DK908b

LAB2 REPORT

NB: for images, please look at the folder "images" in the zip archive

1. Task0

- I used the commands **sudo mongod --dbpath ./mongoServer --port 27217** to lunch the mongo db server.
- I used the command **mongoimport -c movies –port 27217./moviepeople-10.jsonl (Very important client should connect to the same port the server is listening to.)**
- To retrieve all the data, I used **db.movies.find().pretty()** (the pretty at the end allow to have better formatation of theoutput) which returns the contain of the collection movie. The related picture id is **task0_3**

2. Task1

- 1)From the lab directory i ssued the following command : **mongoimport -c collectionName –port 27217 ./lab_2_datasets/filename,** the relatated picture is **task1_1**
- 2-a) The query used is **db.moviepeople.find({"person-name" : "Taxiera,Anabela"})()** and the releated picture id is **task1_2a**
- 2-b) The query used is **: db.moviespeople.find({"person-name" : "Spielberg, Steven"},{"info.birthnotes" : 1, _id : 0})** and the related picture is **task1_2b**
- 2-c)The query used is **db.moviespeople.find("info.birthnotes" : {$exists : "Lisbon"}).count()** . The picture is task1_2c
- 2-d) the query used is the following : **db.find({"info.height" : {$gt : "170 cm"},{_id : 0, "person-name" :1 , "info.hiegth" : 1}).** The result can be seen in the picture **task1_d**
- 2-e) The query used is **db.moviespeople.find({$or : [{"info.birthname" : {$regex : ".*Opera.*"}},{"info.trivia" : {$regex : ".*Opera.*"}},{"info.birthnotes" : {$regex : ".*Opera.*"}},{"info.minibiography" : {$regex : ".Opera.*"}},{"info.magazinecoverphoto" : {$regex : ".*Opera.*"}}]},{"person-name" : 1, _id : 0})**
- 2-f) the query used is **db.moviepeople.qggregqte([{$lookup : {from : "citie" ,localField : "info.brthnotes",foreignField : "name",as "moviePersonCitie"}},{$project : {moviPersonCitie : 1, population : 1,"location.longitude" :1, "location.latitude" :1}}]).pretty() . The result can be seen from the picture task1_2f**

3. Task2

- I created 3 directories **mongoServer1, mongoServer2, mongoServer3**
- The replica set has been create typing the following command in three different terminals : **mongod –replSet small-movie -dbpath ./mongoServer[1-3] --port 2701[1-3]**
- The command used for the initialisation of the replica set is **rs.initiate({_id : “small-movie”,members : [{_id : 1, host : “localhost:27011”},{_id : 2, host : “localhost:27012”}]}) (launched from a shell terminal),** the I added the arbiter using the command rs.addArb(“localhost:27013”) and the server running on port 27011 has been assigned as primary while the one on port 27012 as secondary; we can see the output of the 3 servers from the pictures: **task2OutputServer1, task2OutputServer2, task2OutputServer3**
- I used **rs.conf()** to get the configuration information of the replica set. The output can be seen on the picture **task2OutputConfiguration**
- The output of servers 1 (primary) and 2(secondary) can be seen on the pictures **server1ImportOutput,and server2ImportOutput**
- After stopping master the secondary and the arbiter servers don’t end, they keep trying the to connect to the primary server. The output is on the picture **killedMaster**

4. Task3

- I Used the command mongod –shardsvr –dbpath ./mongoServer[1-2] –port 2701[4-5] to start the 2 servers

The command used for sharding are the following :

```
mongod --configsvr --dbpath ./mongoconfig --port 27016 , mongos --configdb localhost:27016 --chunkSize 1 --port 27020, mongo localhost:27020/admin
db.runCommand( { addshard : "localhost:27014" } ) -> { "shardAdded" : "shard0000", "ok" : 1 }
db.runCommand( { addshard : "localhost:27015" } ) -> { "shardAdded" : "shard0001", "ok" : 1 }
db.runCommand({ enablesharding: "test"}) -> { "ok" : 1 }
db.runCommand( { shardcollection : "test.cities", key : {country : 1} } )
```