MongoDB – Complex Queries

Mongo DB Exercises - With the Restaurants Data Set

mongoimport --uri mongodb+srv://DB_USER13:Pass123@samplecluster.2pkxf.mongodb.net/restaurants --collection restaurants --type json --file F:\restaurants.json

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 **(used restaurants collection as in que. Collection name is restaurants)**

```
C:\Users\PriyankaY Workspace>mongoimport --uri mongodb+srv://DB_USER13:Pass123@samplecluster.
2pkxf.mongodb.net/restaurants --collection restaurants --type json --file F:\restaurants.json

2021-10-25T18:39:36.952+0530    connected to: mongodb+srv://[**REDACTED**]@samplecluster.2pkx
f.mongodb.net/restaurants
2021-10-25T18:39:39.959+0530    [########################] restaurants.restaurants      1.94M
B/1.94MB (100.0%)
2021-10-25T18:39:40.192+0530    [########################] restaurants.restaurants      1.94M
B/1.94MB (100.0%)
2021-10-25T18:39:40.192+0530    3772 document(s) imported successfully. 0 document(s) failed
to import.

C:\Users\PriyankaY Workspace>
```

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

5. Run mongo shell command
6. show databases
7. use restaurants
8. db.addresses.find() should print entire json data

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find()
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    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'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    address: {
      building: '2206',
      coord: [ -74.1377286, 40.6119572 ],
      street: 'Victory Boulevard',
```

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.
   db.restaurants.find()

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find()
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    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'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    address: {
      building: '2206',
      coord: [ -74.1377286, 40.6119572 ],
      street: 'Victory Boulevard',
```

2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.
   db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({},{"restaurant_id" : 1
,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e8"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e9"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ea"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
```

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.
db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({},{"restaurant_id" : 1
,"name":1,"borough":1,"cuisine" :1,"_id":0})
[
  {
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    borough: 'Bronx',
    cuisine: 'American ',
```

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.
   db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"_id":0})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({},{"restaurant_id" : 1
,"name":1,"borough":1,"address.zipcode" :1,"_id":0})
[
  {
    address: { zipcode: '11225' },
    borough: 'Brooklyn',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    address: { zipcode: '10314' },
    borough: 'Staten Island',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    address: { zipcode: '10019' },
    borough: 'Manhattan',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    address: { zipcode: '11369' },
    borough: 'Queens',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    address: { zipcode: '11224' },
    borough: 'Brooklyn',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    address: { zipcode: '10462' },
    borough: 'Bronx',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    address: { zipcode: '10460' },
    borough: 'Bronx',
```

5. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.
   db.restaurants.find({"borough": "Bronx"})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"borough": "Bronx"})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05eb"),
    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("6176ac900820eb032f4d05ec"),
```

6. Write a MongoDB query to display all the restaurant which is in the borough Bronx.
   db.restaurants.find({"borough": "Bronx"}).limit(5);

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"borough": "Bronx"})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05eb"),
    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
      },
```

7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.
   db.restaurants.find({"borough": "Bronx"}).skip(5).limit(5)

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"borough": "Bronx"}).s
kip(5).limit(5);
[
  {
    _id: ObjectId("6176ac900820eb032f4d061f"),
    address: {
      building: '658',
      coord: [ -73.81363999999999, 40.82941100000001 ],
      street: 'Clarence Ave',
      zipcode: '10465'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-21T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2012-07-11T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ],
    name: 'Manhem Club',
    restaurant_id: '40364363'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d063c"),
    address: {
      building: '2222',
      coord: [ -73.84971759999999, 40.8304811 ],
      street: 'Haviland Avenue',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-12-18T00:00:00.000Z"),
        grade: 'A',
        score: 7
```

8. Write a MongoDB query to find the restaurants who achieved a score more than 90.
   db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 90}}}})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({grades : { $elemMatch:
{"score":{$gt : 90}}}})
[
  {
    _id: ObjectId("6176ac900820eb032f4d0744"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West   54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
```

9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.
db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({grades : { $elemMatch:
{"score":{$gt : 80 , $lt :100}}}})
[
  {
    _id: ObjectId("6176ac900820eb032f4d07e5"),
    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
      },
```

10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.
    db.restaurants.find({"address.coord" : {$lt : -95.754168}})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"address.coord" : {$lt
$lt : -95.754168}})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0c2e"),
    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
```

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

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({$and:[{"cuisine" : {$n
$ne :"American "}},{"grades.score" : {$gt : 70}},{"address.coord" : {$lt : -65.754168}}]})
[
  {
    _id: ObjectId("6176ac900820eb032f4d07e5"),
    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',
```

12. 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.
db.restaurants.find({"cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70}, "address.coord" : {$lt : -65.754168}})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"cuisine" : {$ne : "Am
erican "}, "grades.score" :{$gt: 70}, "address.coord" : {$lt : -65.754168}})
[
  {
    _id: ObjectId("6176ac900820eb032f4d07e5"),
    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',
```

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.
    db.restaurants.find({"cuisine" : {$ne : "American "},"grades.grade" :"A","borough": {$ne : "Brooklyn"}}).sort({"cuisine":-1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find( {"cuisine" : {$ne : "A
merican "},"grades.grade" :"A","borough": {$ne : "Brooklyn"}}).sort({"cuisine":-1})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0d06"),
    address: {
      building: '89',
      coord: [ -73.9995899, 40.7168015 ],
      street: 'Baxter Street',
      zipcode: '10013'
    },
    borough: 'Manhattan',
    cuisine: 'Vietnamese/Cambodian/Malaysia',
    grades: [
      {
        date: ISODate("2014-08-21T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-08-31T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-04-11T00:00:00.000Z"),
        grade: 'C',
        score: 3
      },
      {
        date: ISODate("2012-10-17T00:00:00.000Z"),
        grade: 'A',
        score: 4
      },
      {
        date: ISODate("2012-05-15T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ],
    name: 'Thai Son',
    restaurant_id: '40559606'
  },
  {
```

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.
db.restaurants.find({name: /^Wil/},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({name: /^Wil/},{"restau
rant_id" : 1,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05ec"),
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05f3"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  },
  {
    _id: ObjectId("6176ac930820eb032f4d13f4"),
    borough: 'Bronx',
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
Atlas atlas-1483qj-shard-0 [primary] restaurants>
```

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.
db.restaurants.find({name: /ces$/},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({name: /ces$/},{"restau
rant_id" : 1,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0a77"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Pieces',
    restaurant_id: '40399910'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0b39"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'S.M.R Restaurant Services',
    restaurant_id: '40403857'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0b3e"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Good Shepherd Services',
    restaurant_id: '40403989'
  },
  {
    _id: ObjectId("6176ac920820eb032f4d0fee"),
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: "The Ice Box-Ralph'S Famous Italian Ices",
    restaurant_id: '40690899'
  },
  {
    _id: ObjectId("6176ac930820eb032f4d11f1"),
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Alices',
    restaurant_id: '40782042'
  },
  {
    _id: ObjectId("6176ac930820eb032f4d140c"),
    borough: 'Manhattan',
```

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.
db.restaurants.find({"name": /.*Reg.*/},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"name": /.*Reg.*/},{"r
estaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05ef"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d06f8"),
    borough: 'Manhattan',
    cuisine: 'Café/Coffee/Tea',
    name: 'Caffe Reggio',
    restaurant_id: '40369418'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d07fa"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0b14"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Regency Whist Club',
    restaurant_id: '40402377'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0bf9"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Rego Park Cafe',
    restaurant_id: '40523342'
  },
  {
    _id: ObjectId("6176ac930820eb032f4d1267"),
    borough: 'Queens',
    cuisine: 'Pizza',
```

17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.
db.restaurants.find({ "borough": "Bronx" , $or : [{ "cuisine" : "American " },{ "cuisine" : "Chinese" }] })

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({ "borough": "Bronx" ,
$or : [{ "cuisine" : "American " },{ "cuisine" : "Chinese" }] } )
[
  {
    _id: ObjectId("6176ac900820eb032f4d05ec"),
    address: {
      building: '2300',
      coord: [ -73.8786113, 40.8502883 ],
      street: 'Southern Boulevard',
      zipcode: '10460'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-05-28T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-06-19T00:00:00.000Z"),
        grade: 'A',
        score: 4
      },
      {
        date: ISODate("2012-06-15T00:00:00.000Z"),
        grade: 'A',
        score: 3
      }
    ],
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d0607"),
    address: {
      building: '1236',
      coord: [ -73.8893654, 40.81376179999999 ],
      street: '238 Spofford Ave',
      zipcode: '10474'
    },
    borough: 'Bronx',
```

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.
db.restaurants.find({"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"borough" :{$in :["Sta
ten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine
" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e9"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ea"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05eb"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ec"),
    borough: 'Bronx',
```

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.
db.restaurants.find({"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"borough" :{$nin :["St
aten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisin
e" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e8"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05f2"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05f9"),
    borough: 'Manhattan',
    cuisine: 'Delicatessen',
    name: "Bully'S Deli",
    restaurant_id: '40361708'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05fa"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'P & S Deli Grocery',
    restaurant_id: '40362264'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05fb"),
    borough: 'Manhattan',
    cuisine: 'Chicken',
    name: "Harriet'S Kitchen",
    restaurant_id: '40362098'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05fc"),
    borough: 'Manhattan',
```

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.
db.restaurants.find({"grades.score" : { $not: {$gt : 10}}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"grades.score" : {
$not: {$gt : 10}}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05ed"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05f2"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05f5"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies',
    restaurant_id: '40361390'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05fd"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'White Castle',
    restaurant_id: '40362344'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d0611"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: "Sonny'S Heros",
    restaurant_id: '40363744'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d061f"),
    borough: 'Bronx',
```

21. 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'.
db.restaurants.find({$or: [{name: /^Wil/}, {"$and": [{"cuisine" : {$ne :"American "}}, {"cuisine" : {$ne:"Chinees"}}]}]},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({$or: [  {name: /^Wil/}
,   {"$and": [{"cuisine" : {$ne :"American "}}, {"cuisine" : {$ne:"Chinees"}}]}]},{"restauran
t_id" : 1,"name":1,"borough":1,"cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e8"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05eb"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ec"),
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ee"),
    borough: 'Brooklyn',
```

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..
db.restaurants.find({"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A" , "grades.score" : 11}, {"restaurant_id" : 1,"name":1,"grades":1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find( {"grades.date": ISODat
e("2014-08-11T00:00:00Z"), "grades.grade":"A", "grades.score" : 11}, {"restaurant_id" : 1,"na
me":1,"grades":1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d0666"),
    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'
  },
```

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"
db.restaurants.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})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({ "grades.1.date": ISOD
ate("2014-08-11T00:00:00Z"), "grades.1.grade":"A" , "grades.1.score" : 9}, {"restaurant_id" :
 1,"name":1,"grades":1})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0c0f"),
    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'
  }
]
Atlas atlas-1483qj-shard-0 [primary] restaurants>
```

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.. db.restaurants.find({ "address.coord.1": {$gt : 42, $lte : 52}},{"restaurant_id" : 1,"name":1,"address":1,"coord":1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({ "address.coord.1": {$
$gt : 42, $lte : 52}},{"restaurant_id" : 1,"name":1,"address":1,"coord":1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d088e"),
    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("6176ac900820eb032f4d08b1"),
    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("6176ac910820eb032f4d0b09"),
    address: {
      building: '3000',
      coord: [ -87.86567699999999, 42.61150920000001 ],
      street: '47 Avenue',
      zipcode: '11101'
    },
    name: "Di Luvio'S Deli",
    restaurant_id: '40402284'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0d41"),
    address: {
      building: '21972199',
```

25. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.
db.restaurants.find().sort({"name":1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find().sort({"name":1});
[
  {
    _id: ObjectId("6176ac930820eb032f4d1275"),
    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'
  },
  {
```

26. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.
    db.restaurants.find().sort({"name":-1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find().sort({"name":-1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d06a3"),
    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
```

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.
db.restaurants.find().sort({"cuisine":1,"borough" : -1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find().sort({"cuisine":1,"bo
rough" : -1})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0cd0"),
    address: {
      building: '1345',
      coord: [ -73.959249, 40.768076 ],
      street: '2 Avenue',
      zipcode: '10021'
    },
    borough: 'Manhattan',
    cuisine: 'Afghan',
    grades: [
      {
        date: ISODate("2014-10-07T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2013-10-23T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2012-10-26T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-04-26T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-01-12T00:00:00.000Z"),
        grade: 'P',
        score: 10
      }
    ],
    name: 'Afghan Kebab House',
```

28. Write a MongoDB query to know whether all the addresses contains the street or not.
    db.restaurants.find({"address.street" : { $exists : true } })

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"address.street" : { $
exists : true } } )
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    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'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
    address: {
```

29. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.
db.restaurants.find({"address.coord" : {$type : 1} })

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"address.coord" : {
$type : 1} } )
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e6"),
    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'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05e7"),
```

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.
db.restaurants.find({"grades.score" : {$mod : [7,0]} }, {"restaurant_id" : 1,"name":1,"grades":1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({"grades.score" : {
$mod : [7,0]} }, {"restaurant_id" : 1,"name":1,"grades":1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d05e9"),
    grades: [
      {
        date: ISODate("2014-11-15T00:00:00.000Z"),
        grade: 'Z',
        score: 38
      },
      {
        date: ISODate("2014-05-02T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2013-03-02T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-02-10T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d05ea"),
    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
      },
      {
```

31. 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.
db.restaurants.find({ name : { $regex : "mon.*", $options: "i" } }, {"name":1,"borough":1, "address.coord":1, "cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({ name : { $regex : "mo
n.*", $options: "i" } }, {"name":1,"borough":1, "address.coord":1, "cuisine" :1})
[
  {
    _id: ObjectId("6176ac900820eb032f4d0679"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    _id: ObjectId("6176ac900820eb032f4d0681"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d068d"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d06ba"),
    address: { coord: [ -73.9812843, 40.5947365 ] },
    borough: 'Brooklyn',
    cuisine: 'Pizza/Italian',
    name: 'Lb Spumoni Gardens'
  },
  {
    _id: ObjectId("6176ac900820eb032f4d070a"),
    address: { coord: [ -73.951199, 40.7166026 ] },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    name: "Bamonte'S Restaurant"
  },
  {
    _id: ObjectId("6176ac900820eb032f4d074b"),
    address: { coord: [ -73.924072, 40.76108900000001 ] },
    borough: 'Queens',
    cuisine: 'Greek',
```

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.
db.restaurants.find({ name : { $regex : /^Mad/i, } }, {"name":1, "borough":1, "address.coord":1, "cuisine" :1})

```
Atlas atlas-1483qj-shard-0 [primary] restaurants> db.restaurants.find({ name : { $regex : /^M
ad/i, } }, {"name":1, "borough":1, "address.coord":1, "cuisine" :1})
[
  {
    _id: ObjectId("6176ac910820eb032f4d0b1f"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("6176ac910820eb032f4d0bf1"),
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
    _id: ObjectId("6176ac920820eb032f4d0e9e"),
    address: { coord: [ -74.000002, 40.72735 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madame X'
  },
  {
    _id: ObjectId("6176ac920820eb032f4d0f53"),
    address: { coord: [ -73.98171959999999, 40.7499406 ] },
    borough: 'Manhattan',
    cuisine: 'French',
    name: 'Madison Bistro'
  },
  {
    _id: ObjectId("6176ac920820eb032f4d0fdb"),
    address: { coord: [ -73.9717845, 40.6897199 ] },
    borough: 'Brooklyn',
    cuisine: 'African',
    name: 'Madiba'
  },
  {
    _id: ObjectId("6176ac930820eb032f4d12da"),
    address: { coord: [ -73.9040753, 40.9069011 ] },
    borough: 'Bronx',
    cuisine: 'Italian',
```