MongoDB Indexing:

find()
update({},{})
delete({})

We have to find matched documents.

book without index
book with index

shopping 100s of employees

If i need to run only one query from very large data, then do we need to create index as it will also takes time

Book my show
select all movies in given theatre and in a particular city
IRCTC

select all trains from source to destination

100000 trains
Hyderabad

```
db.employees.find({ename: "Sunny"}).pretty()
db.employees.find({ename: "Sunny"}).explain("executionStats")

> db.employees.find({ename: "Sunny"}).explain("executionStats")
{
	"queryPlanner" : {
		"plannerVersion" : 1,
		"namespace" : "durgadb.employees",
		"indexFilterSet" : false,
		"parsedQuery" : {
			"ename" : {
				"$eq" : "Sunny"
			}
		},
		"winningPlan" : {
			"stage" : "COLLSCAN",
			"filter" : {
				"ename" : {
					"$eq" : "Sunny"
				}
			},
			"direction" : "forward"
		},
		"rejectedPlans" : [ ]
	},
	"executionStats" : {
		"executionSuccess" : true,
		"nReturned" : 2,
		"executionTimeMillis" : 0,
		"totalKeysExamined" : 0,
		"totalDocsExamined" : 13,
		"executionStages" : {
			"stage" : "COLLSCAN",
```

```
                "filter" : {
                        "ename" : {
                                "$eq" : "Sunny"
                        }
                },
                "nReturned" : 2,
                "executionTimeMillisEstimate" : 0,
                "works" : 15,
                "advanced" : 2,
                "needTime" : 12,
                "needYield" : 0,
                "saveState" : 0,
                "restoreState" : 0,
                "isEOF" : 1,
                "direction" : "forward",
                "docsExamined" : 13
        }
    },
    "serverInfo" : {
            "host" : "DESKTOP-ECE8V3R",
            "port" : 27017,
            "version" : "4.4.2",
            "gitVersion" :
"15e73dc5738d2278b688f8929aee605fe4279b0e"
    },
    "ok" : 1
}
```


Define index and then find():

---

db.employees.getIndexes()

Returns available indexes

```
> db.employees.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]

db.employees.createIndex({ename: 1})

> db.employees.createIndex({ename: 1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}

> db.employees.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "ename" : 1
                },
                "name" : "ename_1"
        }
]
```

```
> db.employees.find({ename: "Sunny"}).explain("executionStats")
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "durgadb.employees",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "ename" : {
                                "$eq" : "Sunny"
                        }
                },
                "winningPlan" : {
                        "stage" : "FETCH",
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "ename" : 1
                                },
                                "indexName" : "ename_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "ename" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : false,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "ename" : [
                                                "[\"Sunny\", \"Sunny\"]"
                                        ]
                                }
```

```
                }
        },
        "rejectedPlans" : [ ]
},
"executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 2,
        "executionTimeMillis" : 0,
        "totalKeysExamined" : 2,
        "totalDocsExamined" : 2,
        "executionStages" : {
                "stage" : "FETCH",
                "nReturned" : 2,
                "executionTimeMillisEstimate" : 0,
                "works" : 3,
                "advanced" : 2,
                "needTime" : 0,
                "needYield" : 0,
                "saveState" : 0,
                "restoreState" : 0,
                "isEOF" : 1,
                "docsExamined" : 2,
                "alreadyHasObj" : 0,
                "inputStage" : {
                        "stage" : "IXSCAN",
                        "nReturned" : 2,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 3,
                        "advanced" : 2,
                        "needTime" : 0,
                        "needYield" : 0,
                        "saveState" : 0,
                        "restoreState" : 0,
```

```
                        "isEOF" : 1,
                        "keyPattern" : {
                                "ename" : 1
                        },
                        "indexName" : "ename_1",
                        "isMultiKey" : false,
                        "multiKeyPaths" : {
                                "ename" : [ ]
                        },
                        "isUnique" : false,
                        "isSparse" : false,
                        "isPartial" : false,
                        "indexVersion" : 2,
                        "direction" : "forward",
                        "indexBounds" : {
                                "ename" : [
                                        "[\"Sunny\", \"Sunny\"]"
                                ]
                        },
                        "keysExamined" : 2,
                        "seeks" : 1,
                        "dupsTested" : 0,
                        "dupsDropped" : 0
                }
        }
    },
    "serverInfo" : {
        "host" : "DESKTOP-ECE8V3R",
        "port" : 27017,
        "version" : "4.4.2",
        "gitVersion" :
"15e73dc5738d2278b688f8929aee605fe4279b0e"
    },
```

```
        "ok" : 1
}
```

We can define index on multiple fields:

```
db.employees.createIndex({eno:1,ename:-1})
```

How to drop index?

```
db.employees.dropIndex({ename: 1})
```

```
> db.employees.dropIndex({ename: 1})
{ "nIndexesWas" : 2, "ok" : 1 }
> db.employees.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
```

Indexing is DBMS related concept and almost every database provides support.

Indexes supports efficient execution of queries in MongoDB.

Indexing concept is very helpful for find,update and delete queries.

The main objective of indexing is to improve performance.

Without indexes, server will scan all documents present in the collection to find matched documents. This is called Collection Scan(COLLSCAN).

Whenever we are defining index, MongoDB Server will store values of indexed field in B-Tree Data structure in specified sorting order.

To find matched documents, we are not required to scan all

documents,server can identify matched documents directly based Index Scan(IXSCAN).

As the number of documents to be scan, is reduces and hence performance will be improved.

How to check available indexes?

----

db.collection.getIndexes()

How to create Index:

----

db.collection.createIndex({field: 1|-1})

1 means Ascending order

-1 means Descending order

How to drop index:

----

db.collection.dropIndex({field:1|-1})

Is it possible to define index on multiple fields:

----

db.collection.createIndex({field1: 1, field2: -1})

The order of fields is important.

Q. Is it recommended to define index for all fields?

No, performance will be degraded.

Server has to store every indexed field value separately.

Q. Is it recommended to define index for small collections?

No. Instead of improving performance , it will be reduced.

This is like keeping multiple cash counters in paan shop.

Q. How to see execution stats of query?

explain() method with executionStats arguement

executionStats – this mode includes all the information provided by the queryPlanner, plus the statistics. Statistics include details such as the number of documents examined and returned, the execution time in milliseconds, and so on.

db.collection.find({query}).explain("executionStats")

Q. Is any default index for our collection?

For every collection default index is avaialble, and it is based on _id field.

db.employees.getIndexes()
[{ "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" }]

Aggregation Framework:

```
db.employees.insertOne({eno:100,ename:"Sunny",esal:1000,eaddr:"Mumbai"})
db.employees.insertOne({eno:200,ename:"Bunny",esal:2000,eaddr:"Hyderabad"})
db.employees.insertOne({eno:300,ename:"Chinny",esal:3000,eaddr:"Hyderabad"})
db.employees.insertOne({eno:400,ename:"Vinny",esal:4000,eaddr:"Mumbai"})
db.employees.insertOne({eno:500,ename:"Pinny",esal:5000,eaddr:"Chennai"})
```