ dàng xác định các mẫu hoặc nắm bắt của những khái niệm mới. Các công cụ và kỹ thuật trực quan hóa dữ liệu lớn cho phép biểu diễn dữ liệu ở dạng đồ thị, biểu đồ, bản đồ và thậm chí cả video giúp dễ dàng tìm kiếm thông tin liên quan.

3.4. Ứng dụng

Phục vụ trong các ngành như: y tế, Ngân hàng, Thương mại điện tử, Ngành bán lẻ, Digital marketing,...

CHƯƠNG 4: SỰ RA ĐỜI CỦA HỆ THỐNG KẾT HỢP HYBRID

4.1. Nguyên nhân:

Như đã thấy ở phần trên hai hệ cơ sở dữ liệu đều có những ưu điểm và nhược điểm riêng khó có thể khắc phục. Qua sự tìm hiểu và nghiên cứu thì hệ thống Hybrid ra đời để khắc phục cả hai hệ cơ sở dữ liệu trên. Với mong muốn có thể đáp ứng được yêu cầu mà những hệ cơ sở dữ liệu cơ sở trước không thể.

4.2. Thiết kế:

Gồm 2 thành phần chính:

- SQL: Chứa công cụ lưu trữ xử lý việc kiểm soát dữ liệu trong cơ sở dữ liệu MySQL. Tạo thành một tệp nhật ký giao tác và các nhóm tệp chia nhỏ theo thứ bậc. Nó ghi lại thời điểm bắt đầu và kết thúc của mỗi thao tác cũng như mọi sửa đổi được thực hiện trên dữ liệu trong cơ sở.
- MongoDB: Sử dụng bản sao để đảm bảo tính dự phòng và nhất quán. Luồng dữ liệu từ các đích khác nhau và định dạng khác nhau bị hỏng và phân tán đều thành một tập hợp các thiết bị đầu cuối không tĩnh có thể mở rộng.

4.3. Thuật toán

Bước 1: Tải dữ liệu

Bước 2: Xác định lớp của dữ liệu

Bước 3: Khởi tạo giao diện kết hợp DB

Bước 4: Kiểm tra dữ liệu: Nếu

Dữ liệu là dữ liệu cấu trúc, thì :

Lưu trữ trong cơ sở dữ liệu SQL

Dữ liệu là dữ liệu phi cấu trúc, thì

Lưu trữ trong cơ sở dữ liệu MongoDB

Bước 5: Cập nhật cơ sở dữ liệu giao diện kết hợp

Bước 6: Xem, Xóa, Cập nhật, Thoát

4.4. Thiết lập thử nghiệm và kết quả

- Triển khai C# trong visual studio.
- MongoDB và MySql lưu trữ dữ liệu.
- Một số lớp: Person, Symbol, Panel, PatientInfo, FileInfo.
- Trước hết thực hiện lệnh mongod.exe. Máy chủ MongoDB sẽ bắt đầu cổng 27017.
- Sau khi thực hiện một cửa sổ nhắc lệnh riêng biệt thì vỏ máy kết nối với Test.
- Để thực thi dữ liệu, dữ liệu vào đc chia thành 2 loại: cấu trúc và phi cấu trúc trong đĩa cục bộ của hệ thống.
- Truy cập ứng dụng bằng cách nhấn đúp vào ứng dụng. => Hệ thống đề xuất có thể hoạt động.
- Khi dữ liệu lớn được tải lên, hệ thống hybrid sẽ phân loại thành dữ liệu phi cấu trúc hay có cấu trúc để chuyển đến cơ sở dữ liệu tương ứng.

4.5. Kết luận và sự đóng góp

Với kết quả thu được hệ thống có thể lưu trữ và loại bỏ những điểm yếu lớn trong hai cơ sở dữ liệu.

Việc này đã đóng góp những kiến thức sau:

- Phát triển hệ thống Hybrid.
- Cách tiếp cận này cải thiện việc sử dụng cơ sở dữ liệu MongoDB để lưu trữ dữ liệu lớn.
- củng cố các khả năng của hai cơ sở dữ liệu.

CHƯƠNG 5: OrientDB

5.1. Tóm tắt về OrientDB

- Là một hệ thống quản lý cơ sở dữ liệu NoSQL mã nguồn mở viết bằng Java.
- Là một cơ sở dữ liệu đa mô hình, hỗ trợ đồ thị, tài liệu, khóa/ giá trị và mô hình ối tượng, nhưng các mối quan hệ được quản lý như trong cơ sở dữ liệu đồ thị với các kết nối trực tiếp giữa các bản ghi.
- Có 2 phiên bản:
 - + OrientDB community edition là bản miễn phí sử dụng.
 - + OrientDB enterprise edition là bản trả phí cho sự xử lý mạnh mẽ hơn.

5.2. Đặc trưng cơ bản

- Hỗ trợ cơ sở dữ liệu đồ thị cho phép lưu trữ các thực thể và quan hệ giữa các thực thể.
- Điểm mạnh chính là tốc độ thực thi, trả về kết quả các truy vấn rất nhanh, giúp tăng tốc độ ứng dụng

5.3. Tính năng đặc trưng

- Xử lý đáng tin cậy, trường hợp có sự cố, tất cả các tài liệu đang chờ xử lý sẽ được khôi phục.
- Dễ dàng quản lý.
- Hỗ trợ truy vấn SQL với phần mở rộng để xử lý các mối quan hệ mà không cần (JOIN).
- Nguyên bản hỗ trợ HTTP, giao thức RESTful.
- Hỗ trợ đầy đủ cho việc phân tán cơ sở dữ liệu.

5.4. Ứng dụng

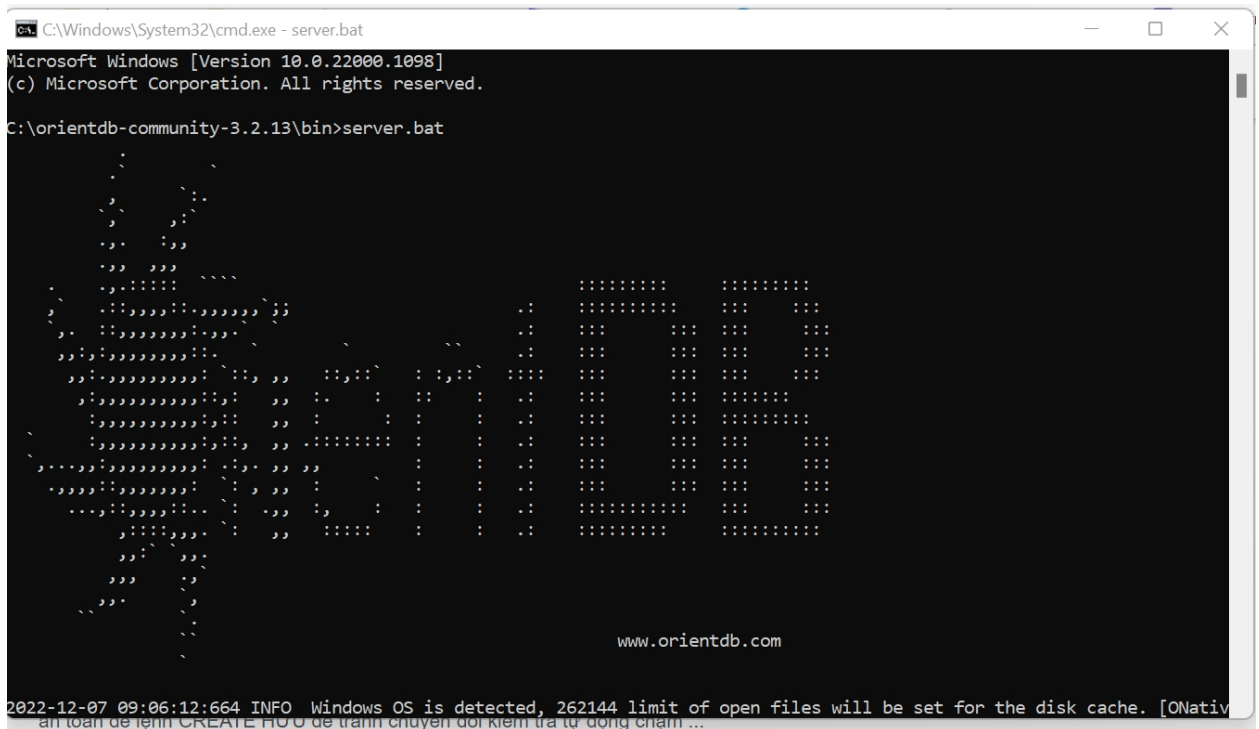
- Ngân hàng.
- Dữ liệu lớn.
- Mạng xã hội.
- Hệ thống quản lý giao thông.

5.5. So sánh với các hệ quản trị cùng dòng

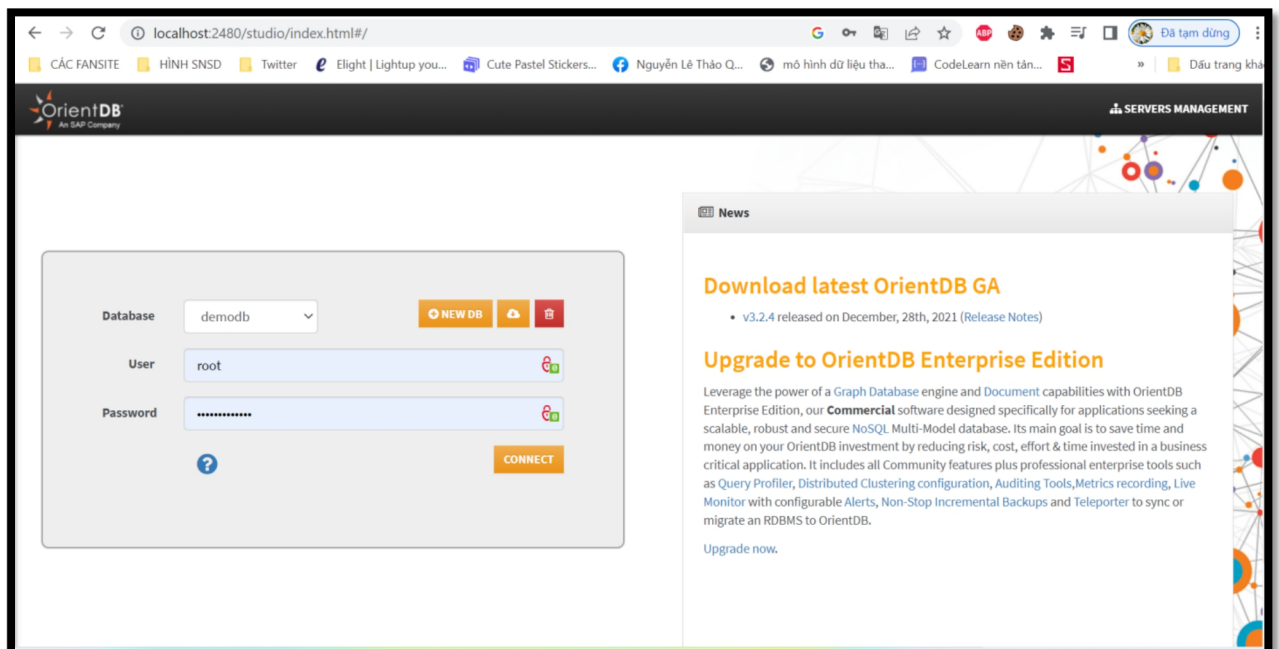
	MongoDB	Hbase	Neo4j	OrientDB
Hệ điều hành	Linux, OS X, Solaris, Windows	Linux, Unix, Windows	Linux, Os X, Solaris, Windows	Linux, OS X, Solaris Windows
Kiến trúc	Dữ liệu lưu trữ theo hàng	Dữ liệu lưu trữ theo hàng	Dữ liệu lưu trữ theo cột	Dữ liệu lưu trữ theo cột
Năm phát hành	2009	2008	2007	2010
Base code	C++	java	java	Java
Ngôn ngữ hỗ trợ	C#, C++, Clojure, Erlang, Go, Node.js, Perl, PHP, Python, Java, Ruby, Scala	C, C#, C++, Groovy, PHP, Python, Scala	.Net, Clojure, Elixir, Go, Groovy, Haskell, JavaScript, Perl, PHP, Python, Ruby, Scala	.Net C C# C++ Python JavaScript Ruby
Phiên bản	Community (Free) And Enterprise	Community	Community	Community
Sử dụng	Quản lý dữ liệu. Hệ thống quản lý nội dung. Phân tích thời gian thực. IoT. Trí tuệ hoạt động	Phân tích nhật ký trực tiếp. Hadoop. Gỡ các ứng dụng lớn.	Quản lý dữ liệu	Dữ liệu lớn Mạng xã hội Quản lý giao thông.

CHƯƠNG 6: Demo

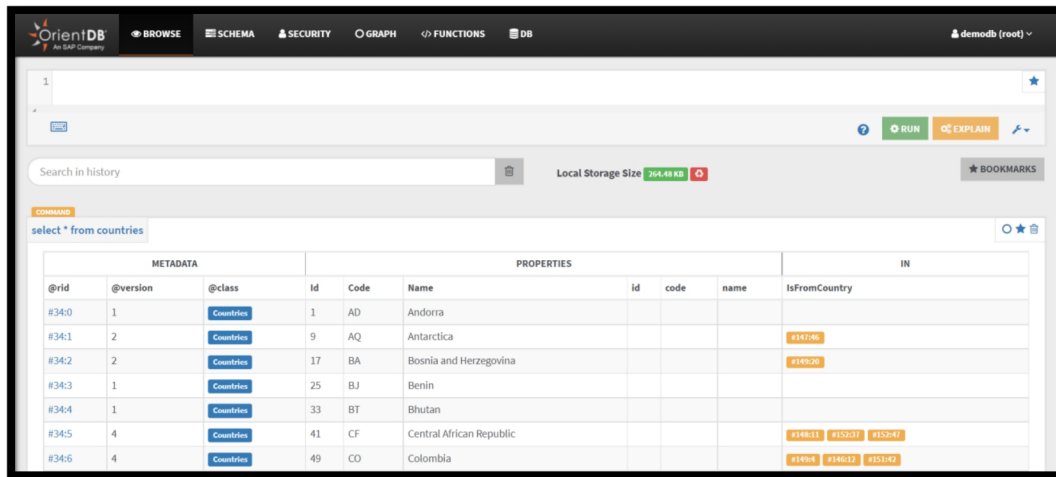
Giao diện running server OrientDB



Liên kết thông qua localhost: 2480 đến OrientDB



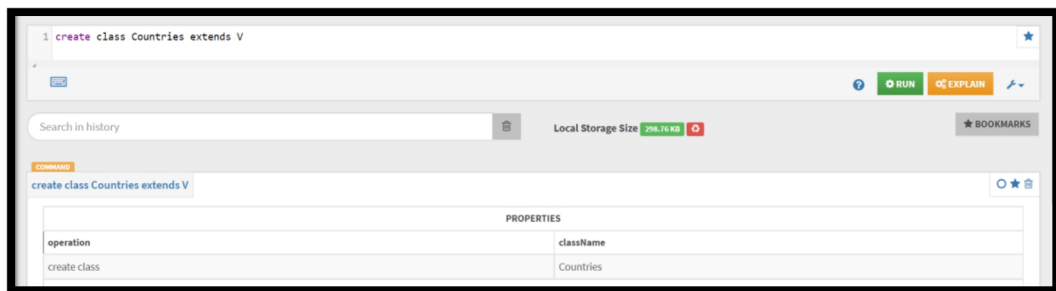
Giao diện sau khi connect với demodb



The screenshot shows the OrientDB web interface with a query result for the command `select * from countries`. The interface includes a top navigation bar with tabs for BROWSE, SCHEMA, SECURITY, GRAPH, FUNCTIONS, and DB. The user is logged in as 'demodb (root)'. The query result is displayed in a table with columns for METADATA and PROPERTIES.

METADATA			PROPERTIES						IN
@rid	@version	@class	id	code	Name	id	code	name	IsFromCountry
#34:0	1	Countries	1	AD	Andorra				
#34:1	2	Countries	9	AQ	Antarctica				#14:0
#34:2	2	Countries	17	BA	Bosnia and Herzegovina				#14:20
#34:3	1	Countries	25	BJ	Benin				
#34:4	1	Countries	33	BT	Bhutan				
#34:5	4	Countries	41	CF	Central African Republic				#14:11 #15:21 #15:247
#34:6	4	Countries	49	CO	Colombia				#14:04 #14:022 #15:142

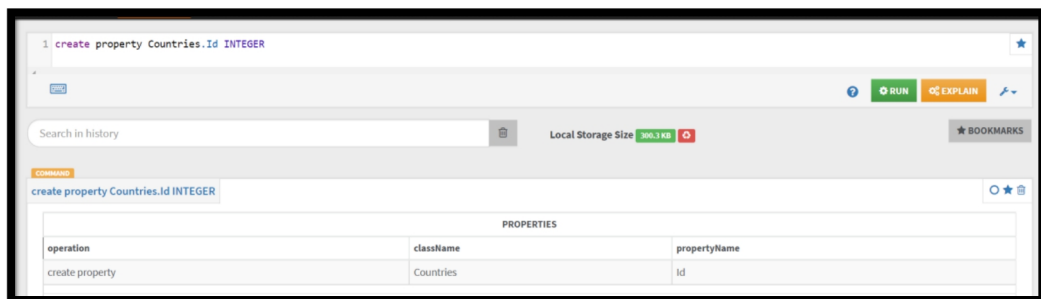
Tạo lớp Countries extends V



The screenshot shows the OrientDB web interface with the command `create class Countries extends V`. The interface displays the properties of the newly created class.

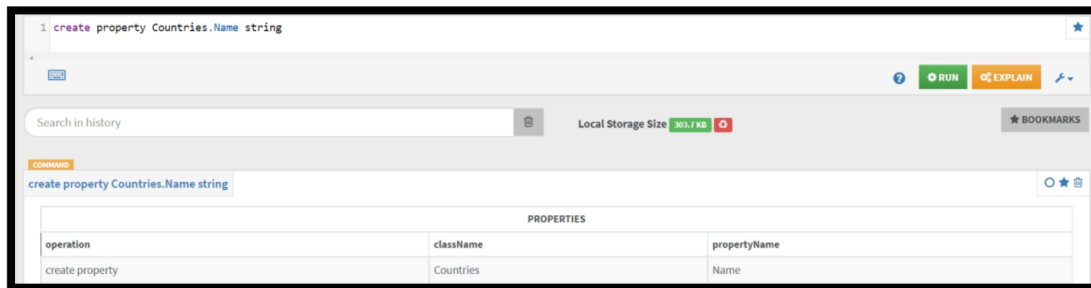
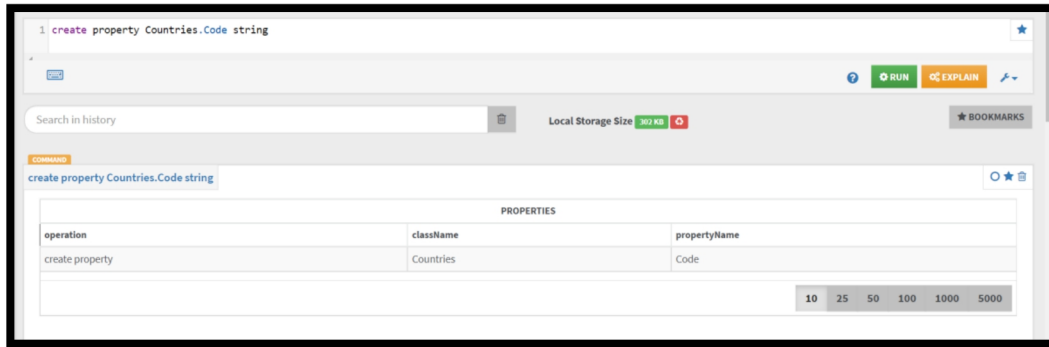
PROPERTIES	
operation	className
create class	Countries

Tạo thuộc tính bằng câu lệnh

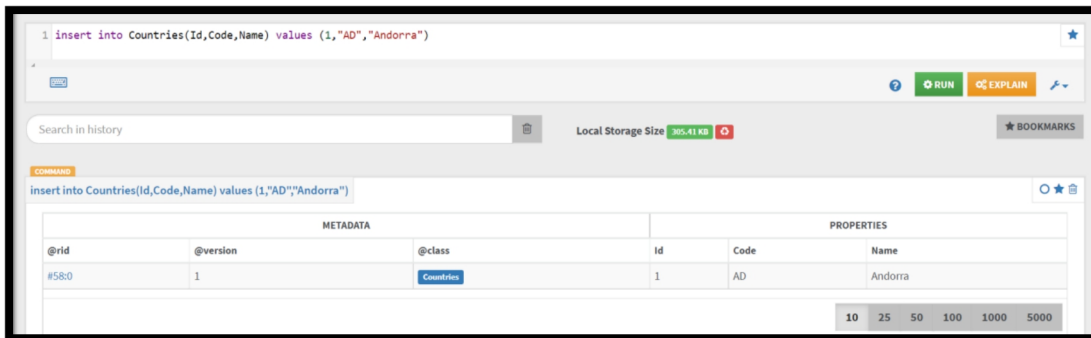


The screenshot shows the OrientDB web interface with the command `create property Countries.Id INTEGER`. The interface displays the properties of the newly created property.

PROPERTIES		
operation	className	propertyName
create property	Countries	Id



Insert dữ liệu vào bảng Countries












- Tạo thêm nhiều đối tượng
- Chạy một số câu truy vấn

KẾT LUẬN:

DB-Engines Ranking - May 2021

Search...

Rank	Name	Type	May 2021	Chart May 2021
1.	Oracle	Relational, Multi-model	1269.94	
2.	MySQL	Relational, Multi-model	1236.38	
3.	Microsoft SQL Server	Relational, Multi-model	992.66	
4.	PostgreSQL	Relational, Multi-model	559.25	
5.	MongoDB	Document, Multi-model	481.01	
6.	IBM Db2	Relational, Multi-model	166.66	
7.	Redis	Key-value, Multi-model	162.17	
8.	Elasticsearch	Search engine, Multi-model	155.35	
9.	SQLite	Relational	126.69	

Sự khác biệt chính giữa cơ sở dữ liệu SQL và NoSQL là ngôn ngữ lập trình được sử dụng để thực hiện các truy vấn. Cơ sở dữ liệu SQL lưu trữ dữ liệu ở dạng có cấu trúc hơn và do đó, chúng được ưu tiên cho các hoạt động phức tạp hơn có thể được hoàn thành bằng các truy vấn SQL phức tạp. NoSQL không có bất kỳ ngôn ngữ truy vấn khai báo nào và ngôn ngữ thay đổi từ cơ sở dữ liệu này sang cơ sở dữ liệu khác, điều này không tốt cho các truy vấn phức tạp nhưng là lựa chọn lý tưởng cho các truy vấn nhanh.

Cả cơ sở dữ liệu SQL và NoSQL đều được sử dụng để đáp ứng các nhu cầu cụ thể. Tùy thuộc vào các mục tiêu và môi trường dữ liệu của một tổ chức, những ưu và nhược điểm cụ thể của chúng có thể được khuếch đại.

Chính vì có sự thiếu sót như thế nên một hệ cơ sở dữ liệu mới được phát triển có tên gọi là cơ sở dữ liệu lai (hybrid database) đang ngày càng được mọi người tin dùng và biết đến. Nó dường như mở ra một tầm nhìn mới về việc quản lý cơ sở dữ liệu. Đến với cơ sở dữ liệu lai nhưng ưu điểm ở những cơ sở dữ liệu như SQL hay NoSQL sẽ được kế thừa và phát triển thêm. Còn với những nhược điểm sẽ được hạn chế và khắc phục. Chính vì thế cơ sở dữ liệu lai đã và đang trở thành sự lựa chọn của nhiều doanh nghiệp. Dẫu rằng hiện tại đôi lúc vẫn còn hạn chế nhưng với sự phát triển vượt bậc của khoa học và công nghệ thì chắc chắn trong khoảng thời gian sau cơ sở dữ liệu này sẽ càng hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

- [1] K. Kline, SQL in a nutshell, 3rd ed. O'Reilly Media, November 2008
- [2] MongoDB. Top 5 considerations when evaluating NoSQL databases. 2015;6:8. (Accessed 5 September 2019)
- [3] Victor Wilkinson, Compare Relational and NoSQL Databases, 25/05/2020
- [4] What is NoSQL database [<https://codelearn.io/sharing/ban-biet-gi-ve-nosql-database>]
- [5] Ưu điểm nhược điểm của SQL [<https://nq.com.vn/uu-diem-va-nhuoc-diem-cua-sql/>]
- [6] Big data là gì? Tất tần tậ về BigData [<https://topdev.vn/blog/big-data/>]
- [7] HYBRID DATABASE SYSTEM FOR BIG DATA STORAGE AND MANAGEMENT Blessing E. James and P.O.Asagba Department of Computer Science, University of Port Harcourt, Choba, Rivers State, Nigeria