SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

DBMS NoSQL - MongoDB - Indexing

Course Name : 23CAT603 - DATA BASE MANAGEMENT SYSTEM

Class : I Year / I Semester

Unit V – MongoDB - Indexing

MongoDB - Indexing

Indexes support the efficient resolution of queries. Without indexes, MongoDB must scan every document of a collection to select those documents that match the query statement. This scan is highly inefficient and require MongoDB to process a large volume of data.
Indexes are special data structures, that store a small portion of the data set in an easy-to-traverse form. The index stores the value of a specific field or set of fields, ordered by the value of the field as specified in the index.

The createIndex() Method
To create an index, you need to use createIndex() method of MongoDB.
Syntax
The basic syntax of **createIndex()** method is as follows().

>db.COLLECTION_NAME.createIndex({KEY:1})

Here key is the name of the field on which you want to create index and 1 is for ascending order. To create index in descending order you need to use -1.
Example

MongoDB - Indexing

```
>db.mycol.createIndex({"title":1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
>
```

In **createIndex()** method you can pass multiple fields, to create index on multiple fields.

```
>db.mycol.createIndex({"title":1,"description":-1})
>
```

This method also accepts list of options (which are optional

MongoDB - Indexing

Parameter	Type	Description
background	Boolean	Builds the index in the background so that building an index does not block other database activities. Specify true to build in the background. The default value is false.
unique	Boolean	Creates a unique index so that the collection will not accept insertion of documents where the index key or keys match an existing value in the index. Specify true to create a unique index. The default value is false.
name	string	The name of the index. If unspecified, MongoDB generates an index name by concatenating the names of the indexed fields and the sort order.
sparse	Boolean	If true, the index only references documents with the specified field. These indexes use less space but behave differently in some situations (particularly sorts). The default value is false.

MongoDB - Indexing

Parameter	Type	Description
expireAfterSeconds	integer	Specifies a value, in seconds, as a TTL to control how long MongoDB retains documents in this collection.
weights	document	The weight is a number ranging from 1 to 99,999 and denotes the significance of the field relative to the other indexed fields in terms of the score.
default_language	string	For a text index, the language that determines the list of stop words and the rules for the stemmer and tokenizer. The default value is **English**.
language_override	string	For a text index, specify the name of the field in the document that contains, the language to override the default language. The default value is language.

MongoDB - Indexing

The dropIndex() method

You can drop a particular index using the dropIndex() method of MongoDB.

Syntax

The basic syntax of DropIndex() method is as follows().

>db.COLLECTION_NAME.dropIndex({KEY:1})

Here, "key" is the name of the file on which you want to remove an existing index. Instead of the index specification document (above syntax), you can also specify the name of the index directly as: dropIndex("name_of_the_index")

MongoDB - Indexing

```
> db.mycol.dropIndex({"title":1})
{
        "ok" : 0,
        "errmsg" : "can't find index with key: { title: 1.0 }",
        "code" : 27,
        "codeName" : "IndexNotFound"
}
```

MongoDB - Indexing

The dropIndexes() method

This method deletes multiple (specified) indexes on a collection.

Syntax

The basic syntax of DropIndexes() method is as follows() −

>db.COLLECTION_NAME.dropIndexes()

Example

Assume we have created 2 indexes in the named mycol collection as shown below −

> db.mycol.createIndex({"title":1,"description":-1})

Following example removes the above created indexes of mycol −

```
>db.mycol.dropIndexes({"title":1,"description":-1})
{ "nIndexesWas" : 2, "ok" : 1 }
>
```

MongoDB - Indexing

The getIndexes() method

This method returns the description of all the indexes int the collection.

Syntax

Following is the basic syntax od the getIndexes() method

db.COLLECTION_NAME.getIndexes()

Example

Assume we have created 2 indexes in the named mycol collection as shown below

- **db.mycol.createIndex({"title":1,"description":-1})**

Following example retrieves all the indexes in the collection mycol

MongoDB - Indexing

```
> db.mycol.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "test.mycol"
        },
        {
                "v" : 2,
                "key" : {
                        "title" : 1,
                        "description" : -1
                },
                "name" : "title_1_description_-1",
                "ns" : "test.mycol"
        }
]
>
```

Indexing in MongoDB

Indexing in **MongoDB** is a crucial feature that enhances query processing efficiency. Without indexing, MongoDB must scan every document in a collection to retrieve the **matching** documents and leading to **slower query performance**. **Indexes** are special data structures that store information about the documents in a way that makes it easier for MongoDB to quickly locate the right data.

In this article, We will learn about **Indexing in MongoDB** by understanding **how to create, drop and get an Index** with the help of examples and so on.

Indexing in MongoDB

MongoDB uses **indexing** to make the query processing more efficient.
If there is no indexing, then MongoDB must scan every document in the collection and retrieve only those documents that match the query.
Indexes are special data structures that store some information related to the documents such that it becomes easy for MongoDB to find the right data file.
The indexes are ordered by the value of the field specified in the index.

Indexing in MongoDB

Creating an Index

MongoDB provides a method called createIndex() that allows users to create an index.

Syntax:

db.COLLECTION_NAME.createIndex({KEY:1}) The key determines the field on the basis of which we want to create an index and 1 (or -1) determines the order in which these indexes will be arranged(ascending or descending).

Example of Indexing in MongoDB

```
db.mycol.createIndex({“age”:1})
{
“createdCollectionAutomatically” : false,
“numIndexesBefore” : 1,
“numIndexesAfter” : 2,
“ok” : 1
}
```

Indexing in MongoDB

The createIndex() method also has a number of optional parameters.
These include:

background (Boolean)
unique (Boolean)
name (string)
sparse (Boolean)
expireAfterSeconds (integer)
hidden (Boolean)
storageEngine (Document)

Indexing in MongoDB

Drop an index

In order to drop an index, MongoDB provides the dropIndex() method.

Syntax:

db.NAME_OF_COLLECTION.dropIndex({KEY:1})

The dropIndex() methods can only delete one index at a time. In order to delete (or drop) multiple indexes from the collection, MongoDB provides the dropIndexes() method that takes multiple indexes as its parameters.

Syntax:

db.NAME_OF_COLLECTION.dropIndexes({KEY1:1, KEY2: 1}) The dropIndex() methods can only delete one index at a time. In order to delete (or drop) multiple indexes from the collection, MongoDB provides the dropIndexes() method that takes multiple indexes as its parameters.

Indexing in MongoDB

Get Description of All Indexes

The getIndexes() method in MongoDB gives a description of all the indexes that exists in the given collection.

Syntax

db.NAME_OF_COLLECTION.getIndexes() It will retrieve all the description of the indexes created within the collection.

References

1. https://www.tutorialspoint.com/mongodb/mongodb_indexing.htm
2. https://www.geeksforgeeks.org/indexing-in-mongodb/
3. https://www.mongodb.com/resources/basics/databases/database-index