

# MongoDB Indexing

## Objectives

After completing this lab you will be able to:

- Measure the time it takes to execute a query with the **explain** function
- Describe the process of creating, listing and deleting indexes
- Evaluate the effectiveness of an index

## Exercise 1 - Getting the environment ready

- Start mongodb server
- Connect to server
- select/create training database
- Create collection bigdata

```
theia@theiadocker-craigtrupp8:/home/project$ mongo -u root -p
ODI3Ny1jcmFpZ3Ry --authenticationDatabase admin local
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017/local
MongoDB server version: 3.6.3
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2023-10-14T16:43:05.126+0000 I STORAGE [initandlisten]
2023-10-14T16:43:05.126+0000 I STORAGE [initandlisten] ** WARNING: Using
the XFS filesystem is strongly recommended with the WiredTiger storage
engine
2023-10-14T16:43:05.126+0000 I STORAGE [initandlisten] **          See
http://dochub.mongodb.org/core/prodnotes-filesystem
2023-10-14T16:43:06.004+0000 I CONTROL [initandlisten]
2023-10-14T16:43:06.004+0000 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
```

```

2023-10-14T16:43:06.004+0000 I CONTROL [initandlisten] **          We
suggest launching mongod like this to avoid performance problems:
2023-10-14T16:43:06.004+0000 I CONTROL [initandlisten] **
numactl --interleave=all mongod [other options]
2023-10-14T16:43:06.004+0000 I CONTROL [initandlisten]
> show dbs;
admin  0.000GB
local  0.000GB
> use training
switched to db training
> show dbs
admin  0.000GB
local  0.000GB
> training.createCollection('bigdata')
2023-10-14T12:44:15.963-0400 E QUERY [thread1] ReferenceError: training
is not defined :
@(shell):1:1
> db.training.createCollection('bigdata')
2023-10-14T12:44:29.510-0400 E QUERY [thread1] TypeError:
db.training.createCollection is not a function :
@(shell):1:1
> db.createCollection('bigdata')
{ "ok" : 1 }
> show collections
bigdata
> db
training

```

- simple db command show which database is currently selected

## Exercise 2 - Insert documents

Let us insert a lot of documents into the newly created collection.

This should take around 3 minutes, so please be patient.

The code given below will insert 200000 documents into the 'bigdata' collection.

Each document would have a field named `account_no` which is a simple auto increment number.

And a field named `balance` which is a randomly generated number, to simulate the bank balance for the account.

```
use training
for
(i=1;i<=200000;i++){print(i);db.bigdata.insert({"account_no":i,"balance":Math.round(Math.random()*1000000)}})}
```

- Verify count in collection

```
> db.bigdata.count()
200000
```

### Exercise 3 - Measure the time taken by a query

- Let us run a query and find out how much time it takes to complete.
- Let us query for the details of account number 58982.
- We will make use of the explain function to find the time taken to run the query in milliseconds.

```
>db.bigdata.find({"account_no":58982}).explain("executionStats").executionStats.executionTimeMillis
66
```

### Exercise 4 - Working with indexes / Create

Before you create an index, choose the field you wish to create an index on.

- It is usually the field that you query most.
- create an index on the field **account\_no**.

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

- get a list of indexes on the 'bigdata' collection.

```

> db.bigdata.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "training.bigdata"
  },
  {
    "v" : 2,
    "key" : {
      "account_no" : 1
    },
    "name" : "account_no_1",
    "ns" : "training.bigdata"
  }
]

```

```

> db.bigdata.find().limit(5)
{ "_id" : ObjectId("652ac6b12c914de81854218b"), "account_no" : 1, "balance" : 974787 }
{ "_id" : ObjectId("652ac6b12c914de81854218c"), "account_no" : 2, "balance" : 446861 }
{ "_id" : ObjectId("652ac6b12c914de81854218d"), "account_no" : 3, "balance" : 956715 }
{ "_id" : ObjectId("652ac6b12c914de81854218e"), "account_no" : 4, "balance" : 470778 }
{ "_id" : ObjectId("652ac6b12c914de81854218f"), "account_no" : 5, "balance" : 160956 }
>

```

- We can see the assigned id to the row/collection after the creation and above is just away to limit a selection of the collection to get a quick insight to what the row objects look like for the collection

## Exercise 5 - Find out how effective an index is

```
> db.bigdata.find({"account_no":  
69271}).explain("executionStats").executionStats.executionTimeMillis  
0
```

- Wow! So we can now see the difference from exercise #3 which showed nearly 70 Millis when using the account\_no in the where type query below translated to mongoDB

## Exercise 6 - Delete an index

```
> db.bigdata.dropIndex({"account_no":1})  
{ "nIndexesWas" : 2, "ok" : 1 }
```

## Practice exercises

- Create an index on the balance field.

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

- Query for documents with a balance of 10000 and record the time taken.

```
>db.bigdata.find({"balance":10000}).explain("executionStats").executionStat  
s.executionTimeMillis  
0  
> db.bigdata.find({"balance":10000}).count()  
0  
> db.bigdata.find({"balance":10000})  
> db.bigdata.find({"balance":974787})  
{ "_id" : ObjectId("652ac6b12c914de81854218b"), "account_no" : 1, "balance"
```

```
: 974787 }
> db.bigdata.find({"balance":974787}).count()
1
>
db.bigdata.find({"balance":974787}).explain("executionStats").executionStats.executionTimeMillis
0
```

- Drop the index you have created
- Then run a similar query for a row with a balance to note the execution time difference

```
> db.bigdata.dropIndex({"balance":1})
{ "nIndexesWas" : 2, "ok" : 1 }
>
db.bigdata.find({"balance":974787}).explain("executionStats").executionStats.executionTimeMillis
74
```

- Then simply run “exit” to disconnect from the mongoDB server