Tên: Mai Chiến Nô

Mssv: 21106521

**Tuần 8: Index**

**Bài 1.** Tìm hiểu vai trò và cách sử dụng index trong cơ sở dữ liệu

1Tạo một cơ sở dữ liệu highestmountains, với một collection peaks gồm các document chứa các thông tin: name, height, location, ascents



2. Dùng phương thức insertMany() method trong MongoDB shell để đồng thời tạo một collection tên peaks và chèn 5 documents



3. Thực hiện truy vấn tìm ngọn núi có độ cao >8700.



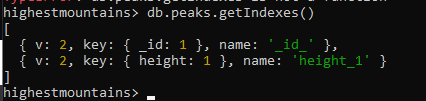
4. Dùng phương thức explain(executionStats) để xem thông tin về cách thực hiện truy vấn. Quan sát kết quả về số lần duyệt các document để tìm được kết quả.



🡪Số lần duyệt: 5(totalDocsExamined)

5. Tạo một chỉ mục (Single Field Index) trên trường height trong peaks collection sử dụng phương thức createIndex(). Viết câu lệnh xem có bao nhiêu chỉ mục đã được xác định trên peaks collection và trạng thái đó khác với trạng thái trước đó như thế nào.





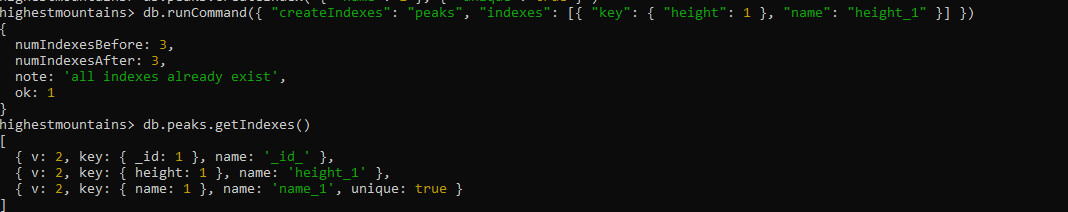


6. Thực hiện lại câu truy vấn tìm ngọn núi có độ cao >8700 dựa trên chỉ mục đã tạo. Dùng phương thức explain(executionStats) để xem thông tin về cách thực hiện truy vấn, so sánh với cách thực thi khi chưa có index.

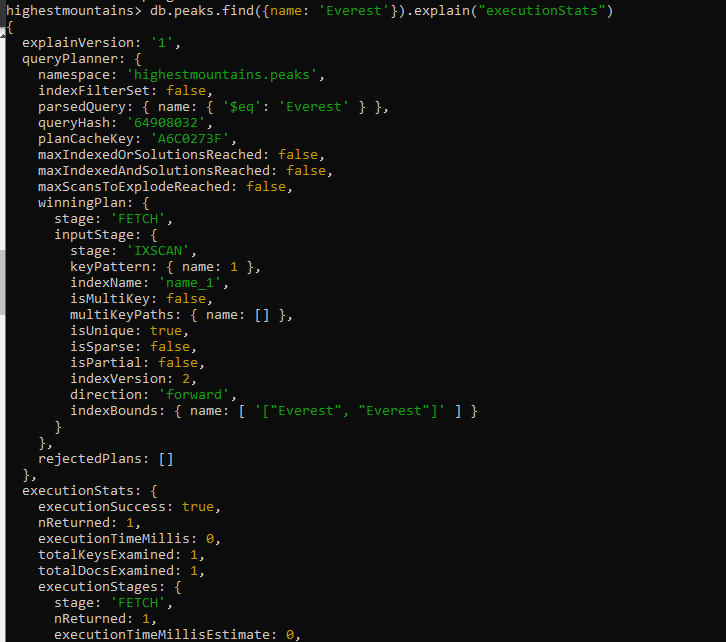


🡪Khác ở mục winningPlan và mục executionStages của executionStats

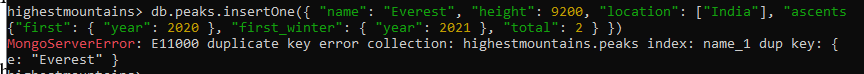
7. Tạo một chỉ mục (Unique Indexes) đảm bảo tên các đỉnh núi (trường Name) là duy nhất. Viết câu lệnh xem kết quả của Index vừa tạo. db.peaks.createIndex( { "name": 1 }, { "unique": true } ) Output { "createdCollectionAutomatically”: false, "numIndexesBefore" : 2, "numIndexesAfter" : 3, "ok": 1 }



8. Thực hiện câu lệnh truy vấn tìm đỉnh núi tên Everest kết hợp với phương thức explain ("executionStats") xem trạng thái thực thi của câu truy vấn trên Index vừa tạo.



9. Dùng phương thức insertOne() chèn thêm một document mới với tên ngọn núi là Everest. Nhận xét kết quả của câu lệnh.



🡪Không insert được bởi vì mình đã tạo chỉ mục với name:Everest là duy nhất

10.Tạo một Index trên Embedded Field

a. Thực hiện truy vấn tìm những ngọn núi có total>300, kết quả sắp xếp theo thứ tự giảm dần.



b. Thực hiện lại truy vấn trên nhưng kết hợp với phương thức explain("executionStats") để xem tình trạng thực thi của câu truy vấn. (giá trị COLLSCAN trong phần này của đầu ra cho biết, MongoDB đã sử dụng quét toàn bộ bộ sưu tập và duyệt qua tất cả các tài liệu từ bộ sưu tập đỉnh để so sánh chúng với các điều kiện truy vấn)



c. Tạo một index trên total field bên trong ascents document (Embedded Field). Viết câu lệnh xem index vừa tạo.



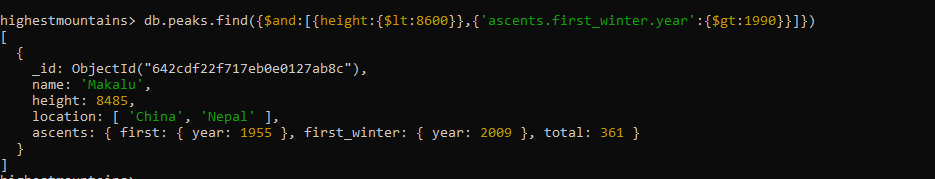
d. Thực thi lại câu truy vấn tìm những ngọn núi có Total>300, kết quả sắp xếp theo thứ tự giảm dần. Viết câu lệnh để kiểm tra xem chỉ mục có giúp MongoDB tránh thực hiện quét toàn bộ collection hay không? (IXSCAN được sử dụng đối với chỉ mục ascents.total\_-1 mới được tạo và chỉ có bốn tài liệu đã được kiểm tra. Đây là cùng một số lượng tài liệu được trả lại và kiểm tra trong chỉ mục, vì vậy không có tài liệu bổ sung nào được truy xuất để hoàn thành truy vấn)



🡪Quét 4 lần so với 5 lần khi chưa tạo chỉ mục

11.Compound Field Index: sử dụng các chỉ mục khi thực hiện các truy vấn trên nhiều trường.

a. Hãy tìm những ngọn núi có chiều cao dưới 8600 mét có winter ascent đầu tiên xảy ra sau năm 1990



b. Thực hiện lại truy vấn trên, thêm phương thức explaion("executionStats") để xem cách MongoDB thực hiện truy vấn.



✓ MongoDB đã sử dụng single field index trên trường height để thu hẹp kết quả từ 5 xuống 3, nhưng sau đó phải quét các document còn lại để kiểm tra ngày đi lên (winter ascent) trong mùa đông đầu tiên.

✓ Mặc dù không có chỉ mục nào có thể ảnh hưởng đến ngày đi lên (winter ascent) trong mùa đông đầu tiên, MongoDB đã sử dụng chỉ mục được tạo trước đó thay vì thực hiện quét toàn bộ collection.

c. Tạo một index dựa trên 2 trường: “height” giảm dần và “ascents.first\_winter.year" tăng dần. Viết câu lệnh xem kết quả index vừa tạo.



d. Thực hiện lại truy vấn tìm những ngọn núi có height< 8600 mét có winter ascent đầu tiên (ascents.first\_winter.year) xảy ra sau năm 1990, kết quả sắp xếp theo chiều giảm dần của trường height. Dùng phương thức explain("executionStats") xem cách Mongodb thực hiện truy vấn, phân tích và nhận xét kết quả.



🡪Chỉ mục quét 3 lần để ra kết quả

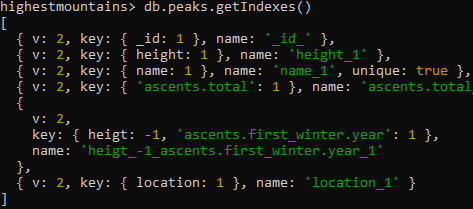
12.Multi-key Index: MongoDB lưu trữ giá trị của trường trực tiếp dưới dạng khóa chỉ mục (index key), giúp chỉ mục có thể duyệt nhanh chóng. Bước này phác thảo cách MongoDB hoạt động khi trường được sử dụng để tạo chỉ mục là trường lưu trữ nhiều giá trị, chẳng hạn như một mảng

a. Viết câu lệnh tìm tất cả các ngọn núi trong collection nằm ở Nepal. Nhận xét: Mỗi đỉnh bao trùm nhiều quốc gia như được biểu thị bằng các trường location, là một mảng gồm nhiều giá trị. Do không có sẵn chỉ mục mở rộng trường location, MongoDB hiện thực hiện quét toàn bộ bộ collection để thực hiện truy vấn.



b. Tạo một index mới cho trường location, viết câu lệnh xem kết quả index vừa tạo.





c. Thực hiện lại câu truy vấn tìm các ngọn núi ở Nepal, kết hợp phương thức explain("executionStats") để xem cách MogoDB thực thi.



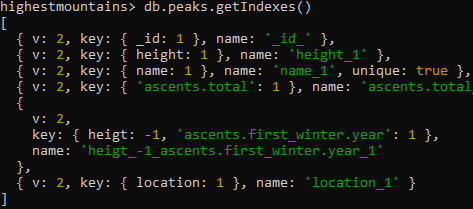
Nhận xét kết quả

• MongoDB đã sử dụng quét chỉ mục ("stage" : "IXSCAN",) làm chiến lược, đề cập đến chỉ mục location\_1 mới được tạo

• Thuộc tính isMultiKey: true. MongoDB tự động tạo một multi-key index cho trường location.

• Đối với document có trường location lưu trữ array [ "China", "Nepal"], hai index entries riêng biệt sẽ xuất hiện cho cùng một tài liệu, một cho China và một cho Nepal. Bằng cách này, MongoDB có thể sử dụng chỉ mục một cách hiệu quả ngay cả khi truy vấn yêu cầu khớp một phần với nội dung mảng.

13.Liệt kê tất cả các index trong collection Peaks, sử dụng phương thức getIndexes()



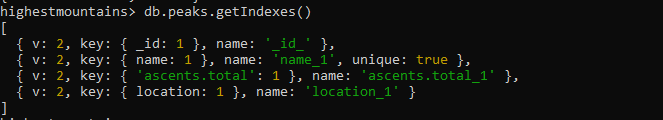
14.Viết câu lệnh xóa index height\_1, sử dụng phương thức dropIndex(), dựa trên tên trường tạo chỉ mục.



15.Viết câu lệnh xóa index dựa trên tên của index ascents.first\_winter.year\_1\_height\_-1.



16.Viết câu lệnh liệt kê tất cả các index còn lại trong collection Peaks



**Bài 2.** Sử dụng collection restaurants thực hiện các thao tác sau đây:

1. Liệt kê danh sách restaurant có cuisine là “Hamburger”, có dùng phương thức explain() để xem thông tin về cách thực hiện truy vấn. Quan xát kết quả về số lần duyệt, thời gian, cách duyệt,…

🡪Không có cuisine là Hamburger, chọn cuisine: Delicatessen

Text

Description automatically generated

Text

Description automatically generated

🡪Số lần duyệt:3772, thời gian: 0

2. Tạo một index trên field cuisine theo thứ tự tăng dần.

Text

Description automatically generated

3. Liệt kê danh sách restaurant có cuisine là “Hamburger”, có dùng phương thức explain() để xem thông tin về cách thực hiện truy vấn. Quan xát kết quả về số lần duyệt, thời gian, cách duyệt,… So sánh với kết quả câu 1.

Text

Description automatically generated



🡪Số lần duyệt: 78, thời gian:0 🡪 Nhanh hơn rất nhiều so với câu trên

4. Liệt kê danh sách các index hiện có trong restaurants, cho biết tên của index đó db.restaurant.getIndex()

Text

Description automatically generated

5. Xóa index hiện có trong collection restaurants

Text

Description automatically generated

🡪 Xóa tất cả chỉ mục trừ chỉ mục \_id không thể xóa

6. Tạo lại index trên trường cuisine theo thứ tự giảm dần và đặt tên là cusine\_inc. Viết câu lệnh xem lại index vừa tạo.





7. Liệt kê danh sách các restaurant có borough là “Brooklyn” khi chưa được lập chỉ mục. Sau đó tạo chỉ mục trên borough, liệt kê lại danh sách và so sánh kết quả.

Text

Description automatically generated

-Khi chưa tạo chỉ mục

Text

Description automatically generated

-Khi tạo chỉ mục:

Text

Description automatically generated

🡪Số lần duyệt: 684, giảm đi 3 lần

8. Tạo chỉ mục compound field index trên trường cuisine (giảm dần) và name (tăng dần). Xem lại các index hiện có restaurants.

Text

Description automatically generated

9. Thực hiện liệt kê danh sách các restaurants có cuisine là “Hamburger” và name bắt đầu là “Wil”.

Text

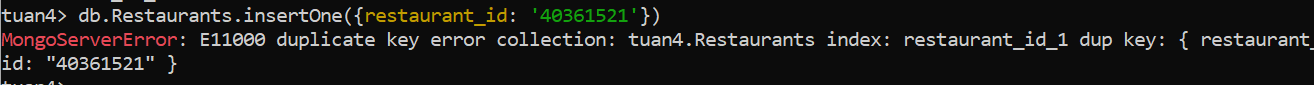
Description automatically generated

10. Cho biết hiện tại trong restaurants có những index nào

Text

Description automatically generated

11. Tạo một Unique Index trên trường restaurant\_id, sau đó chèn một document mới có restaurant\_id trùng với restaurant\_id của một restaurant đã có, cho nhận xét.



🡪Không chèn được do chỉ mục Unique Index chỉ cho phép 1 restaurant\_id duy nhất.

12. Thực hiện tìm các restaurants có restaurant\_id là 40361521, kết hợp phương thức explain(“executionStats”) xem trạng thái thực thi của câu truy vấn.



🡪Chỉ duyệt 1 lần duy nhất

13. Thực hiện ẩn Unique Index vừa tạo ở trên, sau đó chèn một document mới có restaurant\_id trùng với restaurant\_id của một restaurant đã có (ví dụ 40361521). Cho nhận xét.

14. Tạo một Embedded Field Index trên trường address.zipcode theo thứ tự tăng dần.

Text

Description automatically generated

15. Thực hiện tìm các restaurants có zipcode lớn hơn 11000, kết hợp kết hợp phương thức explain(“executionStats”) xem trạng thái thực thi của câu truy vấn.

Text

Description automatically generated

16. Thực hiện xóa tất cả các Index hiện có trong restaurants.

Text

Description automatically generated

17. Tạo index trên trường Name (tăng dần) và trường address.zipcode (tăng dần).



18. Thực hiện liệt kê danh sách các restaurants có zipcode lớn hơn 11000 và xếp theo thức tự tăng dần của tên, kết hợp phương thức explain(“executionStats”) xem trạng thái thực thi của câu truy vấn.

Text

Description automatically generated

19. Thực hiện tạo Multi-key Index trên trường grades tăng dần.

A picture containing text

Description automatically generated

20. Liệt kê danh sách các restaurants có grades.grade là “A”, kết hợp kết hợp phương thức explain(“executionStats”) xem trạng thái thực thi của câu truy vấn.

Text

Description automatically generated

21. Thực hiện tạo Multi-key Index trên trường grades.score tăng dần.

Text

Description automatically generated

22. Thực hiện liệt kê danh sách các restaurant có grades.score từ 5 đến 10, kết hợp kết hợp phương thức explain(“executionStats”) xem trạng thái thực thi của câu truy vấn.

Text

Description automatically generated

Text

Description automatically generated