MongoDB. Home Task 1

Note: if you already have MongoDB installed, please, check that you are running the latest version – 4.0, because it's necessary to complete some of the tasks.

1. Install MongoDB

Follow installation guidelines for your OS at https://docs.mongodb.com/manual/installation/#mongodb-community-edition.

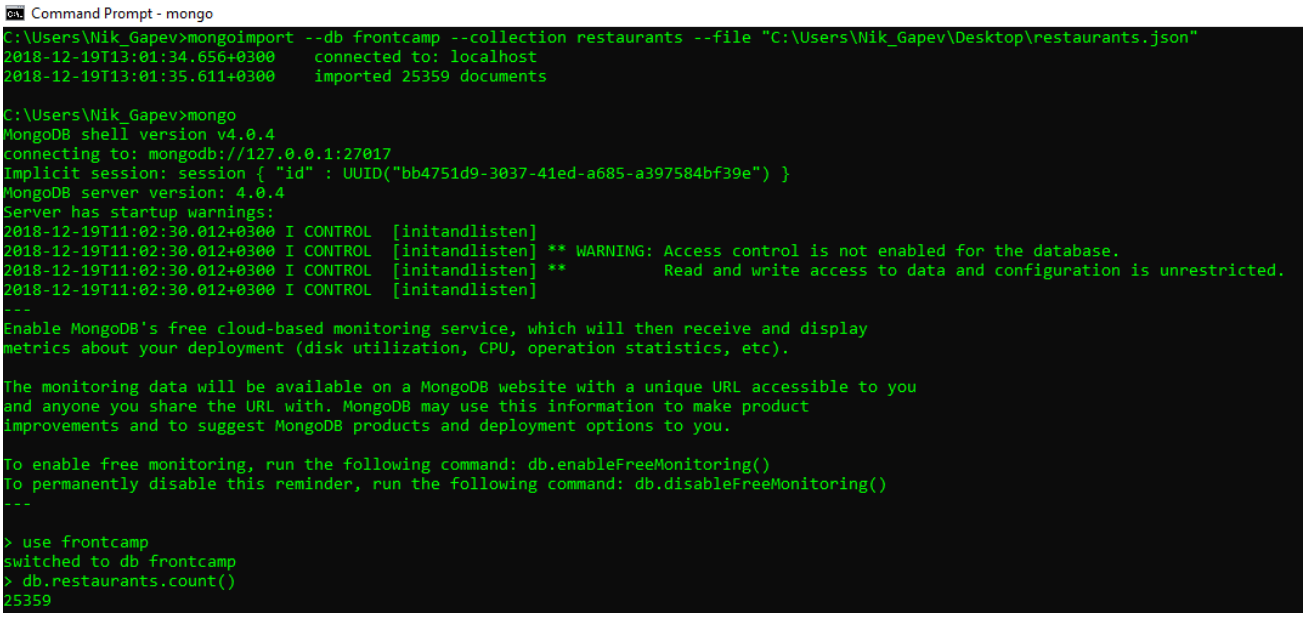
2. Import Restaurants Collection

2.1. Save *restaurants.json* on your PC

2.2. Run local instance of MongoDB and port for the instance run *mongod* without any parameters

2.3. Use *mongoimport* (it's in MongoDB installation folder) to import the collection to the database . Run local MongoDB on the default port the following command should create "restaurants" collection in "frontcamp" database *mongoimport --db frontcamp --collection restaurants --file*

2.4. Verify that collection was correctly imported. Verify that the number of the documents in the restaurants collection is 25359.



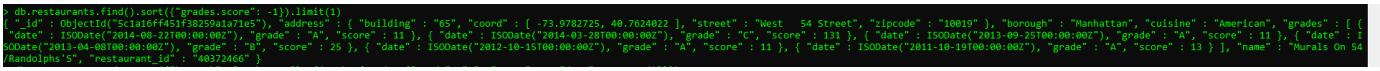
3. Querying Restaurants Collection

Answer the following questions:

3.1. How many “Chinese” (cuisine) restaurants are in “Queens” (borough)?

```
> db.restaurants.find({"cuisine": "Chinese", "borough": "Queens"}).count()
728
```

3.2. What is the _id of the restaurant which has the grade with the highest ever score?



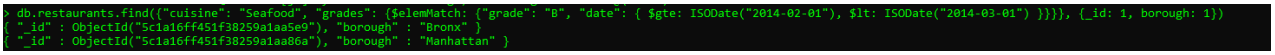
3.3. Add a grade { grade: "A", score: 7, date: ISODate() } to every restaurant in “Manhattan” (borough).

```
> db.restaurants.updateMany({"borough": "Manhattan"}, {$push: {grades:{"grade": "A", "score": 7, "date": ISODate()}}})
{ "acknowledged" : true, "matchedCount" : 10259, "modifiedCount" : 10259 }
```

3.4. What are the names of the restaurants which have a grade at index 8 with score less then 7? Use projection to include only names without _id.

```
> db.restaurants.distinct("name", {"grades.8.score": {$lt: 7}})
[ "Silver Krust West Indian Restaurant", "Pure Food" ]
```

3.5. What are _id and borough of “Seafood” (cuisine) restaurants which received at least one “B” grade in period from 2014-02-01 to 2014-03-01? Use projection to include only _id and borough.



4. Indexing Restaurants Collection

4.1. Create an index which will be used by this query and provide proof (from explain() or Compass UI) that the index is indeed used by the winning plan: db.restaurants.find({ name: "Glorious Food" })

```
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.restaurants.find({name: "Gloriius Food"}).explain()
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "name" : {
                                "$eq" : "Gloriius Food"
                        }
                },
                "winningPlan" : {
                        "stage" : "FETCH",
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "name" : 1
                                },
                                "indexName" : "name_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "name" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : false,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "name" : [
                                                "[\"Gloriius Food\", \"Gloriius Food\"]"
                                        ]
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "EPBYMINW5062",
                "port" : 27017,
                "version" : "4.0.4",
                "gitVersion" : "f288a3bdf201007f3693c58e140056adf8b04839"
        },
        "ok" : 1
}
```


4.2. Drop index from task 4.1

```
> db.restaurants.dropIndex("name_ 1")
{
        "ok" : 0,
        "errmsg" : "index not found with name [name_ 1]",
        "code" : 27,
        "codeName" : "IndexNotFound"
}
> db.restaurants.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.restaurants.find({name: "Gloriius Food"}).explain()
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "name" : {
                                "$eq" : "Gloriius Food"
                        }
                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "name" : {
                                        "$eq" : "Gloriius Food"
                                }
                        },
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "EPBYMINW5062",
                "port" : 27017,
                "version" : "4.0.4",
                "gitVersion" : "f288a3bdf201007f3693c58e140056adf8b04839"
        },
        "ok" : 1
}
```

4.3. Create an index to make this query covered and provide proof (from explain() or Compass UI) that it is indeed covered: db.restaurants.find({ restaurant_id: "41098650" }, { _id: 0, borough: 1 })

```
> db.restaurants.createIndex({restaurant_id: 1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.restaurants.find({restaurant_id: "41098650"}, {_id: 0, borough: 1}).explain()
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "restaurant_id" : {
                                "$eq" : "41098650"
                        }
                },
                "winningPlan" : {
                        "stage" : "PROJECTION",
                        "transformBy" : {
                                "_id" : 0,
                                "borough" : 1
                        },
                        "inputStage" : {
                                "stage" : "FETCH",
                                "inputStage" : {
                                        "stage" : "IXSCAN",
                                        "keyPattern" : {
                                                "restaurant_id" : 1
                                        },
                                        "indexName" : "restaurant_id_1",
                                        "isMultiKey" : false,
                                        "multiKeyPaths" : {
                                                "restaurant_id" : [ ]
                                        },
                                        "isUnique" : false,
                                        "isSparse" : false,
                                        "isPartial" : false,
                                        "indexVersion" : 2,
                                        "direction" : "forward",
                                        "indexBounds" : {
                                                "restaurant_id" : [
                                                        "[\"41098650\", \"41098650\"]"
                                                ]
                                        }
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "EPBYMINW5062",
                "port" : 27017,
                "version" : "4.0.4",
                "gitVersion" : "f288a3bdf201007f3693c58e140056adf8b04839"
        },
        "ok" : 1
```

4.4. Create a partial index on cuisine field which will be used only when filtering on borough equal to “Staten Island”: db.restaurants.find({ borough: "Staten Island", cuisine: "American" }) – uses index db.restaurants.find({ borough: "Staten Island", name: "Bagel Land" }) – does not use index db.restaurants.find({ borough: "Queens", cuisine: "Pizza" }) – does not use index

```
> db.restaurants.createIndex({cuisine: 1}, {partialFilterExpression: {borough: "Staten Island"}})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
>
```

4.5. Create an index to make query from task 3.4 covered and provide proof (from explain() or Compass UI) that it is indeed covered 5.

```
> db.restaurants.find({borough: "Staten Island", cuisine: "American"}).explain()
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "$and" : [
                                {
                                        "borough" : {
                                                "$eq" : "Staten Island"
                                        }
                                },
                                {
                                        "cuisine" : {
                                                "$eq" : "American"
                                        }
                                }
                        ]
                },
                "winningPlan" : {
                        "stage" : "FETCH",
                        "filter" : {
                                "borough" : {
                                        "$eq" : "Staten Island"
                                }
                        },
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "cuisine" : 1
                                },
                                "indexName" : "cuisine_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "cuisine" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : true,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "cuisine" : [
                                                "[\"American\", \"American\"]"
                                        ]
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "EPBYMINW5062",
                "port" : 27017,
                "version" : "4.0.4",
                "gitVersion" : "f288a3bdf201007f3693c58e140056adf8b04839"
        },
        "ok" : 1
```