TÌM HIỂU VỀ MONGODB

1.1. Giới thiệu về hệ cơ sở dữ liệu NoSQL:

1.1.1 Khái niệm NoSQL:

Với hầu hết các thời kỳ web, Hệ quản trị cơ sở dữ liệu quan hệ dựa trên SQL đã thống trị hầu hết các hệ Quản trị Cơ sở dữ liệu. Tuy nhiên, thời gian gần đây, một cách tiếp cận mới đã bắt đầu biết đến là NoSQL, tạo ra sự thay thế cho các hệ quản trị cơ sở dữ liệu truyền thống.

NoSQL còn có nghĩa là Non-Relational - không ràng buộc. Tuy nhiên, thuật ngữ đó ít phổ dụng hơn và ngày nay người ta thường dịch NoSQL thành Not Only SQL - Không chỉ SQLS. NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm.

Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các cơ sở dữ liệu quan hệ nguồn mở nhỏ nhưng không sử dụng SQL cho truy vấn.

Hệ CSDL này có thể lưu trữ, xử lý từ lượng rất nhỏ đến hàng petabytes dữ liệu với khả năng chịu tải, chịu lỗi cao nhưng chỉ đòi hỏi về tài nguyên phần cứng thấp.

NoSQL thích hợp cho các ứng dụng cực lớn (dịch vụ tìm kiếm, mạng xã hội…) và các ứng dụng nhỏ.

Bảng 1.1. So sánh giữa SQL và NoSQL:

|  |  |  |
| --- | --- | --- |
| **Tính năng** | **CSDL quan hệ** | **NoSQL** |
| Hiệu suất | Kém hơn  Truy vấn SQL  Có quan hệ giữa các bảng | Cực cao  Bỏ qua truy vấn SQL  Bỏ qua ràng buộc quan hệ |
| Hiệu suất đọc-ghi | Kém do thiết kế để đảm bảo vào/ra liên tục của dữ liệu. | Tối ưu về tốc độ đọc ghi dữ liệu. |
| Phần cứng | Đòi hỏi về phần cứng cao. | Đòi hỏi phần cứng thấp hơn. |

1.1.2. Ưu nhược điểm của CSDL NoSQL:

1. Ưu điểm của NoSQL:

* 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.
* 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 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 cẩn 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.
* 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 cảu các ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các kết nối mạnh. NoSQL lưu trữ dữ liệu thường mở rộng theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn.
* Được các hãng lớn như Amazon, Facebook và Google sử dụng. Làm nên tảng, cơ sở cho các công ty lớn áp dụng cũng như có sự tin cậy cao về hệ quản trị NoSQL.

1. Nhược điểm của NoSQL:

* Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhấtt từ nhà cung cấp RMBMS(vì được phát triển trước NoSQL một thời gian khó dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được hỗ trợ tốt nhất.
* Chưa được sử dụng rộng rãi vì nó vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu 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.
* Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có tool giao diện tương tác cũng như các phương thức tốt nhất. Dẫn đến hạn chế về tri thức nghiệp vụ.
* Chia sẽ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng 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 nhà cung cấp khác.

1.2. Hệ cơ sở dữ liệu MongoDB:

Trong những gương mặt góp phần làm suy tàn SQL thì MongoDB nổi lên là một CSDL đáng tin cậy và dễ dùng nhất. Mongo viết bằng C++. Nó thích hợp cho các ứng dụng tầm trung trở lên. Nếu tỉ lệ lượng dữ liệu ghi vào CSDL của ứng dụng lớn hơn lượng đọc thì đây càng là lựa chọn hợp lý.

MongoDB là một CSDL có khả năng mở rộng, hiệu suất cao, mã nguồn mở và hướng văn bản.

1.2.1. Giới thiệu về MongoDB:

MongoDB là một mã nguồn mở, dùng để thiết kế cơ sở dữ liệu để dễ dàng phát triển và mở rộng, cung cấp hiệu suất cao, tính sẵn sàng cao.

Cấu trúc của Mongodb bao gồm các cặp “thuộc tính” – “giá trị”, MongoDb document tương tự như đối tượng JSOn. Giá trị của thuộc tính bao gồm các thuộc tính giá trị cụ thể, document, mảng, mảng các đối tượng.

Các tính năng chính:

* Hiệu suất cao: hỗ trợ nhúng dữ liệu dạng mô hình dữ liệu giúp giảm thiểu hoạt động của server, truy vấn dữ liệu sử dụng chỉ mục giúp tối ưu tốc độ truy vấn.
* Dễ dàng tăng tính mở rộng: chế độ sharding tự động ( tự động phân cùng dữ liệu trên máy chủ, đồng bộ hóa dữ liệu tốt.

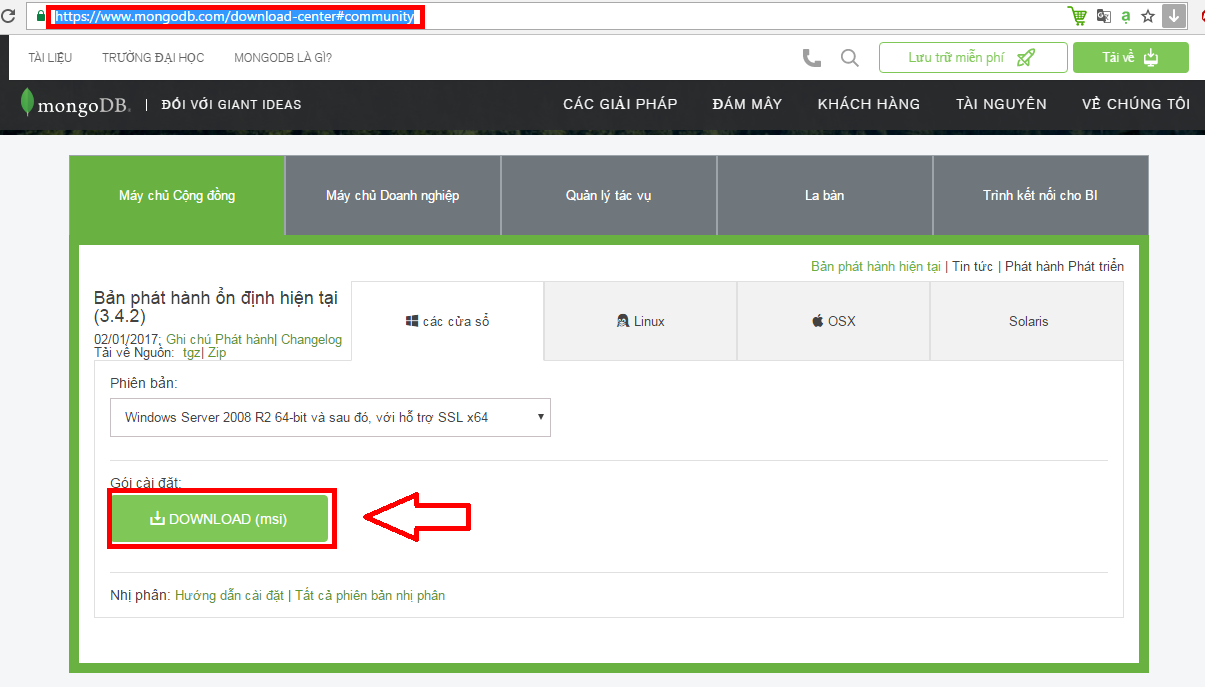
MongoDb chạy trên hầu hết các nền tảng và hỗ trợ kiến trúc 64-bit để sử dụng sản xuất và cả hai 64-bit và 32-bit để thử nghiệm.

Bảng 1.2. Các hệ điều hành hỗ trợ

|  |  |  |
| --- | --- | --- |
|  | MongoDB | MongoDB Enterprise |
| Amazon Linux | Hỗ trợ | Hỗ trợ |
| Debian 7.1 | Hỗ trợ | Hỗ trợ |
| RedHat / CentOS 6.2+ | Hỗ trợ | Hỗ trợ |
| SUSE 11 | Hỗ trợ | Hỗ trợ |
| Ubuntu LTS12.04 | Hỗ trợ | Hỗ trợ |
| Ubuntu LTS14.04 | Hỗ trợ | Hỗ trợ |
| Windows Server 2012 | Hỗ trợ | Hỗ trợ |
| Mac OSX 10.6+ | Hỗ trợ |  |
| RedHat / CentOS 5.5+ | Hỗ trợ |  |
| RedHat / CentOS 5.7+ | Hỗ trợ | Hỗ trợ |
| Windows Server 2008 R2 | Hỗ trợ | Hỗ trợ |

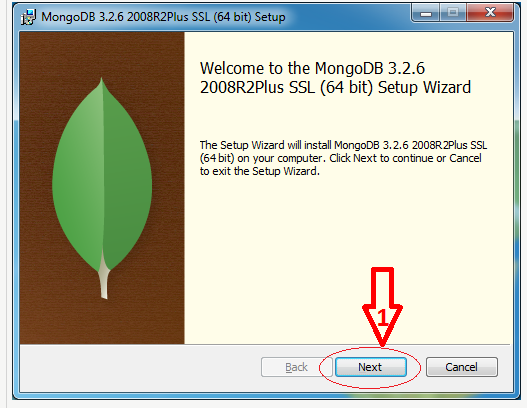
1.2.2. Cài đặt MongoDb:

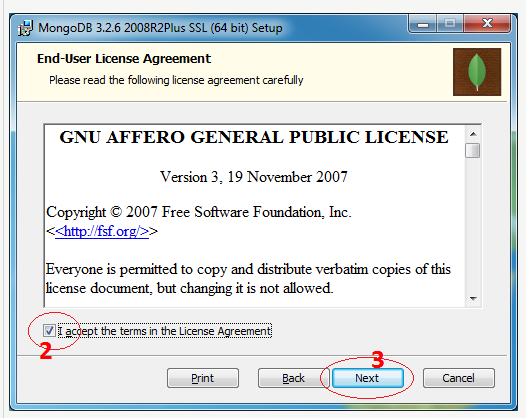
Truy cập vào trang <https://www.mongodb.com/download-center#community> kéo xuống dưới và làm theo hướng dẫn như hình:

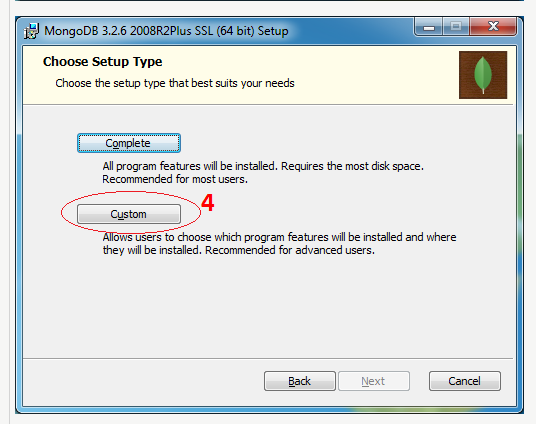


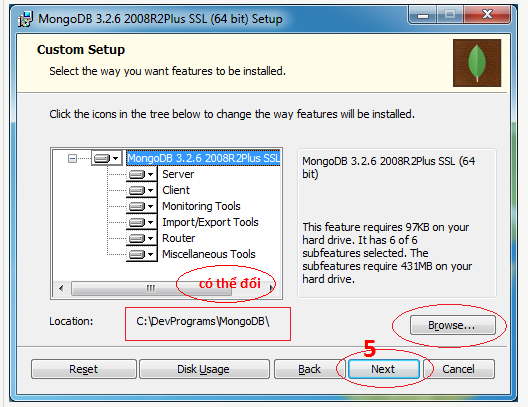
Cài đặt trên Windows:

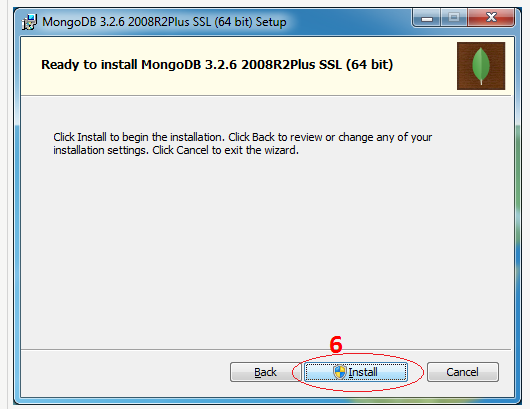
Kích đúp vào file mới tải về và làm như sau:

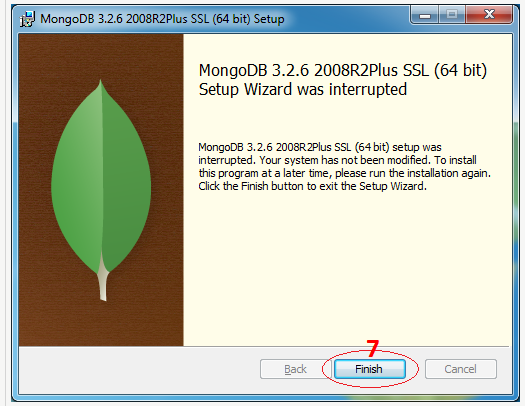






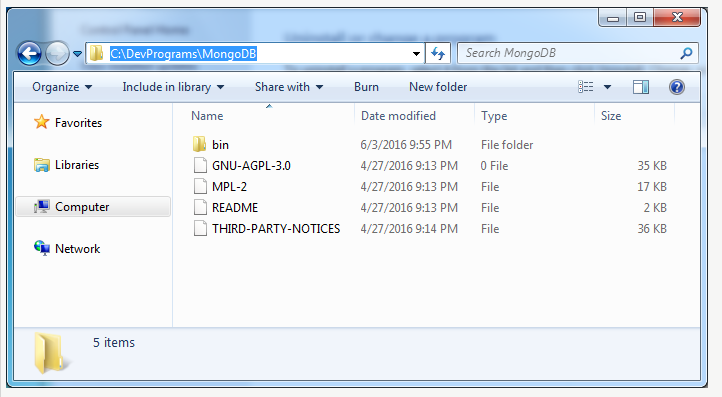




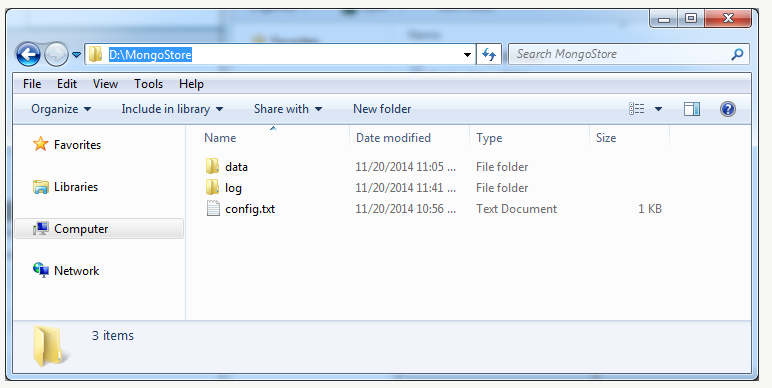


1.2.3. Cấu hình Mongodb trên Windows:

Thư mục vừa cài đặt được như sau:



Bây giờ tạo thư mục cấu hình để lưu dữ liệu, trong thư mục có file log, file cấu hình( tạo thư mục “mongodata” để chứa dữ liệu, thư mục này gồm có: thư mục data, log và một file config.txt). Để tránh mất dữ liệu thì nên tránh ổ C ra, tạo các file và thư mục như hình:



Trong file “config.txt” có nội dung:

##store data

dbpath=D:\MongoStore\data

##all output go here

logpath=D:\MongoStore\log\mongo.log

* Còn thư mục còn lại để trống.
* Vào “cmd” gõ lệnh trỏ đến thư mục bin của mongodb.
* Chạy file cấu hình bằng lệnh “mongod.exe --config D:\MongoStore\config.txt”

Đến bước trên thì việc cấu hình mongodb thành công. Giữ nguyên màn hình “cmd” đừng tắt đi.

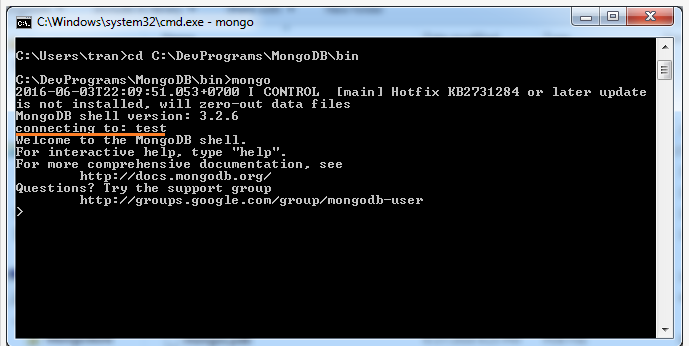
1.2.4. Bắt đầu làm việc với MongoDB:

Tạo cửa sổ “cmd” và gõ lệnh sau:

*cd C:\DevPrograms\MongoDB\bin*

Tiếp theo chạy lệnh : *mongo*

Nếu sau khi chạy lệnh màn hình xuất hiện như hình sau thì bạn đã kết nối vào được database test



\MongoDB\bin52

1.2.5. Công cụ RoboMongo làm việc với MongoDB

* RoboMongo là một công cụ trực quan giúp bạn quản lí Database MongoDB, mó là một phần mềm mã nguồn mở miễn phí, hỗ trợ cả 3 loại hệ điều hành: Linux, Windows, Mac OS.

- Link tải: http://robomongo.org/download.html

1.2.6. MongoDB CRUD

1.2.6.1. Giới thiệu MongoDB CRUD

CRUD là viết tắt của Creat( Tạo), Retrieve(Lấy), Update(Cập nhật), và Delete(Xóa). MongoDB lưu trữ dữ liệu trong các document, với các cặp thuôc tính: giá trị như Json.

{

“MaNhanVien”:NV2, Thuộc tính:Giá trị

“Quyen”:Nhanvien, Thuộc tính:Giá trị

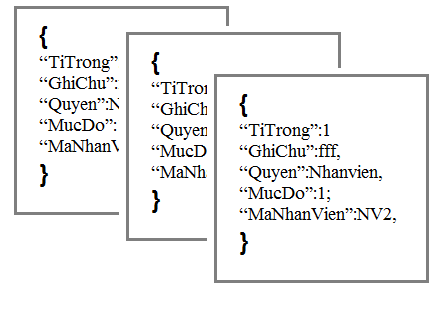
“MucDo”:1; Thuộc tính:Giá trị

“GhiChu”:fff, Thuộc tính:Giá trị

“TiTrong”:1 Thuộc tính:Giá trị

}

MongoDB lưu trữ tất cả các document trong Collection. Collection là một nhóm các doucument giống nhau về chỉ số các thành phần.



1.2.6.2. Các khái niệm trong MongoDB CRUD

* Database là gì ?

Database là một ô chứa dữ liệu ở mức vật lí, mỗi database sẽ có nhiều Collection và được thiết lập để lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDb thường có thể tạo nhiều cơ sở dữ liệu.

* Collection (Bộ sưu tập) là gì?

Collection trong MongoDb là nhóm các tài liệu (document) nó tương đương với một bảng(table) trong cơ sở dữ liệu thông thường nên mỗi collection sẽ chỉ thuộc về một database duy nhất. Tuy nhiên nó có sự khác biệt đó là nó không có sự ràng buộc Relationship như các hệ quản trị khác nên việc truy xuất dữ liệu rất nhanh, chính vì thế mỗi collection có thể chức nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định.

* Document (Văn bản) là gì?

Document trong MongoDb có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp “key-value” nên có tính năng động rất lớn. Ta có thể hiểu nó như các record dữ liệu trong MySQL, tuy nhiên nó có sự khác biệt là các cặp “key-value” có không thể giống nhau ở mỗi document.

MongoDB có thể lưu trữ nhiều CSDL độc lập, mỗi CSDL này có các bộ sưu tập và điều khoản riêng của mình

MongoDB đi kèm với một trình tiện ích JavaScript đơn giản nhưng mạnh mẽ, nó hữu ích trong quản trị và thao tác dữ liệu.

Mỗi văn bản có một khóa đặc biệt, đó là “\_id”, nó là duy nhất trong bộ sưu tập của văn bản.

| **SQL** | **MongoDB** |
| --- | --- |
| Table (Bảng) | Collection ( Bộ sưu tập) |
| Row (Hàng) | Document (Văn bản) |
| Column (Cột) | Field ( một cặp key - value) |
| Primary key  ( Khóa chính) | Primary key (mặc định là \_id do chính Mongo tạo) |

*a. Document:*

Document(Văn bản) là một khái niệm quan trọng trong MongoDB. Văn bản bao gồm tập hợp các khóa với các giá trị tương ứng.

Ví dụ: {"cntt" : "Hello, cong nghe thong tin!"}

Văn bản trên gồm một khóa là “cntt”, với giá trị là “Hello, cong nghe thong tin!”. Các văn bản có thể chứa nhiều cặp khóa/giá trị.

Ví dụ: {"cntt" : "Hello, cong nghe thong tin!", "ba" : 3}

Một số lưu ý:

* Các cặp khóa/ giá trị trong văn bản được sắp xếp. Văn bản trên sẽ khác với văn bản sau

{"ba" : 3, "cntt" : "Hello, cong nghe thong tin!"}

* Khóa trong văn bản là một chuỗi
* MongoDB phân biệt chữ hoa chữ thường
* Văn bản trong MongoDB không được chứa những khóa giống nhau. Ví dụ văn bản sau là không hợp lệ

{"cntt" : "Hello, cong nghe thong tin!", "cntt" : "Hello, tui!"}

*b.Colletion:*

Collection(Bộ sưu tập) là một nhóm các văn bản. Nếu văn bản tương đương với dòng trong CSDL quan hệ thì bộ sưu tập tương đương với bảng.

Các văn bản có hình dạng khác nhau có thể cùng được lưu trữ trong 1 bộ sưu tập.

Bộ sưu tập được xác định bởi tên của nó là một chuỗi UTF-8

Các đặc trưng của MongoDB:

* Lưu trữ hướng văn bản: Văn bản theo phong cách JSON với những lược đồ động đơn giản
* Hỗ trợ chỉ mục đầy đủ: chỉ 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 trên văn bản
* Cập nhật nhanh:
* Map/Reduce
* GridFS: 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 tào, tư vấn.

1.2.7. Data Model

1.2.7.1. Giới thiệu Data Model

Hệ quản trị cơ sở dữ liệu MongoDB là một mô hình dữ liệu linh hoạt. Không giống như các cơ sở dữ liệu SQL, phải xác định và triển khai sơ đồ của một bảng trước khi chèn dữ liệu, Collection trong MongoDB rất linh hoạt không nhất thiết phải cho biết những thuộc tính có trong Collection khi nào cần bổ sung sau vẫn được. Sự linh hoạt này tạo điều kiện cho việc hướng tới một cách lưu trữ như một thực thể hay một đối tượng. Mỗi document có phù hợp với các trường dữ liệu của Collection chứa nó, ngay cả khi thay đổi gâng như toàn bộ dữ liệu. Tuy nhiên, trong thực tế, các Document trong một Collection chung một cấu trúc.

Khi thiết kế cơ sở dữ liệu cần chú ý về cân bằng nhu cầu của các ứng dụng (tức là có nên tạo nhiều Collection lồng nhau), các đặc tính hiệu suất khi thực hiện truy vấn, các mô hình dữ liệu. Khi thiết kế các mô hình dữ liệu, luôn luôn xem xét việc sử dụng các dữ liệu (ví dụ như truy vấn,cập nhật, và xử lý các dữ liệu) cũng như các cấu trúc vốn có của bản thân dữ liệu

1.2.7.2. Các khái niệm cơ bản về Data Model

1. Thiết kế Data Model

* Data Model nhúng

Với MongoDB, bạn có thể nhúng các dữ liệu liên quan vào một Structure hoặc Document, hay còn gọi là mô hình “demormalized”.

Ví dụ:

Tài khoản

{

“MaNhanVien”:NV1,

“TenNhanVien”:aa,

“Quyen”:Q1,

“TrangThai”:nghi phep,

“TaiKhoan”:

{

“TenTaiKhoan”: tk1,

“MatKhau”:\*\*\*,

“MatKhauCap1”: 123

},

“GhiChu” : aaaa,

“SoLuongBanBe”: 5

}

Mô hình dữ liệu nhúng cho phép các ứng dụng lưu trữ phần dữ liệu liên quan của thông tin trong Document của các cơ sở dữ liệu.

Sử dụng nhúng mô hình dữ liệu khi: Cơ sở dữ liệu có mối quan hệ giữa các thực thể.

Nói chung, nhúng cung cấp hiệu suất tốt hơn cho hoạt động đọc, cũng như khả năng yêu cầu và lấy dữ liệu có liên quan trong một hoạt đọng cơ sơ dữ liệu duy nhất.

* Chuẩn hóa Data Model

Tạo liên kết giữa các “id” của các Document với nhau

Ví dụ:

Quyền

{

“id”:<ObjectId>,

“MoTa”: aaa,

“MucDo”: 1,

“LoaiQuyen”: Nhanvien

}

Tài Khoản

{

“id”: NV1,

“TenNhanVien”: nguyen,

“Quyen\_id”: <ObjectId>,

“TrangThai”: nghi phep,

“TaiKhoan”:

{

“TenTaiKhoan”: tk1,

“MatKhau”:\*\*\*,

“MatKhauCap1”: 123

},

“GhiChu” : aaaa,

“SoLuongBanBe”: 5

}

1. Operational Factors and Data Models (Các hoạt động và mô hình dữ liệu)

Mô hình hóa dữ liệu ứngd ụng cho MongoDB phụ thuộc vào dữ liệu, cũng như các đặc điểm của MongoDB.

* Document Growth

Một số cập nhật các Document có thể tăng kích thước của văn bản. Những cập nhật này bao gồm đẩy yếu tố vào một mảng(tức là $push) và thêm các trường mới vào một tài liệu.

Khi sử dụng công cụ lưu trữ MMAPv1, Docunment Growth có thể được xem xét cho một Data Model của bạn. Nếu kích thước dữ liệu vượt quá không gian được chia cho Document đó, MongoDB sẽ di dời các Document trên ổ đĩa.

* Indexes

Sử dụng chỉ số để cải thiện hiệu suất cho các truy vấn phổ biến. Xây dụng các chỉ số trên các thuộc tính mà xuất hiện thường xuyên trong các truy vấn và cho tất cả các yêu cầu đó trả lại kết quả được sắp xếp. MongoDB tự động tạo ra một chỉ số duy nhất trên thuộc tính là \_id.

Khi tạo Indexes cần chú ý:

* Mỗi Indexes yêu cầu tối thiểu 8KB dữ liệu lưu trữ.
* Thêm một Indexes có một số tác động tiêu cực cho các hoạt động ghi. Vì mỗi khi thêm dữ liệu phải cập nhật chỉ số.
* Cần chú ý đến việc thiết lập kích thước cho từng những thuộc tính vì Indexes sẽ chiếm không gian ổ đĩa và ô nhớ.

1.2.8. Một số đặc trưng của MongoDB:

|  |  |
| --- | --- |
| **Thuộc tính** | **Đặc trưng** |
| Lưu trữ hướng văn bản | Văn bản theo phong cách JSON với những lược đồ động đơn giản |
| Hỗ trợ chỉ mục đầy đủ | Chỉ 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 trên văn bản |
| GirdFS | Lưu trữ 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 |

Các lưu ý khi chọn MongoDB:

* Nếu ứng dụng có tính chất Insert cao, bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.
* Ứng dụng ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dụng. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì sẽ bỏ qua phần đó nên sẽ an toàn.
* ứng dụng bạn có nhiều dữ liệu. Bởi vì MongoDB có khả năng tìm kiếm thông tin liên quan cũng khá nhanh.
* Máy chủ không có hệ quản trị CSDL, trường hợp này thường bạn sẽ sử dụng SQL LIFE hoặc MongoDB.

1.2.9. Truy vấn trong MongoDB:

Một trong những tính năng tốt nhất của MongoDB là hỗ trợ truy vấn động. Hệ thống hỗ trợ truy vấn động không yêu cầu bất cứ chỉ mục nào để tìm dữ liệu. Người dùng có thể tìm dữ liệu với việc sử dụng bất kỳ tiêu chuẩn nào. Với CSDL quan hệ, truy vấn động là chuẩn hóa.

MongoDB hỗ trợ một số các đối tượng truy vấn để lấy dữ liệu. Ví dụ, giả sử chúng ta muốn sử dụng trình MongoDB để trả về mọi văn bản trong bộ sưu tập *users.* Truy vấn sẽ được viết như sau:

db.users.find({})

Trong trường hợp này, điều kiện của chúng ta là trống, nó phù hợp với mọi văn bản trong bộ sưu tập.

Ví dụ: db.users.find({'ten’: a})

Ở đây, lựa chọn của chúng ta là tất cả các văn bản mà thuộc tính *ten* là *a.*

1.2.9.1. Các toán tử so sánh:

Ta có bảng các toán tử sau:

|  |  |
| --- | --- |
| $gt | ứng với các giá trị lớn hơn giá trị điều kiện trong truy vấn |
| $gte | ứng với các giá trị lớn hơn hoặc bằng giá trị điều kiện trong truy vấn |
| $in | ứng với bất kỳ giá trị nào nằm trong một tập cho trước |
| $lt | ứng với các giá trị nhỏ hơn giá trị điều kiện trong truy vấn |
| $lte | ứng với các giá trị nhỏ hơn hoặc bằng giá trị điều kiện trong truy vấn |
| $ne | ứng với tất cả các giá trị khác với giá trị điều kiện trong truy vấn |
| $nin | ứng với bất kỳ giá trị nào không nằm trong một tập hợp cho trước |

1.2.9.2. Ví dụ so sánh giữa SQL và MongoDB:

Để nắm rõ hơn các lệnh khi chuyển từ cơ sở dữ liệu SQL sang MongoDB, cần phải tham khảo các lệnh sau đây:

|  |  |
| --- | --- |
| SQL | MONGODB |
| INSERT INTO ten VALUES(3,5) | db.ten.insert({a:3,b:5}) |
| SELECT a,b FROM ten | db.ten.find({},{a:1,b:1}) |
| SELECT \* FROM ten | db.ten.find() |
| SELECT \* FROM ten WHERE tuoi=22 | db.ten.find({tuoi:22}) |
| SELECT a,b FROM ten WHERE tuoi=22 | db.ten.find({tuoi:22},{a:1,b:1}) |
| SELECT \* FROM ten WHERE tuoi=22 ORDER BY ho | db.ten.find(({tuoi:22}),sort({ho:1}) |
| SELECT \* FROM ten WHERE tuoi>22 | db.ten.find({tuoi:{$gt:22}}) |
| SELECT \* FROM ten WHERE tuoi!=22 | db.ten.find({tuoi:{$ne:22}}) |
| SELECT \* FROM ten WHERE ho LIKE “%nguyen%” | db.ten.find({ho:/nguyen/}) |
| SELECT \* FROM ten WHERE ho LIKE “nguyen%” | db.ten.find({ho:/^nguyen/}) |
| SELECT \* FROM ten WHERE tuoi>22 AND tuoi<=40 | db.ten.find({‘tuoi’:{$gt:22,$lte:40}}) |
| SELECT \* FROM ten WHERE a=1 and b=’q’ | db.ten.find({a:1,b:’q’}) |
| SELECT \* FROM ten WHERE a=1 or b=2 | db.ten.find({$or :[{a:1},{b:2}]}) |
| SELECT COUNT(\*y) FROM ten | db.ten.count() |
| SELECT COUNT(\*y) FROM ten where tuoi>22 | db.ten.find({tuoi:{‘$gt’:22}}),count() |
| UPDATE ten SET a=1 WHERE b=’q’ | db.ten.update({b:’q’},{$set:{a:1}},false,true) |
| DELETE FROM ten WHERE z=”abc” | db.ten.remove({z:’abc’}) |

1.2.10. Ưu điểm và nhược điểm của MongoDB:

Ưu điểm

* Khả năng lưu trữ không giới hạn tùy thuộc vào ổ cứng nhưng vẫn đảm bảo hiệu suất cao.
* Khả năng mở rộng theo chiều ngang.
* Lưu trữ theo key – value giúp đẩy nhanh tốc độc truy xuất dữ liệu.
* Tính đáp ứng cao.

Nhược điểm:

* Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao
* 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.