

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

**Giáo viên hướng dẫn**  
(Ký tên và ghi rõ họ tên)

*năm*  
.....  
н а м

## This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Trong thời gian thực hiện đồ án cơ sở ngành: “Tìm hiểu Cơ sở dữ liệu MongoDB và ứng dụng” vừa qua em xin chân thành cảm ơn cô Hà Thị Thúy Vi đã hỗ trợ và hướng dẫn em trong suốt quá trình thực hiện đồ án đồ án. Em rất trân quý những đóng góp hỗ trợ của cô dành cho em. Trong suốt quá trình thực hiện đồ án em còn nhiều thiếu sót. Em rất mong cô thông cảm và đóng góp ý kiến để em khắc phục những thiếu sót này.

Sau cùng em xin chân thành cảm ơn cô vì những đóng góp bổ ích vừa qua.

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	8
1.1. Giới thiệu tổng quan về ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB: ..	8
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT.....	10
2.1. Cơ sở dữ liệu SQL server.....	10
2.1.1. Giới thiệu về cơ sở dữ liệu SQL server.....	10
2.1.2. Cấu trúc của SQL Server.....	11
2.2. Ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB .....	12
2.2.1. Ngôn ngữ NoSQL là gì ?.....	12
2.2.2. Khái niệm về hệ quản trị cơ sở dữ liệu MongoDB .....	14
2.2.3. MongoDB dùng để làm gì .....	15
2.2.4. Bắt đầu với MongoDB .....	15
2.2.5. Khởi động cùng MongoDB .....	19
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	22
3.1. Mô hình hóa dữ liệu trong SQL sever phiên bản 2008 R2 .....	22
3.2. Mô hình hóa dữ liệu trong MongoDB Compass .....	27
3.3 So sánh sự khác nhau của hai ngôn ngữ SQL và NoSQL.....	32
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....	34
4.1. Tạo, sửa, xóa cơ sở dữ liệu SQL .....	34
4.1.1. Tạo cơ sở dữ liệu cho bảng SINHVIEN. ....	34
4.1.2. Sửa dữ liệu trong bảng SINHVIEN .....	34
4.1.3. Xóa toàn bộ dữ liệu trong bảng SINHVIEN .....	35
4.1.4 Các câu lệnh truy vấn trong SQL.....	35
4.2. Tạo, sửa ,xóa collection MongoDB cơ bản.....	39
4.2.1. Tạo collection trong MongoDB.....	39
4.2.2. Xóa collection trong MongoDB .....	39
4.2.3. Thêm dữ liệu trong MongoDB .....	40
4.2.4. Xóa dữ liệu .....	41
4.2.5 Sửa dữ liệu.....	42
4.2.6 LƯU TRỮ, XUẤT FILE TRONG MONGODB.....	43
4.3. TRUY VẤN DỮ LIỆU TRONG MONGODB .....	45
DANH MỤC TÀI LIỆU THAM KHẢO .....	53
PHỤ LỤC .....	54

## DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1: SQL Server.....	10
Hình 2: Cấu trúc của SQL Server .....	11
Hình 3: Mở rộng MongoDB bằng cách sử dụng Sharding trên nhiều máy chủ .....	15
Hình 4: Giao diện bắt đầu SQL.....	22
Hình 5: Tạo mới database .....	22
Hình 6: Tạo database QUANLY_GIANGDAY .....	23
Hình 7: Tạo các tables .....	23
Hình 8: Các tables sau khi tạo xong.....	24
Hình 9: Tạo lưu đồ Diagram.....	25
Hình 10: Lưu đồ Diagram.....	25
Hình 11: Tải file cài đặt.....	27
Hình 12: Bắt đầu quá trình tải.....	27
Hình 13: Thỏa thuận cấp phép cho người dùng .....	28
Hình 14: Chọn kiểu thiết lập.....	28
Hình 15: Cấu hình dịch vụ.....	29
Hình 16: Cài đặt MongoDB.....	29
Hình 17: Bắt đầu quá trình cài đặt .....	30
Hình 18: Kết thúc quá trình cài đặt .....	30
Hình 19: Giao diện khi bắt đầu thao tác trên MongoDB.....	31
Hình 20: Tạo cơ sở dữ liệu .....	31
Hình 21: Kết quả sau khi tạo cơ sở dữ liệu.....	32
Hình 22: Kết quả sau khi thêm dữ liệu.....	34
Hình 23: Kết quả sửa dữ liệu .....	34
Hình 24: Kết quả khi thực hiện lệnh xóa.....	35
Hình 26: Kết quả truy vấn hiển thị danh sách các giảng viên .....	36
Hình 27: Kết quả truy vấn Hiển thị danh sách các giảng viên giảng dạy các môn học .....	37
Hình 28: Kết quả truy vấn hiển thị danh sách các giảng viên không dạy môn học nào trong năm học 2011-2012.....	37
Hình 29: Kết quả truy vấn Hiển thị mã, tên các môn học được sinh viên Nguyễn Thị Hoài Thu đăng ký học.....	38
Hình 30: Kết quả truy vấn đếm số môn học mà sinh viên Trần Ngọc Thanh đã đăng ký học.....	38
Hình 31: Truy vấn toàn bộ dữ liệu trong một collection .....	45
Hình 32: Truy vấn dữ liệu với điều kiện 1 .....	46
Hình 33: Truy vấn dữ liệu với điều kiện 2 .....	46
Hình 34: Truy vấn sử dụng điều kiện logic .....	47
Hình 35: Truy vấn sắp xếp kết quả tăng dần .....	48
Hình 36: Truy vấn sắp xếp kết quả giảm dần .....	48
Hình 37: Truy vấn giới hạn số lượng kết quả .....	49
Hình 38: Truy vấn sử dụng projection (chọn lọc các trường) .....	50
Hình 39: Truy vấn sử dụng regex .....	50
Hình 40: Truy vấn sử dụng logical AND và OR .....	51

## TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Tóm tắt vấn đề nghiên cứu:

- Tìm hiểu về hệ quản trị CSDL MongoDB, ngôn ngữ truy vấn NoSQL.
- Thiết kế CSDL cụ thể bằng 2 ngôn ngữ SQL và NoSQL.
- So sánh sự khác nhau của 2 ngôn ngữ SQL và NoSQL.

Hướng tiếp cận:

- Sử Dụng MongoDB:

MongoDB là một công cụ đồ họa giúp bạn quản lý, tương tác, và trực quan hóa cơ sở dữ liệu MongoDB của mình.

- Sử dụng SQL Server Management Studio (SSMS):

SSMS là một công cụ quản lý SQL Server chính thức từ Microsoft. Nó cung cấp một giao diện đồ họa để quản lý cơ sở dữ liệu, tạo, chỉnh sửa và thực hiện truy vấn SQL.

Cách giải quyết vấn đề:

- Sử dụng MongoDB để tạo Documents và các Collection với tập dữ liệu QUANLY\_GIANGDAY. Sử dụng lệnh để liệt kê dữ liệu có trong tập dữ liệu.
- Sử dụng SQL Server tạo các tables, nhập dữ liệu và sử dụng các câu truy vấn để xuất dữ liệu có trong tập dữ liệu QUANLY\_GIANGDAY đã được tạo.

Một số kết quả đạt được:

- Sử dụng MongoDB xây dựng tập dữ liệu QUANLY\_GIANGDAY. Sử dụng các lệnh liệt kê chi tiết giúp cho người dùng dễ dàng tìm kiếm cơ sở dữ liệu một cách nhanh chóng.
- Sử dụng SQL Server tạo thành công tập dữ liệu QUANLY\_GIANGDAY. Dùng lệnh thêm dữ liệu, xóa dữ liệu, sửa dữ liệu và các câu lệnh truy vấn để xuất các thông tin trong tập dữ liệu đã được tạo.

## MỞ ĐẦU

### Lý do chọn đề tài:

NoSql hiện nay đã được sử dụng ở rất nhiều công ty, tập đoàn lớn. Ngay cả Facebook cũng sử dụng điều này. Google cũng cần đến một dạng của NoSql để phát triển BigTable.

Có nhiều công ty toàn cầu sử dụng MongoDB để lưu trữ lượng dữ liệu “khổng lồ” của họ như Facebook, Nokia, eBay, Adobe, Google. MongoDB là một công cụ có thể quản lý thông tin hướng document cũng như lưu trữ hoặc truy xuất thông tin. MongoDB giúp lưu trữ lượng lớn dữ liệu trong khi vẫn hoạt động nhanh chóng, tích hợp một lượng lớn dữ liệu đa dạng.

Vì NoSQL là cơ sở dữ liệu phi quan hệ nên người dùng dễ dàng tiếp cận sử dụng một cách nhanh chóng đó là lý do cho các công ty lựa chọn cơ sở dữ liệu để thực hiện các dự án.

### Mục đích sử dụng:

Xây dựng cơ sở dữ liệu QUANLY\_GIANGDAY bằng ngôn ngữ SQL và ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB. Cho thấy được điểm khác nhau khi sử dụng hai ngôn ngữ.

### Đối tượng nghiên cứu:

a) Nghiên cứu lý thuyết: MongoDB là một cơ sở dữ liệu đa nền tảng hoạt động trên các khái niệm collection và document giúp người dùng thêm, sửa, xóa dữ liệu một cách nhanh chóng.

b) Nghiên cứu thực nghiệm:

- Phân tích dữ liệu: xây dựng , thêm , sửa , xóa và truy vấn dữ liệu
- Trực quan hóa dữ liệu : xây dựng luồng dữ liệu để trực quan hóa một cách dễ dàng.

**Phạm vi nghiên cứu:** Thêm dữ liệu, xóa dữ liệu, sửa dữ liệu và viết các lệnh truy vấn cho ngôn ngữ SQL và ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB.

## CHƯƠNG 1: TỔNG QUAN

### 1.1. Giới thiệu tổng quan về ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB:

Cơ sở dữ liệu SQL là cơ sở dữ liệu quan hệ bằng việc lưu trữ thông tin trong các bảng riêng biệt. Khi cần dữ liệu sẽ được truy vấn từ nhiều bảng để đưa ra kết quả của câu lệnh thực thi. Vì vậy mà khi thao tác tạo cơ sở dữ liệu chúng ta cần lưu ý nắm rõ các khóa chính, khóa ngoại và các ràng buộc.

Ngược lại với ngôn ngữ SQL cơ sở dữ liệu quan hệ là ngôn ngữ NoSQL được coi là cơ sở dữ liệu phi quan hệ. NoSQL được phát triển vào cuối những năm 2000 để khắc phục các vấn đề và hạn chế của SQL. Ngôn ngữ NoSQL không bị giới hạn bởi không gian dữ liệu, không yêu cầu một lược đồ quan hệ cố định, tránh các phép nối và dễ dàng cho người dùng tìm kiếm thông tin dữ liệu một cách nhanh chóng. Cơ sở dữ liệu NoSQL được sử dụng cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu khổng lồ. Hệ thống cơ sở dữ liệu NoSQL bao gồm nhiều loại công nghệ cơ sở dữ liệu khác nhau như MongoDB là cơ sở dữ liệu dưới dạng tài liệu, Apache Cassandra là cơ sở dữ liệu phân tán dựa trên cột, Amazon DynamoDB là cơ sở dữ liệu quản lý key-value trên nền tảng AWS, InfluxDB là hệ thống cơ sở dữ liệu chuyên dụng cho việc lưu trữ và truy vấn dữ liệu chuỗi thời gian. Các cơ sở dữ liệu này được phát triển để đáp ứng nhu cầu trình bày cho sự phát triển của các ứng dụng hiện đại. Khác với database SQL là cơ sở dữ liệu dựa trên bảng thì database NoSQL có thể dựa trên tài liệu, cặp khóa-giá trị, cơ sở dữ liệu biểu đồ. Tuy nhiên ngôn ngữ NoSQL không phù hợp với các câu lệnh truy vấn phức tạp. NoSQL phụ thuộc vào hệ thống phần mềm được sử dụng để quản lý và tổ chức cơ sở dữ liệu còn được gọi là hệ quản trị cơ sở dữ liệu. Hệ quản trị cơ sở dữ liệu cung cấp một giao diện giữa cơ sở dữ liệu và người dùng hoặc các ứng dụng để thực hiện các hoạt động như tạo, sửa đổi, truy vấn và xóa dữ liệu. Nhanh hơn SQL thì NoSQL không chuẩn hóa cho phép bạn lấy được tất cả thông tin về một mục cụ thể với các điều lệ mà không cần tham gia liên quan hoặc truy vấn SQL phức tạp. Vậy NoSQL phù hợp với những dự án yêu cầu dữ liệu không liên quan, khó xác định, đơn giản mềm dẻo khi đang phát triển.

MongoDB là cơ sở dữ liệu dưới dạng tài liệu có thể được cài đặt cục bộ hoặc được lưu trữ trên đám mây. MongoDB lưu trữ dữ liệu trong các tài liệu linh hoạt. Thay vì có nhiều bảng như SQL, MongoDB chỉ cần giữ tất cả dữ liệu liên



quan của mình lại với nhau. Việc này giúp ích cho người dùng đọc và tra cứu dữ liệu nhanh chóng. MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1. Cơ sở dữ liệu SQL server



Hình 1: SQL Server

#### 2.1.1. Giới thiệu về cơ sở dữ liệu SQL server

SQL Server là một hệ quản trị cơ sở dữ liệu (DBMS) do Microsoft phát triển. Nó là một trong những hệ quản trị cơ sở dữ liệu phổ biến và mạnh mẽ nhất trên thị trường. Phát triển và lịch sử: SQL Server được Microsoft phát triển và ra mắt lần đầu vào năm 1989. Nó đã trải qua nhiều phiên bản và cập nhật từ đó, bao gồm SQL Server 2000, 2005, 2008, 2012, 2014, 2016, 2017, 2019 và các phiên bản Azure SQL Database dựa trên đám mây.

Các phiên bản và chức năng: SQL Server có nhiều phiên bản dành cho các mục đích khác nhau, từ phiên bản Express miễn phí cho ứng dụng nhỏ đến phiên bản Enterprise dành cho các hệ thống lớn. Nó hỗ trợ các tính năng như truy vấn dữ liệu, lập trình trên cơ sở dữ liệu, tích hợp với các ứng dụng và nhiều tính năng bảo mật mạnh mẽ....

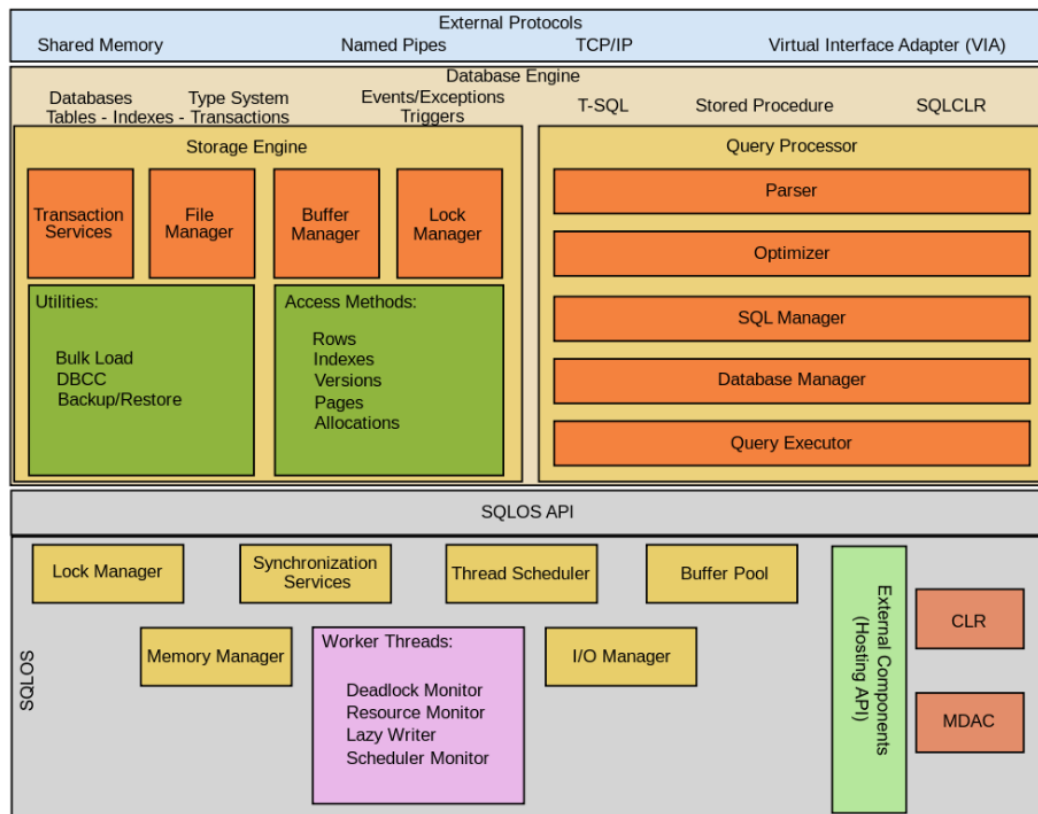
Ngôn ngữ SQL trong SQL Server:

SQL (Structured Query Language): SQL Server sử dụng ngôn ngữ SQL để tương tác với cơ sở dữ liệu. SQL là một ngôn ngữ lập trình dùng để truy vấn và quản lý dữ liệu trong cơ sở dữ liệu. Ngôn ngữ này bao gồm các câu lệnh và cú pháp cho phép bạn thực hiện các thao tác như truy vấn dữ liệu, thêm, cập nhật, xóa và tạo bảng.

Tính năng mạnh mẽ: SQL Server hỗ trợ nhiều tính năng mạnh mẽ của SQL, bao gồm truy vấn dữ liệu phức tạp, gán kết dữ liệu từ nhiều bảng, xử lý giao dịch và quản lý bảo mật thông qua quyền truy cập và phân quyền.

### 2.1.2. Cấu trúc của SQL Server

Sơ đồ dưới đây minh họa cấu trúc của SQL Server:



Hình 2: Cấu trúc của SQL Server

## 2.2. Ngôn ngữ NoSQL hệ quản trị cơ sở dữ liệu MongoDB

### 2.2.1. Ngôn ngữ NoSQL là gì ?

Thuật ngữ NoSQL (viết tắt của “Not Only SQL” hoặc “Not SQL”) được sử dụng lần đầu tiên vào năm 1998 cho một cơ sở dữ liệu không có giao diện SQL. NoSQL phát triển mạnh suốt những năm 2000 trong sự mở rộng nhanh chóng của Internet.

Cơ sở dữ liệu NoSQL là một Hệ thống quản lý dữ liệu không quan hệ có lược đồ linh hoạt, dễ mở rộng. Mục đích chính của việc sử dụng cơ sở dữ liệu NoSQL là dành cho các kho dữ liệu phân tán với nhu cầu lưu trữ dữ liệu lớn. NoSQL được sử dụng cho dữ liệu lớn và ứng dụng web thời gian thực. Chẳng hạn các công ty như Twitter, Facebook và Google thu thập hàng terabyte dữ liệu người dùng mỗi ngày.

- Cơ sở dữ liệu NoSQL có các đặc tính sau:
  1. Mô hình cơ sở dữ liệu là phi quan hệ.
  2. Tập trung cho khả năng phân tán và mở rộng thuộc tính theo chiều rộng.
  3. Quy tắc cho lược đồ thì yếu hoặc không có.
  4. Sao chép dữ liệu dễ dàng.
  5. Dễ dàng truy cập được cung cấp thông qua một API.
  6. Mô hình nhất quán không phải là ACID.
- Các dạng cơ sở dữ liệu NoSQL

Có nhiều hệ thống cơ sở dữ liệu NoSQL như: MongoDB, RavenDB, Redis, Neo4j,... Xong có thể chia NoSQL thành 4 loại:

1. Key-Value stores gồm có các đặc tính sau:
  - Có một tập các đối tượng dữ liệu nhận dạng, các khóa
  - Đối với mỗi khóa, có chính xác một đối tượng dữ liệu mô tả có liên quan, giá trị cho khóa đó
  - Chỉ định một khóa cho phép truy vấn giá trị liên quan trong cơ sở dữ liệu.

Database tiêu biểu: : Riak, Redis, MemCache, Project Voldemort, CouchBase

Ứng dụng: Do tốc độ truy xuất nhanh, key-value stores thường được dùng để làm cache cho ứng dụng.

2. Document stores gồm có các đặc tính sau:

Dữ liệu sẽ được lưu trữ dưới dạng BSON/JSON/XML dưới database. Dữ liệu không có lược đồ cứng như SQL. Do đó có thể thêm, sửa các thuộc tính, thay đổi bảng, ... rất nhanh và đơn giản. Cơ sở dữ liệu dạng này có tốc độ truy vấn nhanh, có thể thực hiện các câu truy vấn phức tạp, dễ mở rộng. Mỗi cơ sở dữ liệu có một kiểu truy vấn riêng.

Database tiêu biểu: MongoDB, RavenDB, CouchDB, TerraStone, OrientDB.

Ứng dụng: Do nhanh và linh động, document database thường đóng vai trò làm cơ sở dữ liệu cho các ứng dụng prototype, big data, e-commerce, CMS. Ngoài ra, ta còn dùng nó để lưu nhật ký hoặc lịch sử.

3. Column-Family Databases gồm có các đặc tính sau:

Dữ liệu được lưu trong CSDL dưới dạng các cột, thay vì các hàng như SQL. Mỗi hàng sẽ có một key/id riêng. Điểm đặc biệt là các hàng trong một bảng sẽ có số lượng cột khác nhau. Câu lệnh truy vấn của nó khá giống SQL.

Database tiêu biểu: Cassandra (Phát triển bởi Facebook), HyperTable, Apache Hbase.

Ứng dụng: Column-Family Database được sử dụng khi ta cần ghi một số lượng lớn dữ liệu. Nó còn được ứng dụng trong một số hệ thống quản trị nội dung và ứng dụng thương mại điện tử.

4. Graph Databases gồm có các đặc tính sau:

Dữ liệu trong graph database được lưu dưới dạng các nút. Mỗi nút sẽ có một nhãn, một số thuộc tính như một hàng trong SQL. Graph database tập trung nhiều vào mối quan hệ giữa các nút, áp dụng nhiều thuật toán duyệt nút để tăng tốc độ.

Database tiêu biểu: Neo4j, InfiniteGraph, OrientDB, HYPERGRAPHDB

Ứng dụng: Khi cần truy vấn các mối quan hệ, graph database truy vấn nhanh và dễ hơn nhiều so với database. Nó được dùng trong các hệ thống: mạng nơ ron, chuyển tiền bạc, mạng xã hội, giới thiệu sản phẩm.

- Một số thuật ngữ tương đương trong Relational Database và NoSQL database

<b>RELATIONAL</b>	<b>KEY-VALUE (RIAK)</b>	<b>DOCUMENT (MONGODB)</b>	<b>COLUMN-FAMILY (CASSANDRA)</b>	<b>GRAPH (NEO4J)</b>
instance	cluster	mongodb	cluster	instance
table	bucket	collection	column-family	label
row	key-value	document	row	node
row-id	key	_id		
schema		database		schema

Thuật ngữ có trong NoSQL

### 2.2.2. Khái niệm về hệ quản trị cơ sở dữ liệu MongoDB

MongoDB là một cơ sở dữ liệu hướng tài liệu, không phải là cơ sở dữ liệu quan hệ. Lý do chính khiến chúng ta chuyển từ mô hình quan hệ truyền thống sang là để làm cho quá trình mở rộng dễ dàng hơn, tuy nhiên, còn có một số ưu điểm khác.

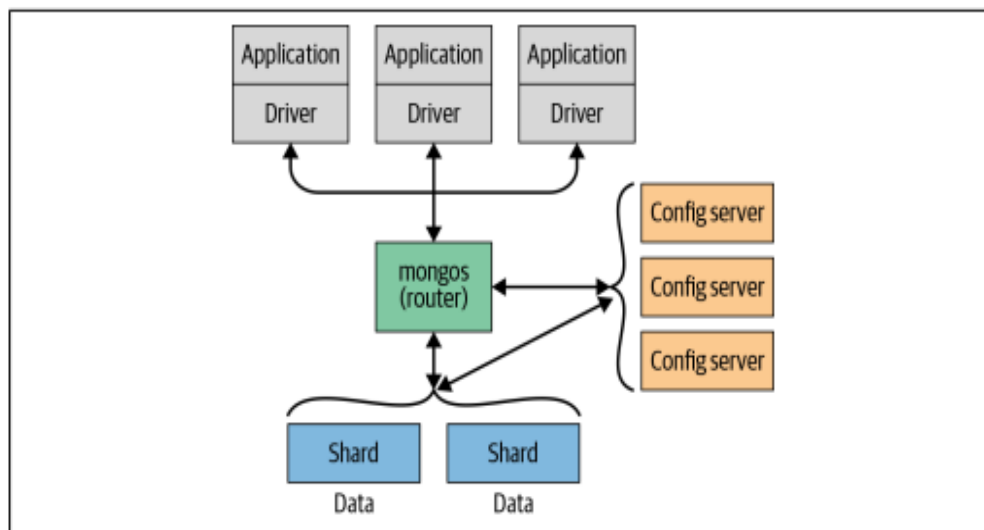
Trong cơ sở dữ liệu hướng tài liệu, khái niệm "dòng" được thay thế bằng một mô hình linh hoạt hơn, gọi là "tài liệu". Bằng cách cho phép nhúng tài liệu và mảng, phương pháp hướng tài liệu cho phép biểu diễn các mối quan hệ phức tạp theo cách hiệu quả với một bản ghi duy nhất. Điều này phù hợp tự nhiên với cách mà các nhà phát triển trong các ngôn ngữ lập trình hướng đối tượng hiện đại nghĩ về dữ liệu của họ.

Một điểm nổi bật khác của cơ sở dữ liệu hướng tài liệu là sự thiếu hụt của các mô hình mẫu cố định. Trong các cơ sở dữ liệu này, các khóa và giá trị của một tài liệu không bị ràng buộc vào các loại hoặc kích thước cố định. Điều này cho phép thêm hoặc loại bỏ trường khi cần trở nên dễ dàng. Điều này, trong tương lai, giúp tăng tốc quá trình phát triển, vì các nhà phát triển có thể nhanh chóng lặp lại và thử

nghiệm các mô hình dữ liệu khác nhau. Họ có thể khám phá nhiều mô hình dữ liệu và chọn lựa mô hình phù hợp nhất với nhu cầu của họ.

### 2.2.3. MongoDB dùng để làm gì

MongoDB được thiết kế để mở rộng ra. Mô hình dữ liệu hướng tài liệu giúp dễ dàng chia dữ liệu thành nhiều máy chủ. MongoDB tự động quản lý việc cân bằng dữ liệu và tải trên một cụm, tự động phân phối các tài liệu lại và định tuyến các hoạt động đọc và ghi đúng vào các máy tính thích hợp.



Hình 3: Mở rộng MongoDB bằng cách sử dụng Sharding trên nhiều máy chủ

### 2.2.4. Bắt đầu với MongoDB

Một số khái niệm cơ bản về MongoDB:

- **Document**

Là một tập hợp có thứ tự của các khóa kèm theo giá trị tương ứng. Biểu diễn của một tài liệu có thể thay đổi theo ngôn ngữ lập trình, nhưng hầu hết các ngôn ngữ đều có cấu trúc dữ liệu phù hợp, chẳng hạn như một bản đồ (map), bảng băm (hash), hoặc từ điển (dictionary).

Trong JavaScript, ví dụ, tài liệu được biểu diễn dưới dạng các đối tượng (objects):

```
{"greeting" : "Hello, world!"}
```

Tài liệu đơn giản này chứa một khóa duy nhất, "greeting", với giá trị là "Hello, world!". Hầu hết các tài liệu sẽ phức tạp hơn tài liệu đơn giản này và thường sẽ chứa nhiều cặp khóa/giá trị:

```
{"greeting" : "Hello, world!", "views" : 3}
```

Giá trị trong các tài liệu không chỉ là "blobs" (chuỗi vô hình). Trong ví dụ này, giá trị cho "greeting" là một chuỗi (string), trong khi giá trị cho "views" là một số nguyên (integer).

Các khóa trong một tài liệu là chuỗi. Bất kỳ ký tự UTF-8 nào cũng được phép trong một khóa, với một số ngoại lệ đáng chú ý:

- Khóa không được chứa ký tự \0 (ký tự null). Ký tự này được sử dụng để chỉ thị kết thúc của một khóa.

- Các ký tự . và \$ có một số tính năng đặc biệt và nên được sử dụng chỉ trong những tình huống cụ thể, như mô tả trong các chương sau. Nói chung, chúng nên được coi là đã được dự trữ, và các trình điều khiển sẽ phản nản nếu chúng được sử dụng một cách không thích hợp.

MongoDB là một hệ thống nhạy cảm với kiểu dữ liệu và phân biệt chữ hoa. Ví dụ, những tài liệu sau đây là khác biệt:

```
{ "count" : 5 }  
{ "count" : "5" }
```

và cũng khác nhau như sau:

```
{ "count" : 5 }  
{ "Count" : 5 }
```

Một điều quan trọng cuối cùng cần lưu ý là tài liệu trong MongoDB không thể chứa các khóa trùng lặp. Ví dụ, sau đây không phải là một tài liệu hợp lệ:

```
{ "greeting" : "Hello, world!", "greeting" : "Hello, MongoDB!" }
```

- **Collection:**

Một bộ sưu tập (Collection) là một nhóm các tài liệu. Nếu một tài liệu là phiên bản MongoDB tương tự của một hàng trong cơ sở dữ liệu quan hệ, thì một bộ sưu tập có thể được coi là phiên bản tương tự của một bảng.

- **Dynamic Schemas**

Các bộ sưu tập (Collections) có schema động. Điều này có nghĩa là các tài liệu trong cùng một bộ sưu tập có thể có bất kỳ số lượng "hình dạng" (shape) khác nhau nào. Ví dụ, cả hai tài liệu sau đây có thể được lưu trữ trong cùng một bộ sưu tập:

```
{ "greeting" : "Hello, world!", "views": 3 }  
{ "signoff": "Good night, and good luck" }
```



Lưu ý rằng các tài liệu trước đó có các khóa khác nhau, số lượng khóa khác nhau và giá trị có các kiểu khác nhau. Vì bất kỳ tài liệu nào cũng có thể được đặt vào bất kỳ bộ sưu tập.

Có một số lý do tốt:

- Lưu trữ các loại tài liệu khác nhau trong cùng một bộ sưu tập có thể làm phiền cho nhà phát triển và quản trị viên. Nhà phát triển cần đảm bảo rằng mỗi truy vấn chỉ trả về các tài liệu tuân thủ một schema cụ thể hoặc rằng mã ứng dụng thực hiện truy vấn có thể xử lý các tài liệu có hình dạng khác nhau. Nếu chúng ta đang truy vấn các bài viết trên blog, việc phải loại bỏ các tài liệu chứa dữ liệu tác giả là một phiền toái.
- Nhanh hơn để có được danh sách các bộ sưu tập thay vì trích xuất danh sách các loại tài liệu trong một bộ sưu tập. Ví dụ, nếu chúng ta có một trường "type" trong mỗi tài liệu xác định loại tài liệu đó là "skim," "whole," hoặc "chunky monkey," việc tìm kiếm ba giá trị này trong một bộ sưu tập duy nhất sẽ chậm hơn nhiều so với việc có ba bộ sưu tập riêng biệt và truy vấn chính xác bộ sưu tập cần.
- Gom nhóm các tài liệu cùng loại vào cùng một bộ sưu tập cho phép tận dụng dữ liệu cục bộ. Lấy một số bài viết từ một bộ sưu tập chỉ chứa các bài viết sẽ có ít truy cập đĩa hơn so với việc lấy cùng các bài viết từ một bộ sưu tập chứa cả bài viết và dữ liệu tác giả.
- Đặt một số cấu trúc lên các tài liệu khi chúng ta tạo các chỉ mục. Các chỉ mục này được định nghĩa trên mỗi bộ sưu tập. Bằng cách đặt chỉ tài liệu của một loại vào cùng một bộ sưu tập, chúng ta có thể tạo chỉ mục cho các bộ sưu tập của mình một cách hiệu quả hơn.

Có nhiều lý do hợp lý để tạo schema và nhóm các loại tài liệu liên quan lại với nhau. Mặc dù không bắt buộc mặc định, việc định nghĩa schema cho ứng dụng của bạn là một phương pháp tốt và có thể được thực hiện thông qua việc sử dụng chức năng xác nhận tài liệu của MongoDB và các thư viện ánh xạ đối tượng-tài liệu có sẵn cho nhiều ngôn ngữ lập trình.

Có những lý do hợp lý để tạo schema và nhóm các loại tài liệu liên quan lại với nhau. Mặc dù không bắt buộc theo mặc định, việc định nghĩa schema cho ứng dụng

của bạn là một thực hành tốt và có thể được thực hiện thông qua sử dụng chức năng xác thực tài liệu của MongoDB và các thư viện ánh xạ đối tượng-tài liệu có sẵn cho nhiều ngôn ngữ lập trình.

- **Naming**

Một bộ sưu tập được xác định bằng tên của nó. Tên bộ sưu tập có thể là bất kỳ chuỗi UTF-8 nào, với một số hạn chế:

- Chuỗi trống ("" ) không phải là một tên bộ sưu tập hợp lệ.
- Tên bộ sưu tập không được chứa ký tự \0 (ký tự null), vì ký tự này đánh dấu kết thúc của tên bộ sưu tập.
- Bạn không nên tạo bất kỳ bộ sưu tập nào có tên bắt đầu bằng "system.", một tiền tố dành cho các bộ sưu tập nội bộ. Ví dụ, bộ sưu tập "system.users" chứa thông tin về người dùng của cơ sở dữ liệu, và bộ sưu tập "system.namespaces" chứa thông tin về tất cả các bộ sưu tập của cơ sở dữ liệu.
- Các bộ sưu tập do người dùng tạo ra không nên chứa ký tự \$ đã được dự trữ trong tên của họ. Các trình điều khiển khác nhau cho cơ sở dữ liệu hỗ trợ việc sử dụng \$ trong tên bộ sưu tập vì một số bộ sưu tập được tạo ra bởi hệ thống chứa nó, nhưng bạn không nên sử dụng \$ trong tên trừ khi bạn đang truy cập một trong những bộ sưu tập này.

- **Subcollections**

Một quy ước cho việc tổ chức các bộ sưu tập là sử dụng các bộ sưu tập con được chia tách bằng ký tự ".". Ví dụ, một ứng dụng chứa một blog có thể có một bộ sưu tập được đặt tên là "blog.posts" và một bộ sưu tập riêng biệt được đặt tên là "blog.authors". Điều này chỉ để phục vụ mục đích tổ chức - không có mối quan hệ giữa bộ sưu tập blog (nó thậm chí không cần tồn tại) và các "con cái" của nó.

Mặc dù các bộ sưu tập con không có bất kỳ tính năng đặc biệt nào, chúng rất hữu ích và được tích hợp vào nhiều công cụ MongoDB. Ví dụ:

- GridFS, một giao thức để lưu trữ các tệp lớn, sử dụng các bộ sưu tập con để lưu trữ dữ liệu về tệp riêng biệt so với các phần nội dung.

- Hầu hết các trình điều khiển cung cấp một số cú pháp đường dẫn cho việc truy cập vào các bộ sưu tập con của một bộ sưu tập đã cho. Ví dụ trong bảng shell cơ sở dữ liệu, `db.blog` sẽ cung cấp cho bạn bộ sưu tập blog, và `db.blog.posts` sẽ cung cấp cho bạn bộ sưu tập blog.posts.

Các bộ sưu tập con là một cách tốt để tổ chức dữ liệu trong MongoDB cho nhiều trường hợp sử dụng.

### 2.2.5. Khởi động cùng MongoDB

- Insert Documents

Có hai phương pháp để chèn tài liệu vào cơ sở dữ liệu:

Để chèn một tài liệu sử dụng insertOne()

```
db.SINHVIEN.insertOne({
  {
    "_id": {
      "$oid": "6586f5e9ba1ae3b1627d9351"
    },
    "MaSV": "K6100001",
    "MaMon": "DSTT",
    "NamHoc": "2012-2013",
    "HocKy": "2",
    "MaGV": "15.111.05",
    "Diem": "7"
  }
})
```

Để chèn nhiều tài liệu cùng một lúc sử dụng insertMany()

```
db.SINHVIEN.insertMany [  
  {  
    "MaSV": "K6100001",  
    "HoTenSV": "Phạm Văn Bình",  
    "NgàySinh": "02-24-1990",  
    "MaLop": "DA11TO12A1"  
  },  
  {  
    "MaSV": "K6100001",  
    "HoTenSV": "Phạm Văn Bình",  
    "NgàySinh": "02-24-1990",  
    "MaLop": "DA11TO12A1"  
  },  
  {  
    "MaSV": "K6100003",  
    "HoTenSV": "Trần Ngọc Thanh",  
    "NgàySinh": "04-15-1990",  
    "MaLop": "DA12DL12A1"  
  }  
]
```

- Find Data

Có hai phương pháp để tìm và chọn dữ liệu:

Để chọn dữ liệu từ bộ siêu tập trong MongoDB sử dụng find()

```
db.SINHVIEN.find()
```

Để chỉ chọn một tài liệu sử dụng findOne( )

```
db.SINHVIEN.findOne()
```

- Querying Data

Để truy vấn hoặc lọc dữ liệu có thể bao gồm một truy vấn trong các phương thức find() hoặc truy vấn một bản ghi đầu tiên phù hợp với điều kiện nhất định từ cơ sở dữ liệu findOne()

```
db.SINHVIEN.find({category: "News"})
```

- Projection

Cả hai phương thức tìm tài liệu đều chấp nhận tham số thứ hai được gọi là chiếu.

Tham số này mô tả những trường nào sẽ được đưa vào kết quả.

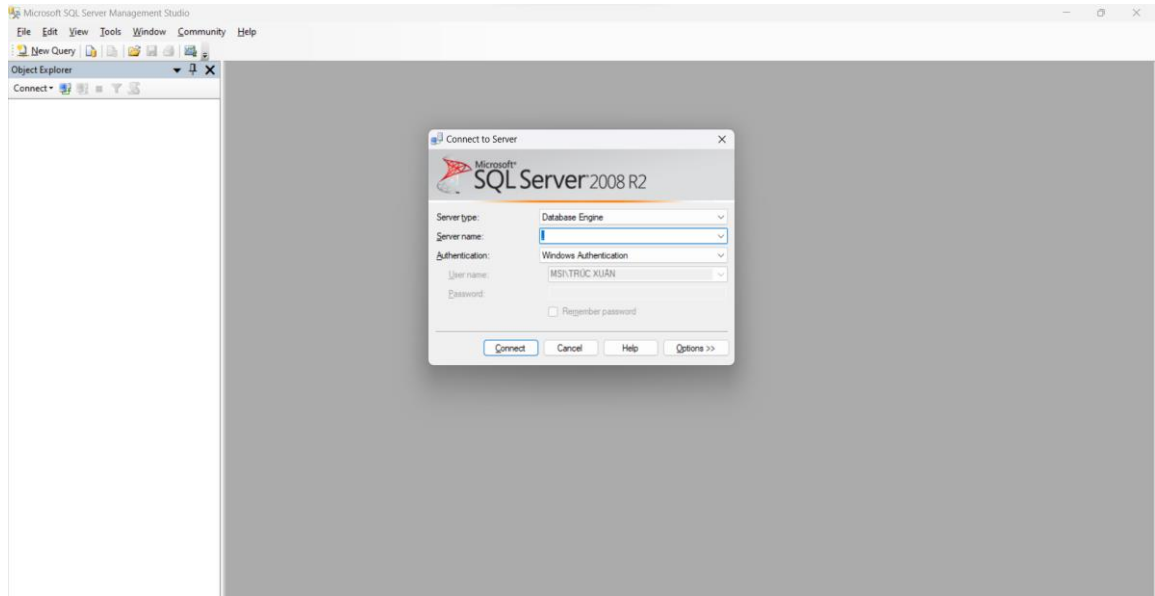
```
db.SINHVIEN.find({}, {HoTenSV: 1, NgaySinh: 1})
```

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô hình hóa dữ liệu trong SQL sever phiên bản 2008 R2

Hướng dẫn sử dụng SQL sever 2008 R2:

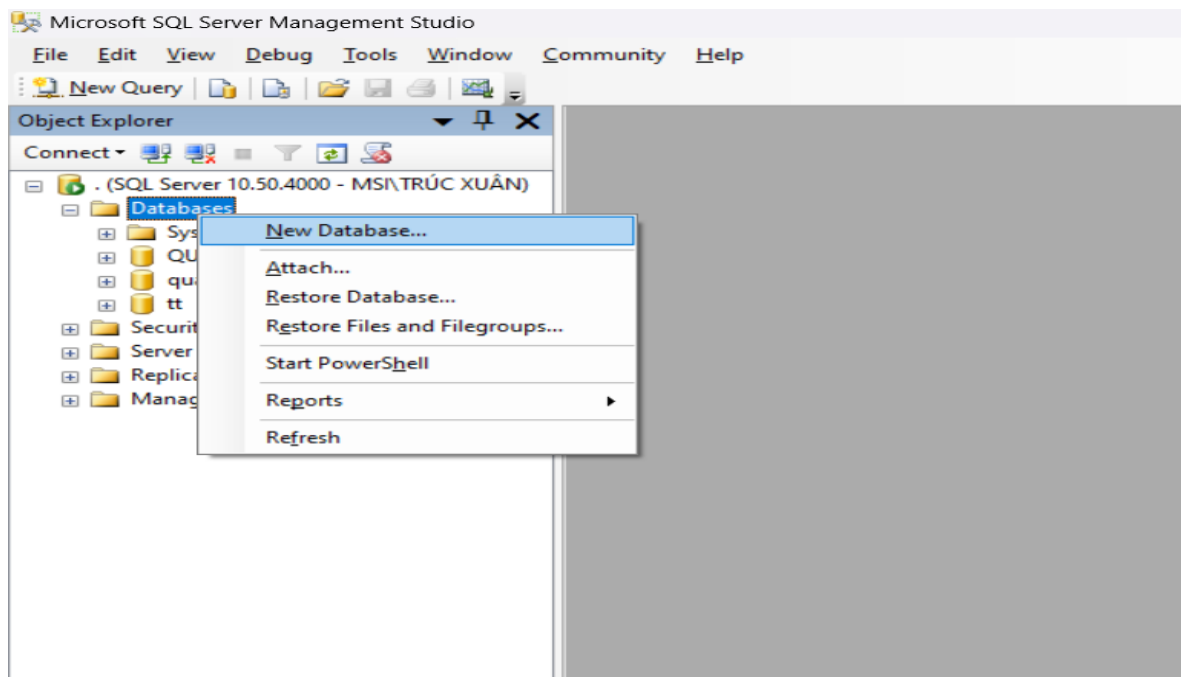
Start => Microsoft SQL Server 2008 R2 => SQL Server Management Studio



Hình 4: Giao diện bắt đầu SQL

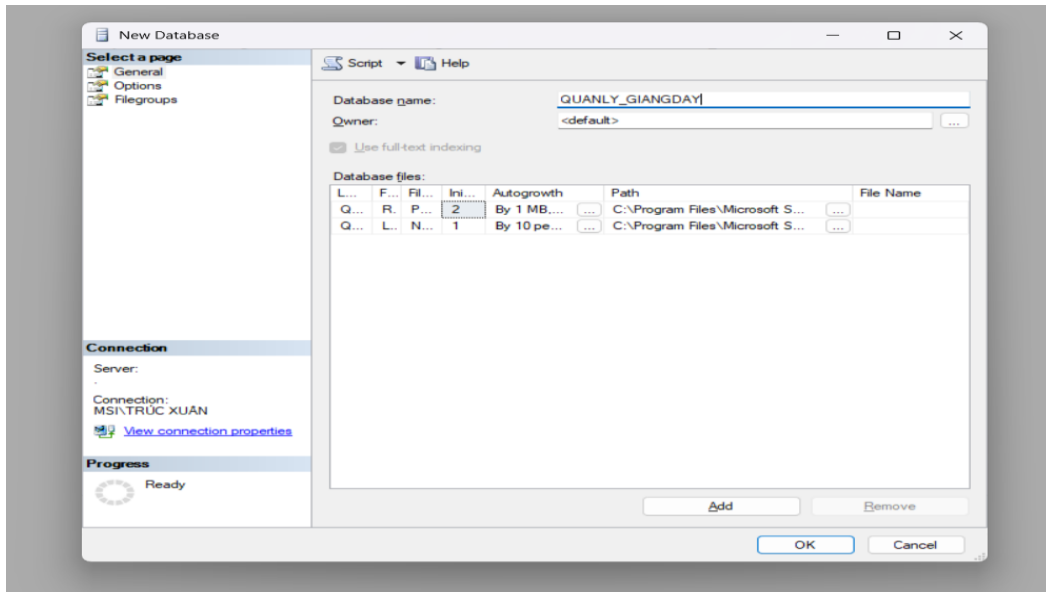
#### Tạo mới một database mới

- + Bước 1: Đầu tiên các bạn mở SQL lên rồi kết nối đến SQL Server.
- Các bạn chuột phải vào mục Database.
- Chọn New Database trong danh sách sổ xuống để tạo mới một database.



Hình 5: Tạo mới database

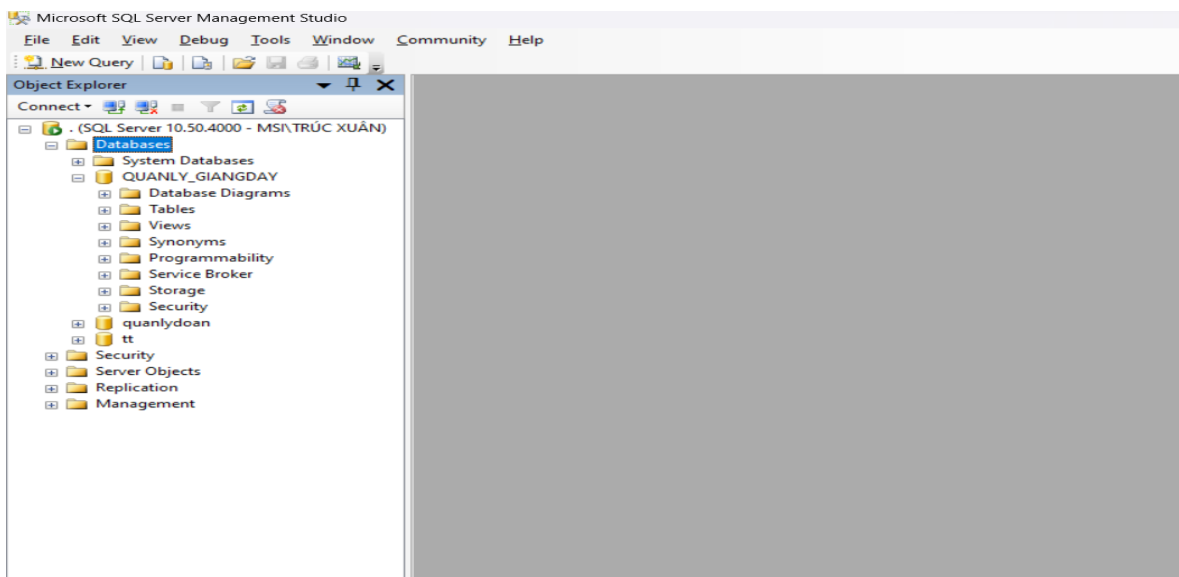
+ Bước 2: Tiếp theo các bạn đặt tên cho database QUANLY\_GIANGDAY trong mục Database name như hình bên dưới => Sau đó bấm OK để xác nhận tạo mới một database.



Hình 6: Tạo database QUANLY\_GIANGDAY

+ Bước 3: Sau khi tạo đây chính là cấu trúc chung của một database trong SQL Server. Về cơ bản cũng có các thành phần như Tables (các bảng), Views (các views)...

Ngoài ra thì SQL Server còn có thêm phần Database Diagrams (thiết kế của database) được tích hợp sẵn giúp trực quan hóa quan hệ giữa các bảng trong database.



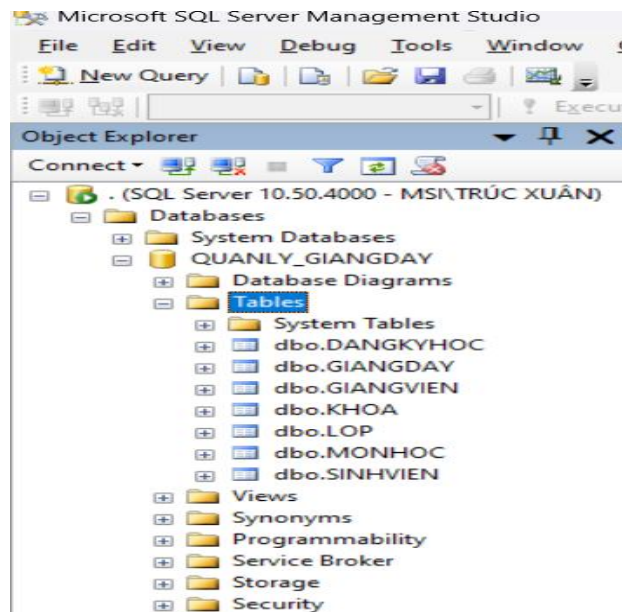
Hình 7: Tạo các tables

Cách tạo mới một table:

Tạo bảng bằng câu lệnh:

```
Create table SINHVIEN  
(  
MaSV char (8) primary key not null,  
HoTenSV nvarchar (30),  
NgàySinh datetime,  
MaLop char(10) constraint RB_ML foreign key (Malop) references LOP(MaLop)  
)
```

Sau khi thực thi xong câu lệnh sẽ xuất hiện các tables đã được tạo:



Hình 8: Các tables sau khi tạo xong

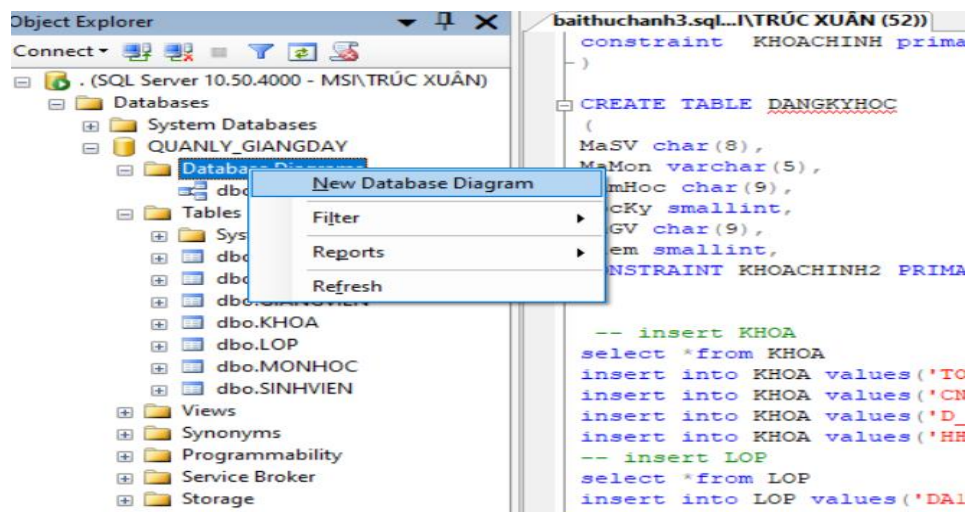
Sau khi tạo bảng xong chúng ta cần tạo lưu đồ Diagrams

Diagrams là một cửa sổ hiển thị mối quan hệ giữa các bảng của một cơ sở dữ liệu.

Tạo diagram ta thực hiện như sau: Chọn tên database cần tạo → Right Click → Database Diagrams → New Database Diagram

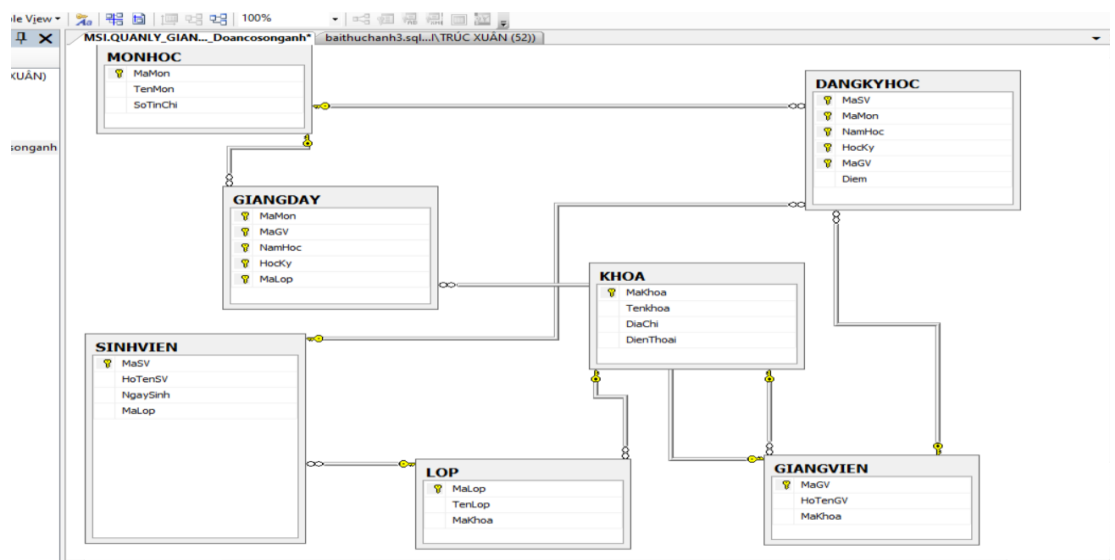
Vì SQL là cơ sở dữ liệu quan hệ nên khi tạo lưu đồ Diagram cần lưu ý về các khóa chính, khóa ngoại để khi truy vấn đạt được kết quả tốt. Đây là bước rất quan trọng khi xây dựng cơ sở dữ liệu bằng ngôn ngữ SQL và cũng là khuyết điểm gây khó khăn đối với người dùng mới sử dụng SQL.





Hình 9: Tạo lưu đồ Diagram

Sau khi thực thi xong sẽ xuất hiện các bảng trong Database Diagram và sau cùng nối các Database Diagram lại với nhau để tiến hành nhập liệu và truy vấn cơ sở dữ liệu:



Hình 10: Lưu đồ Diagram

Sau khi vẽ lưu đồ Database Diagram kết thúc chúng ta tiến hành nhập liệu vào trong cơ sở dữ liệu.

Lưu ý khi tiến hành nhập liệu vì SQL theo một cấu trúc nhất định nên người dùng phải tuân thủ đúng cách thực thi thêm dữ liệu.

```
insert into SINHVIEN values('K6100001','N'Phạm Văn Bình','02-24-1990','DA11TO12A1')
insert into SINHVIEN values('K6100002','N'Nguyễn Thị Hoài Thu','04-12-1991','DA12TT12A1')
insert into SINHVIEN values('K6100003','N'Trần Ngọc Thanh','04-15-1990','DA12DL12A1')
insert into SINHVIEN values('K6100004','N'Nguyễn Tấn Hùng','02-03-1992','DA12TT12A2')
insert into SINHVIEN values('K6100005','N'Trương Thành Sang','12-04-1990','DA12DL12A2')
insert into SINHVIEN values('K6100006','N'Nguyễn Anh Dũng','03-03-1982','DA12HH12A1')
```

Một số câu lệnh cần thiết khi sử dụng SQL.

-- 1) Hiển thị bảng :

```
select *
from SINHVIEN
```

-- 2) Để xóa bảng dùng lệnh:

```
DROP TABLE SINHVIEN;
```

-- 3) Sửa đổi cấu trúc bảng :

```
ALTER TABLE SINHVIEN
ADD HoTenSV nvarchar (10)
```

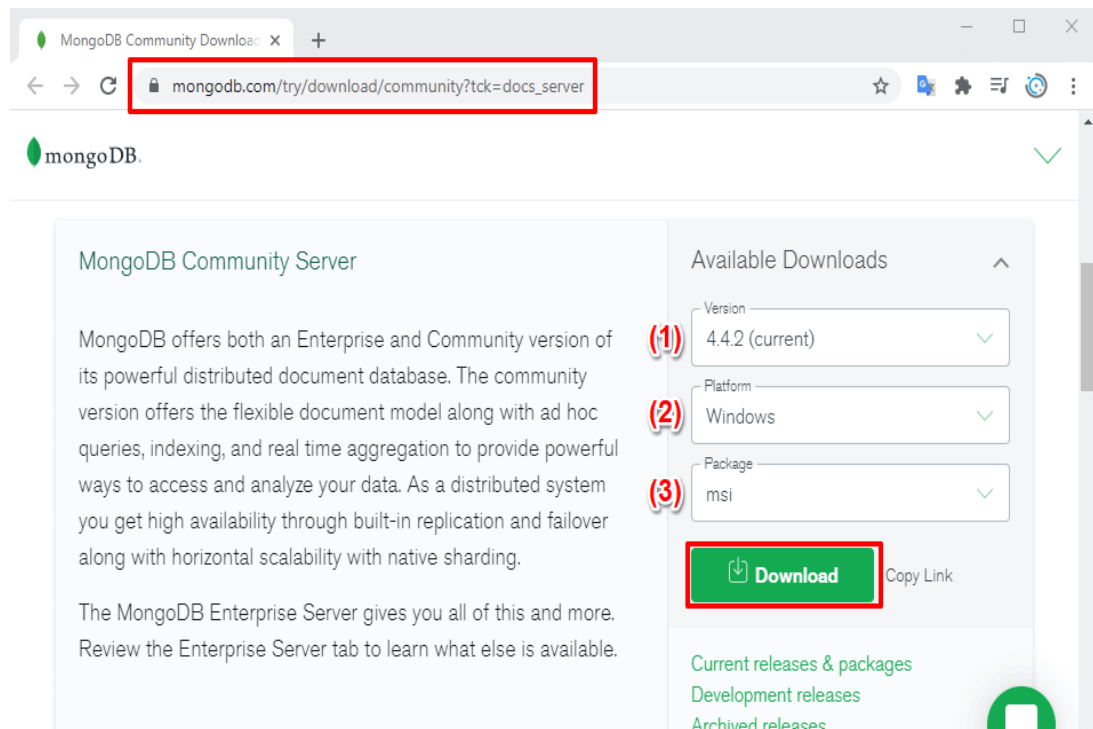
-- 4) Thêm bảng:

```
Create table SINHVIEN
```

### 3.2. Mô hình hóa dữ liệu trong MongoDB Compass

+ Bước 1: Đầu tiên các bạn vào đường dẫn sau để download file cài đặt của MongoDB tại đây:

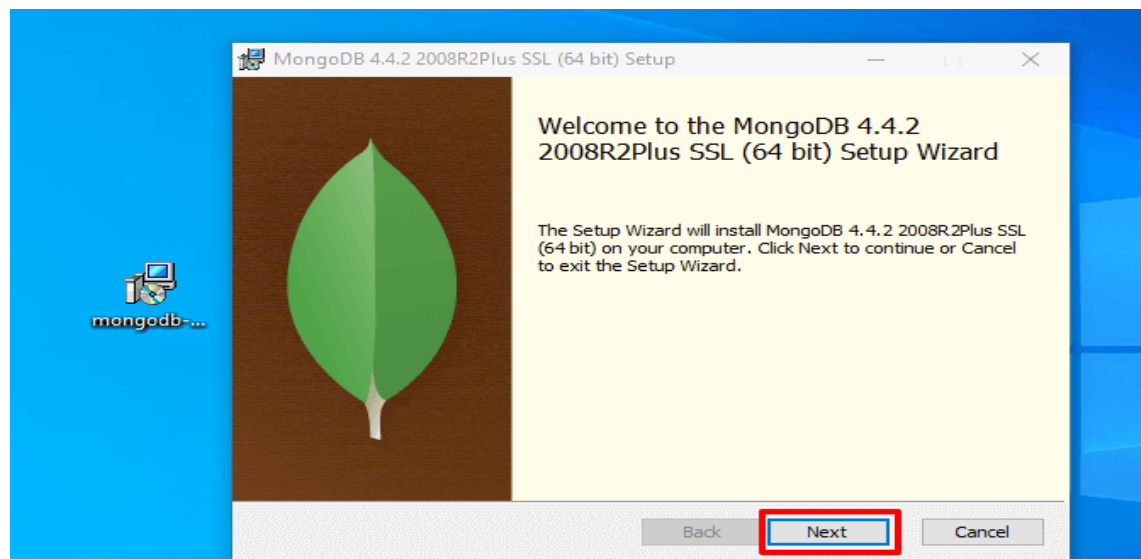
[https://www.mongodb.com/try/download/community?tck=docs\\_server](https://www.mongodb.com/try/download/community?tck=docs_server)



Hình 11: Tải file cài đặt

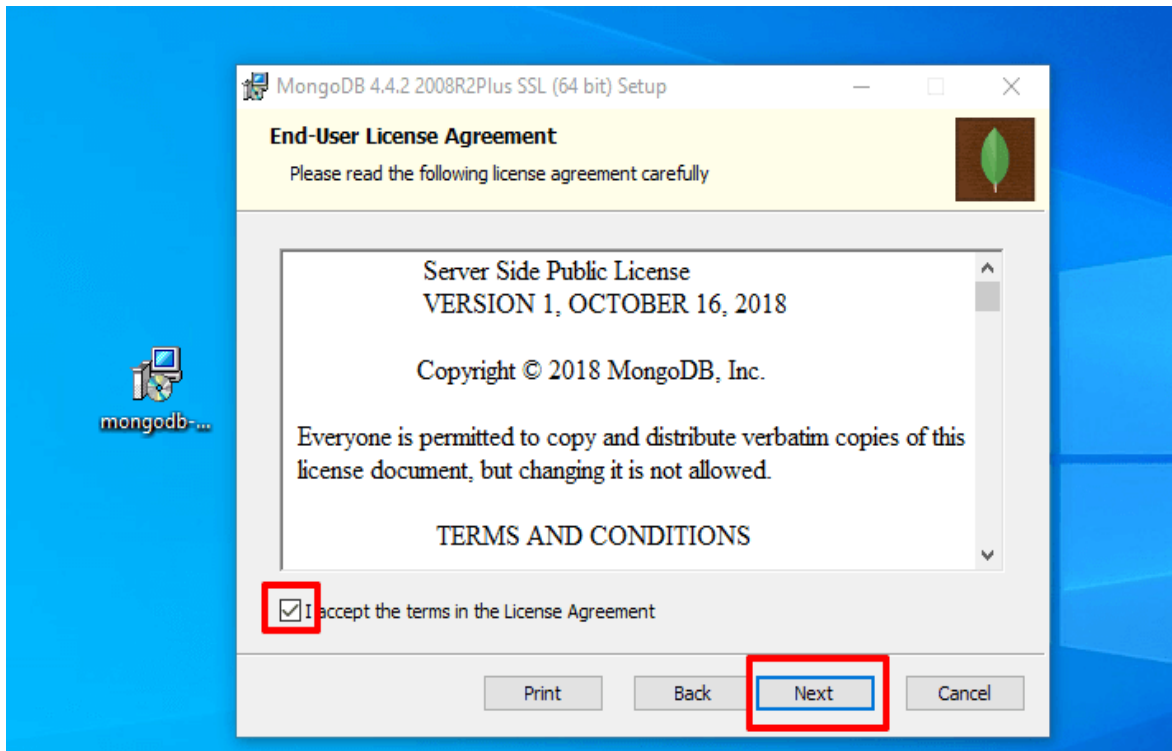
+ Bước 2: Bấm vào biểu tượng mũi tên bên cạnh để chọn các option phù hợp => rồi bấm vào nút Download để tải file cài đặt về.

Sau khi tải file cài đặt về, các bạn nhấp đúp chuột vào file để bắt đầu quá trình cài đặt => và bấm Next để chuyển sang bước tiếp theo.



Hình 12: Bắt đầu quá trình tải

+ Bước 3: Ở bước này các bạn tích vào I accept the terms in the License Agreement => sau đó bấm vào Next để sang bước tiếp theo.

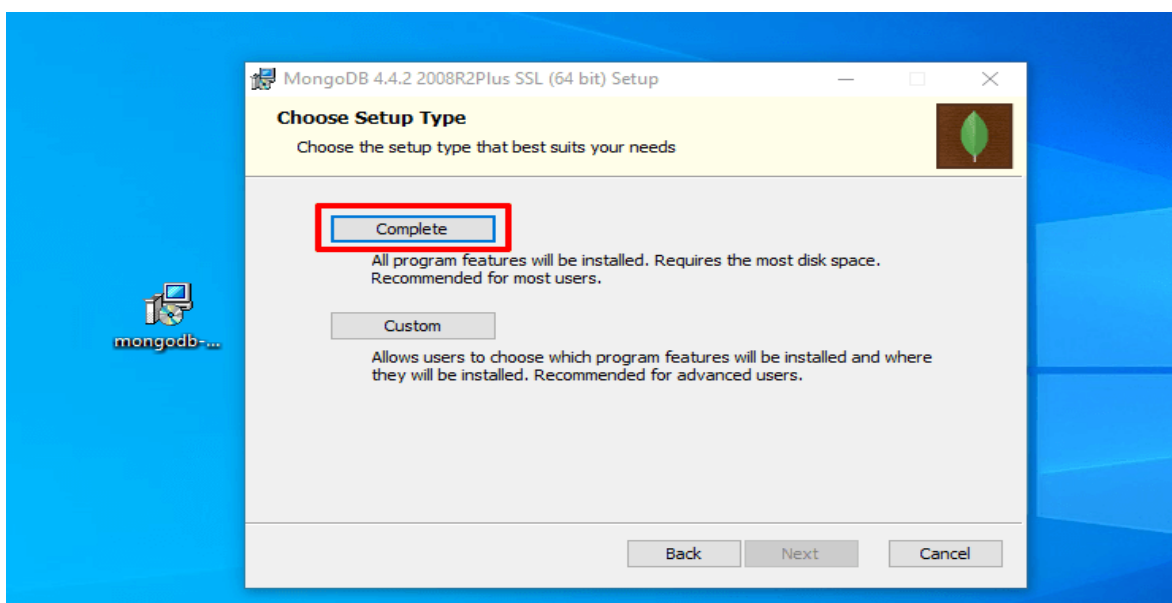


Hình 13: Thỏa thuận cấp phép cho người dùng

+ Bước 4: Nếu bạn chọn Complete thì tất cả các chức năng sẽ được cài đặt và tất nhiên là sẽ chiếm nhiều dung lượng ổ cứng hơn.

Còn nếu bạn chọn Custom thì bạn sẽ chỉ chọn những chức năng bạn muốn cài đặt và tùy thuộc vào chức năng bạn chọn dung lượng ổ cứng cài đặt sẽ giảm đi.

=> Bấm vào Complete để tiếp tục .

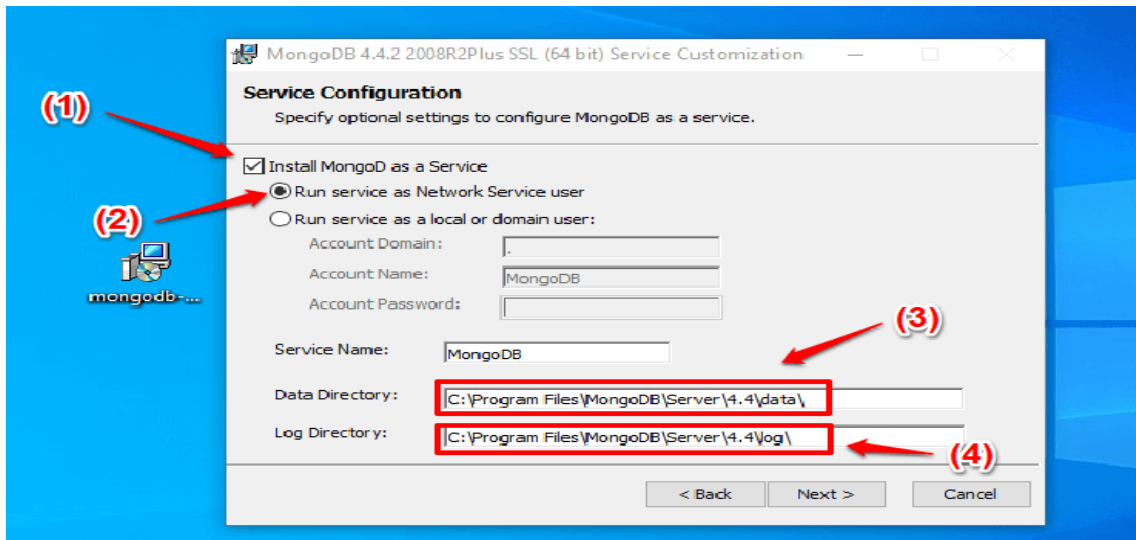


Hình 14: Chọn kiểu thiết lập

+ Bước 5: Tiếp đến bước cấu hình service.

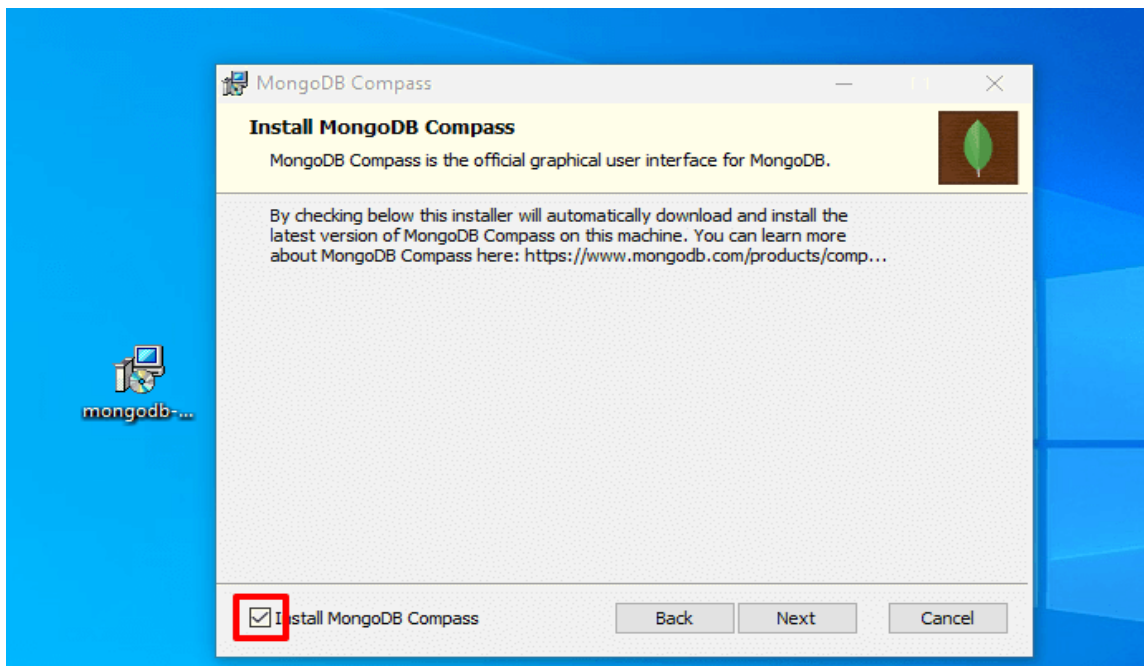
## Tìm hiểu Cơ sở dữ liệu MongoDB và ứng dụng.

- (1) Cài đặt MongoDB như là một service. Nó giống như những service bình thường của hệ điều hành. Có thể bật/tắt được.
  - (2) Chạy service với vai trò là người sử dụng mạng dịch vụ.
  - (3) Nơi lưu trữ Database.
  - (4) Nơi chứa file log
- => Sau đó bấm Next để tiếp tục.



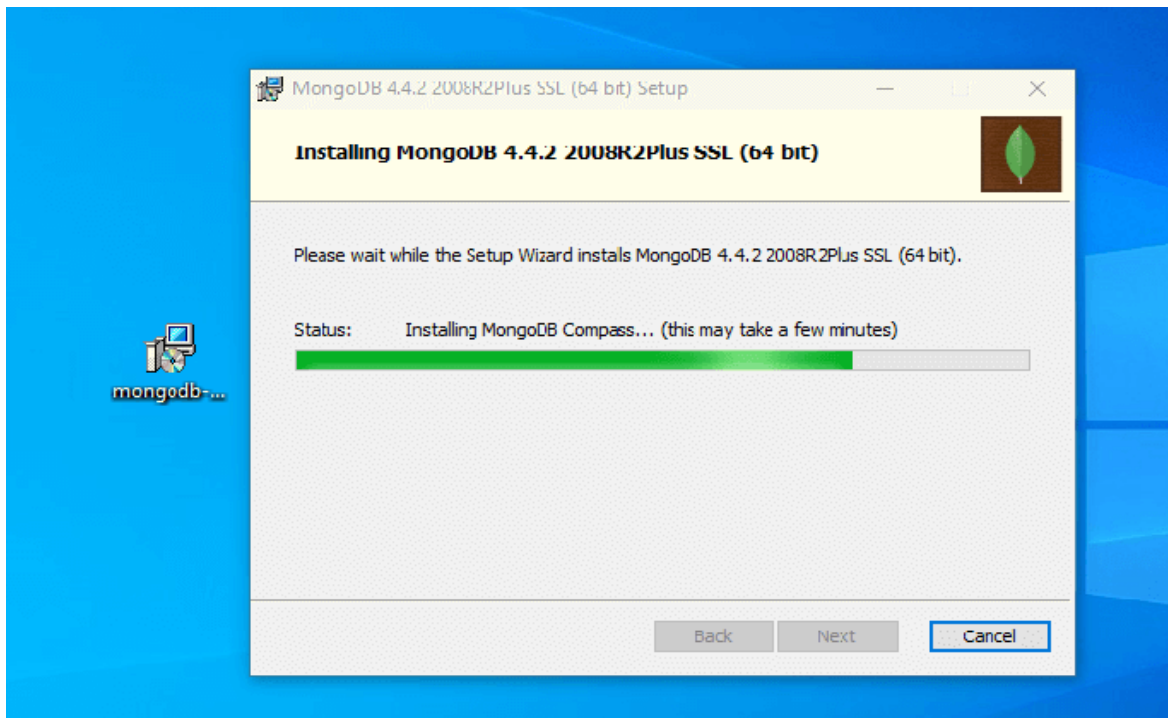
Hình 15: Cấu hình dịch vụ

+ Bước 6: => Tích vào ô Install MongoDB Compass để cài đặt => và bấm Next để đến bước cài đặt.

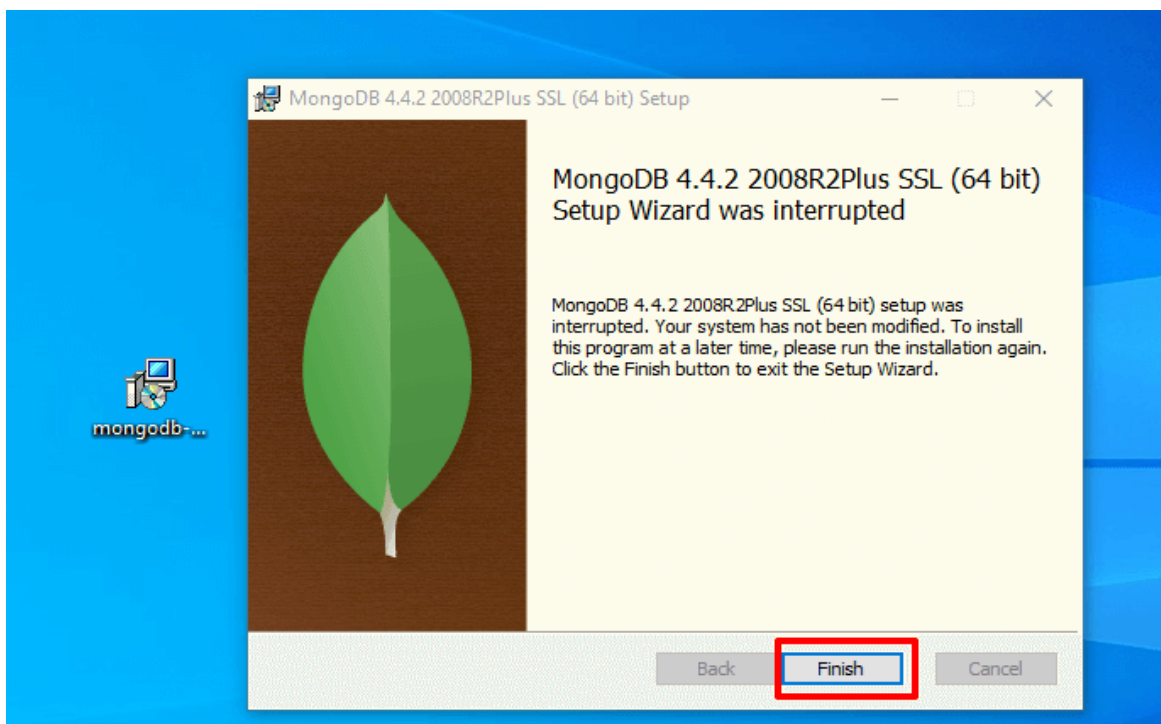


Hình 16: Cài đặt MongoDB

+ Bước 7: Đến bước này thì quá trình cấu hình đã xong và bắt đầu cài đặt. Bấm vào Install để bắt đầu quá trình cài đặt.



Hình 17: Bắt đầu quá trình cài đặt  
+ Bước 8: Sau khi cài đặt xong bấm vào Finish để hoàn tất quá trình cài đặt.



Hình 18: Kết thúc quá trình cài đặt  
Sau khi cài đặt hoàn tất chúng ta tiến hành mở giao diện

Compass sẽ hiện thị màn hình như sau:



The screenshot shows the 'New Connection' dialog in MongoDB Compass. At the top, it says 'New Connection' and 'Connect to a MongoDB deployment'. There is a 'FAVORITE' toggle with a star icon. Below this, there is a 'URI' field with a help icon and a toggle for 'Edit Connection String'. The URI field contains 'mongodb://localhost:27017/'. Below the URI field is a link for 'Advanced Connection Options'. At the bottom, there are three buttons: 'Save', 'Save & Connect', and 'Connect'.

Hình 19: Giao diện khi bắt đầu thao tác trên MongoDB  
Đề lại giá trị mặc định trừ khi đã thay đổi port mà phiên bản MongoDB của người dùng đang kết nối. Sau đó nhấn Connect để tiếp tục thao tác.

The screenshot shows the 'Create Database' dialog in MongoDB Compass. It has a title bar with a close button. The main content area has two input fields: 'Database Name' and 'Collection Name'. There are callouts pointing to these fields: 'Nhấn vào Database Name để tạo tên cơ sở dữ liệu' (Click on Database Name to create the database name) and 'Nhấn vào Collection Name để tạo tên bộ sưu tập' (Click on Collection Name to create the collection name). Below the input fields, there is a checkbox for 'Time-Series' with a description and a link to 'Learn More'. There is also a link for 'Additional preferences'. At the bottom, there is a blue information box stating: 'Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information'. At the very bottom, there are 'Cancel' and 'Create Database' buttons.

Hình 20: Tạo cơ sở dữ liệu  
Các bộ sưu tập trong MongoDB cũng giống như các bảng trong SQL nhưng việc thực hiện thì MongoDB không cần các câu lệnh hay tạo các khóa giống cấu trúc của

SQL.MongoDB hỗ trợ người dùng về việc xác định các cặp khóa để tránh các lỗi truy vấn về sau.



Hình 21: Kết quả sau khi tạo cơ sở dữ liệu

Để tạo dữ liệu chỉ cần chọn ADD DATA sau đó nhấn vào Insert document sẽ xuất hiện giao diện:

```
{
  "_id": {
    "$oid": "65980102ef76ade85e31230a"
  },
  "MaSV": "K6100002",
  "HoTenSV": "Nguyễn Thị Hoài Thu",
  "NgàySinh": "04-12-1991",
  "MaLop": "DA12TT12A1"
}
```

Khi xuất hiện giao diện có một số dòng lệnh hướng dẫn nhập dữ liệu chỉ cần nhập dữ liệu sau đó nhấn Insert là đã tạo được dữ liệu cho bộ sưu tập.

### 3.3 So sánh sự khác nhau của hai ngôn ngữ SQL và NoSQL.

Điểm khác nhau rõ ràng nhất giữa hai ngôn ngữ là ngôn ngữ SQL chủ yếu được gọi là RDBMS còn được gọi là cơ sở dữ liệu quan hệ còn ngôn ngữ NoSQL thuộc về cơ sở dữ liệu phi quan hệ không theo cấu trúc nhất định. Cơ sở dữ liệu quan hệ truyền thống sử dụng cú pháp và truy vấn SQL để phân tích và lấy dữ liệu để có thêm thông tin chi tiết. Về hệ thống cơ sở dữ liệu phi quan hệ tuy được hình thành sau cơ sở dữ liệu quan hệ nhưng nó mang lại nhiều lợi ích giúp người dùng dễ dàng sử dụng và tiếp cận hơn. Hệ thống cơ sở dữ liệu NoSQL bao gồm nhiều loại công



nghe cơ sở dữ liệu khác nhau. Các cơ sở dữ liệu này được phát triển để đáp ứng nhu cầu trình bày cho sự phát triển của ứng dụng hiện đại. Trong mặt cấu trúc thì SQL database là cơ sở dữ liệu dựa trên bảng. Còn NoSQL database có thể dựa trên tài liệu, cặp khóa-giá trị, cơ sở dữ liệu biểu đồ. Cấu trúc lược đồ cho thấy SQL database có lược đồ được xác định trước. NoSQL database sử dụng lược đồ động cho dữ liệu phi cấu trúc. SQL phù hợp cho môi trường truy vấn phức tạp còn NoSQL thì không phù hợp với truy vấn phức tạp. Sự ra đời của SQL được phát triển vào những năm 1970 để giải quyết các vấn đề với lưu trữ tệp phẳng. NoSQL được phát triển vào cuối những năm 2000 để khắc phục các vấn đề và hạn chế của SQL. Khi người dùng cần một ngôn ngữ dễ sử dụng có thể truy suất dữ liệu nhanh, nơi lưu trữ những dữ liệu quan trọng, có mã nguồn mở dễ dàng cho các đồ án, dự án nhóm thì NoSQL là một lựa chọn tốt nhất. Nhanh hơn SQL thì NoSQL sẽ không chuẩn hóa cho phép người dùng lấy tất cả thông tin về một mục có thể với các mã mà không cần tham gia liên quan hoặc truy vấn SQL phức tạp. SQL phù hợp với những dự án có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể được xác định trước. NoSQL phù hợp với những dự án yêu cầu dữ liệu không liên quan, khó xác định.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1. Tạo, sửa, xóa cơ sở dữ liệu SQL

#### 4.1.1. Tạo cơ sở dữ liệu cho bảng SINHVIEN.

Create table SINHVIEN

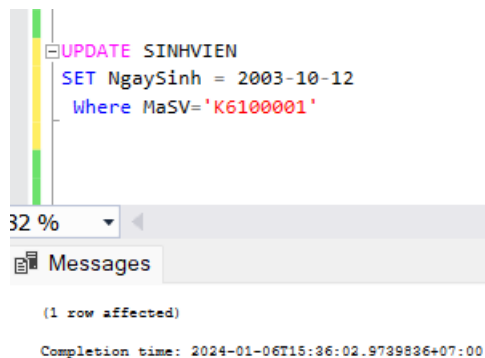
```
(  
MaSV char (8) primary key not null,  
HoTenSV nvarchar (30),  
NgàySinh datetime,  
MaLop char(10) constraint RB_ML foreign key (MaLop) references LOP(MaLop)  
)
```

	MaSV	HoTenSV	NgàySinh	MaLop
1	K6100001	Phạm Văn Bình	1990-02-24 00:00:00.000	DA11TO12A1
2	K6100002	Nguyễn Thị Hoài Thu	1991-04-12 00:00:00.000	DA12TT12A1
3	K6100003	Trần Ngọc Thanh	1990-04-15 00:00:00.000	DA12DL12A1
4	K6100004	Nguyễn Tấn Hùng	1992-02-03 00:00:00.000	DA12TT12A2
5	K6100005	Trương Thành Sang	1990-12-04 00:00:00.000	DA12DL12A2
6	K6100006	Nguyễn Anh Dũng	1982-03-03 00:00:00.000	DA12HH12A1
7	K6100007	Phạm Hồng Ánh	1990-02-24 00:00:00.000	DA11TO12A1
8	K6100008	Lê Thị Liễu	1991-04-12 00:00:00.000	DA12TT12A1
9	K6100009	Phạm Ngọc Sương	1990-04-17 00:00:00.000	DA12DL12A1
10	K6100010	Nguyễn Tấn	1992-12-03 00:00:00.000	DA12TT12A2
11	K6100011	Thạch Thành Sang	1990-02-04 00:00:00.000	DA12DL12A2
12	K6100012	Nguyễn Hồng Anh Ngọc	1992-12-03 00:00:00.000	DA12HH12A1
13	K6100013	Nguyễn Hồng Ngọc	1992-12-13 00:00:00.000	DA12TT12A2
14	K6100014	Kim Thanh Sang	1990-02-04 00:00:00.000	DA12DL12A2
15	K6100015	Hồng Anh Ngọc	1992-12-13 00:00:00.000	DA12HH12A1

Hình 22: Kết quả sau khi thêm dữ liệu

#### 4.1.2. Sửa dữ liệu trong bảng SINHVIEN

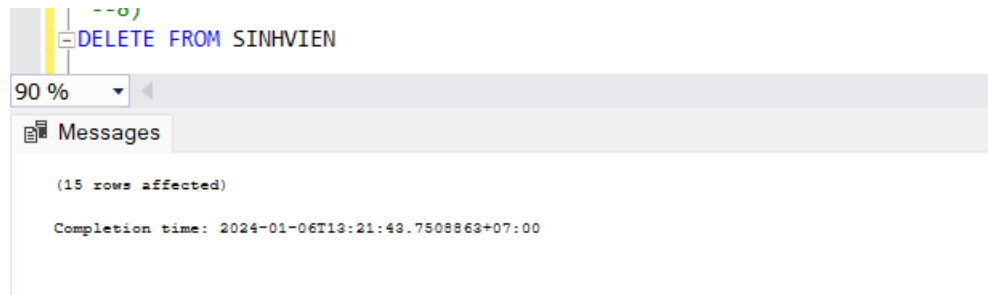
```
UPDATE SINHVIEN  
SET NgàySinh = 2003-10-12  
Where MaSV='K6100001'
```



Hình 23: Kết quả sửa dữ liệu

### 4.1.3. Xóa toàn bộ dữ liệu trong bảng SINHVIEN

```
DELETE FROM SINHVIEN
```



Hình 24: Kết quả khi thực hiện lệnh xóa

### 4.1.4 Các câu lệnh truy vấn trong SQL

Việc sử dụng các câu lệnh truy vấn gây ra các khó khăn nhất định cho người dùng khi chưa nắm rõ được các câu lệnh truy vấn sau đây là vài câu lệnh được truy vấn từ dữ liệu QUANLY\_GIANGDAY.

-Hiển thị danh sách tất cả các sinh viên (gồm: họ tên sinh viên, tên lớp, tên khoa).

```
select HoTenSV,TenLop,TenKhoa
from SINHVIEN,LOP,KHOA
group by HoTenSV,TenLop,TenKhoa
```

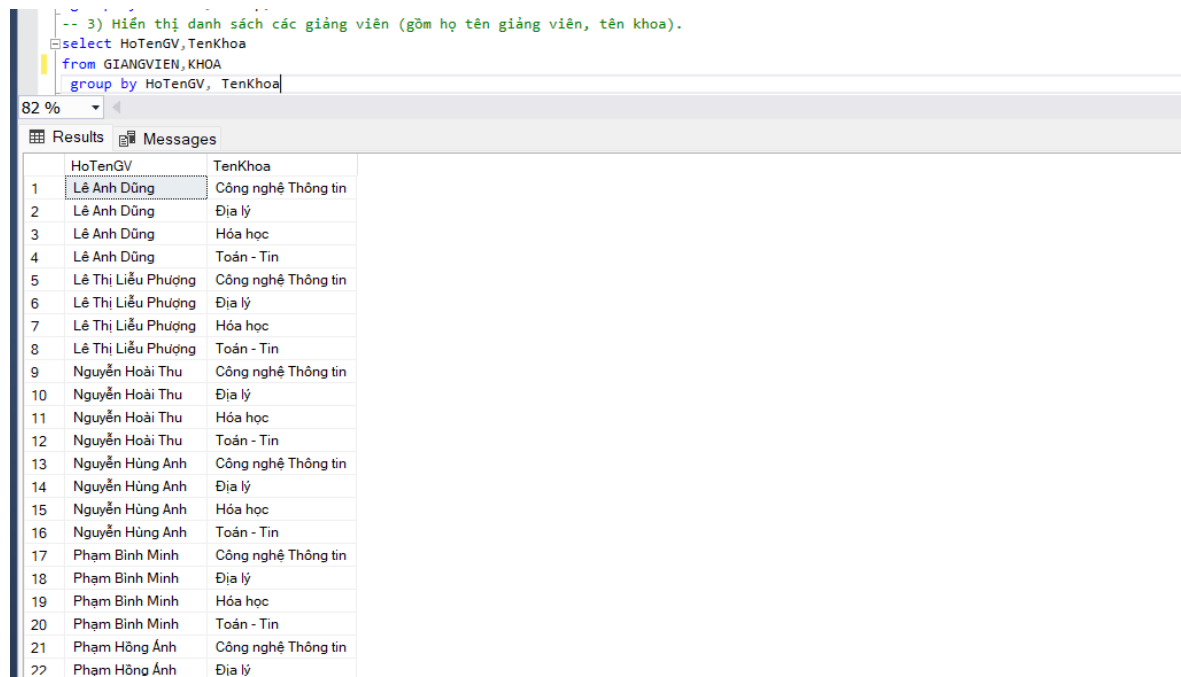


	HoTenSV	TenLop	TenKhoa
1	Hồng Anh Ngọc	ĐH CNTT A1 2012	Công nghệ Thông tin
2	Hồng Anh Ngọc	ĐH CNTT A1 2012	Địa lý
3	Hồng Anh Ngọc	ĐH CNTT A1 2012	Hóa học
4	Hồng Anh Ngọc	ĐH CNTT A1 2012	Toán - Tin
5	Hồng Anh Ngọc	ĐH CNTT A2 2012	Công nghệ Thông tin
6	Hồng Anh Ngọc	ĐH CNTT A2 2012	Địa lý
7	Hồng Anh Ngọc	ĐH CNTT A2 2012	Hóa học
8	Hồng Anh Ngọc	ĐH CNTT A2 2012	Toán - Tin
9	Hồng Anh Ngọc	ĐH Địa lý A1 2012	Công nghệ Thông tin
10	Hồng Anh Ngọc	ĐH Địa lý A1 2012	Địa lý
11	Hồng Anh Ngọc	ĐH Địa lý A1 2012	Hóa học
12	Hồng Anh Ngọc	ĐH Địa lý A1 2012	Toán - Tin
13	Hồng Anh Ngọc	ĐH Địa lý A2 2012	Công nghệ Thông tin
14	Hồng Anh Ngọc	ĐH Địa lý A2 2012	Địa lý
15	Hồng Anh Ngọc	ĐH Địa lý A2 2012	Hóa học
16	Hồng Anh Ngọc	ĐH Địa lý A2 2012	Toán - Tin
17	Hồng Anh Ngọc	ĐH Hóa học A1 2012	Công nghệ Thông tin
18	Hồng Anh Ngọc	ĐH Hóa học A1 2012	Địa lý
19	Hồng Anh Ngọc	ĐH Hóa học A1 2012	Hóa học
20	Hồng Anh Ngọc	ĐH Hóa học A1 2012	Toán - Tin
21	Hồng Anh Ngọc	ĐH Toán A1 2012	Công nghệ Thông tin
22	Hồng Anh Ngọc	ĐH Toán A1 2012	Địa lý
23	Hồng Anh Ngọc	ĐH Toán A1 2012	Hóa học
24	Hồng Anh Ngọc	ĐH Toán A1 2012	Toán - Tin
25	Kim Thanh Sa...	ĐH CNTT A1 2012	Công nghệ Thông tin

Hình 25: Kết quả truy vấn hiển thị danh sách tất cả các sinh viên

-Hiển thị danh sách các giảng viên (gồm họ tên giảng viên, tên khoa).

```
select HoTenGV,TenKhoa
from GIANGVIEN,KHOA
group by HoTenGV, TenKhoa
```



The screenshot shows a MongoDB query interface with the following SQL query:

```
-- 3) Hiển thị danh sách các giảng viên (gồm họ tên giảng viên, tên khoa).
select HoTenGV,TenKhoa
from GIANGVIEN,KHOA
group by HoTenGV, TenKhoa
```

The results are displayed in a table with two columns: HoTenGV and TenKhoa. The table contains 22 rows of data, showing lecturers grouped by their names and departments.

	HoTenGV	TenKhoa
1	Lê Anh Dũng	Công nghệ Thông tin
2	Lê Anh Dũng	Địa lý
3	Lê Anh Dũng	Hóa học
4	Lê Anh Dũng	Toán - Tin
5	Lê Thị Liễu Phượng	Công nghệ Thông tin
6	Lê Thị Liễu Phượng	Địa lý
7	Lê Thị Liễu Phượng	Hóa học
8	Lê Thị Liễu Phượng	Toán - Tin
9	Nguyễn Hoài Thu	Công nghệ Thông tin
10	Nguyễn Hoài Thu	Địa lý
11	Nguyễn Hoài Thu	Hóa học
12	Nguyễn Hoài Thu	Toán - Tin
13	Nguyễn Hùng Anh	Công nghệ Thông tin
14	Nguyễn Hùng Anh	Địa lý
15	Nguyễn Hùng Anh	Hóa học
16	Nguyễn Hùng Anh	Toán - Tin
17	Phạm Bình Minh	Công nghệ Thông tin
18	Phạm Bình Minh	Địa lý
19	Phạm Bình Minh	Hóa học
20	Phạm Bình Minh	Toán - Tin
21	Phạm Hồng Ánh	Công nghệ Thông tin
22	Phạm Hồng Ánh	Địa lý

Hình 26: Kết quả truy vấn hiển thị danh sách các giảng viên

-Hiển thị danh sách các giảng viên giảng dạy các môn học (Tên giảng viên, tên khoa, tên môn học).

```
select HoTenGV , TenKhoa, TenMon
from GIANGVIEN,KHOA,MONHOC
```

## Tìm hiểu Cơ sở dữ liệu MongoDB và ứng dụng.

```
-- 4) Hiển thị danh sách các giảng viên giảng dạy các môn học (Tên giảng viên, tên khoa, tên môn học).
select HoTenGV , TenKhoa, TenMon
from GIANGVIEN, KHOA, MONHOC
```

	HoTenGV	TenKhoa	TenMon
10	Võ Anh Tài	Công nghệ Thông tin	Đại số tuyến tính
11	Phạm Bình Minh	Công nghệ Thông tin	Giải tích 1
12	Nguyễn Hoài Thu	Công nghệ Thông tin	Giải tích 1
13	Trần Ngọc Ân	Công nghệ Thông tin	Giải tích 1
14	Nguyễn Hùng Anh	Công nghệ Thông tin	Giải tích 1
15	Phạm Ngọc Diệp	Công nghệ Thông tin	Giải tích 1
16	Lê Anh Dũng	Công nghệ Thông tin	Giải tích 1
17	Phạm Hồng Anh	Công nghệ Thông tin	Giải tích 1
18	Lê Thị Liễu Phương	Công nghệ Thông tin	Giải tích 1
19	Phạm Ngọc Thu Sương	Công nghệ Thông tin	Giải tích 1
20	Võ Anh Tài	Công nghệ Thông tin	Giải tích 1
21	Phạm Bình Minh	Công nghệ Thông tin	Hình học AFIN
22	Nguyễn Hoài Thu	Công nghệ Thông tin	Hình học AFIN
23	Trần Ngọc Ân	Công nghệ Thông tin	Hình học AFIN
24	Nguyễn Hùng Anh	Công nghệ Thông tin	Hình học AFIN
25	Phạm Ngọc Diệp	Công nghệ Thông tin	Hình học AFIN
26	Lê Anh Dũng	Công nghệ Thông tin	Hình học AFIN
27	Phạm Hồng Anh	Công nghệ Thông tin	Hình học AFIN
28	Lê Thị Liễu Phương	Công nghệ Thông tin	Hình học AFIN
29	Phạm Ngọc Thu Sương	Công nghệ Thông tin	Hình học AFIN
30	Võ Anh Tài	Công nghệ Thông tin	Hình học AFIN
31	Phạm Bình Minh	Công nghệ Thông tin	Kỹ thuật lập trình
32	Nguyễn Hoài Thu	Công nghệ Thông tin	Kỹ thuật lập trình
33	Trần Ngọc Ân	Công nghệ Thông tin	Kỹ thuật lập trình
34	Nguyễn Hùng Anh	Công nghệ Thông tin	Kỹ thuật lập trình
35	Phạm Ngọc Diệp	Công nghệ Thông tin	Kỹ thuật lập trình
36	Lê Anh Dũng	Công nghệ Thông tin	Kỹ thuật lập trình
37	Phạm Hồng Anh	Công nghệ Thông tin	Kỹ thuật lập trình
38	Lê Thị Liễu Phương	Công nghệ Thông tin	Kỹ thuật lập trình
39	Phạm Ngọc Thu Sương	Công nghệ Thông tin	Kỹ thuật lập trình
40	Võ Anh Tài	Công nghệ Thông tin	Kỹ thuật lập trình

Hình 27: Kết quả truy vấn Hiển thị danh sách các giảng viên giảng dạy các môn học

- Hiển thị danh sách các giảng viên không dạy môn học nào trong năm học 2011-2012.

```
select a.MaGV, HoTenGV, NamHoc
from GIANGVIEN a, GIANGDAY
where NamHoc like '2011-2012'
except
select a.MaGV, HoTenGV, NamHoc
from GIANGVIEN a, GIANGDAY b , MONHOC c
where a.MaGV = b.MaGV and b.MaMon = c.MaMon
```

```
-- 5) Hiển thị danh sách các giảng viên không dạy môn học nào trong năm học 2011-2012.
select a.MaGV, HoTenGV, NamHoc
from GIANGVIEN a, GIANGDAY
where NamHoc like '2011-2012'
except
select a.MaGV, HoTenGV, NamHoc
from GIANGVIEN a, GIANGDAY b , MONHOC c
where a.MaGV = b.MaGV and b.MaMon = c.MaMon
```

	MaGV	HoTenGV	NamHoc
1	15.111.02	Nguyễn Hoài Thu	2011-2012
2	15.111.03	Trần Ngọc Ân	2011-2012
3	15.111.04	Nguyễn Hùng Anh	2011-2012
4	15.111.05	Phạm Ngọc Diệp	2011-2012
5	15.111.06	Lê Anh Dũng	2011-2012
6	15.111.07	Phạm Hồng Anh	2011-2012
7	15.111.08	Lê Thị Liễu Phương	2011-2012
8	15.111.10	Võ Anh Tài	2011-2012

Hình 28: Kết quả truy vấn hiển thị danh sách các giảng viên không dạy môn học nào trong năm học 2011-2012.

- Hiển thị mã, tên các môn học được sinh viên Nguyễn Thị Hoài Thu đăng ký học.

```
select HoTenSV, MaMon , TenMon
from SINHVIEN, MONHOC
```

```
where HoTenSV like N'Nguyễn Thị Hoài Thu'
```

```
-- 6) Hiển thị mã, tên các môn học được sinh viên Nguyễn Thị Hoài Thu đăng ký học.
select HoTenSV, MaMon ,TenMon
from SINHVIEN, MONHOC
where HoTenSV like N'Nguyễn Thị Hoài Thu'
```

82 %

Results Messages

	HoTenSV	MaMon	TenMon
1	Nguyễn Thị Hoài Thu	DSTT	Đại số tuyến tính
2	Nguyễn Thị Hoài Thu	GTA1	Giải tích 1
3	Nguyễn Thị Hoài Thu	HHAf	Hình học Afn
4	Nguyễn Thị Hoài Thu	KTLT	Kỹ thuật lập trình
5	Nguyễn Thị Hoài Thu	THDC	Tin học đại cương
6	Nguyễn Thị Hoài Thu	XSTK	Xác suất thống kê

Hình 29: Kết quả truy vấn Hiển thị mã, tên các môn học được sinh viên Nguyễn Thị Hoài Thu đăng ký học.

-Đếm số môn học mà sinh viên Trần Ngọc Thanh đã đăng ký học.

```
select HoTenSV , COUNT(MaMon) as SoMaMon_SVDKH
from SINHVIEN , MONHOC
group by HoTenSV
having HoTenSV like N'Trần Ngọc Thanh'
order by SoMaMon_SVDKH
```

```
-- 7) Đếm số môn học mà sinh viên Trần Ngọc Thanh đã đăng ký học.
select HoTenSV , COUNT(MaMon) as SoMaMon_SVDKH
from SINHVIEN , MONHOC
group by HoTenSV
having HoTenSV like N'Trần Ngọc Thanh'
order by SoMaMon_SVDKH
```

82 %

Results Messages

	HoTenSV	SoMaMon_SVDKH
1	Trần Ngọc Thanh	6

Hình 30: Kết quả truy vấn đếm số môn học mà sinh viên Trần Ngọc Thanh đã đăng ký học.

## 4.2. Tạo, sửa ,xóa collection MongoDB cơ bản

### 4.2.1. Tạo collection trong MongoDB

- Cú pháp cơ bản của phương thức createCollection() không có tham số options như sau:

```
db.createCollection(name)
```

Ví dụ:

```
db.createCollection("SINHVIEN")
```

- Sử dụng phương thức createCollection() với một số option

```
db.createCollection(name, options)
```

Ví dụ:

```
db.createCollection("SINHVIEN",  
  MaSV: 'K6100001',  
  HoTenSV: 'Phạm Văn Bình',  
  NgaySinh: '02-24-1990',  
  MaLop: 'DA11TO12A1')
```

Và ngoài ra MongoDB có thể tự động tạo collection khi người dùng insert một document

### 4.2.2. Xóa collection trong MongoDB

```
db.COLLECTION_NAME.drop()
```

Ví dụ

```
db.SINHVIEN.drop()
```

Sau khi thực thi phương thức drop() sẽ trả về true collection đã chọn bị xóa thành công, nếu không thành công nó sẽ trả về false.

### 4.2.3. Thêm dữ liệu trong MongoDB

Trong MongoDB dữ liệu được thêm vào bảng thông qua lệnh `insertOne` (thêm một bản ghi) hoặc `insertMany` (thêm nhiều bản ghi)

- Thêm một bản ghi vào collection:

```
db. COLLECTION_NAME.insertOne ({  
  field1: value1,  
  field2:value 2,  
  //...  
});
```

Ví dụ:

```
db.SINHVIEN.insertOne({  
  "MaSV": "K6100001",  
  "HoTenSV": "Phạm Văn Bình",  
  "NgàySinh": "02-24-1990",  
  "MaLop": "DA11TO12A1"  
});
```

- Thêm nhiều bản ghi vào collection:

```
db. COLLECTION_NAME insertMany ([ {  
  field1: value1,  
  field2:value 2,  
  //...  
},  
{ field1: value1,  
  field2:value 2,  
  //...  
}  
]);
```



Ví dụ:

```
db.SINHVIEN.insertMany([ { "MaSV": "K6100001",
    "HoTenSV": "Phạm Văn Bình",
    "NgàySinh": "02-24-1990",
    "MaLop": "DA11TO12A1"
},
{ "MaSV": "K6100002",
    "HoTenSV": "Nguyễn Thị Hoài Thu",
    "NgàySinh": "04-12-1991",
    "MaLop": "DA12TT12A1"},
{ "MaSV": "K6100003",
    "HoTenSV": "Trần Ngọc Thanh",
    "NgàySinh": "04-15-1990",
    "MaLop": "DA12DL12A1"}
])
```

#### 4.2.4. Xóa dữ liệu trong MongoDB

Có thể xóa dữ liệu bằng hai phương thức sau: `deleteOne()` và `deleteMany()`.

- Phương thức `deleteOne()` sẽ xóa tài liệu đầu tiên phù hợp với truy vấn được cung cấp.

```
db.COLLECTION_NAME.deleteOne({options})
```

Ví dụ:

```
db.SINHVIEN.deleteOne ({ "MaSV": "K6100002" })
```

- Phương thức `deleteMany()` sẽ xóa tất cả tài liệu phù hợp với truy vấn được cung cấp.

```
db.COLLECTION_NAME.deleteMany({options})
```

Ví dụ:

```
db.SINHVIEN.deleteMany ({ "MaLop": "DA12DL12A1" })
```

#### 4.2.5 Sửa dữ liệu trong MongoDB

Để cập nhật một tài liệu hiện có người dùng có thể sử dụng các phương thức `updateOne()` hoặc `updateMany` và có thể sửa hoặc thêm document nếu không tìm thấy.

- Phương thức `updateOne()` sẽ cập nhật tài liệu đầu tiên được tìm thấy phù hợp với truy vấn được cung cấp.

```
db. COLLECTION_NAME.updateOne( { field1: value1 }, { $set: { field2:value 2} } )
```

Ví dụ :

```
db.SINHVIEN.updateOne( { "HoTenSV": "Phạm Văn Bình"}, { $set: { "HoTenSV": "Nguyễn Thị Hoài Thu" } } )
```

- Phương thức `updateMany()` sẽ cập nhật tất cả các tài liệu phù hợp với truy vấn được cung cấp.

```
db. COLLECTION_NAME.updateMany ( { }, { $inc: { field1:value 1 } } )
```

Ví dụ

```
db.SINHVIEN.updateMany ( { }, { $inc: { "HoTenSV": "Trần Ngọc Thanh" } } )
```

Trong phương thức này điều kiện tìm kiếm là `{ }` không có điều kiện cụ thể nào được xác định nên tất cả các bản ghi trong bảng sẽ được cập nhật.

- Sửa một hoặc thêm document nếu không tìm thấy.

```
db. COLLECTION_NAME.updateOne(
  { field1: value1 },
  {
    $set:
    {
      field1: value1    }
  }
```

```
},  
{ upsert: true }  
)
```

Ví dụ

```
db.SINHVIEN.updateOne(  
  { "MaSV": "K6100001"},  
  {  
    $set:  
    {  
      "MaSV": "K6100001",  
      "HoTenSV": "Phạm Văn Bình",  
      "NgàySinh": "02-24-1990",  
      "MaLop": "DA11TO12A1"  
    }  
  },  
  { upsert: true }  
)
```

#### 4.2.6 LƯU TRỮ, XUẤT FILE TRONG MONGODB ❖ IMPORT DATA

➤ Import từ tệp JSON vào collection:

```
mongoimport --db ten_database --collection ten_collection --file  
duong_dan/tên_tệp_tin.json
```

Ví dụ:

```
mongoimport --db QUANLY_GIANGDAY --collection SINHVIEN --file  
/path/to/QUANLY_GIANGDAY.json
```

➤ Import từ tệp CSV vào collection:

```
mongoimport --db ten_database --collection ten_collection --type csv --headerline --  
file duong_dan/tên_tệp_tin.csv
```

Ví dụ:

```
mongoimport --db QUANLY_GIANGDAY --collection SINHVIEN --type csv --  
headerline --file /path/to/ QUANLY_GIANGDAY.csv
```

--type csv: Chỉ định định dạng của tệp là CSV.

--headerline: Cho biết tệp tin có hàng đầu tiên là dòng chứa tên

### ❖ EXPORT DATA

#### ➤ Export toàn bộ collection vào tệp tin JSON:

```
mongoexport --db ten_database --collection ten_collection --out  
duong_dan/tên_tệp_tin.json
```

**Ví dụ:**

```
mongoexport --db QUANLY_GIANGDAY --collection SINHVIEN --out  
/path/to/ QUANLY_GIANGDAY.json
```

#### ➤ Export toàn bộ cơ sở dữ liệu vào tệp tin JSON:

```
mongoexport --db ten_database --out duong_dan/tên_tệp_tin.json
```

**Ví dụ:**

```
mongoexport --db QUANLY_GIANGDAY --out /path/to/  
QUANLY_GIANGDAY.json
```

#### ➤ Export từ collection với query vào tệp tin JSON:

```
mongoexport --db ten_database --collection ten_collection --query '{ "field":  
"value" }' --out duong_dan/tên_tệp_tin.json
```

**Ví dụ:**

```
mongoexport --db QUANLY_GIANGDAY --collection SINHVIEN --query '{  
"HoTenSV": "Phạm Văn Bình" }' --out /path/to/ QUANLY_GIANGDAY.json
```

#### ➤ Export từ collection với trường chọn lọc (projection) vào tệp tin JSON:

```
mongoexport --db ten_database --collection ten_collection --fields field1,field2  
--out duong_dan/tên_tệp_tin.json
```

**Ví dụ:**

```
mongoexport --db QUANLY_GIANGDAY --collection SINHVIEN --fields  
MaSV,HoTenSV,NgaySinh,MaLop --out /path/to/ QUANLY_GIANGDAY.json
```

➤ Export từ collection vào tệp tin CSV:

```
mongoexport --db ten_database --collection ten_collection --type csv --fields field1,field2 --out duong_dan/tên_tệp_tin.csv
```

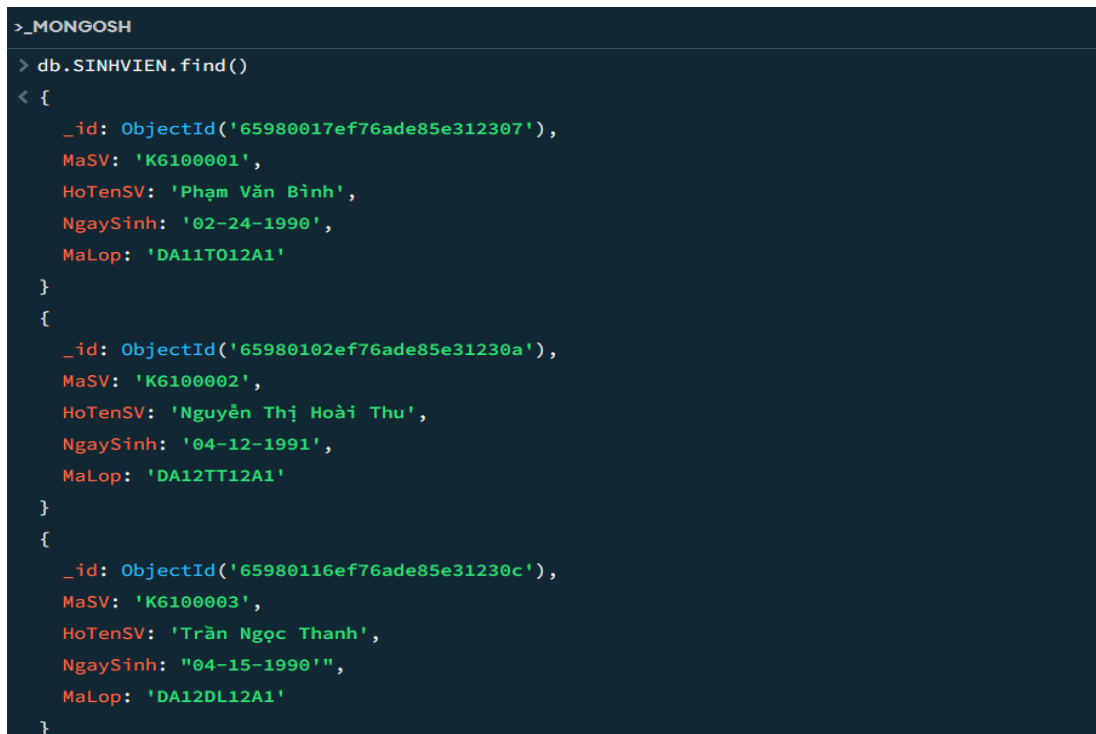
**Ví dụ:**

```
mongoexport --db QUANLY_GIANGDAY --collection SINHVIEN --type csv -  
-fields MaSV,HoTenSV,NgaySinh,MaLop --out /path/to/  
QUANLY_GIANGDAY.csv
```

### 4.3. TRUY VẤN DỮ LIỆU TRONG MONGODB

#### 1. Truy vấn toàn bộ dữ liệu trong một collection:

```
db.SINHVIEN.find()
```



```
>_MONGOSH  
> db.SINHVIEN.find()  
< {  
  _id: ObjectId('65980017ef76ade85e312307'),  
  MaSV: 'K6100001',  
  HoTenSV: 'Phạm Văn Bình',  
  NgaySinh: '02-24-1990',  
  MaLop: 'DA11T012A1'  
}  
{  
  _id: ObjectId('65980102ef76ade85e31230a'),  
  MaSV: 'K6100002',  
  HoTenSV: 'Nguyễn Thị Hoài Thu',  
  NgaySinh: '04-12-1991',  
  MaLop: 'DA12TT12A1'  
}  
{  
  _id: ObjectId('65980116ef76ade85e31230c'),  
  MaSV: 'K6100003',  
  HoTenSV: 'Trần Ngọc Thanh',  
  NgaySinh: '04-15-1990',  
  MaLop: 'DA12DL12A1'  
}
```

Hình 31: Truy vấn toàn bộ dữ liệu trong một collection

## 2. Truy vấn dữ liệu với điều kiện:

- Tìm người có tên Phạm Văn Bình

```
db.SINHVIEN.find({ HoTenSV: 'Phạm Văn Bình'})
```

```
> db.SINHVIEN.find({ HoTenSV: 'Phạm Văn Bình'})
< {
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: '02-24-1990',
  MaLop: 'DA11T012A1'
}
```

Hình 32: Truy vấn dữ liệu với điều kiện 1

- Tìm người có ngày sinh: 02/24/1990

```
db.SINHVIEN.find({ NgaySinh: '02-24-1990'})
```

```
> db.SINHVIEN.find({ NgaySinh: '02-24-1990'})
< {
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: '02-24-1990',
  MaLop: 'DA11T012A1'
}
```

Hình 33: Truy vấn dữ liệu với điều kiện 2

### 3. Truy vấn sử dụng điều kiện logic:

Tìm người có ngày sinh từ 02-24-1990 đến 04-12-1991

```
db.SINHVIEN.find({
  NgaySinh: {
    $gte: new ISODate("1990-02-24"),
    $lte: new ISODate("1991-04-12")
  }
})
```

```
> db.SINHVIEN.find({
  NgaySinh: {
    $gte: new ISODate("1990-02-24"),
    $lte: new ISODate("1991-04-12")
  }
})
< {
  _id: ObjectId('659cd5cbbe22f2cef48c7d2a'),
  MaSV: 'K6100002',
  HoTenSV: 'Nguyễn Thị Hoài Thu',
  NgaySinh: 1991-04-11T17:00:00.000Z,
  MaLop: 'DA12TT12A1'
}
{
  _id: ObjectId('659cd5cbbe22f2cef48c7d2b'),
  MaSV: 'K6100003',
  HoTenSV: 'Trần Ngọc Thanh',
  NgaySinh: 1990-04-14T17:00:00.000Z,
  MaLop: 'DA12DL12A1'
}
```

Hình 34: Truy vấn sử dụng điều kiện logic

#### 4. Truy vấn sắp xếp kết quả:

- Sắp xếp tăng dần theo ngày sinh

```
db.SINHVIEN.find().sort({ NgaySinh: 1 })
```

```
>_MONGOSH
> db.SINHVIEN.find().sort({ NgaySinh:1 })
< {
  _id: ObjectId('659cd787be22f2cef48c7d2e'),
  MaSV: 'K6100006',
  HoTenSV: 'Nguyễn Anh Dũng',
  NgaySinh: 1982-03-02T17:00:00.000Z,
  MaLop: 'DA12HH12A1'
}
{
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
{
  _id: ObjectId('659cd787be22f2cef48c7d2f'),
  MaSV: 'K6100007',
  HoTenSV: 'Phạm Hồng Ánh',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
```

Hình 35: Truy vấn sắp xếp kết quả tăng dần

- Sắp xếp giảm dần theo ngày sinh

```
db.SINHVIEN.find().sort({ NgaySinh:- 1 })
```

```
>_MONGOSH
}
{
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
{
  _id: ObjectId('659cd787be22f2cef48c7d2f'),
  MaSV: 'K6100007',
  HoTenSV: 'Phạm Hồng Ánh',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
{
  _id: ObjectId('659cd787be22f2cef48c7d2e'),
  MaSV: 'K6100006',
  HoTenSV: 'Nguyễn Anh Dũng',
  NgaySinh: 1982-03-02T17:00:00.000Z,
  MaLop: 'DA12HH12A1'
}
```

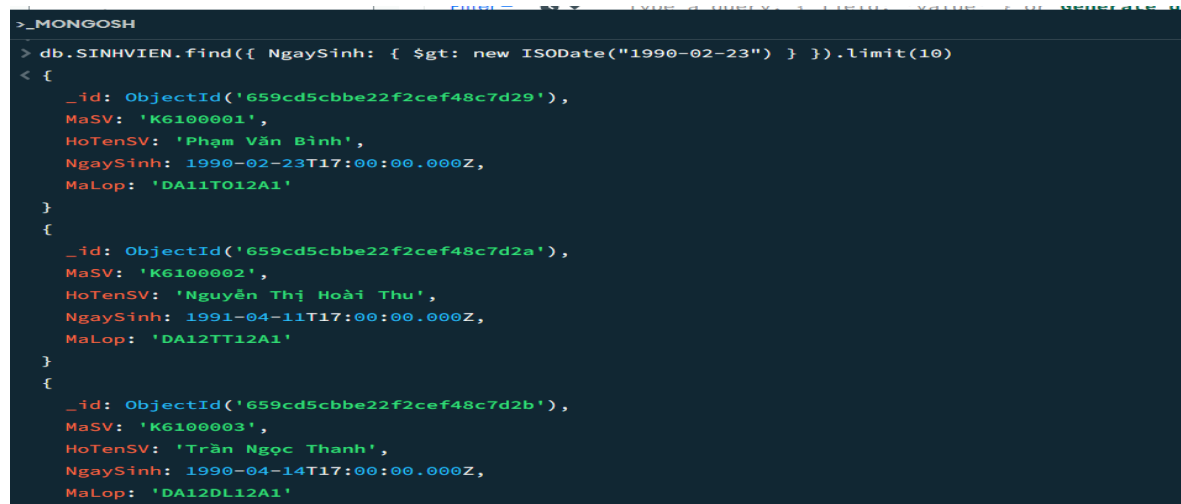
Hình 36: Truy vấn sắp xếp kết quả giảm dần



## 5. Truy vấn giới hạn số lượng kết quả:

Lấy ra các bản ghi từ bảng SINHVIEN có trường NgaySinh lớn hơn ngày 23 tháng 2 năm 1990 và giới hạn kết quả trả về 10 bản ghi.

```
db.SINHVIEN.find({ NgaySinh: { $gt: new ISODate("1990-02-23") } }).limit(10)
```



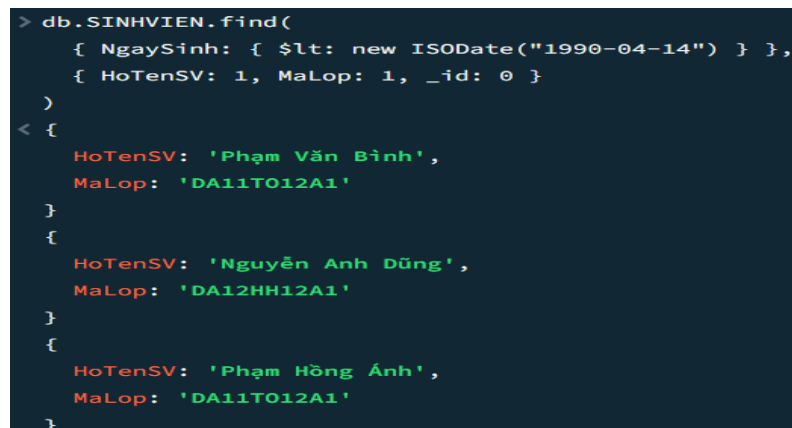
```
>_MONGOSH
> db.SINHVIEN.find({ NgaySinh: { $gt: new ISODate("1990-02-23") } }).limit(10)
< {
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11TO12A1'
}
{
  _id: ObjectId('659cd5cbbe22f2cef48c7d2a'),
  MaSV: 'K6100002',
  HoTenSV: 'Nguyễn Thị Hoài Thu',
  NgaySinh: 1991-04-11T17:00:00.000Z,
  MaLop: 'DA12TT12A1'
}
{
  _id: ObjectId('659cd5cbbe22f2cef48c7d2b'),
  MaSV: 'K6100003',
  HoTenSV: 'Trần Ngọc Thanh',
  NgaySinh: 1990-04-14T17:00:00.000Z,
  MaLop: 'DA12DL12A1'
}
```

Hình 37: Truy vấn giới hạn số lượng kết quả

## 6. Truy vấn sử dụng projection (chọn lọc các trường):

Lấy ra các bản ghi từ bảng SINHVIEN có trường NgaySinh nhỏ hơn ngày 14 tháng 4 năm 1990, và chỉ hiển thị trường HoTenSV và MaLop, loại bỏ trường \_id.

```
db.SINHVIEN.find(
  { NgaySinh: { $lt: new ISODate("1990-04-14") } },
  { HoTenSV: 1, MaLop: 1, _id: 0 }
)
```




```
> db.SINHVIEN.find(
  { NgaySinh: { $lt: new ISODate("1990-04-14") } },
  { HoTenSV: 1, MaLop: 1, _id: 0 }
)
< {
  HoTenSV: 'Phạm Văn Bình',
  MaLop: 'DA11TO12A1'
}
{
  HoTenSV: 'Nguyễn Anh Dũng',
  MaLop: 'DA12HH12A1'
}
{
  HoTenSV: 'Phạm Hồng Ánh',
  MaLop: 'DA11TO12A1'
}
```

Hình 38: Truy vấn sử dụng projection (chọn lọc các trường)

### 7. Truy vấn sử dụng regex:

Trả về tất cả các bản ghi trong bảng SINHVIEN mà tên sinh viên (HoTenSV) chứa chữ "B" bất kỳ ở bất cứ vị trí nào:

```
db.SINHVIEN.find({ HoTenSV: /B/ })
```



```
> db.SINHVIEN.find({ HoTenSV: /B/ })
< {
  _id: ObjectId('659cd5cbbe22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11TO12A1'
}
```

Hình 39: Truy vấn sử dụng regex

### 8. Truy vấn sử dụng logical AND và OR:

Lấy ra tất cả các bản ghi từ bảng (collection) SINHVIEN trong đoạn thời gian từ ngày 23 tháng 2 năm 1990 đến ngày 14 tháng 4 năm 1990, và có MaLop là 'DA11TO12A1' hoặc 'DA12TT12A1'.

```
db.SINHVIEN.find({
  $and: [
    { NgaySinh: { $gte: new ISODate("1990-02-23"), $lte: new ISODate("1990-04-14") } },
    { $or: [{ MaLop: 'DA11TO12A1' }, { MaLop: 'DA12TT12A1' }] }
  ]
})
```

```
> db.SINHVIEN.find({
  $and: [
    { NgaySinh: { $gte: new ISODate("1990-02-23"), $lte: new ISODate("1990-04-14") } },
    { $or: [{ MaLop: 'DA11T012A1' }, { MaLop: 'DA12TT12A1' }] }
  ]
})
< {
  _id: ObjectId('659cd5cbb22f2cef48c7d29'),
  MaSV: 'K6100001',
  HoTenSV: 'Phạm Văn Bình',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
{
  _id: ObjectId('659cd787be22f2cef48c7d2f'),
  MaSV: 'K6100007',
  HoTenSV: 'Phạm Hồng Ánh',
  NgaySinh: 1990-02-23T17:00:00.000Z,
  MaLop: 'DA11T012A1'
}
```

Hình 40: Truy vấn sử dụng logical AND và OR

## **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

Nhìn chung hiện nay hệ quản trị cơ sở dữ liệu NoSQL cơ sở dữ liệu MongoDB đang phát triển rất mạnh mẽ so với hệ quản trị cơ sở dữ liệu SQL. Hệ quản trị cơ sở dữ liệu SQL truyền thống sử dụng cú pháp và câu lệnh truy vấn để phân tích và lấy dữ liệu để có thêm thông tin chi tiết. Ngược lại với SQL thì hệ thống NoSQL bao gồm nhiều loại công nghệ cơ sở dữ liệu khác nhau. Các cơ sở dữ liệu này được phát triển để đáp ứng nhu cầu cho sự phát triển của ứng dụng hiện đại. Hệ quản trị cơ sở dữ liệu SQL hỗ trợ cho việc xây dựng dự án đã có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể xác định được trước. Hệ quản trị cơ sở dữ liệu NoSQL thì phù hợp với những dự án yêu cầu dữ liệu không liên quan, khó xác định, đơn giản mềm dẻo khi đang phát triển. Qua bài báo cáo cho thấy được những lợi thế mạnh mẽ giữa hai hệ quản trị cơ sở dữ liệu. Việc sử dụng hệ quản trị cơ sở dữ liệu NoSQL cơ sở dữ liệu MongoDB giúp người dùng dễ dàng áp dụng cho những dự án cá nhân hoặc trong phạm vi công ty và doanh nghiệp.

Hướng phát triển tiếp theo sẽ tiếp tục nghiên cứu chi tiết rõ ràng về cơ sở dữ liệu MongoDB, nâng cao chất lượng sử dụng MongoDB giúp người dùng dễ dàng sử dụng và nhận ra được nhiều điểm mạnh hơn nữa của cơ sở dữ liệu MongoDB.

### **DANH MỤC TÀI LIỆU THAM KHẢO**

1. [https://www.w3schools.com/mongodb/mongodb\\_mongosh\\_delete.php](https://www.w3schools.com/mongodb/mongodb_mongosh_delete.php)
2. Shannon Bradshaw, Eoin Brazil, Kristina Chodorow - MongoDB\_ The Definitive Guide\_ Powerful and Scalable Data Storage-O\_Reilly Media (2019)
3. 2022\_BT thuc hanh SQL\_DA21TTA\_2022 biên soạn Phan Thị Phương Nam
4. Tai lieu giang day CSDL\_HaThiThuyVi

## PHỤ LỤC