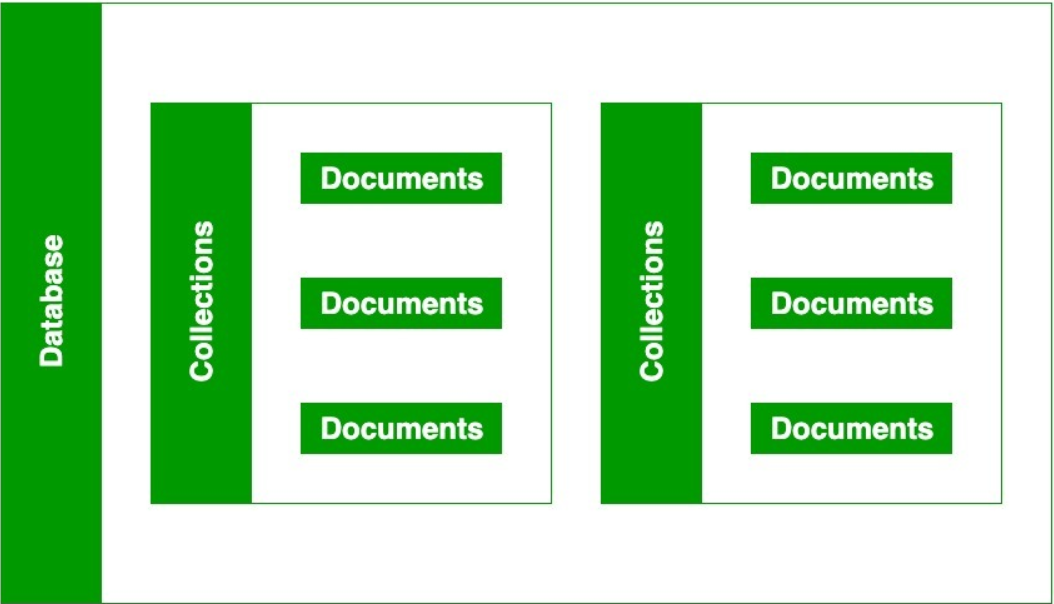
1. SECTION1: Introduction to MongoDB
2. Giới thiệu về NoSQL

* Là hệ quản trị dữ CSDL được thiết kế để xử lý và lưu trữ khối lượng lớn của các loại dữ liệu không có cấu trúc hoặc bán cấu trúc
* Không giống RDBMS sử dụng các bảng có lược đồ dể xác định trước cấu trúc dữ liệu cần lưu , NoSQL sử dung các data Model có thể đáp ứng việc thay đổi cấu trúc dữ liệu và có khả năng mở rộng theo chiều ngang để xử lú khối lượng dữ liệu lớn.
* NoSQL DB được chia thành 4 loại chính:
  + Document DB: lưu trữ các dữ liệu dạng văn bản bán cấu trúc (semi-structed document) như JSON hoặc XML, có thể sử dụng ngôn ngữ truy vấn hướng văn bản (Document -Oriented query language). Ví dụ MongoDB, CouchBase…
  + Key-Value Store: lưu trữ dạng key-value, được tối ưu cho các thao tác đọc/ghi trở nên nhanh và đơn giản, ví dụ redis, DynamoDB
  + Column-family store: Các cơ sở dữ liệu này lưu trữ dữ liệu dưới dạng các họ cột, là các tập hợp các cột được coi là một thực thể duy nhất. Chúng được tối ưu hóa để truy vấn nhanh và hiệu quả lượng dữ liệu lớn, ví dụ Apache Casandra, Hbase
  + Graph DataBase: Lưu trữ dữ liệu như là các node(entiy) và các edge(relationship), được thiết kế để xử lý quan hệ phức tạp giữ các dữ liệu, ví dụ NEO4j, Amazon Nep
* Sử dụng:
  + Sử dụng với các ứng dụng có khối lượng dữ liệu lớn, cần xử lý và phân tích theo thời gian thực
  + Không thích hợp sử dụng cho các ứng dụng cần dòi hỏi tính nhát quán của dữ liệu (thái mà dữ liệu trong hệ thống là chính xác, đúng đắn, và đồng nhất trên tất cả các nút (nodes) hoặc phiên bản (replicas)) hoặc đảm bảo giao dịch được thực thi một cách đang tin cậy và nhất quán (ACID)
* Đặc điểm quan trọng của NOSQL:
  + Dynamic schema:

1. Cơ bản về MongoDB
2. MongoDB – Database, Collection, and Document



* Database: chứa 1 tập hợp các document
  + Hiển thị DB:  *show db*
* Đặt tên cho DB:
  + Tên DB là case insensitive, tuy nhiên không nên đặt tên các BD giống nhau và chỉ khác ở kí tự in hoa, in thường
  + Không chứa các kí tự: /\. "$\*:|?
  + Không được chứa kí tự null
  + Không thể rỗng
  + Không được vượt quá 64 ký tự
* Collection: Giống như các bảng trong RDBMS chứ dữ liệu dưới dạng document , 1 DB có thể có nhiều collection
  + Schemaless: trong 1 Collection các document có thể có lược đồ khác nhau
  + Đặt tên cho collection:
    - Phải bắt đầu với ký tự \_ hoặc ký tự chữ (a-z, A-Z)
    - Không được bắt đầu bằng số, không được chứa: $, rỗng, null và không được bắt đầu với *system.*
    - Độ dài tối đa là 120 byte
  + Tạo collecion: ***db.collection\_name.insertOne({..})***
* Document : bản ghi dữ liệu được lưu trữ dưới dạng BSON document (binary representation of JSON documents)
  + Lưu trữ dưới dạng cặp field-value hoặc key-value trong đó value là dữ liệu thuộc kiểu BSON
  + Quy tắc đặt tên:
    - Field phải là string
    - \_id được sử dụng như là một primary key. Giá trị của trường này phải là duy nhất, không được thay đổi và có thể là bất kỳ kiểu dữ liệu nào ngoài kiểu mảng
  + Kích thước tối đa của document là 16MB để đảm bảo 1 document không chứa quá nhiều RAM hoặc băng thông. Nếu có kích thước lớn hơn thì MongoDB cung cấp API GridFS để lưu trữ
* Lưu ý quan trọng:
  + Một document có thể chứa nhiều trường trùng lặp
  + MongoDB luôn lưu trữ thứ tự của các trường ngoại trừ \_id (luôn đứng đầu). Việc đổi tên các trường có thể thay đổi thứ tự hiện thị của nó.
  + \_id: có thể thiết lập bởi người dùng hoặc hệ thống nhưng nó phải là duy nhât, nếu được thiết lập bởi hệ thống, nó sẽ tạo 1 ObjectId cho trường \_id này)

1. MongoDB Cursor

* Là một con trỏ tham chiếu đến document của 1 collection và được trả về hởi hàm find()
* Duyệt thủ công bằng con trỏ trong MongoDB
  + Gán cursor cho từ khóa var hoặc 1 biến javaScript: *var name = db.collection\_name.find().* Lưu ý nếu không được gán thì con trỏ có thể duyệt tối da 20 document . Nếu con trỏ inactive trong 10p thì MongoDB sẽ tự động đóng con trỏ đó.
  + Sử dụng hàm next()

var mycursor = db.student.find({studentId:{$gt:1}});  
> while(mycursor.hasNext()){  
... print(tojson(mycursor.next()));  
... }

* + Sử dụng hàm forEach()
* Các hàm của cursor:

1. Kiểu dữ liệu trong MongoDB

* Sử dụng BSON (Binary JSON) để lưu trữ dữ liệu
* **String**: định dạng UTF-8
* **Integer:** kiểu số nguyên 4byte không dấu hoặc 8byte không dấu
* **Double:** Kiểu dấu phây động
* **Boolean:** chỉ có 2 giá trị true hoặc false
* **Null:** giá trị null
* **Array:** 1 mảng các giá trị cùng kiểu nằm trong []
* **Object:** Lưu trữ 1 document,
* **ObjectId:** khi tạo mới 1 document thì MongoDB sẽ tự động tạo một ObjectId duy nhất và được gắn vào trường \_id, có độ dài 12 ký tự:
  + 4 byte timestamp
  + 3 byte machineId , 2 byte processID
  + 3 byte counter
* **Undefined:** kiểu giá trị không xác định
* **Kiểu dữ liệu nhị phân:**
* **Date:**

1. MongoDB Query

* Cho phép truy vấn dữ liệu trong MongoDB
* Tìm kiếm 1 bản ghi : *db.collection\_name.findOne ()*
* Select tất các bản ghi : *db.collection\_name.find().pretty()*
* Điều kiện bằng: sử dụng ký tự ($eq): *db.article.find({author:{$eq:"devil"}}).pretty()*
* Điều kiện lớn hơn hoặc nhỏ hơn:
  + Lớn hơn ($gte): b.collection\_name.find({< key > : {$gte : < value >}})
  + Nhỏ hơn ($lte): db.collection\_name.find({< key > : {$lte : < value >}})
* Kiểm tra điều kiện có tồn tại : db.article.find({time:{$exists:"true"}}).pretty()

1. BSON

* Ưu điểm của BSON:
  + Có thể paser dễ dàng và nhanh chóng vì nó hỗ trợ mã hóa sẵn thông tin kiểu và độ dài
  + Có tính di động cao và nhẹ, thích hợp cho việc truyền tải dữ liệu
  + Tốn ít không gian và scan nhanh hơn so với JSON
  + Hỗ trợ nhiều kiểu dữ liệu
* Nhược điểm: chứa thêm các thông tin bổ sung (độ dài, kiểu dữ liệu) dẫn tới tăng kích thước

1. CRUD TRONG MONGODB
2. CRUD