**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**

5. Run mongo shell command

C:\WINDOWS\system32>mongosh "mongodb+srv://mongodb-course.ailhn.mongodb.net/myFirstDatabase" --username diksharai

Enter password: \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Current Mongosh Log ID: 61f14ae157cb69db9db9827a

Connecting to: mongodb+srv://mongodb-course.ailhn.mongodb.net/myFirstDatabase?appName=mongosh+1.1.9

Using MongoDB: 4.4.11

Using Mongosh: 1.1.9

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

6. show databases

Atlas atlas-68i9f5-shard-0 [primary] myFirstDatabase> show dbs

mongo\_practice 295 kB

population 1.68 MB

admin 336 kB

local 4.7 GB

7. use restaurants

Atlas atlas-68i9f5-shard-0 [primary] myFirstDatabase> use restaurants

switched to db restaurants

Atlas atlas-68i9f5-shard-0 [primary] restaurants> show dbs

mongo\_practice 295 kB

population 1.68 MB

restaurants 8.19 kB

admin 336 kB

local 4.7 GB

8. db.addresses.find() should print entire json data

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.createCollection("addresses")

{ ok: 1 }

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find()

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa6"),

address: {

building: '1007',

coord: [ -73.856077, 40.848447 ],

street: 'Morris Park Ave',

zipcode: '10462'

},

borough: 'Bronx',

cuisine: 'Bakery',

grades: [

{

date: ISODate("2014-03-03T00:00:00.000Z"),

grade: 'A',

score: 2

},

{

date: ISODate("2013-09-11T00:00:00.000Z"),

grade: 'A',

score: 6

},

{

date: ISODate("2013-01-24T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2011-11-23T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2011-03-10T00:00:00.000Z"),

grade: 'B',

score: 14

}

],

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa7"),

address: {

building: '469',

coord: [ -73.961704, 40.662942 ],

street: 'Flatbush Avenue',

zipcode: '11225'

},

borough: 'Brooklyn',

cuisine: 'Hamburgers',

grades: [

{

date: ISODate("2014-12-30T00:00:00.000Z"),

grade: 'A',

score: 8

},

{

date: ISODate("2014-07-01T00:00:00.000Z"),

grade: 'B',

score: 23

},

{

date: ISODate("2013-04-30T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-05-08T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: "Wendy'S",

restaurant\_id: '30112340'

}]

Type "it" for more

9. Then start working on the following exercises and submit your queries as the answers to the questions

**Exercise Questions**

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> show collections

Addresses

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find()

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa6"),

address: {

building: '1007',

coord: [ -73.856077, 40.848447 ],

street: 'Morris Park Ave',

zipcode: '10462'

},

borough: 'Bronx',

cuisine: 'Bakery',

grades: [

{

date: ISODate("2014-03-03T00:00:00.000Z"),

grade: 'A',

score: 2

},

{

date: ISODate("2013-09-11T00:00:00.000Z"),

grade: 'A',

score: 6

},

{

date: ISODate("2013-01-24T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2011-11-23T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2011-03-10T00:00:00.000Z"),

grade: 'B',

score: 14

}

],

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa7"),

address: {

building: '469',

coord: [ -73.961704, 40.662942 ],

street: 'Flatbush Avenue',

zipcode: '11225'

},

borough: 'Brooklyn',

cuisine: 'Hamburgers',

grades: [

{

date: ISODate("2014-12-30T00:00:00.000Z"),

grade: 'A',

score: 8

},

{

date: ISODate("2014-07-01T00:00:00.000Z"),

grade: 'B',

score: 23

},

{

date: ISODate("2013-04-30T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-05-08T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: "Wendy'S",

restaurant\_id: '30112340'

}]

Type "it" for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa6"),

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa7"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: "Wendy'S",

restaurant\_id: '30112340'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa8"),

borough: 'Manhattan',

cuisine: 'Irish',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa9"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

\_id: ObjectId("61f14cf491be67206f1e9faa"),

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fab"),

borough: 'Queens',

cuisine: 'American ',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fac"),

borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fad"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: "Wilken'S Fine Food",

restaurant\_id: '40356483'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fae"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649'

}]

Type "it" for more

3. 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-68i9f5-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'

},

{

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: "Wendy'S",

restaurant\_id: '30112340'

},

{

borough: 'Manhattan',

cuisine: 'Irish',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841'

},

{

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068'

},

{

borough: 'Queens',

cuisine: 'American ',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151'

},

{

borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442'

},

{

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: "Wilken'S Fine Food",

restaurant\_id: '40356483'

},

{

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649'

},

{

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731'

},

{

borough: 'Bronx',

cuisine: 'American ',

name: 'Wild Asia',

restaurant\_id: '40357217'

},

{

borough: 'Brooklyn',

cuisine: 'American ',

name: 'C & C Catering Service',

restaurant\_id: '40357437'

},

{

borough: 'Brooklyn',

cuisine: 'Chinese',

name: 'May May Kitchen',

restaurant\_id: '40358429'

},

{

borough: 'Manhattan',

cuisine: 'American ',

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480'

}]

Type "it" for more

4. 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-68i9f5-shard-0 [primary] restaurants> Atlas atlas-68i9f5-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'

},

{

address: { zipcode: '11225' },

borough: 'Brooklyn',

name: "Wendy'S",

restaurant\_id: '30112340'

},

{

address: { zipcode: '10019' },

borough: 'Manhattan',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841'

},

{

address: { zipcode: '11224' },

borough: 'Brooklyn',

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

address: { zipcode: '11374' },

borough: 'Queens',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068'

},

{

address: { zipcode: '11369' },

borough: 'Queens',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151'

}]

Type "it" for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.aggregate({$match:{borough:"Bronx"}},{$group:{\_id:{RestaurantName:"$name"}}},{$limit:5})

[

{ \_id: { RestaurantName: "Artie'S" } },

{ \_id: { RestaurantName: 'The Williamsbridge Tavern' } },

{ \_id: { RestaurantName: 'Great Wall Restaurant' } },

{ \_id: { RestaurantName: "Johnny'S O'S" } },

{ \_id: { RestaurantName: 'Fordham University - Student Deli' } }

]

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.aggregate({$match:{borough:"Bronx"}},{$group:{\_id:{RestaurantName:"$name"}}})

[

{ \_id: { RestaurantName: "Zaro'S Bread Basket" } },

{ \_id: { RestaurantName: "Spoto'S Restaurant" } },

{ \_id: { RestaurantName: 'Wembley Athletic Club' } },

{ \_id: { RestaurantName: "Ali'S Roti Shop" } },

{ \_id: { RestaurantName: 'La Perla Mexicana' } },

{ \_id: { RestaurantName: 'Villa Barone' } },

{ \_id: { RestaurantName: 'African Terrace' } },

{ \_id: { RestaurantName: 'New Rainbow Restaurant' } },

{ \_id: { RestaurantName: "Fella'S Bar" } },

{ \_id: { RestaurantName: "John Mulligan'S Fireside Pub" } },

{ \_id: { RestaurantName: "Lolita'S Restaurant" } },

{ \_id: { RestaurantName: 'Terrace Cafe' } },

{ \_id: { RestaurantName: 'Sugar City Bakery' } },

{ \_id: { RestaurantName: 'Cafe Al Mercato' } },

{ \_id: { RestaurantName: 'Yankee Jz Pizza' } },

{ \_id: { RestaurantName: 'Shamrock Inn' } },

{ \_id: { RestaurantName: 'Black Whale' } },

{ \_id: { RestaurantName: 'Woodlawn Cafe' } },

{ \_id: { RestaurantName: "Jimmy Ryan'S" } },

{ \_id: { RestaurantName: 'Seashore Restaurant' } }

]

Type "it" for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.aggregate({$match:{borough:"Bronx"}},{$group:{\_id:{RestaurantName:"$name"}}},{$skip:5},{$limit:5})

[

{ \_id: { RestaurantName: "Mr Mcgoo'S" } },

{ \_id: { RestaurantName: 'Nb. National Bakery' } },

{ \_id: { RestaurantName: "Jackie'S West Indian Bakery" } },

{

\_id: { RestaurantName: 'Feeding Tree Style West Indian Restaurant' }

},

{ \_id: { RestaurantName: 'Aqueduct North' } }

]

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{$gt:90}},{"name":1,"restaurant\_id":1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea104"),

name: "Murals On 54/Randolphs'S",

restaurant\_id: '40372466'

},

{

\_id: ObjectId("61f14cf491be67206f1ea1a5"),

name: 'Gandhi',

restaurant\_id: '40381295'

},

{

\_id: ObjectId("61f14cf491be67206f1ea308"),

name: 'Bella Napoli',

restaurant\_id: '40393488'

}

]

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{"$gt":80,"$lt":100}},{"name":1,"restaurant\_id":1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea104"),

name: "Murals On 54/Randolphs'S",

restaurant\_id: '40372466'

},

{

\_id: ObjectId("61f14cf491be67206f1ea1a5"),

name: 'Gandhi',

restaurant\_id: '40381295'

},

{

\_id: ObjectId("61f14cf491be67206f1ea308"),

name: 'Bella Napoli',

restaurant\_id: '40393488'

},

{

\_id: ObjectId("61f14cf591be67206f1eab74"),

name: 'West 79Th Street Boat Basin Cafe',

restaurant\_id: '40756344'

}

]

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"address.coord.0" : {$lt : -95.754168}})

[

{

\_id: ObjectId("61f14cf491be67206f1ea5ee"),

address: {

building: '3707',

coord: [ -101.8945214, 33.5197474 ],

street: '82 Street',

zipcode: '11372'

},

borough: 'Queens',

cuisine: 'American ',

grades: [

{

date: ISODate("2014-06-04T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2013-11-07T00:00:00.000Z"),

grade: 'B',

score: 19

},

{

date: ISODate("2013-05-17T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2012-08-29T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2012-04-03T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2011-11-16T00:00:00.000Z"),

grade: 'A',

score: 7

}

],

name: 'Burger King',

restaurant\_id: '40534067'

},

{

\_id: ObjectId("61f14cf591be67206f1ea959"),

address: {

building: '15259',

coord: [ -119.6368672, 36.2504996 ],

street: '10 Avenue',

zipcode: '11357'

}]

Type “it” for more

11. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$and : [{"cuisine" : {$ne : "American "}}, {"address.coord.0" : {$lt : -65.754168}}, {"grades.score" : {$gt : 70}}]})

[

{

\_id: ObjectId("61f14cf491be67206f1ea1a5"),

address: {

building: '345',

coord: [ -73.9864626, 40.7266739 ],

street: 'East 6 Street',

zipcode: '10003'

},

borough: 'Manhattan',

cuisine: 'Indian',

grades: [

{

date: ISODate("2014-09-15T00:00:00.000Z"),

grade: 'A',

score: 5

},

{

date: ISODate("2014-01-14T00:00:00.000Z"),

grade: 'A',

score: 8

},

{

date: ISODate("2013-05-30T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2013-04-24T00:00:00.000Z"),

grade: 'P',

score: 2

},

{

date: ISODate("2012-10-01T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2012-04-06T00:00:00.000Z"),

grade: 'C',

score: 92

},

{

date: ISODate("2011-11-03T00:00:00.000Z"),

grade: 'C',

score: 41

}

],

name: 'Gandhi',

restaurant\_id: '40381295'

},

{

\_id: ObjectId("61f14cf491be67206f1ea308"),

address: {

building: '130',

coord: [ -73.984758, 40.7457939 ],

street: 'Madison Avenue',

zipcode: '10016'

},

borough: 'Manhattan',

cuisine: 'Pizza/Italian',

grades: [

{

date: ISODate("2014-12-24T00:00:00.000Z"),

grade: 'Z',

score: 31

},

{

date: ISODate("2014-06-17T00:00:00.000Z"),

grade: 'C',

score: 98

},

{

date: ISODate("2013-12-12T00:00:00.000Z"),

grade: 'C',

score: 32

},

{

date: ISODate("2013-05-22T00:00:00.000Z"),

grade: 'B',

score: 21

},

{

date: ISODate("2012-05-02T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: 'Bella Napoli',

restaurant\_id: '40393488'

}]

Type “it” for more

13. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$and:[{cuisine:{$ne:"American"}},{"grades.grade":"A"},{borough:{$ne:"Brooklyn"}}]},{"name":1,"restaurant\_id":1}).sort({cuisine:-1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea6b2"),

name: 'Thai Son',

restaurant\_id: '40559606'

},

{

\_id: ObjectId("61f14cf491be67206f1ea76b"),

name: 'Pho Bac Vietnamese Seafood Cuisine',

restaurant\_id: '40578058'

},

{

\_id: ObjectId("61f14cf591be67206f1eab54"),

name: 'Nha-Trang Centre Vietnam Restaurant',

restaurant\_id: '40751226'

},

{

\_id: ObjectId("61f14cf491be67206f1ea256"),

name: 'Angelica Kitchen',

restaurant\_id: '40388281'

},

{

\_id: ObjectId("61f14cf491be67206f1ea40b"),

name: 'Candle Cafe',

restaurant\_id: '40399007'

},

{

\_id: ObjectId("61f14cf491be67206f1ea573"),

name: 'Village Yogurt',

restaurant\_id: '40512123'

},

{

\_id: ObjectId("61f14cf491be67206f1ea613"),

name: 'Village Natural',

restaurant\_id: '40536786'

},

{

\_id: ObjectId("61f14cf591be67206f1ea8b4"),

name: 'Caravan Of Dreams',

restaurant\_id: '40635781'

},

{

\_id: ObjectId("61f14cf591be67206f1ea8da"),

name: 'Mana',

restaurant\_id: '40644922'

},

{

\_id: ObjectId("61f14cf591be67206f1ead62"),

name: 'Red Bamboo',

restaurant\_id: '40861201'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fc2"),

name: 'The Country Cafe',

restaurant\_id: '40362715'

},

{

\_id: ObjectId("61f14cf491be67206f1ea41d"),

name: 'Turkish Kitchen',

restaurant\_id: '40399372'

},

{

\_id: ObjectId("61f14cf491be67206f1ea6ad"),

name: 'Uskudar Restaurant',

restaurant\_id: '40559172'

},

{

\_id: ObjectId("61f14cf491be67206f1ea6b1"),

name: 'Pasha Turkish Restaurant',

restaurant\_id: '40559546'

},

{

\_id: ObjectId("61f14cf591be67206f1ea846"),

name: 'Kebab House',

restaurant\_id: '40608102'

},

{

\_id: ObjectId("61f14cf591be67206f1ea956"),

name: 'Turkish Cuisine',

restaurant\_id: '40668464'

},

{

\_id: ObjectId("61f14cf591be67206f1eab40"),

name: 'Turkuaz Turkish Cuisine',

restaurant\_id: '40749081'

},

{

\_id: ObjectId("61f14cf591be67206f1eac6d"),

name: 'Saharas Turkish Cuisine',

restaurant\_id: '40815015'

},

{

\_id: ObjectId("61f14cf491be67206f1ea125"),

name: 'Pongsri Thai Restaurant',

restaurant\_id: '40374088'

},

{

\_id: ObjectId("61f14cf491be67206f1ea2ed"),

name: 'Jaiya Thai Oriental Restaurant',

restaurant\_id: '40392724'

}

]

Type "it" for more

14. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({name:/^Wil/},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fad"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: "Wilken'S Fine Food",

restaurant\_id: '40356483'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb0"),

borough: 'Bronx',

cuisine: 'American ',

name: 'Wild Asia',

restaurant\_id: '40357217'

},

{

\_id: ObjectId("61f14cf591be67206f1eadb5"),

borough: 'Bronx',

cuisine: 'Pizza',

name: 'Wilbel Pizza',

restaurant\_id: '40871979'

}

15. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({name:/ces$/},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea439"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Pieces',

restaurant\_id: '40399910'

},

{

\_id: ObjectId("61f14cf491be67206f1ea4f8"),

borough: 'Queens',

cuisine: 'American ',

name: 'S.M.R Restaurant Services',

restaurant\_id: '40403857'

},

{

\_id: ObjectId("61f14cf491be67206f1ea4fe"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Good Shepherd Services',

restaurant\_id: '40403989'

},

{

\_id: ObjectId("61f14cf591be67206f1ea9b1"),

borough: 'Queens',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: "The Ice Box-Ralph'S Famous Italian Ices",

restaurant\_id: '40690899'

},

{

\_id: ObjectId("61f14cf591be67206f1eabb3"),

borough: 'Brooklyn',

cuisine: 'Jewish/Kosher',

name: 'Alices',

restaurant\_id: '40782042'

},

{

\_id: ObjectId("61f14cf591be67206f1eadcf"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Re: Sources',

restaurant\_id: '40876068'

}

]

16. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({name:/Reg/},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fae"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649'

},

{

\_id: ObjectId("61f14cf491be67206f1ea0ab"),

borough: 'Manhattan',

cuisine: 'Café/Coffee/Tea',

name: 'Caffe Reggio',

restaurant\_id: '40369418'

},

{

\_id: ObjectId("61f14cf491be67206f1ea1ba"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Regency Hotel',

restaurant\_id: '40382679'

},

{

\_id: ObjectId("61f14cf491be67206f1ea4d7"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Regency Whist Club',

restaurant\_id: '40402377'

},

{

\_id: ObjectId("61f14cf491be67206f1ea5ba"),

borough: 'Queens',

cuisine: 'American ',

name: 'Rego Park Cafe',

restaurant\_id: '40523342'

},

{

\_id: ObjectId("61f14cf591be67206f1eac28"),

borough: 'Queens',

cuisine: 'Pizza',

name: 'Regina Pizza',

restaurant\_id: '40801325'

},

{

\_id: ObjectId("61f14cf591be67206f1eae3f"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Regal Entertainment Group',

restaurant\_id: '40891782'

}

]

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({borough:"Bronx",$or:[{cuisine:"American"},{cuisine:"Chinese"}]},{restaurant\_id:1,name:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fc9"),

name: 'Happy Garden',

restaurant\_id: '40363289'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fdb"),

name: 'Happy Garden',

restaurant\_id: '40364296'

},

{

\_id: ObjectId("61f14cf491be67206f1ea553"),

name: 'China Wok Ii',

restaurant\_id: '40510328'

},

{

\_id: ObjectId("61f14cf491be67206f1ea5de"),

name: 'Dragon City',

restaurant\_id: '40529203'

},

{

\_id: ObjectId("61f14cf491be67206f1ea68d"),

name: 'Hunan Balcony',

restaurant\_id: '40551996'

},

{

\_id: ObjectId("61f14cf491be67206f1ea68e"),

name: 'Great Wall Restaurant',

restaurant\_id: '40552226'

},

{

\_id: ObjectId("61f14cf491be67206f1ea720"),

name: 'Lucky House Restaurant',

restaurant\_id: '40571587'

},

{

\_id: ObjectId("61f14cf491be67206f1ea72e"),

name: 'New Wah Kitchen',

restaurant\_id: '40573101'

},

{

\_id: ObjectId("61f14cf591be67206f1ea9ea"),

name: 'New Hing Restaurant',

restaurant\_id: '40701165'

},

{

\_id: ObjectId("61f14cf591be67206f1eabaa"),

name: 'Hong Kong Restaurant',

restaurant\_id: '40765358'

},

{

\_id: ObjectId("61f14cf591be67206f1eac35"),

name: "Kristy'S Restaurant",

restaurant\_id: '40804049'

},

{

\_id: ObjectId("61f14cf591be67206f1eacaf"),

name: 'East Dynasty',

restaurant\_id: '40827529'

},

{

\_id: ObjectId("61f14cf591be67206f1eacf3"),

name: 'Lin Home Chinese Restaura',

restaurant\_id: '40842437'

},

{

\_id: ObjectId("61f14cf591be67206f1ead11"),

name: 'Peacock Restaurant',

restaurant\_id: '40849313'

},

{

\_id: ObjectId("61f14cf591be67206f1ead45"),

name: "Lin'S Garden",

restaurant\_id: '40857406'

},

{

\_id: ObjectId("61f14cf591be67206f1eae5b"),

name: 'New Rainbow Restaurant',

restaurant\_id: '40899178'

}

]

18. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$or:[{borough:"Staten Island"},{borough:"Queens"},{borough:"Bronx"},{borough:"Brooklyn"},]},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa6"),

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa7"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: "Wendy'S",

restaurant\_id: '30112340'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa9"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

\_id: ObjectId("61f14cf491be67206f1e9faa"),

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068'

}]

Type “it” for more

19. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({borough:{$nin:["Staten Island","Queens","Bronx","Brooklyn"]}},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa8"),

borough: 'Manhattan',

cuisine: 'Irish',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb3"),

borough: 'Manhattan',

cuisine: 'American ',

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb8"),

borough: 'Manhattan',

cuisine: 'American ',

name: 'Glorious Food',

restaurant\_id: '40361521'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fbb"),

borough: 'Manhattan',

cuisine: 'Delicatessen',

name: "Bully'S Deli",

restaurant\_id: '40361708'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fbd"),

borough: 'Manhattan',

cuisine: 'Chicken',

name: "Harriet'S Kitchen",

restaurant\_id: '40362098'

}]

Type “it” for more

20. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"grades.score":{$not:{$gt:10}}},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[ name: 'Angelika Film Center',

{ restaurant\_id: '40362274'

\_id: ObjectId("61f14cf491be67206f1e9fb1"),

borough: 'Brooklyn',

cuisine: 'American ',f491be67206f1e9fc2"),

name: 'C & C Catering Service',

restaurant\_id: '40357437'

},name: 'The Country Cafe',

{ restaurant\_id: '40362715'

\_id: ObjectId("61f14cf491be67206f1e9fb3"),

borough: 'Manhattan',

cuisine: 'American ',f491be67206f1e9fc4"),

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480'

},name: 'Downtown Deli',

{ restaurant\_id: '40363021'

\_id: ObjectId("61f14cf491be67206f1e9fb7"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',1be67206f1e9fc8"),

name: 'Nordic Delicacies',

restaurant\_id: '40361390'

},name: "Olive'S",

{ restaurant\_id: '40363151'

\_id: ObjectId("61f14cf491be67206f1e9fc0"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',491be67206f1e9fca"),

name: 'White Castle',

restaurant\_id: '40362344'

},name: 'Cafe Metro',

{ restaurant\_id: '40363298'

\_id: ObjectId("61f14cf491be67206f1e9fd3"),

borough: 'Brooklyn',

cuisine: 'American ',f491be67206f1e9fcc"),

name: "Sonny'S Heros",

restaurant\_id: '40363744'ds/Mixed Buffet',

},name: 'Lexler Deli',

{ restaurant\_id: '40363426'

\_id: ObjectId("61f14cf491be67206f1e9fe3"),

borough: 'Bronx',

cuisine: 'American ',f491be67206f1e9fd0"),

name: 'Manhem Club',,

restaurant\_id: '40364363'

},name: "Lorenzo & Maria'S",

{ restaurant\_id: '40363630'

\_id: ObjectId("61f14cf491be67206f1e9ff1"),

borough: 'Staten Island',

cuisine: 'American ',f491be67206f1e9fd1"),

name: 'Great Kills Yacht Club',

restaurant\_id: '40364610'

},name: "Domino'S Pizza",

{ restaurant\_id: '40363644'

\_id: ObjectId("61f14cf491be67206f1e9ff8"),

borough: 'Manhattan',

cuisine: 'American ',f491be67206f1e9fd2"),

name: 'Serendipity 3',

restaurant\_id: '40364863'

},name: 'Berkely',

{ restaurant\_id: '40363685'

\_id: ObjectId("61f14cf491be67206f1e9ffc"),

borough: 'Manhattan',

cuisine: 'American ',f491be67206f1e9fd6"),

name: 'White Horse Tavern',

restaurant\_id: '40364958'

},name: "Domino'S Pizza",

Type “it” for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$or:[{name:/^Wil/},{$and:[{"cuisine":{$ne:"American"}},{"cuisine":{$ne:"Chinese"}}]}]},{restaurant\_id:1,name:1,borough:1,cuisine:1})

[

{

\_id: ObjectId("61f14cf491be67206f1e9fa6"),

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa7"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: "Wendy'S",

restaurant\_id: '30112340'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa8"),

borough: 'Manhattan',

cuisine: 'Irish',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fa9"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

\_id: ObjectId("61f14cf491be67206f1e9faa"),

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fab"),

borough: 'Queens',

cuisine: 'American ',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fac"),

borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fad"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: "Wilken'S Fine Food",

restaurant\_id: '40356483'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fae"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649'

},

{

\_id: ObjectId("61f14cf491be67206f1e9faf"),

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb0"),

borough: 'Bronx',

cuisine: 'American ',

name: 'Wild Asia',

restaurant\_id: '40357217'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb1"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'C & C Catering Service',

restaurant\_id: '40357437'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb3"),

borough: 'Manhattan',

cuisine: 'American ',

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480'

}]

Type “it” for more

22. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$and:[{"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11}]},{restaurant\_id:1,name:1,grades:1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea024"),

grades: [

{

date: ISODate("2014-08-11T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2013-07-22T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2013-03-14T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-07-02T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2012-02-02T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2011-08-24T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: "Neary'S Pub",

restaurant\_id: '40365871'

},

{

\_id: ObjectId("61f14cf491be67206f1ea0ff"),

grades: [

{

date: ISODate("2014-08-11T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2013-12-10T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2013-06-10T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-06-08T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2012-01-25T00:00:00.000Z"),

grade: 'A',

score: 8

},

{

date: ISODate("2011-09-13T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: 'Don Filippo Restaurant',

restaurant\_id: '40372417'

}]

Type “it” for more

23. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$and:[{"grades.1.grade":"A"},{"grades.score":9},{"grades.1.date":ISODate("2014-08-11T00:00:00Z")}]},{restaurant\_id:1,name:1,grades:1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea5d1"),

grades: [

{

date: ISODate("2015-01-12T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2014-08-11T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2014-01-14T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2013-02-07T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2012-04-30T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: 'Club Macanudo (Cigar Bar)',

restaurant\_id: '40526406'

},

{

\_id: ObjectId("61f14cf591be67206f1ea86e"),

grades: [

{

date: ISODate("2015-01-05T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2014-08-11T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2014-03-04T00:00:00.000Z"),

grade: 'B',

score: 22

},

{

date: ISODate("2013-09-13T00:00:00.000Z"),

grade: 'A',

score: 5

},

{

date: ISODate("2013-01-16T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: "Gene'S Coffee Shop",

restaurant\_id: '40614916'

}

]

24. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({$and:[{"address.coord.1":{$gt:42}},{"address.coord.1":{$lte:52}}]},{restaurant\_id:1,name:1,address:1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea248"),

address: {

building: '47',

coord: [ -78.877224, 42.89546199999999 ],

street: 'Broadway @ Trinity Pl',

zipcode: '10006'

},

name: "T.G.I. Friday'S",

restaurant\_id: '40387990'

},

{

\_id: ObjectId("61f14cf491be67206f1ea274"),

address: {

building: '1',

coord: [ -0.7119979, 51.6514664 ],

street: 'Pennplaza E, Penn Sta',

zipcode: '10001'

},

name: 'T.G.I. Fridays',

restaurant\_id: '40388936'

},

{

\_id: ObjectId("61f14cf491be67206f1ea4cd"),

address: {

building: '3000',

coord: [ -87.86567699999999, 42.61150920000001 ],

street: '47 Avenue',

zipcode: '11101'

},

name: "Di Luvio'S Deli",

restaurant\_id: '40402284'

}]

Type “it” for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find().sort({name:1})

[

{

\_id: ObjectId("61f14cf591be67206f1eac36"),

address: {

building: '129',

coord: [ -73.962943, 40.685007 ],

street: 'Gates Avenue',

zipcode: '11238'

},

borough: 'Brooklyn',

cuisine: 'Italian',

grades: [

{

date: ISODate("2014-03-06T00:00:00.000Z"),

grade: 'A',

score: 5

},

{

date: ISODate("2013-08-29T00:00:00.000Z"),

grade: 'A',

score: 2

},

{

date: ISODate("2013-03-08T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2012-06-27T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2011-11-17T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: '(Lewis Drug Store) Locanda Vini E Olii',

restaurant\_id: '40804423'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fb3"),

address: {

building: '1',

coord: [ -73.96926909999999, 40.7685235 ],

street: 'East 66 Street',

zipcode: '10065'

},

borough: 'Manhattan',

cuisine: 'American ',

grades: [

{

date: ISODate("2014-05-07T00:00:00.000Z"),

grade: 'A',

score: 3

},

{

date: ISODate("2013-05-03T00:00:00.000Z"),

grade: 'A',

score: 4

},

{

date: ISODate("2012-04-30T00:00:00.000Z"),

grade: 'A',

score: 6

},

{

date: ISODate("2011-12-27T00:00:00.000Z"),

grade: 'A',

score: 0

}

],

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480'

},

{

\_id: ObjectId("61f14cf591be67206f1ea7c5"),

address: {

building: '10016',

coord: [ -73.8216636, 40.5838155 ],

street: 'Rockaway Beach Boulevard',

zipcode: '11694'

},

borough: 'Queens',

cuisine: 'Delicatessen',

grades: [

{

date: ISODate("2014-01-28T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2013-08-20T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2013-03-19T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2012-03-06T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2011-03-01T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: '101 Deli',

restaurant\_id: '40591271'

},

{

\_id: ObjectId("61f14cf491be67206f1ea6b9"),

address: {

building: '10018',

coord: [ -74.0343092, 40.6125569 ],

street: '4 Avenue',

zipcode: '11209'

},

borough: 'Brooklyn',

cuisine: 'Italian',

grades: [

{

date: ISODate("2014-12-01T00:00:00.000Z"),

grade: 'A',

score: 5

},

{

date: ISODate("2013-11-29T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2013-06-03T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2012-06-05T00:00:00.000Z"),

grade: 'B',

score: 15

},

{

date: ISODate("2012-01-10T00:00:00.000Z"),

grade: 'A',

score: 10

}

],

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find().pretty().sort({name:-1})

[

{

\_id: ObjectId("61f14cf491be67206f1ea065"),

address: {

building: '6946',

coord: [ -73.8811834, 40.7017759 ],

street: 'Myrtle Avenue',

zipcode: '11385'

},

borough: 'Queens',

cuisine: 'German',

grades: [

{

date: ISODate("2014-09-24T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2014-04-17T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2013-03-12T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2012-10-02T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2012-05-09T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2011-12-28T00:00:00.000Z"),

grade: 'B',

score: 24

}

],

name: 'Zum Stammtisch',

restaurant\_id: '40367377'

},

{

\_id: ObjectId("61f14cf591be67206f1eab9f"),

address: {

building: '107109',

coord: [ -73.9744668, 40.731155 ],

street: 'Avenue C',

zipcode: '10009'

},

borough: 'Manhattan',

cuisine: 'German',

grades: [

{

date: ISODate("2014-03-04T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2013-08-21T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2013-02-27T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2012-06-05T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2011-12-20T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2011-07-15T00:00:00.000Z"),

grade: 'A',

score: 6

}

],

name: 'Zum Schneider',

restaurant\_id: '40763382'

},

{

\_id: ObjectId("61f14cf591be67206f1eadd9"),

address: {

building: '2905',

coord: [ -73.91337299999999, 40.775064 ],

street: '23 Avenue',

zipcode: '11105'

},

borough: 'Queens',

cuisine: 'Greek',

grades: [

{

date: ISODate("2014-09-29T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2014-04-18T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2013-03-19T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-10-03T00:00:00.000Z"),

grade: 'B',

score: 20

},

{

date: ISODate("2012-05-07T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: "Zorba'S",

restaurant\_id: '40877247'

}]

Type “it” for more

27. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({}, {\_id:0, cuisine:1, borough:1}).sort({cuisine: 1, borough: -1})

[

{ borough: 'Manhattan', cuisine: 'Afghan' },

{ borough: 'Manhattan', cuisine: 'Afghan' },

{ borough: 'Manhattan', cuisine: 'Afghan' },

{ borough: 'Manhattan', cuisine: 'Afghan' },

{ borough: 'Queens', cuisine: 'African' },

{ borough: 'Brooklyn', cuisine: 'African' },

{ borough: 'Bronx', cuisine: 'African' },

{ borough: 'Bronx', cuisine: 'African' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' },

{ borough: 'Staten Island', cuisine: 'American ' }

]

Type "it" for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"address.street": {$regex: /Street/}}).pretty()

[

{

\_id: ObjectId("61f14cf491be67206f1e9fc2"),

address: {

building: '60',

coord: [ -74.0085357, 40.70620539999999 ],

street: 'Wall Street',

zipcode: '10005'

},

borough: 'Manhattan',

cuisine: 'Turkish',

grades: [

{

date: ISODate("2014-09-26T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2013-09-18T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2012-09-21T00:00:00.000Z"),

grade: 'A',

score: 9

},

{

date: ISODate("2012-05-09T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: 'The Country Cafe',

restaurant\_id: '40362715'

},

{

\_id: ObjectId("61f14cf491be67206f1e9fe0"),

address: {

building: '1',

coord: [ -73.97166039999999, 40.764832 ],

street: 'East 60 Street',

zipcode: '10022'

},

borough: 'Manhattan',

cuisine: 'American ',

grades: [

{

date: ISODate("2014-10-16T00:00:00.000Z"),

grade: 'B',

score: 24

},

{

date: ISODate("2014-05-02T00:00:00.000Z"),

grade: 'A',

score: 4

},

{

date: ISODate("2013-04-02T00:00:00.000Z"),

grade: 'A',

score: 13

},

{

date: ISODate("2012-10-19T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2012-04-27T00:00:00.000Z"),

grade: 'B',

score: 17

},

{

date: ISODate("2011-11-29T00:00:00.000Z"),

grade: 'A',

score: 11

}

],

name: 'Metropolitan Club',

restaurant\_id: '40364347'

}]

Type “it” for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"address.coord": {$type: "double"}}, {\_id:0,name:1, address:1})

[

{

address: {

building: '2780',

coord: [ -73.98241999999999, 40.579505 ],

street: 'Stillwell Avenue',

zipcode: '11224'

},

name: 'Riviera Caterer'

},

{

address: {

building: '6409',

coord: [ -74.00528899999999, 40.628886 ],

street: '11 Avenue',

zipcode: '11219'

},

name: 'Regina Caterers'

},

{

address: {

building: '469',

coord: [ -73.961704, 40.662942 ],

street: 'Flatbush Avenue',

zipcode: '11225'

},

name: "Wendy'S"

}]

Type “it” for more

30. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({"grades.score": {$mod: [7,0]}},{\_id:0, restaurant\_id:1, name:1, grades:1})

[

{

grades: [

{

date: ISODate("2014-06-10T00:00:00.000Z"),

grade: 'A',

score: 5

},

{

date: ISODate("2013-06-05T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2012-04-13T00:00:00.000Z"),

grade: 'A',

score: 12

},

{

date: ISODate("2011-10-12T00:00:00.000Z"),

grade: 'A',

score: 12

}

],

name: 'Riviera Caterer',

restaurant\_id: '40356018'

},

{

grades: [

{

date: ISODate("2014-02-25T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2013-08-14T00:00:00.000Z"),

grade: 'A',

score: 11

},

{

date: ISODate("2012-08-07T00:00:00.000Z"),

grade: 'A',

score: 7

},

{

date: ISODate("2012-03-26T00:00:00.000Z"),

grade: 'A',

score: 10

},

{

date: ISODate("2011-11-04T00:00:00.000Z"),

grade: 'A',

score: 0

},

{

date: ISODate("2011-06-29T00:00:00.000Z"),

grade: 'A',

score: 4

}

],

name: 'Snack Time Grill',

restaurant\_id: '40363590'

}]

Type “it” for more

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

Atlas atlas-68i9f5-shard-0 [primary] restaurants> db.addresses.find({name:/mon/},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})

[

{

address: { coord: [ -73.9901605, 40.7526176 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: "Delmonico'S Kitchen"

},

{

address: { coord: [ -74.10465599999999, 40.58834 ] },

borough: 'Staten Island',

cuisine: 'American ',

name: 'Richmond County Country Club'

},

{

address: { coord: [ -73.8221418, 40.7272376 ] },

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Shimons Kosher Pizza'

},

{

address: { coord: [ -73.9812843, 40.5947365 ] },

borough: 'Brooklyn',

cuisine: 'Pizza/Italian',

name: 'Lb Spumoni Gardens'

},

{

address: { coord: [ -73.951199, 40.7166026 ] },

borough: 'Brooklyn',

cuisine: 'Italian',

name: "Bamonte'S Restaurant"

},

{

address: { coord: [ -73.98306099999999, 40.7441419 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: "Desmond'S Tavern"

}]

Type “it” for more

32. 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-68i9f5-shard-0 [primary] restaurants> db.addresses.find({name:/^Mad.\*/},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})

[

{

address: { coord: [ -73.9860597, 40.7431194 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Madison Square'

},

{

address: { coord: [ -73.98302199999999, 40.742313 ] },

borough: 'Manhattan',

cuisine: 'Indian',

name: 'Madras Mahal'

},

{

address: { coord: [ -73.98171959999999, 40.7499406 ] },

borough: 'Manhattan',

cuisine: 'French',

name: 'Madison Bistro'

},

{

address: { coord: [ -74.000002, 40.72735 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Madame X'

},

{

address: { coord: [ -73.9717845, 40.6897199 ] },

borough: 'Brooklyn',

cuisine: 'African',

name: 'Madiba'

},

{

address: { coord: [ -73.9040753, 40.9069011 ] },

borough: 'Bronx',

cuisine: 'Italian',

name: "Madison'S"

},

{

address: { coord: [ -73.9886598, 40.7565811 ] },

borough: 'Manhattan',

cuisine: 'Hotdogs',

name: "Madame Tussaud'S"

},

{

address: { coord: [ -73.95623719999999, 40.7761697 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Mad River Bar & Grille'

}

]

Happy Coding!!!