**Tài Liệu Nhóm: 69**

Nguyễn Văn Anh Quốc – 42.01.104.125

Nguyễn Hoài Thanh – 42.01.104.159

Nguyễn Minh Hiếu Bốn – 42.01.104.014

Trịnh Hoàng Huy – 42.01.104.228

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++.

MongoDB là một trong top các các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được pahts triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc..

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các document dạng JSON với schema động rất linh hoạt. Nghĩa là bạn có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường, kiểu của trường lưu trữ. Tài liệu MongoDB tương tự như các đối tượng JSON.

{

"\_id": "\_id",

"name": "name",

"price": "price",

"status": "status",

"gallery": [

{"image\_path": "image\_path"},

{"image\_path": "image\_path2"},

]

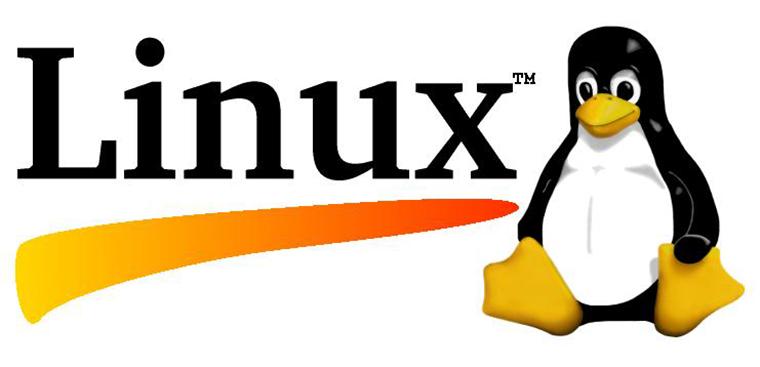
}

MongoDB hỗ trợ các kiểu dữ liệu sau:

* **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.
* **Code:** Kiểu dữ liệu này được sử dụng để lưu giữ JavaScrip code vào trong Document.
* **Regular expression:** Kiểu dữ liệu này được sử dụng để lưu giữ Regular Expresion.

**Nền tảng và ngôn ngữ hỗ trợ**

MongoDB là một cơ sở dữ liệu NoSQL hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.

RDBMS (là viết tắt của Relational Database Management System có nghĩa là hệ quản trị cơ sỡ dữ liệu quan hệ)

\*\*Schema:

* Schema là khái niệm mới có trong Sql server 2005 trở lên
* Schema mặc định của các object trong database là dbo
* Có thể hiểu schema là một namespace đối với các đối tượng database
* Một schema định nghĩa một ranh giới mà trong đó tất cả các tên là duy nhất
* Bên trong một database, tên của schema là duy nhất, luôn có tên với đường đầy đủ dạng server.database.schema.object

## Tại sao là MongoDB?

Như bạn biến, RDBMS lưu trữ dữ liệu trong các định dạng bảng và sử dụng ngôn ngữ truy vấn cấu trúc (SQL) để truy vấn cơ sở dữ liệu. RDBMS cũng định nghĩa sẵn các schema theo yêu cầu và tập hợp các quy tắc để định nghĩa quan hệ giữa các trường trong bảng.

Nhưng MongoDB lưu trữ dữ liệu trong các tài liệu thay vì các bảng. Bạn có thể thay đổi cấu trúc các bản ghi (được gọi là document trong MongoDB) đơn giản chỉ cần thêm mới các trường hoặc xóa các trường có sẵn. Khả năng này của MongoDB giúp bạn trình bày các quan hệ dạng thứ bậc, để lưu trữ mảng, và các cấu trúc phức tạp khác một cách dễ dàng. MongoDB cung cấp hiệu suất cao, tính sẵn có cao và dễ dàng mở rộng.

Tóm lại:

**MongoDB có ưu điểm là:**

#### Open Source:

* MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn

#### ****Hiệu năng cao:****

* Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).
* Thử nghiệm cho thấy tốc độ insert, tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL ([So sánh hiệu suất của MongoDB với MySQL](https://quantrimang.com/so-sanh-hieu-suat-hoat-dong-cua-mongodb-va-sql-server-2008-84584))

Tại sao MongoDB có hiệu năng cao như thế? có các lý do sau:

* MongoDB lưu dữ liệu dạng JSON, khi bạn insert nhiều đối tượng thì nó sẽ là insert một mảng JSON gần như với trường hợp insert 1 đối tượng
* Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong RDBMS, khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan như trong RDBMS.
* Dữ liệu trong MongoDB được [đánh chỉ mục](https://stackjava.com/mongodb/danh-chi-muc-la-gi-danh-chi-muc-trong-mongodb.html) ([đánh index](https://stackjava.com/mongodb/danh-chi-muc-la-gi-danh-chi-muc-trong-mongodb.html)) nên khi truy vấn nó sẽ tìm rất nhanh.
* Khi thực hiện insert, find… MongoDB sẽ khóa các thao tác khác lại, ví dụ khi nó thực hiện find(), trong quá trình find mà có thêm thao tác insert, update thì nó sẽ dừng hết lại để chờ find() xong đã.

#### ****Dữ liệu linh hoạt:****

* MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng field, kiểu dữ liệu… bạn có thể insert thoải mái dữ liệu mà mình muốn.

#### ****Là Rich Query Language:****

* MongoDB là một rich query language tức là nó có sẵn các method để thực hiện create, read, update, delete dữ liệu (CRUD)

#### ****Tính sẵn có:****

* MongoDB hỗ trợ [replica set](https://docs.mongodb.com/manual/replication/) nhằm đảm bảo việc sao lưu và khôi phục dữ liệu

#### ****Khả năng mở rộng Horizontal Scalability:****

* Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

\*\*Một trong nhưng lợi ích chính của SQL là khả năng kết hợp dữ liệu từ hai hay nhiều bảng lại với nhau. Việc kết hợp các bảng lại như vậy gọi là JOIN

**NHược điểm:**

* MongoDB không có các tính chất ràng buộc như trong RDBMS –> dễ bị làm sai dữ liệu
* Không hỗ trợ join giống như RDBMS nên khi viết function join trong code ta phải làm bằng tay khiến cho tốc độ truy vấn bị giảm.
* Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên sẽ bị dữ thừa dữ liệu (trong RDBMS thì ta chỉ cần lưu 1 bản ghi rồi các bản ghi khác tham chiếu tới còn trong MongoDB thì không)
* Bị giới hạn kích thước bản ghi: mỗi document không được có kích thước > 16Mb và không mức độ các document con trong 1 document không được > 100

so sánh sự khác nhau dữ MongoDB và RDBMS

|  |  |
| --- | --- |
| **SQL DB** | **MongoDB** |
| Table | Collection |
| Row | Document |
| Column | Field |
| Joins | Embeded document, linking |
| Primary key | Primary key |

**Khi nào thì nên xài MongoDB**

MongoDB dùng cho các hệ thống:

Hệ thống realtime (thời gian thực) yêu cầu phản hồi nhanh

Các hệ thống bigdata với yêu cầu truy vấn nhanh.

Các hệ thống có tần suất write/insert lớn

Sử dụng làm search engine.

**Các truy vấn trong mongodb:**

|  |  |  |
| --- | --- | --- |
| **Câu Lệnh** | **SQL** | **MongoDB** |
| **Create Table** | CREATE TABLE people (id MEDIUMINT NOT NULL AUTO\_INCREMENT, user\_id Varchar(30), age Number, status char(1), PRIMARY KEY (id)) | db.people.insertOne({User\_id: “abc123”, Age: 55, Status: “A”}) |
| **Drop Table** | DROP TABLE people | dp.people.drop() |
| **Insert records into table** | INSERT INTO people(user\_id, age, status) VALUES ("bcd001", 45, "A") | db.people.insertOne( { user\_id: "bcd001", age: 45, status: "A" }) |
| **Select** | SELECT \*FROM people | SELECT \*FROM people |
|  | SELECT id,user\_id, status FROM people | db.people.find( { }, { user\_id: 1, status: 1 } ) |
|  | SELECT \* FROM people WHERE status = "A" | db.people.find( { status: "A" } ) |
|  | SELECT \* FROM people WHERE status = "A" AND age = 50 | db.people.find( { status: "A", age: 50 } ) |
|  | SELECT \* FROM people WHERE status = "A" OR age = 50 | db.people.find( { $or: [ { status: "A" } , { age: 50 } ] } ) |
|  | SELECT \* FROM people WHERE user\_id like "%bc%" | db.people.find( { user\_id: /bc/ } )  db.people.find( { user\_id: { $regex: /bc/ } } ) |
|  | SELECT COUNT(user\_id) FROM people | db.people.count( { user\_id: { $exists: true } } )  db.people.find( { user\_id: { $exists: true } } ).count() |
| **Update records** | UPDATE people SET status = "C" WHERE age > 25 | db.people.updateMany( { age: { $gt: 25 } }, { $set: { status: "C" } } ) |
|  | UPDATE people SET age = age + 3 WHERE status = "A" | db.people.updateMany( { status: "A" } , { $inc: { age: 3 } } ) |
| **Delete records** | DELETE FROM people WHERE status = "D" | db.people.deleteMany( { status: "D" } ) |
|  | DELETE FROM people | db.people.deleteMany({}) |