Indexes

Open windows prompt   and connect to mongos

```
mongo --port 27200

mongos> use mongoMart

mongos> db.restaurants.getIndexes()

mongos> db.restaurants.createIndex({name : 1})

mongos> db.restaurants.createIndex({cuisine : 1, name : 1})

mongos> db.restaurants.getIndexes()

mongos> db.restaurants.find({cuisine: "Sushi", name: /^Sushi/})

mongos> db.restaurants.find({cuisine: "Sushi", name: /^Sushi/}).count()

mongos> db.restaurants.dropIndex({name : 1})

mongos> db.restaurants.getIndexes()

mongos> db.restaurants.createIndex({name : 1}, {background:1})

mongos> use cars

mongos> db.list2.find()

mongos> db.list.getIndexes()

mongos> db.list.createIndex({mileage:1,year:-1},{})

mongos> db.list.find({mileage:{$gte:100}}).explain()

mongos> db.list.find({year:2000}).explain()

mongos>  db.list.find({year:2000}).hint('mileage_1_year_-1').explain()

mongos> db.list.createIndex({brand:'hashed'})

mongos> db.list.getIndexes()
mongos> db.list.find({brand:'Toyota'}).explain()

mongos> db.list.find({brand:{$in:['Toyota', 'Nissan']}}).explain()

mongos> db.list.dropIndex('brand_hashed')

mongos> db.arr.find()

mongos> db.arr.createIndex({cities:1})

mongos> db.arr.find({cities:{$size:2}})

mongos> db.arr.find({cities:{$size:2}}).explain()

mongos> db.arr.find({cities:{$all:['London']}})

mongos> db.arr.find({cities:{$all:['London']}}).explain()

mongos> db.arr.find({cities:{$elemMatch:{$eq: 'London'}}})

mongos> db.arr.find({cities:{$elemMatch:{$eq: 'London'}}}).explain()

mongos> db.nested.findOne()

mongos> db.nested.createIndex({'models.engine':1})

mongos> db.nested.find({models:{$elemMatch:{engine:'Petrol'}}})

mongos> db.nested.find({models:{$elemMatch:{engine:'Petrol'}}}).explain()

mongos> db.nested.find({models:{$exists:true}})

mongos> db.nested.find({models:{$exists:true}}).explain()

mongos> db.nested.createIndex({models:1})

mongos> db.nested.find({models:{$exists:1}})

mongos> db.nested.find({models:{$exists:1}}).explain()

mongos> db.list.find({},{mileage:1})
{ "_id" : ObjectId("60ad5cfec19cbcf173c7519e") }
{ "_id" : ObjectId("60ad5cfec19cbcf173c7519f"), "mileage" : 120000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a0") }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a1"), "mileage" : 5000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a2"), "mileage" : NumberLong(200000) }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a3"), "mileage" : NumberLong(65000) }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a4"), "mileage" : 7000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a5"), "mileage" : 15000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a6"), "mileage" : 27000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a7"), "mileage" : 35000 }

mongos> db.list.getIndexes()
mongos> db.list.dropIndex('mileage_1_year_-1')

mongos> db.list.createIndex({mileage:1},{name:'mileage_1_non_sparse'})

mongos> db.list.createIndex({mileage:1},{sparse:true,name:'mileage_1_sparse'})

mongos> db.list.createIndex({mileage:-1},{sparse:true,name:'mileage_-1_sparse'})

mongos> db.list.getIndexes()
[
	{
		"v" : 2,
		"key" : {
			"_id" : 1
		},
		"name" : "_id_"
	},
	{
		"v" : 2,
		"key" : {
			"mileage" : 1
		},
		"name" : "mileage_1_non_sparse"
	},
	{
		"v" : 2,
		"key" : {
			"mileage" : -1
		},
		"name" : "mileage_-1_sparse",
		"sparse" : true
	}
]
```

```
mongos> db.list.find({mileage:{$ne:0}},{mileage:1})
{ "_id" : ObjectId("60ad5cfec19cbcf173c7519e") }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a0") }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a1"), "mileage" : 5000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a4"), "mileage" : 7000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a5"), "mileage" : 15000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a6"), "mileage" : 27000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a7"), "mileage" : 35000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a3"), "mileage" : NumberLong(65000) }
{ "_id" : ObjectId("60ad5cfec19cbcf173c7519f"), "mileage" : 120000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a2"), "mileage" : NumberLong(200000) }

mongos> db.list.find({mileage:{$ne:0}},{mileage:1}).explain()


mongos> db.list.find({mileage:{$ne:0}},{mileage:1}).hint('mileage_-1_sparse')
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a2"), "mileage" : NumberLong(200000) }
{ "_id" : ObjectId("60ad5cfec19cbcf173c7519f"), "mileage" : 120000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a3"), "mileage" : NumberLong(65000) }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a7"), "mileage" : 35000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a6"), "mileage" : 27000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a5"), "mileage" : 15000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a4"), "mileage" : 7000 }
{ "_id" : ObjectId("60ad5cfec19cbcf173c751a1"), "mileage" : 5000 }


mongos> db.ttl.insert({msg:'hi',clock: new Date()})
WriteResult({ "nInserted" : 1 })
mongos> db.ttl.insert({msg:'no clock field'})
WriteResult({ "nInserted" : 1 })
mongos> db.ttl.insert({msg:'illegal clock val',clock: 'now'})
WriteResult({ "nInserted" : 1 })

mongos> db.ttl.find()
{ "_id" : ObjectId("60aff1d5712a9c7703ff61af"), "msg" : "hi", "clock" : ISODate("2021-05-27T19:24:05.281Z") }
{ "_id" : ObjectId("60aff218712a9c7703ff61b0"), "msg" : "no clock field" }
{ "_id" : ObjectId("60aff257712a9c7703ff61b1"), "msg" : "illegal clock val", "clock" : "now" }

mongos> db.ttl.createIndex({clock:1},{expireAfterSeconds:5})
{
	"raw" : {
		"shard3rs/localhost:2007,localhost:2008,localhost:2009" : {
```

```
                        "createdCollectionAutomatically" : false,
                        "numIndexesBefore" : 1,
                        "numIndexesAfter" : 2,
                        "commitQuorum" : "votingMembers",
                        "ok" : 1
                }
        },
        "ok" : 1,
        "operationTime" : Timestamp(1622143714, 7),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622143714, 7),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> db.ttl.find()
{ "_id" : ObjectId("60aff1d5712a9c7703ff61af"), "msg" : "hi", "clock" : ISODate("2021-05-27T19:24:05.281Z") }
{ "_id" : ObjectId("60aff218712a9c7703ff61b0"), "msg" : "no clock field" }
{ "_id" : ObjectId("60aff257712a9c7703ff61b1"), "msg" : "illegal clock val", "clock" : "now" }
mongos> db.ttl.find()
{ "_id" : ObjectId("60aff218712a9c7703ff61b0"), "msg" : "no clock field" }
{ "_id" : ObjectId("60aff257712a9c7703ff61b1"), "msg" : "illegal clock val", "clock" : "now" }


mongos> load('text_index.js')
true
mongos> arr
[
        {
                "name" : "apple pie",
                "msg" : "An apple pie is made with apples, flour, sugar, etc."
        },
        {
                "name" : "apple juice",
                "msg" : "Apple juice tastes great. It is made with water, sugar and apple."
        },
        {
                "name" : "orange cake",
                "msg" : "To make an orange cake, what do I need?"
        },
        {
                "name" : "café",
                "msg" : "Café sells coffee"
        }
]


mongos> db.text.insert(arr)
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 4,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})

mongos> db.text.find({},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }
{ "name" : "orange cake", "msg" : "To make an orange cake, what do I need?" }
{ "name" : "café", "msg" : "Café sells coffee" }


mongos> db.text.createIndex({name:'text',msg:'text'},{})
{
        "raw" : {
                "shard3rs/localhost:2007,localhost:2008,localhost:2009" : {
                        "createdCollectionAutomatically" : false,
                        "numIndexesBefore" : 1,
                        "numIndexesAfter" : 2,
                        "commitQuorum" : "votingMembers",
                        "ok" : 1
                }
        },
        "ok" : 1,
        "operationTime" : Timestamp(1622146744, 8),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622146744, 8),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}

mongos> db.text.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "_fts" : "text",
                        "_ftsx" : 1
                },
                "name" : "name_text_msg_text",
                "weights" : {
                        "msg" : 1,
                        "name" : 1
                },
                "default_language" : "english",
                "language_override" : "language",
                "textIndexVersion" : 3
        }
]

mongos> db.text.find({$text:{$search:'apple'}},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }



mongos> db.text.find({$text:{$search:'apple orange'}},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }
{ "name" : "orange cake", "msg" : "To make an orange cake, what do I need?" }
```

```
mongos> db.text.find({$text:{$search:'apple pie'}},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }

mongos> db.text.find({$text:{$search:"\"apple pie\""}},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }

mongos> db.text.find({$text:{$search:'apple -pie'}},{_id:0})
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }

mongos> db.text.find({$text:{$search:'an apple'}},{_id:0})
{ "name" : "apple pie", "msg" : "An apple pie is made with apples, flour, sugar, etc." }
{ "name" : "apple juice", "msg" : "Apple juice tastes great. It is made with water, sugar and
apple." }
```