

*Trường Đại học Sư phạm thành phố Hồ Chí Minh *

**BÁO CÁO ĐỒ ÁN**

**Cơ sở dữ liệu nâng cao || Giáo viên : Lương Trần Hy Hiến**

**Nội dung đề tài : *MongoDB***

****

**Danh sách nhóm :**

***Cao Cảnh Linh-42.01.104.078***

***Nguyễn Minh Vương-42.01.104.195***

***Nhan Thái Dương-42.01.107.217***

***Nguyễn Cát Tường-42.01.104.156***

***Huỳnh Phú Thành--42.01.104.161***

**Mục lục**

[CHƯƠNG 1 : TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL 1](#_Toc528688425)

[1.1 Khái niệm NOSQL 1](#_Toc528688426)

[1.2 Ưu nhược điểm của CSDL NOSQL 2](#_Toc528688427)

[1.3 So sánh NOSQL với cơ sở dữ liệu quan hệ 3](#_Toc528688428)

[1.4 Phân Loại NOSQL 4](#_Toc528688429)

[CHƯƠNG 2 : MongoDB 5](#_Toc528688430)

[2.1 Khái niệm MongoDB 5](#_Toc528688431)

[2.2 Ưu nhược điểm của MongoDB 6](#_Toc528688432)

[2.3 Kiến trúc tổng quát 7](#_Toc528688433)

[2.4 Những thuật ngữ đầu tiên trong MongoDB 8](#_Toc528688434)

[2.5 Các kiểu dữ liệu trong MongoDB 9](#_Toc528688435)

[2.6 Một số đặc trưng của MongoDB 10](#_Toc528688436)

[2.7 Cài đặt MongoDB 11](#_Toc528688437)

[TÀI LIỆU THAM KHẢO 14](#_Toc528688438)

# CHƯƠNG 1 : TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL

* 1. **Khái niệm NOSQL**

NoSQL là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ. (RDBMS). RDBMS vốn tồn tại khá nhiều nhược điểm như có hiệu năng không tốt nếu kết nối dữ liệu nhiều bảng lại hay khi dữ liệu trong một bảng là rất lớn.

NoSQL là cơ sở dữ liệu không quan hệ, ràng buộc giữa các Collection (hay còn gọi là bảng trong cơ sở dữ liệu bình thường). Tức là giữa các Collection (Bảng) sẽ không có khóa chính, khóa ngoại như trong cơ sở dữ liệu bình thường. Việc này sẽ giảm bớt rối khi cơ sở dữ liệu chúng ta có nhiều bảng. NoSQL đặc biệt nhấn mạnh đến mô hình lưu trữ cặp giá trị - khóa và hệ thống lưu trữ phân tán.

NoSQL ra đời năm 1998 bởi Carlo Strozzi khi ông lập mới một hệ cơ sở dữ liệu quan hệ mã nguồn mở nhanh và nhẹ không liên quan đến SQL Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: phân tán (distributed) + không ràng buộc (non-relational).

**Đặc điểm**

* NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin – Mô hình phân tán dưới sự kiểm soát phần mềm
* Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau
* Một truy vấn sẽ được gửi tới nhiều máy cùng lúc, do đó khi một máy nào đó không phục vụ được sẽ không ảnh hưởng lắm đến chất lượng trả về kết quả
* Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu
* Tính nhất quán không theo thời gian thực: Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.
  1. **Ưu nhược điểm của CSDL NOSQL**

**Ưu điểm**

1. Là mã nguồn mở: không phải mất chi phí và có xu hướng tin cậy, an ninh và nhanh hơn để triển khai so với các hệ quản trị cơ sở dữ liệu độc quyền.
2. Linh hoạt trong việc mở rộng và phát triển: Về phía công ty quản lý thì giúp dễ dàng mở rộng máy chủ khi dữ liệu càng ngày càng lớn hoặc lượng truy cập, tải dữ liệu quá lớn, thay vì thuê một máy chủ lớn hơn để thế máy chủ trước thì công ty chỉ cần thuê thêm một máy chủ khác. Về phía người lập trình thì giúp dễ dàng thêm Collection (Bảng) hoặc cặp dữ liệu (Cột). Dễ dàng trong việc thống kê, truy vấn nhanh.
3. Áp dụng được công nghệ điện toán đám mây: dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu của ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các các kết nối mạnh. NoSQL lưu trữ dữ liệu thường được mở rộng phạm vi theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn. iv. Được các hãng lớn sử dụng như Amazon, BBC, Facebook và Google. Làm nền tảng, cơ sở cho các công ty lớn áp dụng cũng như có sự tin cậy cao về hệ quản trị NoSQL.

**Nhược điểm**

1. Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhất từ nhà cung cấp RMBMS (vì được phát triển trước NoSQL một thời gian khá dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được sự hỗ trợ tốt nhất.
2. Chưa được sử dụng rộng rãi vì NoSQL vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu đã được xây dựng từ lâu, nên việc chuyển đổi cũng là vấn đề khó với nhiều doanh nghiệp và chưa được hỗ trợ tốt về chức năng cũng như sự ổn định như RMBMS.
3. Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có tool hỗ trợ giao diện tương tác cũng như các phương thức tốt nhất. Dẫn đến hạn chế về tri thức nghiệp vụ.
4. Chia sẻ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng API riêng của mình. Sự thiếu hụt các tiêu chuẩn có nghĩa là nó không có khả năng để chuyển một cách đơn giản từ một nhà cung cấp này sang một nhà cung cấp khác.
   1. **So sánh NOSQL với cơ sở dữ liệu quan hệ**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí đánh giá** | **RDBMS** | **NoSQL** |
| Đánh chỉ mục lượng lớn dữ liệu, phân trang hoặc phân phối luồng dữ liệu media | Có nhiều điểm yếu kém | Thực hiện khá tốt |
| Mô hình dữ liệu | Table | Key/Value, Column-Based-Tabular |
| Yêu cầu phần cứng cho dữ liệu lớn | Yêu cầu phần cứng máy chủ cao | Không nhất thiết phần cứng phải cao, có thể mở rộng máy chủ ở nhiều nơi với tài nguyên phần cứng thấp |
| Cách lấy dữ liệu từ client | Thông qua chuỗi kết nối | Thông qua API |
| Hiệu suất | Kém hơn NoSQL | Rất tốt  Bỏ qua SQL và các ràng buộc dữ liệu |
| Khả năng mở rộng | Hạn chế về lượng | Hỗ trợ lượng lớn các node |
| Phù hợp với công nghệ điện toán đám mây | Không | Có |
| Hiệu suất đọc ghi | Kém | Tốt |
| Thay đổi số node trong hệ thống | Phải shutdown cả hệ thống, thay đổi phức tạp | Không cần shutdown cả hệ thống, thay đổi đơn giản, không ảnh hưởng đển hệ thống |
| Hỗ trợ lập trình hướng đối tượng | Kém | Tốt |

* 1. **Phân Loại NOSQL**

**Key – value data stores**: Dữ liệu lưu dưới dạng cặp key – value. Giá trị được truy xuất

thông qua key.

* Ví dụ : Redis, Dynomite, Project Voldemort.
* Thường cho: Content caching Applications
* Ưu điểm: Tìm kiếm rất nhanh
* Nhược điểm: Lưu dữ liệu không theo khuôn dạng (schema) nhất định

**Column-based – Tabular**: Cơ sở dữ liệu tổ chức dưới dạng các bảng. Gần giống với mô

hình RDBMS. Tuy nhiên, Chúng lưu dữ liệu bởi các cột chứ không phải bằng các dòng.

Nó khá thích hợp với để hiển thị bằng các phần mềm quản lý kho dữ liệu

* Ví dụ : Apache Hbase, Apache Cassandra, Hypertable
* Thường cho: các hệ phân tán file
* Ưu điểm: Tìm kiếm nhanh, Phân tán dữ liệu tốt
* Nhược điểm: Hỗ trợ được với rất ít phần mềm

**Document-based**: Dữ liệu (bán cấu trúc hay semi-structured) được lưu trữ và tổ chức

dưới dạng một tập hợp các document. Các document này linh hoạt, mỗi document có một

tập nhiều trường.

* Ví dụ : Apache CouchDB và MongoDB
* Thường cho: Web applications
* Ưu điểm: Dùng khi dữ liệu nguồn không được mô tả đầy đủ
* Nhược điểm: Hiệu năng truy vấn, Không có cú pháp chuẩn cho câu truy vấn dữ liệu

**Graph-based data-stores**: Những CSDL này áp dụng lý thuyết đồ thị trong khoa học máy tính để lưu trữ và truy xuất dữ liệu. Chúng tập trung vào tính rời rạc giữa các phần dữ

liệu. Các phần tử đơn vị dữ liệu được biểu thị như một nút và liên kết với các thành phần

khác bằng các cạnh.

* Ví dụ : Neo4j, InfiniteGraph, DEX
* Thường cho: Social networking, Hệ trợ giúp
* Ưu điểm: Ứng dụng các thuật toán trên đồ thị như Đường đi ngắn nhất, liên thông,…
* Nhược điểm: Phải duyệt nội bộ đồ thị, để trả lời lại các truy vấn. Không dễ để phân tán

**CHƯƠNG 2 : MongoDB**

## 2.1 Khái niệm MongoDB

MongoDB (bắt nguồn từ “humongous”) là một hệ cơ sở dữ liệu NoSQL mã nguồn mở.

 Thay cho việc lưu trữ dữ liệu vào các bảng có quan hệ với nhau như truyền thống, MongoDB lưu các dữ liệu cấu trúc dưới dạng giống với JSON(JavaScript Object Notation) và gọi tên là BSON. Dự án được bắt đầu triển khai vào tháng 10 năm 2007 bởi 10gen trong khi công ty này đang xây dựng một nền tảng như là dịch vụ (Platform as a Service) giống như Google App Engine. Phải đến năm 2009, dự án này được tách độc lập. Hệ thống có thể chạy trên Windows, Linux, OS X và Solaris. Nó được một số tổ chức sử dụng trong thực tế như:

* Caigslist : Công ty làm việc trong lịch vực môi giới quảng cáo trên các website khác (giống adMicro của Việt Nam). MongoDB giúp cho công ty này quản lý hàng tỉ các bản ghi quảng cáo thuận tiện và nhanh chóng.
* Foursquare là một mạng xã hội gắn các thông tin địa lý. Công ty này cần lưu dữ liệu của rất rất nhiều vị trí của các địa điểm như quán cafe, nhà hàng, điểm giải trí, lịch sử, … và ghi lại những nơi mà người sử dụng đã đi qua.
* CERN : Trung tâm nghiên cứu năng lượng nguyên tử của Châu Âu, sử dụng MongoDB để lưu trữ lại các kết quả, dữ liệu thí nghiệm của mình. Đây là một lượng dữ liệu khổng lồ sẽ dùng để sử dụng trong tương lai.
* MTV Networks, Disney Interactive Media Group, bit.ly, The New York Times, The Guardian, SourceForge, Barclays, …

**Tại sao lại chọn MongoDB?**

MongoDB có những ưu điểm sau đây:

* Dễ học, có một số nét khá giống với CSDL quan hệ – Quản lý bằng command line hoặc bằng GUI như RockMongo hoặc phpMoAdmin
* Linh động, không cần phải định nghĩa cấu trúc dữ liệu trước khi tiến hành lưu trữ nó -> rất tốt khi ta cần làm việc với các dạng dữ liệu không có cấu trúc.
* Khả năng mở rộng tốt (distributed horizontally), khả năng cân bằng tải cao, tích hợp các công nghệ quản lý dữ liệu vẫn tốt khi kích thước và thông lượng trao đổi dữ liệu tăng.
* Miễn phí

## 2.2 Ưu nhược điểm của MongoDB

**Ưu điểm :**

* Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng. Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
* MongoDB rất dễ mở rộng.
* Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB.
* MongoDB phù hợp cho các ứng dụng realtime.

**Nhược điểm :**

Điều đầu tiên phải kể đến ở đây là MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận. MongoDB sử dụng sẽ hao tốn tài nguyên của hệ thống nhiều hơn RDBMS. Nhưng đến thời điểm hiện tại thì vấn đề này không còn là điều lo ngại nữa.

## 2.3 Kiến trúc tổng quát

Một MongoDB Server sẽ chứa nhiều database. Mỗi database lại chứa một hoặc nhiều colection. Đây là một tập các documnents, về mặt logic thì chúng gần tương tự như các table trong CSDL quan hệ. Tuy nhiên, điểm hay ở đây là ta không cần phải định nghĩa trước cấu trúc của dữ liệu trước khi thao tác thêm, sửa dữ liệu… Một document là một đơn vị dữ liệu – một bản ghi (không lớn hơn 16MB). Mỗi chúng lại chứa một tập các trước hoặc các cặp key – value. Key là một chuỗi ký tự, dùng để truy xuất giá trị dạng : string, integer, double, … Dưới đây là một ví dụ về MongoDB document

{

\_id : ObjectId("4db31fa0ba3aba54146d851a"),

username : "joegunchy",

email : "joe@mysite.org",

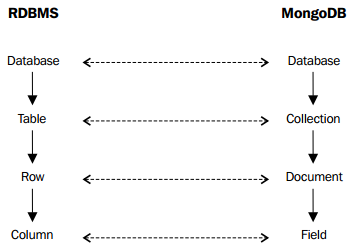
age : 26,

is\_admin : true,

created : "Sun Apr 24 2011 01:52:58 GMT+0700 (BDST)"

}

Cấu trúc có vẻ khá giống JSON, tuy nhiên, khi lưu trữ document này ra database, MongoDB sẽ serialize dữ liệu thành một dạng mã hóa nhị phân đặc biệt – BSON. Ưu điểm của BSON là hiệu quả hơn các dạng format trung gian như XML hay JSON cả hệ tiêu thụ bộ nhớ lẫn hiệu năng xử lý. BSON hỗ trợ toàn bộ dạng dữ liệu mà JSON hỗ trợ (string, integer, double, Boolean, array, object, null) và thêm một số dạng dữ liệu đặc biệt như regular expression, object ID, date, binary, code.



**2.4 Những thuật ngữ đầu tiên trong MongoDB**

1. **id** – Là trường bắt buộc có trong mỗi document. Trường \_id đại diện cho một giá trị duy nhất trong document MongoDB. Trường \_id cũng có thể được hiểu là khóa chính trong document. Nếu bạn thêm mới một document thì MongoDB sẽ tự động sinh ra một \_id đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.
2. **Collection** – Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong cơ sở dữ liệu RDBMS (Relational Database Management System). Collection nằm trong một cơ sở dữ liệu duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.
3. **Cursor** – Đây là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.
4. **Database** – Nơi chứa các Collection, giống với cơ sở dữ liệu RDMS chúng chứa các bảng. Mỗi Database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một mấy chủ MongoDB có thể chứa nhiều Database.
5. **Document** – Một bản ghi thuộc một Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường tên và giá trị.
6. **Field** – Là một cặp name – value trong một document. Một document có thể có không hoặc nhiều trường. Các trường giống các cột ở cơ sở dữ liệu quan hệ.
7. **JSON** – Viết tắt của JavaScript Object Notation. Con người có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang hỗ trợ rất nhiều ngôn ngữ lập trình.
8. **Index** – Là những cấu trúc dữ liệu đặc biệt, dùng để chứa một phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Chỉ số lưu trữ giá trị của một fields cụ thể hoặc thiết lập các fields, sắp xếp theo giá trị của các fields này. Index hỗ trợ độ phân tích một cách hiệu quả các truy vấn. Nếu không có chỉ mục, MongoDB sẽ phải quét tất cả các documents của collection để chọn ra những document phù hợp với câu truy vấn. Quá trình quét này là không hiệu quả và yêu cầu MongoDB để xử lý một khối lượng lớn dữ liệu.

Hãy lưu ý sự khác biệt của các trường và **\_id** trong một document. Một **\_id** được dùng để đại diện cho một document và chúng được sinh ra khi thêm một Document vào Collection.

**2.5 Các kiểu dữ liệu trong MongoDB**

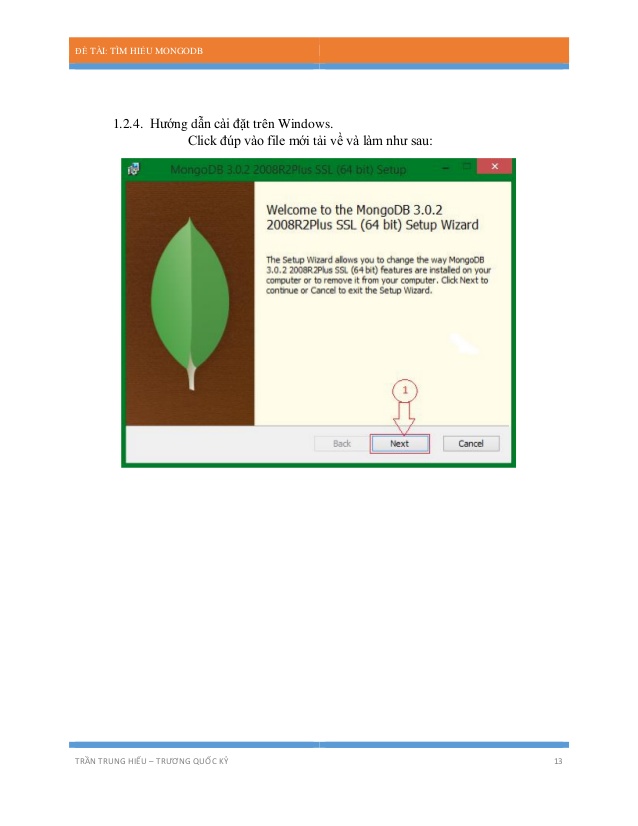
* **Chuỗi** : Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.
* **Số nguyên** : Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.
* **Boolean** : Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).
* **Double** : Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực dấu chấm động.
* **Min/ Max keys**: Kiểu dữ liệu này được sử dụng để so sánh một giá trị với các phần tử BSON thấp nhất và cao nhất.
* **Mảng**: Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.
* **Timestamp** : Đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.
* **Object**: Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.
* **Null**: Kiểu dữ liệu này được sử dụng để lưu một giá trị Null.
* **Symbol**: Kiểu dữ liệu này được sử dụng giống như một chuỗi
* **Date** : Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time.
* **Object ID**: Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.
* **Binary data**: Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân.

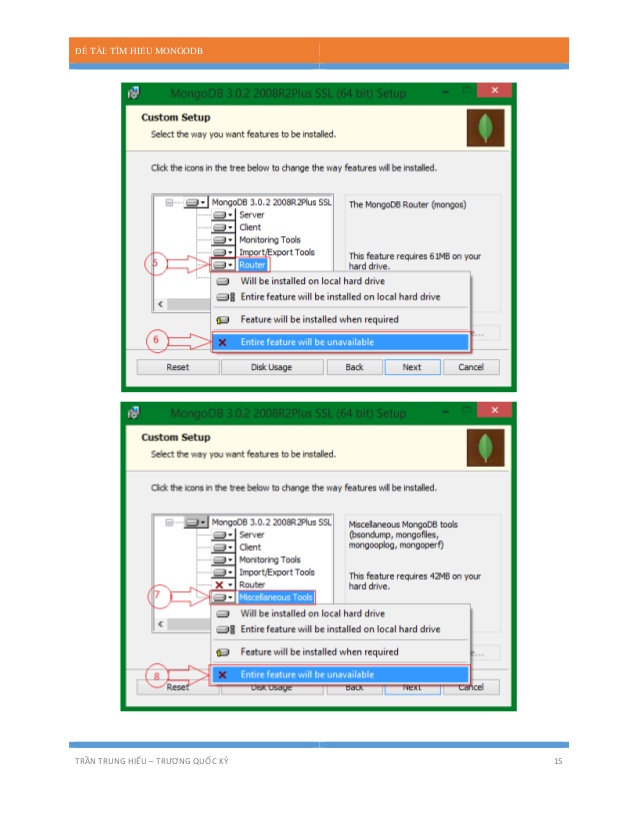
## 2.6 Một số đặc trưng của MongoDB

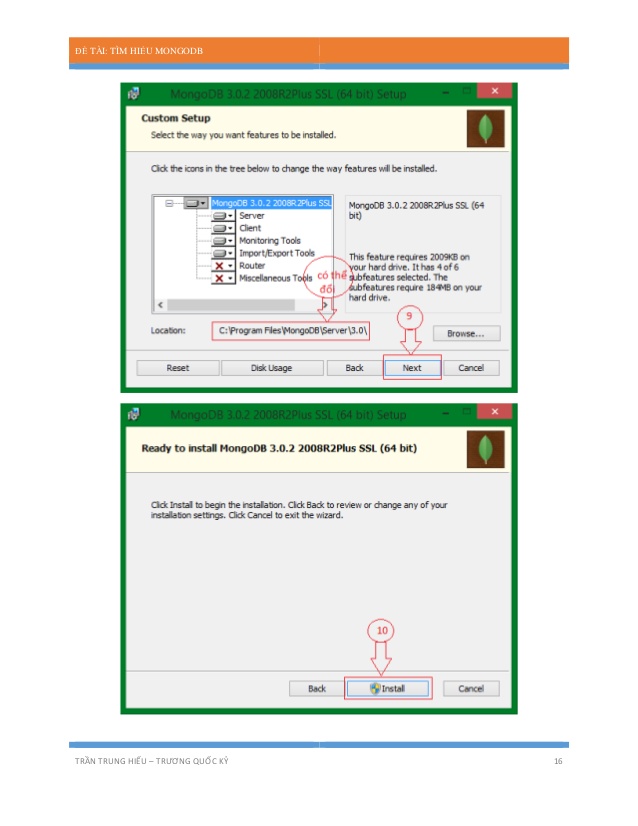
|  |  |
| --- | --- |
| **Thuộc tính** | **Đặc trưng** |
| Lưu trữ hướng văn bản | Văn bản theo phong cách JSON với nhứng lược đồ động đơn giản |
| Hỗ trợ chỉ mục đầy đủ | Định mục trên bất kỳ các thuộc tính |
| Tính sao lặp và tính sẵn sàng cao | Mở rộng |
| Auto-Sharding | Mở rộng theo chiều ngang mà không ảnh hưởng đến chức năng |
| Truy vấn | Đa dạng, truy vấn dựa |
| GrindFS | Lưu trữ file với bất kỳ kích cỡ nào mà không làm phức tạp ngăn xếp |
| Hỗ trợ thương mại | Hỗ trợ doanh nghiệp đào tạo, tư vấn |

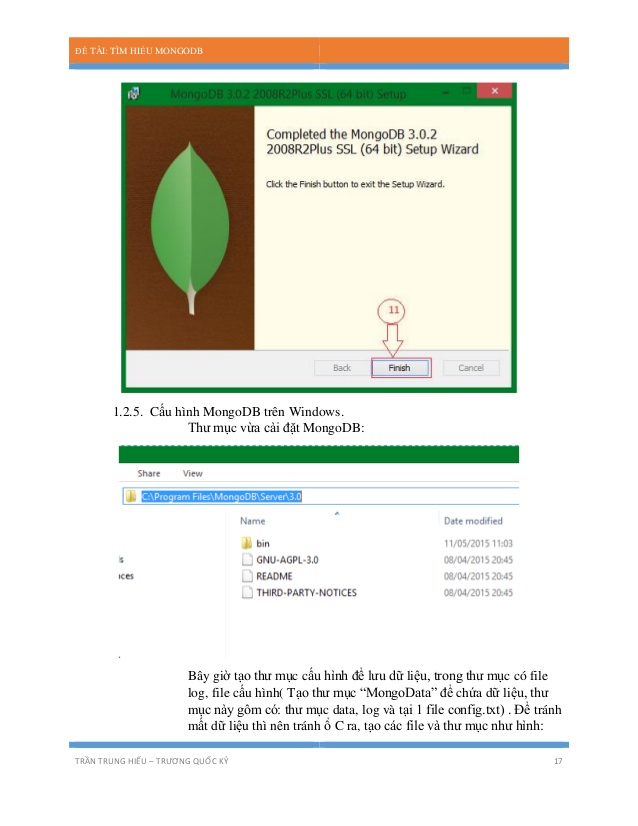
## 2.7 Cài đặt MongoDB

Vào trang <https://www.mongodb.com/download-center/>









# TÀI LIỆU THAM KHẢO

<https://viblo.asia/p/mongodb-va-nhung-dieu-can-biet-phan-1-m68Z03VQKkG?fbclid=IwAR3YElIAFxcq_QBoR3XgibhVa8Fu1VyjSayGePuAYfQ0DGWoqixi8UvOW1U>

<https://www.mongodb.com/>

<https://bigsonata.wordpress.com/2014/06/05/mongodb/?fbclid=IwAR11idLoCRr3MxB2XYDnrGzHVo7YrAxepr30UkdDVQTYZ_IVfkZ3KGfG_tU>

<https://www.slideshare.net/MrNoo1/tm-hiu-mongodb?fbclid=IwAR1dujH-gGrIruwVVu5m91LfcaDypmJ3mslZM1W69meDBzOtIxx63MYZ0JU>