MongoDB – Complex Queries

# Mongo DB Exercises - With the Restaurants Data Set

1. Download the restaurants.zip file
2. Unzip the file, you will see restaurants.json file
3. Run the mongod server
4. Run the following command to import the json file provided. It will load the json file into the mongodb with database name - restaurants, collections name - addresses

# mongoimport --db restaurants --collection addresses --file restaurants.json

1. Run mongo shell command
2. show databases
3. use restaurants
4. db.addresses.find() should print entire json data
5. Then start working on the following exercises and submit your queries as the answers to the questions

# Query Reference Links and Cheat sheets

1. <https://docs.mongodb.com/manual/crud/>

# Exercise Questions

1. Write a MongoDB query to display all the documents in the collection restaurants.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find()

[

{

\_id: ObjectId("61db1b08dbcc722135b5b287"),

address: {

building: '1007',

coord: [ -73.856077, 40.848447 ],

street: 'Morris Park Ave',

zipcode: '10462'

},

[

Type "it" for more

1. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1,"\_id":0});

[

{

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

]

Type "it" for more

1. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1,"\_id":0});

[

{

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

]

Type "it" for more

1. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"\_id":0});

[

{

address: { zipcode: '10462' },

borough: 'Bronx',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

]

Type "it" for more

1. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"\_id":0});

[

{

address: { zipcode: '10462' },

borough: 'Bronx',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

1. Write a MongoDB query to display all the restaurant which is in the borough Bronx.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"borough": "Bronx"}).skip(5).limit(5);

[

{

\_id: ObjectId("61db1b08dbcc722135b5b2c4"),

address: {

building: '658',

coord: [ -73.81363999999999, 40.82941100000001 ],

street: 'Clarence Ave',

zipcode: '10465'

},

borough: 'Bronx',

cuisine: 'American ',

grades:

1. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({grades : { $elemMatch:{"score":{$gt : 90}}}});

[

{

\_id: ObjectId("61db1b08dbcc722135b5b3e5"),

address: {

building: '65',

coord: [ -73.9782725, 40.7624022 ],

street: 'West 54 Street',

zipcode: '10019'

},

1. Write a MongoDB query to find the restaurants who achieved a score more than 90.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({grades : { $elemMatch:{"score":{$gt : 90}}}});

[

{

\_id: ObjectId("61db1b08dbcc722135b5b3e5"),

address: {

building: '65',

coord: [ -73.9782725, 40.7624022 ],

street: 'West 54 Street',

zipcode: '10019'

},

borough: 'Manhattan',

cuisine: 'American ',

grades:

1. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});

[

{

\_id: ObjectId("61db1b08dbcc722135b5b486"),

address: {

building: '345',

coord: [ -73.9864626, 40.7266739 ],

street: 'East 6 Street',

zipcode: '10003'

},

borough: 'Manhattan',

cuisine: 'Indian',

grades:

1. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"address.coord" : {$lt : -95.754168}});

1. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({$and: [{"cuisine" : {$ne :"American "}},{"grades.score": {$gt : 70}},{"address.coord": {$lt : -65.754168}}]});

1. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"cuisine" : {$ne : "American "},"grades.score" :{$g$gt: 70}, "address.coord" :{$lt : -65.754168}});

1. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find( {"cuisine" : {$ne : "American "},"grades.grade" :"A","borough": {$ne : "Brooklyn"}} ).sort({"cuisine":-1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({name: /^Wil/},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({name: /ces$/},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"name": /.\*Reg.\*/},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({ "borough": "Bronx" , $or : [{ "cuisine" : "American " },{ "cuisine" : "Chinese" }] } );

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"grades.score" : { $not: {$gt : 10}}},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({$or: [{name: /^Wil/},{"$and": [{"cuisine" : {$ne :"American "}},{"cuisine" : {$ne :"Chinees"}}]}]},{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates..

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find( {"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A" , "grades.score" : 11}, {"restaurant\_id" : 1,"name":1,"grades":1});

1. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({ "grades.1.date": ISODate("2014-08-11T00:00:00Z"), "grades.1.grade":"A" , "grades.1.score" : 9}, {"restaurant\_id" : 1,"name":1,"grades":1});

1. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52..

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find( { "address.coord.1": {$gt : 42, $lte : 52}},{"restaurant\_id" : 1,"name":1,"address":1,"coord":1});

1. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find().sort({"name":1});

1. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.restaurants.find().sort({"name":-1});db.restaurants.find().sort({"name":-1});

1. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find().sort({"cuisine":1,"borough" : -1,});

1. Write a MongoDB query to know whether all the addresses contains the street or not.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"address.street" : { $exists : true } } );db.addresses.find({"address.street" : { $exists : true } } );

1. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({"address.coord" : {$type : 1}} );

1. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find( {"grades.score" : {$mod : [7,0]}},{"restaurant\_id" : 1,"name":1,"grades":1});

1. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({ name :{ $regex : "mon.\*", $options: "i" } },{"name":1,"borough":1, "address.coord":1,"cuisine" :1});

1. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

Atlas atlas-ojy0k0-shard-0 [primary] restaurants> db.addresses.find({ name :{ $regex : /^Mad/i, }},{"name":1,"borough":1,"address.coord":1,"cuisine" :1});

Happy Coding!!!