

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

—o0o—



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
Document Store (MongoDB, CouchDB, SimpleDB)

Giảng viên hướng dẫn: Thầy Trần Quang

Sinh viên thực hiện:

Họ và tên	MSSV
Trương Ngọc Anh	1410141
Lê Nguyễn Minh Trí	1414207
Nguyễn Quốc Kim Hoàng	141xxxx
Nguyễn Khánh Bình	141xxxx

Mục lục

1	Giới thiệu Document store và MongoDB	4
1.1	4
1.2	4
2	Thiết kế cơ sở dữ liệu với MongoDB	5
2.1	Index trong MongoDB	5
2.1.1	Index là gì?	5
2.1.2	Các loại index	6
2.2	Các phép toán	9
2.2.1	Aggregation framework	9
3	Sharding	10
4	Các cơ sở dữ liệu Document store	11
4.1	Tổng quan về CouchDB	11
4.2	Tổng quan về SimpleDB	11
4.3	So sánh	11

Danh sách hình vẽ

2.1	Câu lệnh tạo index hỗn hợp các khóa	6
2.2	Kết quả	7
2.3	tạo chỉ mục duy nhất kết hợp với option sparse	8

Danh sách bảng

Chương 1

Giới thiệu Document store và MongoDB

1.1

1.2

Chương 2

Thiết kế cơ sở dữ liệu với MongoDB

2.1 Index trong MongoDB

2.1.1 Index là gì?

Trong cơ sở dữ liệu, ta đã biết Index(chỉ mục) là một file cấu trúc lưu trữ trên ổ cứng tương ứng với một table hoặc một view nào đó trong cơ sở dữ liệu nhằm mục đích cải thiện tốc độ truy xuất dữ liệu từ table hoặc view đó.

Ở mục này, chúng ta sẽ tìm hiểu cách đánh index cũng như các loại index có trong MongoDB.

2.1.2 Các loại index

Compound index (Chỉ mục hỗn hợp các khóa)

Đây là chỉ mục được đánh trên nhiều trường khác nhau trong database. Loại index này hữu ích khi chúng ta thực hiện câu truy vấn trên đòi hỏi nhiều trường trên dữ liệu. Ví dụ câu lệnh sau đây giúp tạo chỉ mục hỗn hợp các khóa trên MongoDB.

Trong câu lệnh trên, ta giả sử có một thư mục data có nhiều trường trong

```
> db.users.ensureIndex({"age" : 1, "username" : 1})
```

Hình 2.1: Câu lệnh tạo index hỗn hợp các khóa

đó có trường username và age. Câu lệnh trên sẽ tạo ra chỉ mục đánh trên hai trường username và age đó. Số ứng với mỗi khóa trong câu lệnh chỉ mục trên thể hiện hướng của dữ liệu trên trường đó được sắp xếp như thế nào, với 1 là tăng dần, -1 là giảm dần. Trong ví dụ này thì age và username đều được đánh chỉ mục tăng dần.

Nếu ta có một tập dữ liệu gồm mà mỗi document trong dữ liệu đó bao gồm username có các giá trị như *user01*, *user02*, *user03*... và age bao gồm các giá trị nguyên như *1*, *2*, *3*, *4*.... Thì chỉ mục sẽ có hình dạng như thế này Mỗi

```

[0, "user100309"] -> 0x0c965148
[0, "user100334"] -> 0xf51f818e
[0, "user100479"] -> 0x0fd7934
...
[0, "user99985" ] -> 0xd246648f
[1, "user100156"] -> 0xf78d5bdd
[1, "user100187"] -> 0x68ab28bd
[1, "user100192"] -> 0x5c7fb621
...
[1, "user999920"] -> 0x67ded4b7
[2, "user100141"] -> 0x3996dd46
[2, "user100149"] -> 0xfce68412
[2, "user100223"] -> 0x91106e23
...

```

Hình 2.2: Kết quả

index entry sẽ chứa lần lượt các giá trị của các trường là age ứng với giá trị nguyên và trường username ứng với các username *user1*, *user2*,.... Và trong file index ấy do câu lệnh tạo index phía trên với trường age đứng trước tiên nên các entry được xếp tăng dần theo trường age và tiếp đến là tăng dần theo trường username có nghĩa là trước tiên chúng sắp xếp tăng dần theo trường age là số tuổi, tiếp đó với các user có cùng số tuổi thì sẽ sắp xếp tăng dần theo username. Bên cạnh đó, mỗi entry còn trỏ đến địa chỉ (*được biểu diễn bằng chuỗi Hex*) của document được lưu trên đĩa.

Unique index (Chỉ mục duy nhất)

Với các hệ cơ sở dữ liệu như oracle, mySQL,... ta có primary index là dạng index đối với các trường có các giá trị không trùng nhau(unique). Với mongoDB, ta có Unique index hay còn gọi là chỉ mục duy nhất, các trường được đánh index này không được trùng nhau. Trong collection của mongoDB, trường "id" sẽ được tự động đánh là unique index.

- **Compound unique index:** đây là sự kết hợp chỉ mục duy nhất với chỉ mục hỗn hợp các khóa. Cụ thể là, nếu có hai trường như name và address được đánh compound unique index thì một trong hai trường có thể có giá trị trùng nhau giữa các document nhưng không có document nào mà giống nhau ở cả hai trường đã được đánh index.
- **Các giá trị null:** trong quá trình insert dữ liệu, nếu một trường không có giá trị nào khi insert thì mongoDB sẽ mặc định giá trị đó là null. Nhưng nếu trường nói trên được đánh chỉ mục duy nhất thì nếu có thêm một trường bị gán giá trị null thì sẽ báo lỗi insert do trùng giá trị null.

Sparse index (Chỉ mục thưa)

Ở phần trên đề cập đến việc những trường được đánh chỉ mục duy nhất nhưng lại xuất hiện giá trị null. Trong những trường hợp có một trường nào đó trong collection mà có hoặc không cần insert giá trị nhưng bắt buộc trường đó là unique thì khi đó ta tạo chỉ mục duy nhất kết hợp với option sparse(thưa).

Ví dụ dưới đây là câu lệnh thực hiện việc tạo ra chỉ mục duy nhất trên trường email mà trường này không nhất thiết phải có giá trị. Các giá trị cấu hình cho unique và sparse lần lượt là "1" và "true":

```
> db.ensureIndex({"email" : 1}, {"unique" : true, "sparse" : true})
```

Hình 2.3: tạo chỉ mục duy nhất kết hợp với option sparse

2.2 Các phép toán

Sau khi data đã được lưu và thực hiện tạo index cần thiết trên mongoDB, trong công việc chúng ta có thể cần thực hiện nhiều tác vụ như phân tích theo nhiều cách khác nhau. Chính vì vậy mà mongoDB đã cung cấp một số công cụ để thực hiện các việc đó.

2.2.1 Aggregation framework

Đây là công cụ giúp tạo ra một đường dẫn nhằm mục đích xử lý dòng dữ liệu trong các document thông qua việc kết hợp nhiều tác vụ được xây dựng sẵn như: phép lọc (filter), phép chiếu(projecting), sắp xếp (sorting), nhóm(grouping),...

Các phép toán:

- \$match

Chương 3

Sharding

Cái này tự chỉnh lại tiêu đề cho hợp lý

Chương 4

Các cơ sở dữ liệu Document store

4.1 Tổng quan về CouchDB

4.2 Tổng quan về SimpleDB

4.3 So sánh

Tài liệu tham khảo

- [1] Rémi Cadène, Nicolas Thome, and Matthieu Cord. Master's Thesis : Deep Learning for Visual Recognition. 2016.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. Chapter 6:Deep Feedforward Networks.
- [3] Andrej Karpathy. CS231n: Convolutional Neural Networks for Visual Recognition. module 1 - Neural Networks and module 2 - Convolutional Neural Network, 2016.
- [4] Andrej Karpathy. CS231n: Convolutional Neural Networks for Visual Recognition. convolutional layers, 2016.
- [5] Andrej Karpathy. CS231n: Convolutional Neural Networks for Visual Recognition. transfer learning, 2016.
- [6] Andrej Karpathy. CS231n: Convolutional Neural Networks for Visual Recognition. CNN overview, 2016.
- [7] Andrej Karpathy. CS231n: Convolutional Neural Networks for Visual Recognition. CNN architecture, 2016.

- [8] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. 2013.
- [9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed and Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. 2014.
- [10] tensorflow.org. Image retraining.
- [11] tensorflow.org. Tensorflow programmer guide.
- [12] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 2013.