* Đối mới mongoDB thì ta Lớn nhấn là cluster trong cluster thì có các database, trong database thì có collection. Và collection chính là những object mình cần lưu như employee, user … Khi ta tạo object mới thì không cần id vì mongoDB sẽ tự tạo 1 cái duy nhất cho mỗi object trong collection.
* Sử dụng powershell để học

+ mongsh : để biết thông tin về mongoshell và mongDB

1. Insert document phải là 1 json (mảng, object)

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

* show databases: đưa ra những database
* user nameDatabase: vào database đó

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, Đồ họa

Mô tả được tạo tự động

* show collections
* khi vào được DB thì dùng câu lệnh **db.nameCollection.insertOne()** để thêm 1 object vào collection
* Để thêm nhiều object cùng 1 lúc thì dùng **db.nameCollection.insertMany([]),** truyền vào 1 array object
* Đối với mongoDB khi ta thêm 1 object mới vào 1 collection chưa có sẵn thì nó sẽ tự tạo collection đó luôn

1. Find document

* Khi vào được database bookstore dùng câu lệnh **db.collection.find().** Nếu không có điều kiện gì thì nó sẽ show 20 record đầu tiên.
* **db.collection.find({key: value, key: value…})** => trả về những object có key: value trùng khớp**.**
* **db.collection.find({key: value, …}, {key: value, …})** gồm 2 phần ({điều kiện}, {projection}). Giải thích chi tiết:
  + Điều kiện tìm kiếm: Tìm tất cả các tài liệu trong collection books có trường key : value
  + Phép chiếu: Hiển thị chỉ các trường key và 1 or 0, ngoài ra, trường \_id cũng sẽ được hiển thị mặc định. Trong đó 1 là hiển thị 0 là không hiển thị
  + Kết quả: MongoDB sẽ trả về các tài liệu có key: value nhưng chỉ bao gồm các thuộc tính \_id, key : 1, trong kết quả trả về.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Vì object đầu tiên mình không điền gì nên nó trả về all nhưng chỉ hiển thị title còn \_id là mặc định
* **db.books.find({key: value}).count():** trả về số lượng kq được tìm thấy
* **db.books.find({key: value}).limit(number):** trả về số lượng kq được tìm thấy với number kết quả
* MongoDB Shell: Trả về 20 bản ghi đầu tiên khi sử dụng db.books.find() để giúp quản lý việc hiển thị dữ liệu dễ dàng hơn.
* Mongoose/Node.js: Trả về tất cả các bản ghi khớp với điều kiện tìm kiếm khi sử dụng find() mà không có giới hạn mặc định về số lượng bản ghi.
* **db.collection.findOne({key: value})** => trả về đối tượng đầu tiên phù hợp

1. Sort documents

* **db.books.find().sort({title: 1}):** Câu lệnh này sẽ trả về all object trong collection vì find ko có đc sau đó nó sẽ sắp xếp dựa vào title là acsending **(vì có value là 1, nếu value là -1 thì nó sẽ là decsending)** theo anphabetically, đương nhiên nếu mình cho đk vào find thì nó sẽ cho ra kết quả tìm kiếm sau đó sắp xếp.
* Ngoài ra mình có thể dùng kết hợp giữa find, sort and limit.
* **db.books.find().sort({title: 1}).limit(4):** Ví dụ câu lệnh này nó sẽ thực hiện find trả về all object trong collection, sau đó sắp xếp title theo ascending sau đó giới hạn lại hiển thị 4

1. Nested documents

* Documents lồng nhau thôi

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

1. Operators and complex queries

* **db.books.find({rating: {$gt: 9.5}}): gt (greater than)** câu lệnh này sẽ trả về rating > 9.5, tương tự **lt(less than)** thì nó sẽ trả về giá trị nhỏ hơn 9.5
* **db.books.find({rating: {$gte: 5.5}}): gte (greater than equal)** câu lệnh này trả về rating > 5.5 hoặc bằng 5.5, tương tự lts (less than equal) thì nó trả về giá trị nhỏ hơn hoặc bằng 5.5
* **db.books.find({rating: {$gte: 9}, author : "Thế Bảo"}):** mình có thể kết hợp như vầy, câu lệnh trên trả về rating >= 9 và author là Thế Bảo
* **db.books.find({$or: [{rating: 9.2}, {rating: 9}]}):** Câu lệnh này trả về object có rating là 9.2 hoặc 9 nhờ sử dụng $or nó sẽ trả về object nếu đúng 1 trong những đk trong mảng object
* **db.books.find({$or: [{pages: {$lt: 300}}, {pages: {$gt: 400}}]})**: ta có thể kết hợp như vầy, câu lệnh này trả về danh sách object có pages nhỏ hơn 300 hoặc pages lớn hơn 400, nếu ta .count() thì nó sẽ trả về số lượng

1. $in và $nin

* In và not in:
  + **db.books.find({rating: {$in: [5.2,8,9]}}):** trả về những rating nào có value là 5.2 or 8 or 9. Vậy in này nó sẽ trả về những object có giá trị nằm trong mảng giá trị của thuộc tính điều kiện
  + **db.books.find({rating: {$nin: [5.2,8,9]}}):** ngược lại với cái trên nó trả về những object mà có giá trị không nằm trong những giá trị của thuộc tính điều kiện

1. Querying Arrays

* **db.books.find({genres: "fantasy"})**: trong đó genres là 1 mảng, nếu ta dùng câu lệnh này thì nó sẽ trả về object có thuộc tính **genres**: chứa giá trị fantasy cho dù giá trị này nó nằm trong mảng hay nó chỉ là 1 string. Nhưng nếu mình dùng **db.books.find({genres: [“fantasy”]})**: thêm dấu [] thì nó trả về object **chỉ chứa** giá trị fantasy nằm trong mảng của thuộc tính genres
* **db.books.find({genres: {$all: ["fantasy", "sci-fi"]}})**: đây là câu lệnh để tìm những object nào chứa cả 2 giá fantasy và sci-fi **db.books.find({genres: "fantasy", "sci-fi"})** thì sẽ bị lỗi.
* **db.books.find({"reviews.name": "Ganon"}):** Câu lệnh này là để tìm giá

1. Deleting documents

* **db.books.deleteOne({\_id: ObjectId('66b8b38bba49b59c45947068'**)}): xóa 1 cái theo id,
* **db.books.deleteOne({ genres: 'fantasy'}):** tương tự mình cũng có thể dùng 1 đk xóa khác nhưng vì dùng deleteOne nên nó chỉ xóa object đầu tiên thỏa điều kiện
* **db.books.deleteMany({author: "Thế Bảo"}):** ở đây ta dùng deleteMany thì nó sẽ xóa all object thỏa điều kiện

1. updating documents

* **db.books.updateOne({ \_id: ObjectId('66b8b38bba49b59c45947068')}, {$set: {rating: 1, pages: 10}})**: ở đây mình update 1 object dựa vào \_id của nó và dùng từ khóa $set để updata 2 trường rating: new value, pages: new value
* **db.books.updateMany({author: "Thế Bảo"}, {$set: {author: "Thế Bảo mới update nè"}}):** ở đây mình dùng update nhiều object dựa vào author là Thế Bảo vế sau tương tự như ở trên
* **db.books.updateOne({\_id: ObjectId('66b76d75818aebda4329ba23')}, {$inc: {pages: 2}})**: ở đây mình cũng dùng update 1 object và dựa vào id nhưng mình không dùng từ khóa $set mà là $inc (increment), cái cần update ở đây là pages với value là 2 thì sau khi update xong thì giá trị của pages này sẽ tăng lên 2. Đường nhiên vì là tăng giá trị lên nên chỉ có tác dụng với numeric type còn nếu khác bị lỗi. Tương tự mình dùng inc mà value âm thì nó có nghĩa là giảm
* **db.books.updateOne({\_id: ObjectId('66b76d75818aebda4329ba23')}, {$pull: {genres: "tình cảm"}}):** Nếu mình dùng câu lệnh này với từ khóa $pull thì nó sẽ xóa (kéo) value “tình cảm” ra khỏi key genres của object có id kia
* **db.books.updateOne({\_id: ObjectId('66b76d75818aebda4329ba23')}, {$push: {genres: "tình cảm"}}):** Nếu mình dùng câu lệnh này với từ khóa $push ngược lại ở trên thì nó sẽ thêm (đẩy) value “tình cảm” vào key genres của object có id kia
* **db.books.updateOne({\_id: ObjectId('66b76d75818aebda4329ba23')}, {$push: {genres: {$each: ["1", "2"]}}}):** Dùng từ khóa $each : array thì sẽ thêm được nhiều giá trị cùng 1 lúc vào mảng genres
* **db.books.updateOne({\_id: ObjectId('66b76d75818aebda4329ba23')}, {$push: {genres: ["1", "2"]}}):** Còn nếu mình dùng như vầy không có từ khóa $each thì sẽ thêm 1 mảng mới là mảng con nằm trong mảng ganres

1. indexex
2. **Lợi ích**:
   * **Tăng hiệu suất**: Truy vấn nhanh hơn đáng kể khi có chỉ mục phù hợp.
   * **Tránh collection scan**: Giảm số lượng tài liệu MongoDB cần duyệt qua để tìm kết quả.
3. **Chi phí**:
   * **Tốn tài nguyên**: Chỉ mục tiêu tốn thêm dung lượng lưu trữ.
   * **Ảnh hưởng đến tốc độ ghi**: Các thao tác ghi (insert, update, delete) có thể chậm hơn vì MongoDB phải cập nhật chỉ mục mỗi khi tài liệu thay đổi.

**Tổng kết:**

* **Chỉ mục (Indexes)** trong MongoDB là một công cụ mạnh mẽ giúp tăng tốc độ truy vấn.
* Có nhiều loại chỉ mục khác nhau như single field index, compound index, multikey index, text index, và hashed index, mỗi loại phù hợp với các loại truy vấn khác nhau.
* Mặc dù chỉ mục giúp tăng hiệu suất truy vấn, nhưng chúng cũng đi kèm với chi phí về dung lượng lưu trữ và có thể ảnh hưởng đến hiệu suất của các thao tác ghi.
* db.books.createIndex({rating: 8}): tạo chỉ mục cho rating
* db.books.getIndexes(): liệt kê all chỉ mục hiện tại