Course name: A level SUBJECT:DATABASE TECHNOLOGIES

Topic:MongoDB DATE:12/5/2020

Indexing in MongoDb

In SQL, index is a special type of data structure which is used to easily and quickly locate the record in a given table of the database without traversing every record in the give table. Indexed may be generated for one or multiple fields of the table. Binary Tree data structures are used by indexes.

Indexes plays a vital role in execution of queries in MongoDB. If no index is defined, MongoDB has to scan every record/ document of the specified collection and if index is there, it has to scan less number of documents resulting in a faster result to select those documents that match the query criteria. The index in MongoDB is similar to the indexes used in other relational databases.

In MongoDB, indexes are special data structure, which stores the value of a specific field or set of fields of a collection, ordered by the value of the field. The ordering of the index, ascending or descending, supports efficient equality matches and range-based query operations apart from returning sorted results by using the ordering in the index. The indexes are defined at collection level in MongoDB on any field or sub-field of the documents. MongoDB indexes also uses a B-tree data structure.

It is very important and suggested to create the index on the field that will be frequently searched in a collection.

Default Index in MongoDB i.e. _id

By default, MongoDB creates a _id field if not specified in each collection and create a unique index on the **_id** field during the creation of a collection. The _id field acts like Primary Key of other RDBMS and prevents clients from inserting two documents with the same value for the _id field. The index on the _id field automatically created by MongoDB cannot be removed or dropped.

Single Field Index

MongoDb provides the facility to create index on any user defied key in the collection apart from the default **_id** index. This user created index can be ascending/descending indexes on a single field of a document as defined by the user.

Syntax

db.collection.createIndex({ <field>: <Value>})

where

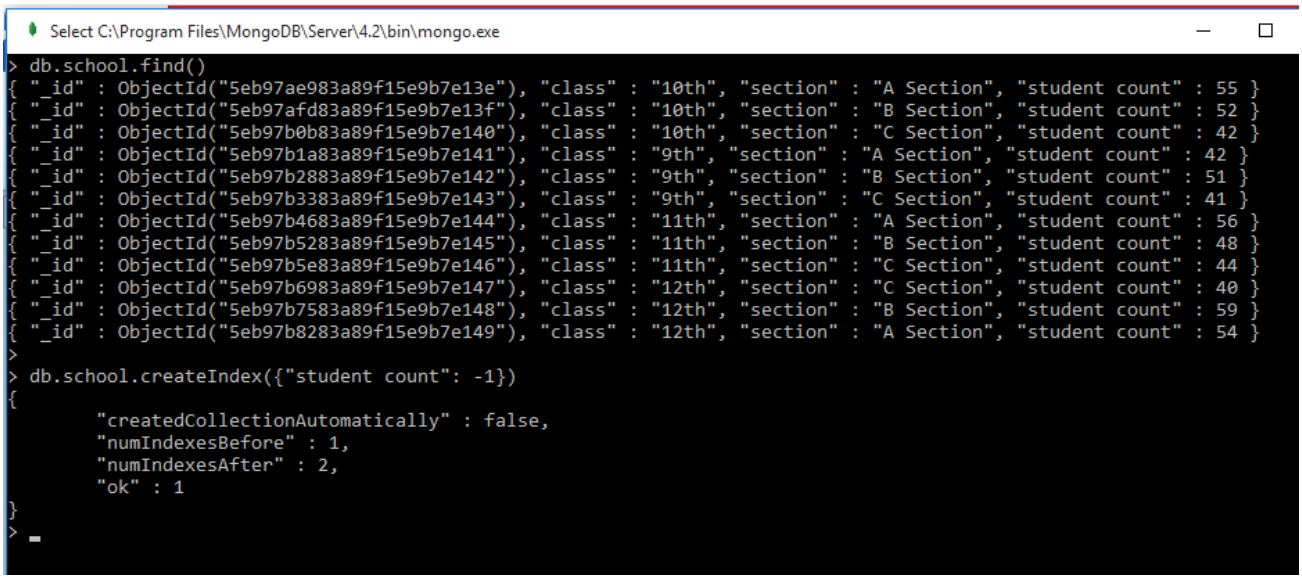
Field : is the name of the key in the collection

Value : 1 for ascending , -1 for descending

Example

Command to create a descending order index on field “student count”

db.school.createIndex({"student count": -1})



After executing the command, it displays

"createdCollectionAutomatically" : false -- It is created by user

"numIndexesBefore" : 1 -- No of indexes before the command (only _id)

"numIndexesAfter" : 2 -- No of indexes after the command (_id and the other one)

"ok" : 1 – Command executed Successfully

Here, an index has been created on “student count” which means when someone searches the document based on the “student count”, the search will be faster because the created index will be used for this search.

Compound Indexes

A compound index is an index on two or more fields of a collection, and it can support queries based on those fields. In compound indexes, Fields can be sorted inside other fields.

Syntax

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

where

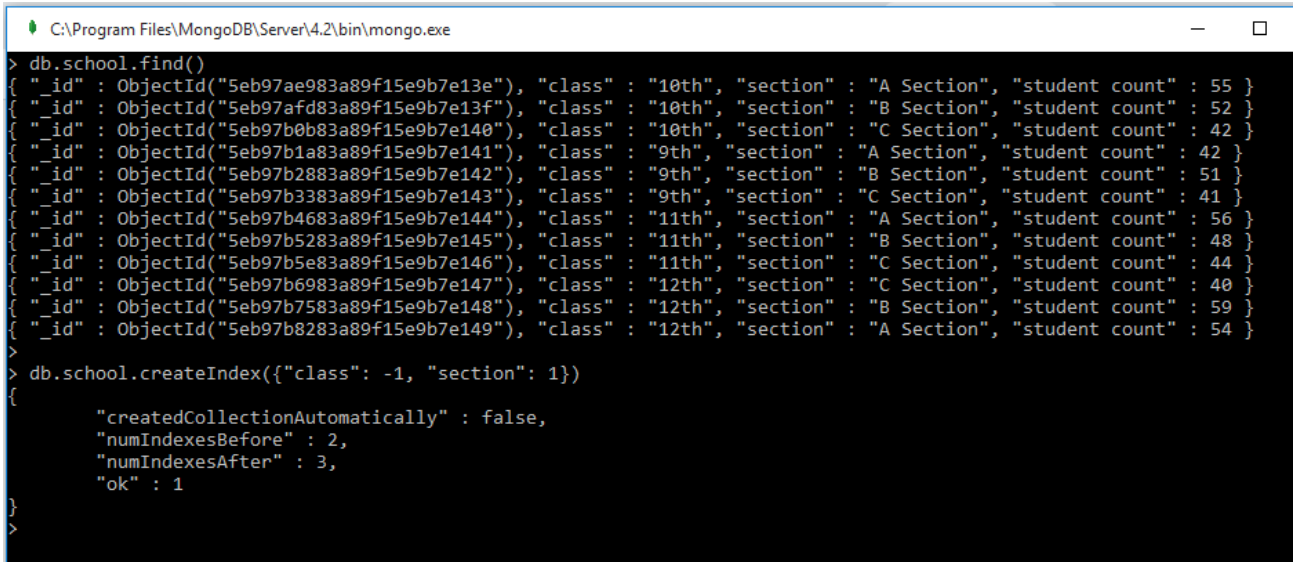
Field : is the name of the key in the collection

Value : 1 for ascending , -1 for descending

Example

We have to create a compound index on two fields, i.e. class (descending order) and section(ascending order), to do so, the command will be

db.school.createIndex({"class": -1, "section": 1})



After executing the command, it displays

"createdCollectionAutomatically" : false -- It is created by user

"numIndexesBefore" : 2 -- No of indexes before the command (_id and “student count” created just before)

"numIndexesAfter" : 3 -- No of indexes after the command (_id, “student count” and compound index of “**class and section”**)

"ok" : 1 – Command executed Successfully

This compound index of “**class + section”** sorts the collection first by “**class**” and then, within each “**class**” value, sort by “**section**”.

Assignment

1. What is indexing?

2. What is the difference between default index, single index, compound index?