33333

TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

**--🕮--**

**MÔN: CƠ SỞ DỮ LIỆU NÂNG CAO**

**BÁO CÁO CUỐI KÌ MONGODB**



**Giảng viên hướng dẫn:** Ths. :Lương Trần Hy Hiến

**Nhóm sinh viên thực hiện:**

Nguyễn Công Khanh 43.01.104.075

Ngô Xuân Hải 43.01.104.041

Tạ Hoàng Trí 43.01.104.189

Hoàng Văn Quý 43.01.104.143

Nguyễn Văn Hiếu 43.01.104.049

Hồ Chí Minh, ngày 20 tháng 11 năm 2019

MỤC LỤC

[1.Giới thiệu về MongoDB 1](#_Toc25178382)

[2. Các thuật ngữ hay sử dụng trong MongoDB 2](#_Toc25178383)

[3. So sánh giữa RDBMS (Relational database management system) và MongoDB 4](#_Toc25178384)

[4. Các kiểu dữ liệu trong MongoDB 4](#_Toc25178385)

[5. Một số câu lệnh dùng trong MongoDB 5](#_Toc25178386)

[6. MongoDB hoạt động như thế nào: 6](#_Toc25178387)

[7. Lợi thế của MongoDB 8](#_Toc25178388)

[8. Khi nào NÊN sử dụng MongoDB ? 10](#_Toc25178389)

[9. Khi nào KHÔNG NÊN sử dụng MongoDB ? 10](#_Toc25178390)

[10. Các công cụ quản trị MongoDB 10](#_Toc25178391)

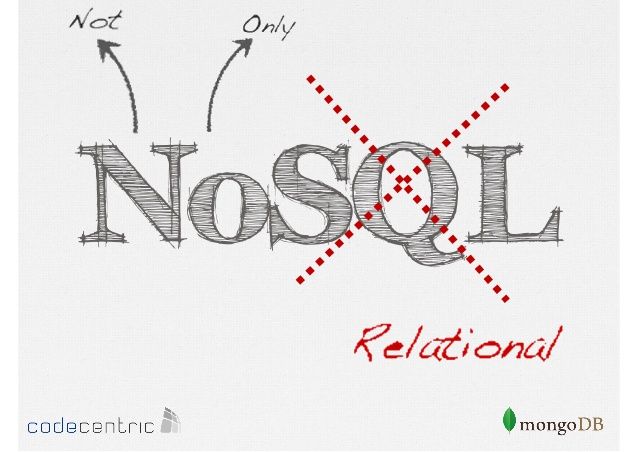
[11. Cài đặt MongoDB 10](#_Toc25178392)

[Một số ví dụ về mongo 11](#_Toc25178393)

# 1.Giới thiệu về MongoDB

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++ và 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.

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng Transact-SQL để truy vấn thông tin.NoSQL viết tắt bởi: None-Relational SQL, hay có nơi thường gọi là Not-Only SQL.CSDL này được phát triển trên Javascript Framework với kiểu dữ liệu JSON. (Cú pháp của JSON là “key:value”)

Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng

# 2. Các thuật ngữ hay sử dụng trong MongoDB

**\_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

**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.

**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.

**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.

**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.

**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.

**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.

# 3. So sánh giữa RDBMS (Relational database management system) và MongoDB

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

### 

# 4. Các kiểu dữ liệu trong MongoDB

MongoDB hỗ trợ các kiểu dữ liệu sau:

|  |  |  |
| --- | --- | --- |
| Type | Number | Alias |
| Double | 1 | “double” |
| String | 2 | “String” |
| Object | 3 | “Object” |
| Array | 4 | “Array” |
| Binary data | 5 | “Bindata” |
| Undefined | 6 | “Undefined” |
| ObjectId | 7 | “ObjectId” |
| Boolean | 8 | “bool” |
| Date | 9 | “date” |
| Null | 10 | “null” |
| 32-bit integer | 16 | “int” |
| Timestamp | 17 | “Timestamp” |
| 64-bit integer | 18 | “long” |
| Min key | -1 | “minKey” |
| Max key | 123 | “maxkey” |

# 5. Một số câu lệnh dùng trong MongoDB

|  |  |
| --- | --- |
| Hiển thị database | show dbs |
| Tạo database | use database\_name |
| Xóa database | db.dropDatabase() |
| Tạo collection | db.createCollection(name, options) |
| Drop collection | db.COLLECTION\_NAME.drop() |
| Insert document | db.COLLECTION\_NAME.insert(document) |
|  | db.COLLECTION\_NAME.insertOne(document) |
|  | db.COLLECTION\_NAME.insertMany(document) |
| Truy vấn Document | db.COLLECTION\_NAME.find(query) |
|  | db.COLLECTION\_NAME.findOne(query) |
| Cập nhật Document | db.COLLECTION\_NAME.update(query, update, options) |
|  | db.COLLECTION\_NAME.updateOne(query, update, options) |
|  | db.COLLECTION\_NAME.updateMany(query, update, options) |
| Save | db.COLLECTION\_NAME.save({\_id:ObjectId(),NEW\_DATA}) |
| Xóa documnet | db.COLLECTION\_NAME.remove(query) |
| Giới hạn bản ghi | db.COLLECTION\_NAME.find().limit(NUMBER) |
| Sắp xếp bản ghi | db.COLLECTION\_NAME.find().sort({KEY:1}) |
| Tạo chỉ mục (index) | db.COLLECTION\_NAME.ensureIndex({KEY:1}) |
| …. |  |

+ query: tiêu chí để update.

+ update: document muốn update.

+ options: các document khác không bắt buộc.

|  |  |
| --- | --- |
| **Phép toán** | **Cú pháp** |
| Bằng(Equality) | {key: value} |
| Nhỏ hơn (Less Than) | {key: {$lt: value}} |
| Nhỏ hơn bằng (Less Than Equals) | {key: {$lte: value}} |
| Lơn hơn (Greater Than) | {key: {$gt: value}} |
| Lớn hơn bằng (Greater Than Equals) | {key: {$gte: value}} |
| Khác (Not Equals) | {key: {$ne: value}} |
| Trong ( In) | {key: {$in: [value1, value2,..]}} |
| Không Thuộc (Not In) | {key: {$nin: [value1, value2,..]}} |

# 6. MongoDB hoạt động như thế nào:

### 

### Ưu điểm

MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau.  
Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

**Nhược điểm**

Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn

Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao

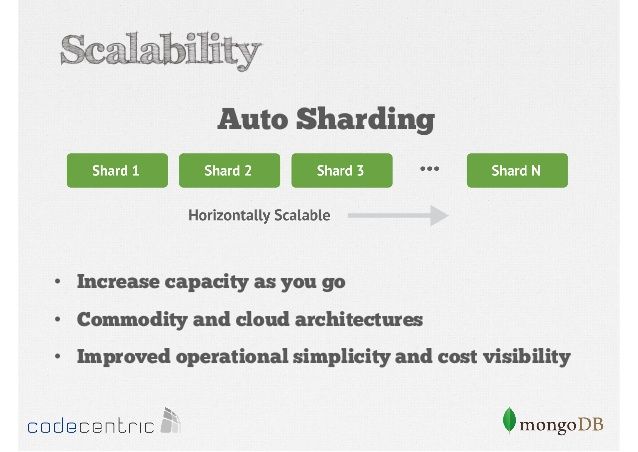
# 7. Lợi thế của MongoDB

### Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong mongodb thì chỉ 1 collection ta có thể chứa nhiều document khác nhau . Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.

Cấu trúc của một đối tượng rõ ràng.

Không có các Join phức tạp.

Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.



### Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy cập dữ liệu nhanh hơn. Việc cập nhật được thực hiện nhanh gọn nhờ update tại chỗ (in-place).

# 8. Khi nào NÊN sử dụng MongoDB ?

Sử dụng MongoDB trong trường hợp:

Nếu website của bạn có tính chất INSERT cao  
Bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.Website của bạn ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dung. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì nó sẽ bỏ qua phần đó nên sẽ an toàn.

Website bạn có nhiều dữ liệu quá  
Giả sử web bạn có đến 10 triệu records thì đó là cơn ác mộng với MYSQL. Bởi vì MongoDB có khả năng tìm kiến thông tin liên quan cũng khá nhanh nên trường hợp này nên dùng nó.

Máy chủ không có hệ quản trị CSDL  
Trường hợp này thường bạn sẽ sử dụng SQLITE hoặc là MongoDB.

# 9. Khi nào KHÔNG NÊN sử dụng MongoDB ?

Các ứng dụng cần sử dụng nhiều transaction (như ngân hàng) do Mongodb không có cơ chế transaction (giao dịch) để phục vụ cho các ứng dụng ngân hàng

Các ứng dụng cần SQL (sử dụng joins).

# 10. Các công cụ quản trị MongoDB

RoboMongo

MongoExplorer

Mongodb Compass

# 11. Cài đặt MongoDB

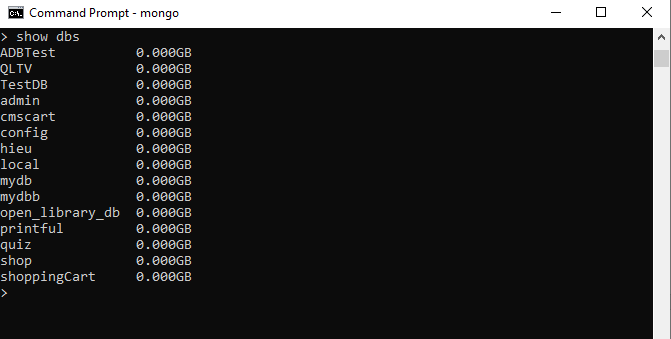
Trên Windows:

1. Tải MongoDB từ link bên dưới:  
   <https://www.mongodb.org/downloads>
2. Chạy file vừa tải về
3. Sau khi chạy xong file, vào thư mục C:\Program Files\MongoDB\Server\3.4\bin , chạy file mongod để khởi động mongoDB
4. Dùng địa chỉ 127.0.0.1:27017 để tạo connection đến mongoDB

Chi tiết tại: <https://o7planning.org/vi/10265/huong-dan-cai-dat-va-cau-hinh-database-mongodb>

# Một số ví dụ về mongo

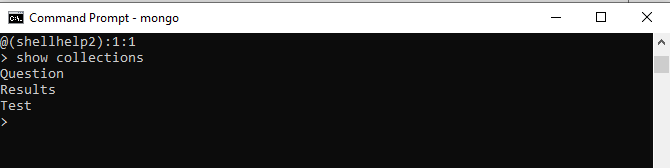
Show db



Tạo db ( lưu ý phải tạo ít nhất 1 collection thì mới hiển thị db)



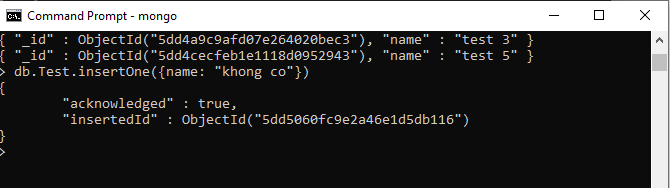
Show collection



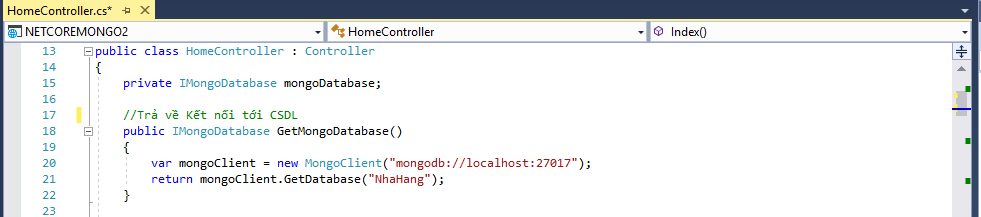
Find



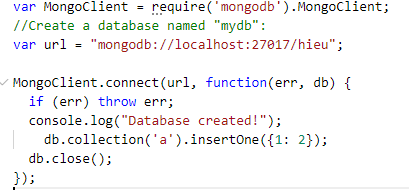
Insert



Tạo kết nối c# với mongodb



Tạo kết nối với nodejs



Tài liệu tham khảo

1. <https://docs.mongodb.com/manual/tutorial/>
2. <https://en.wikipedia.org/wiki/MongoDB>
3. <https://vietjack.com/mongodb/>