MongoDB–ComplexQueries

MongoDBExercises-WiththeRestaurantsDataSet

Downloadtherestaurants.zipfile

Unzipthefile, youwill seerestaurants.jsonfile

Runthemongodserver

Run the following command to import the json file provided. It will load thejson file into the mongodb with database name - restaurants, collectionsname-addresses

mongoimport--dbrestaurants--collectionaddresses--filerestaurants.json

Runmongoshellcommand

showdatabases

userestaurants

db.addresses.find()shouldprintentirejsondata

Then start working on the following exercises and submit your queries as theanswersto thequestions

QueryReferenceLinksandCheatsheets

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

ExerciseQuestions

Write a MongoDB query to display all the documents in the collectionrestaurants.

**db.addresses.find()**

Write a MongoDB query to display the fieldsrestaurant\_id, name, boroughandcuisineforall the documentsinthecollectionrestaurant.

**db.addresses.find({},{restaurant\_id:1, name:1, borough:1,cuisine:1})**

Write a MongoDB query to display the fieldsrestaurant\_id, name, boroughand cuisine, but exclude the field \_id for all the documents in the collectionrestaurant.

**db.addresses.find({},{restaurant\_id:1, name:1, borough:1,\_id:0})**

Write a MongoDB query to display the fieldsrestaurant\_id, name, boroughand zip code, but exclude the field \_id for all the documents in the collectionrestaurant.

**db.addresses.find({},{restaurant\_id:1, name:1, borough:1,"address.zipcode":1,\_id:0})**

Write a MongoDB query to display the first 5 restaurant which is in theboroughBronx.

**db.addresses.find({borough:'Bronx'}).limit(5)**

Write a MongoDB query to display all the restaurant which is in the boroughBronx.

**db.addresses.find({borough:'Bronx'})**

Write a MongoDB query to display the next 5 restaurants after skipping first 5whichareintheborough Bronx.

**db.addresses.find({borough:'Bronx'}).skip(5).limit(5)**

Write a MongoDB query to find the restaurants who achieved a score morethan90.

**db.addresses.find({"grades.score":{$gt:90}})**

Write a MongoDB query to find the restaurants that achieved a score, morethan80but lessthan 100.

**db.addresses.find({"grades.score":{$gt:80,$lt:100}})**

Write a MongoDB query to find the restaurants which locate in latitude valuelessthan -95.754168.

**db.addresses.find({"address.coord.0":{$lt: -95.754168}})**

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

**db.addresses.find({$and:[{cuisine:{$ne:'American'}},{"grades.score":{$gt:70}},{"address.coord.0":{$lt:-65.754168}}]})**

Write a MongoDB query to find the restaurants which do not prepare anycuisine of 'American' and achieved a score more than 70 and located in thelongitudelessthan-65.754168.

**db.addresses.find({$and:[{cuisine:{$ne:'American'}},{"grades.score":{$gt:70}},{"address.coord.1":{$lt:-65.754168}}]})**

Write a MongoDB query to find the restaurants which do not prepare anycuisine of 'American ' and achieved a grade point 'A' not belongs to theboroughBrooklyn.Thedocumentmustbedisplayedaccordingtothecuisineindescending order.

**db.addresses.find({$and:[{cuisine:{$ne:'American'}},{"grades.grade":'A'},{borough:{$ne:'Brooklyn'}}]}).sort({cuisine:-1})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisineforthoserestaurantswhichcontain'Wil'as firstthreelettersforitsname.

**db.addresses.find({name:{$regex: /^Wil/i}},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisineforthoserestaurantswhichcontain'ces'as lastthreelettersforitsname.

**db.addresses.find({name:{$regex: /.\*ces$/}},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

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

**db.addresses.find({name:{$regex: 'Reg'}},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

Write a MongoDB query to find the restaurants which belong to the boroughBronxandpreparedeither American orChinesedish.

**db.addresses.find({$and:[{borough:'Bronx'},{$or:[{cuisine:'American'},{cuisine:'Chinese'}]}]})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisineforthoserestaurantswhichbelongtotheboroughStatenIslandorQueensorBronxorBrooklyn.

**db.addresses.find({$or:[{borough:'Staten Island'},{borough:'Queens'},{borough:'Bronoxr Brooklyn'}]},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisineforthoserestaurantswhicharenotbelongingtotheboroughStatenIslandorQueensor Bronxor Brooklyn.

**db.addresses.find({$or:[{borough:{$ne:'Staten Island'}},{borough:{$ne:'Queens'}},{borough:{ne:'Bronoxr Brooklyn'}}]},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

**db.addresses.find({borough:{$nin:['Staten Island','Queens','Bronoxr Brooklyn']}},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisineforthoserestaurantswhichachievedascore whichis not morethan10.

**db.addresses.find({"grades.score":{$lte:10}},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

Write a MongoDB query to find the restaurant Id, name, borough and cuisinefor those restaurants which prepared dish except 'American' and 'Chinees' orrestaurant'snamebeginswithletter'Wil'.

**db.addresses.find({$nor:[{cuisine:{$in:['American','Chinese']}},{name:{$regex: /^Wil.\*/}}]},{\_id:0,restaurant\_id:1,name:1,borough:1,cuisine:1})**

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

**db.addresses.find({$and:[{"grades.date":ISODate("2014-08-11T00:00:00Z")},{"grades.grade":'A'},{"grades.score":11}]},{\_id:0,restaurant\_id:1,name:1,grades:1})**

Write a MongoDB query to find the restaurant Id, name and grades for thoserestaurants where the 2nd element of grades array contains a grade of "A"andscore9on anISODate"2014-08-11T00:00:00Z"

**db.addresses.find({$and:[{"grades.1.date":ISODate("2014-08-11T00:00:00Z")},{"grades.1.grade":'A'},{"grades.1.score":9}]},{\_id:0,restaurant\_id:1,name:1,grades:1})**

Write a MongoDB query to find the restaurant Id, name, address andgeographical location for those restaurants where 2nd element of coord arraycontainsavaluewhichismorethan 42andupto 52.

**db.addresses.find({$and:[{"address.coord.1":{$gt:42}},{"address.coord.1":{$lte:52}}]},{\_id:0,restaurant\_id:1,name:1,address:1})**

Write a MongoDB query to arrange the name of the restaurants in ascendingorderalong withallthecolumns.

**db.addresses.find().sort({name:1})**

Write a MongoDB query to arrange the name of the restaurants in descendingalongwithallthecolumns.

**db.addresses.find().sort({name:-1})**

Write a MongoDB query to arranged the name of the cuisine in ascendingorderandforthatsamecuisineboroughshouldbeindescendingorder.

**db.addresses.find().sort({cuisine:1,borough:-1})**

WriteaMongoDBqueryto knowwhether alltheaddressescontainsthestreetor not.

**db.addresses.find({'address.street':{$exists:true}})**

Write a MongoDB query which will select all documents in the restaurantscollectionwherethecoordfieldvalueisDouble.

**db.addresses.find({'address.coord':{$type:'double'}})**

Write a MongoDB query which will select the restaurant Id, name and gradesfor those restaurants which returns 0 as a remainder after dividing the scoreby7.

**db.addresses.find({'grades.score':{$mod:[7,0]}},{\_id:0,restaurant\_id:1,name:1,grades:1})**

Write a MongoDB query to find the restaurant name, borough, longitude andattitude and cuisine for those restaurants which contains 'mon' as three letterssomewhereinitsname.

**db.addresses.find({name:{$regex:'mon'}},{\_id:0,name:1,borough:1,cuisine:1,"address.coord":1})**

Write a MongoDB query to find the restaurant name, borough, longitude andlatitude and cuisine for those restaurants which contain 'Mad' as first threelettersof itsname.

**db.addresses.find({name:{$regex:/^Mad.\*/}},{\_id:0,name:1,borough:1,cuisine:1,"address.coord":1})**