Map Reduce in MongoDB

In MongoDB Documentation, Map-Reduce is a data processing system that condenses a large amount of data into useful overall results. MongoDB uses mapReduce command for Map-Reduce operation. In general, Map Reduce is used to handle large data sets.

In MongoDB Documentation, **Map-Reduce** is a data processing system that condenses a large amount of data into useful overall results. MongoDB uses mapReduce command for **Map-Reduce operation** . In general, Map Reduce is used to handle large data sets.

MapReduce command in MongoDB

The basic syntax of mapReduce command is as follows:

```
> db . collection . mapReduce ( function () { emit ( key , value );}, //map func
```

First, the function of Map Reduce Query Collection, then map the resulting Document to emit (Emit) key-value pairs that are then shortened based on the keys that have multiple values.

In the above syntax:

map is a JavaScript function that maps a value to a key and emits a key-value pair.

reduce is a JavaScript function that shortens or groups all Documents with the same key.

out determines the location of Map-Reduce query results.

The query specifies the arbitrary selection criteria to select the documents.

Sort sorting criteria arbitrarily arranged.

limit determines the number of arbitrary maximum documents to be returned.

Use MapReduce in MongoDB

You follow the structure of the Document to keep User Post. This Document stores the user's user_name and status.

```
{ "post_text" : "tutorialspoint is an awesome website for tutorials" , "user_nam
```

Now, we will use a mapReduce function on the posts collection to select all active posts, group them based on user_name and then count the number of posts of each user by using the following code:

```
> db . posts . mapReduce ( function () {  emit ( this . user_id , 1 ); }, functi
```

The mapReduce query on will result:

```
{
"result": "post_total",
"timeMillis": 9,
"counts": {
"input": 4,
"emit": 4,
"reduce": 2,
"output": 2
},
"ok": 1,
}
```

The results show that, the total of 4 documents have been connected to the query (status: "active"), the map function emits 4 documents with key-value pairs and ultimately reduce the function of mapped Documents with the same key go in 2.

To see the results of this mapReduce query, you use the find operator:

```
> db . posts . mapReduce ( function () {  emit ( this . user_id , 1 ); }, functi
```

The above query provides results indicating that both users **tom** and **mark** have two posts in the state that is active.

```
{ "_id" : "tom" , "value" : 2 } { "_id" : "mark" , "value" : 2 }
```

In the same manner, MapReduce queries can be used to build complex Aggregation queries. The use of custom JavaScript functions makes using MapReduce more flexible and powerful.

According to Tutorialspoint

Previous article: ObjectId in MongoDB

Next article: Text Search in MongoDB