NIELIT Gorakhpur

Course name: A level **SUBJECT:DATABASE TECHNOLOGIES**

Topic: MongoDB **DATE:14/5/2020**

Multikey Index

MongoDB uses multikey indexes to index the content stored in arrays. To index a field that holds an array value, MongoDB creates an index key for each element in the array, i.e. creates separate index entries for every element of the array

These *multikey* indexes support efficient queries against array fields. Multikey indexes can be constructed over arrays that hold both scalar values (e.g. strings, numbers) *and* nested documents.

The multikey indexes, created by MongoDB, allow queries to select documents that contain arrays by matching on element or elements of the arrays.

Syntax

```
db.collection_name.createIndex( { <field>: value } )
```

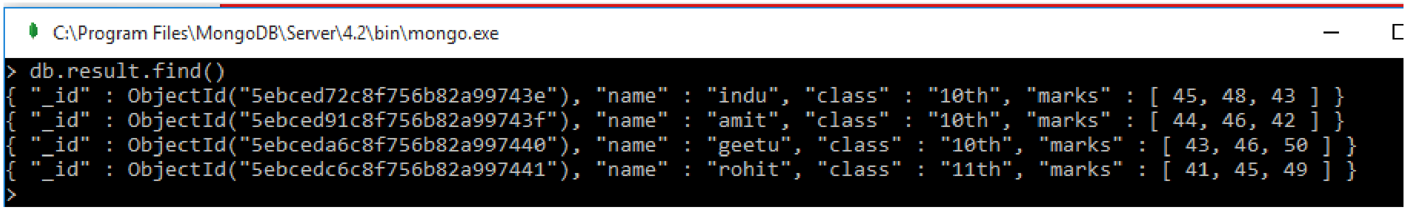
where
Field : is the name of the key in the collection
Value : 1 for ascending , -1 for descending

While Creating an Index, it is not required to explicitly specify the multikey type, MongoDB automatically determines whether to create a multikey index, if the specified indexed field contains an array value.

Example:

Lets consider a collection **result** with following documents:

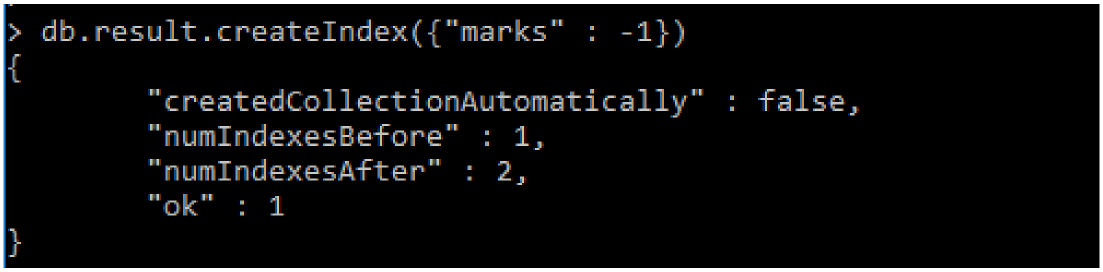
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [45, 48, 43] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [44, 46, 42] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [43, 46, 50] }
{ "_id" : ObjectId("5ebcedc6c8f756b82a997441"), "name" : "rohit", "class" : "11th", "marks" : [41, 45, 49] }



```
> db.result.find()
{ "_id" : ObjectId("5ebced72c8f756b82a99743e"), "name" : "indu", "class" : "10th", "marks" : [ 45, 48, 43 ] }
{ "_id" : ObjectId("5ebced91c8f756b82a99743f"), "name" : "amit", "class" : "10th", "marks" : [ 44, 46, 42 ] }
{ "_id" : ObjectId("5ebceda6c8f756b82a997440"), "name" : "geetu", "class" : "10th", "marks" : [ 43, 46, 50 ] }
{ "_id" : ObjectId("5ebcedc6c8f756b82a997441"), "name" : "rohit", "class" : "11th", "marks" : [ 41, 45, 49 ] }
>
```

Lets create an index on “**marks**” key, which is an array on descending order

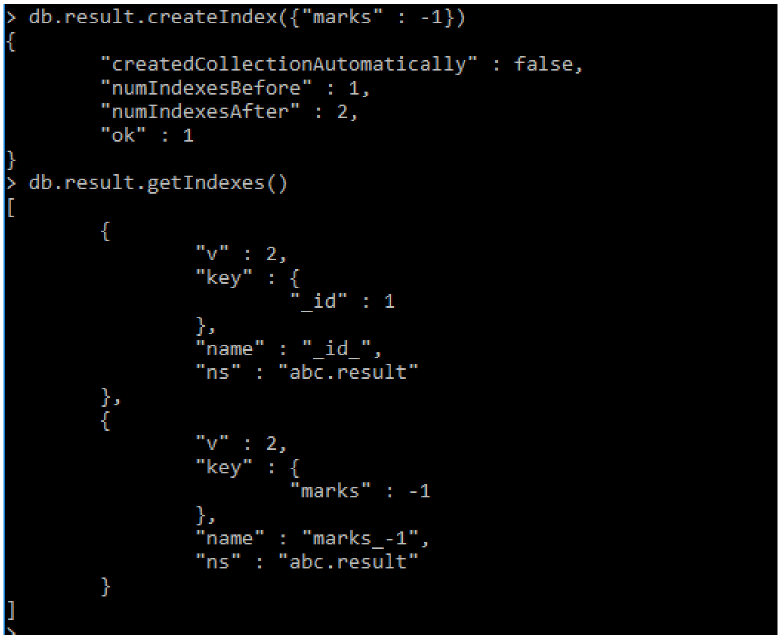
> db.result.createIndex({"marks" : -1})



```
> db.result.createIndex({"marks" : -1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

And if we run the command **getindexes**(), it will display all the indexes in the collection **result**

> db.result.getIndexes()



```
> db.result.createIndex({"marks" : -1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.result.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "abc.result"
        },
        {
                "v" : 2,
                "key" : {
                        "marks" : -1
                },
                "name" : "marks_-1",
                "ns" : "abc.result"
        }
]
>
```

It shows that an index on **marks key** has been created with name "**marks_-1**" and stored internally, but it will be a multikey index as it has been created on an array key.

Compound Multikey Index

Just like, creating a compound index, we may create a compound multikey index by specifying multiple keys in createIndex method.

Syntax

db.collection.createIndex({ <field1>: <value>, <field2>: <value>, ... })

where

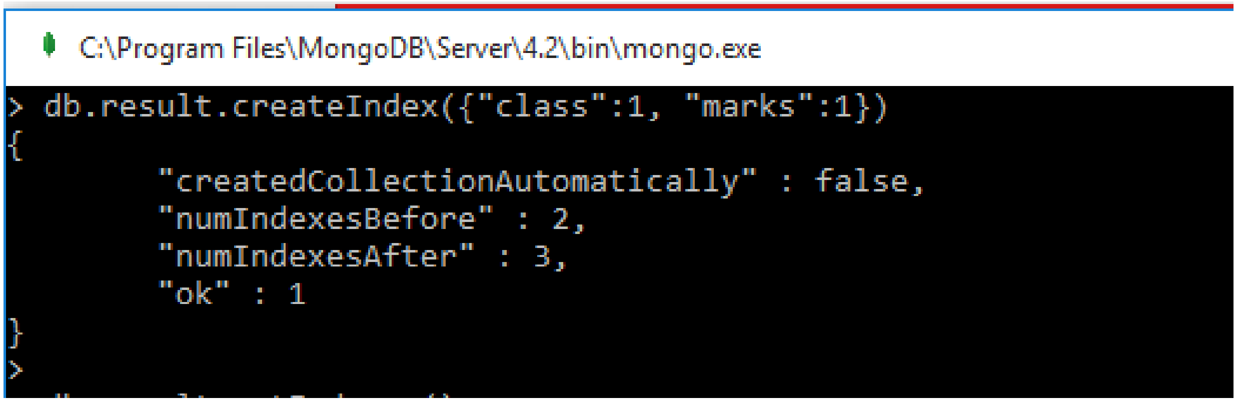
Field : is the name of the key in the collection

Value : 1 for ascending , -1 for descending

Example

Lets create a compound multikey index on class and marks key, both in ascending order.

> db.result.createIndex({"class":1, "marks":1})

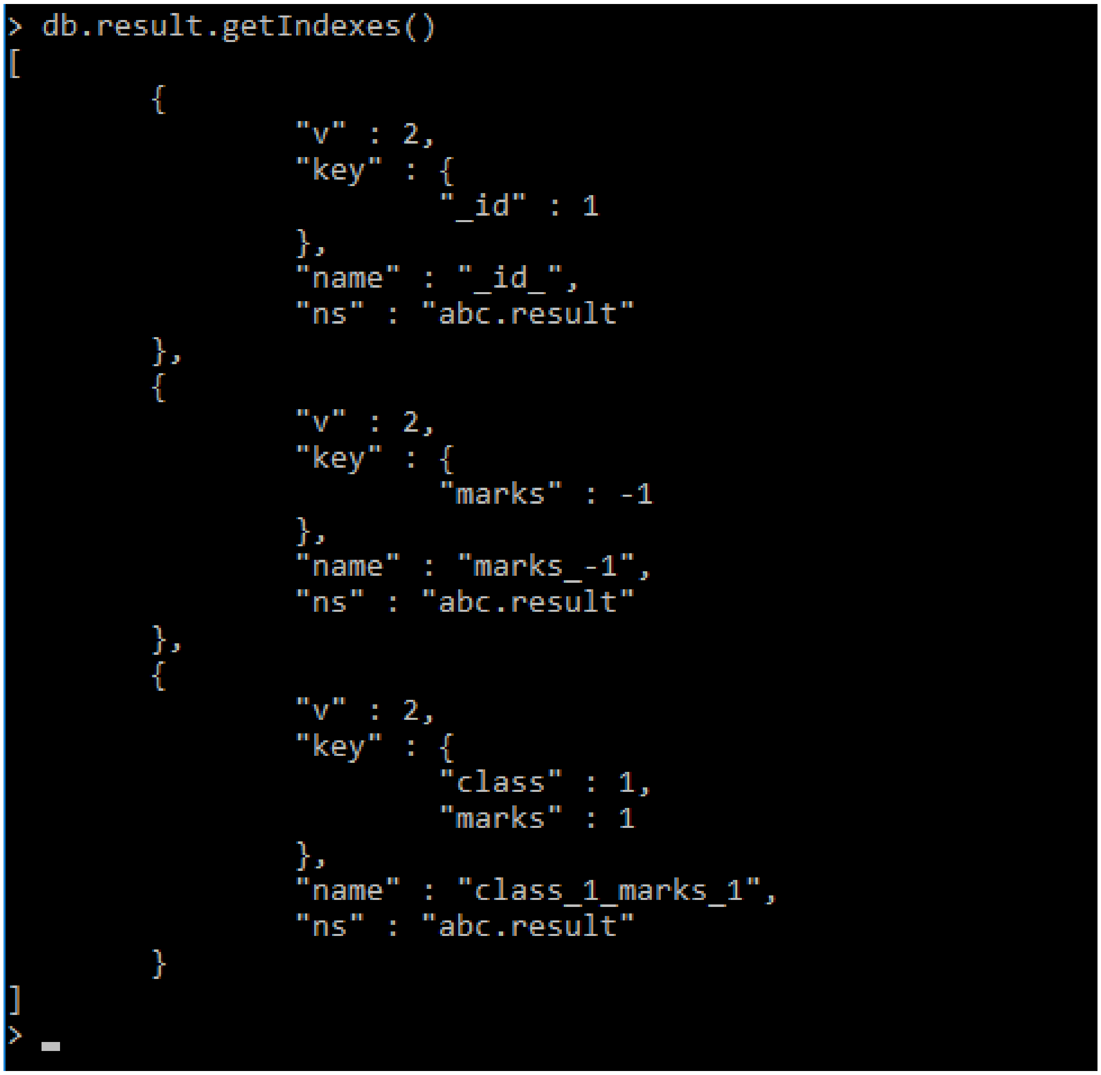


C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
> db.result.createIndex({"class":1, "marks":1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
>
```

This will create a compound index combining both the specified key , both in ascending order as we have specified the value as 1 in both the cases.

And if we display the created indexes on the result collection, with getIndexes() method, it will display both the created indexes.



```
> db.result.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "abc.result"
        },
        {
                "v" : 2,
                "key" : {
                        "marks" : -1
                },
                "name" : "marks_-1",
                "ns" : "abc.result"
        },
        {
                "v" : 2,
                "key" : {
                        "class" : 1,
                        "marks" : 1
                },
                "name" : "class_1_marks_1",
                "ns" : "abc.result"
        }
]
> _
```

Assignments

1. What are Multikey Indexes? How it is different from Compound index.
2. To create a multikey index, what is the syntax ? Do we need to specify **multikey** keyword while creating a multikey index in MongoDB?