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

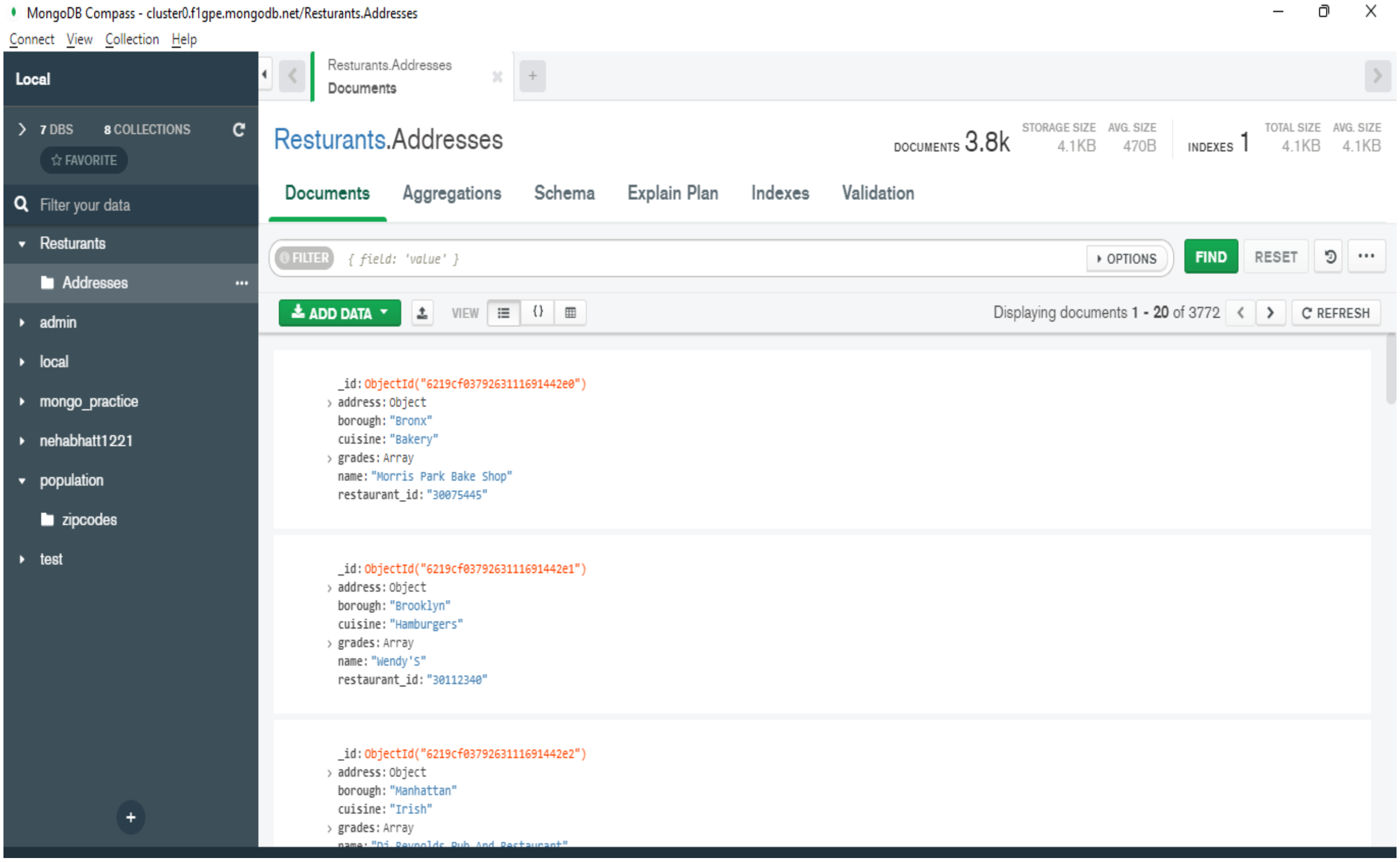
4. Run the following command to import the json file provided. It will load the json file into the mongodb with database name - restaurants, collections name – addresses

5. Run mongo shell command

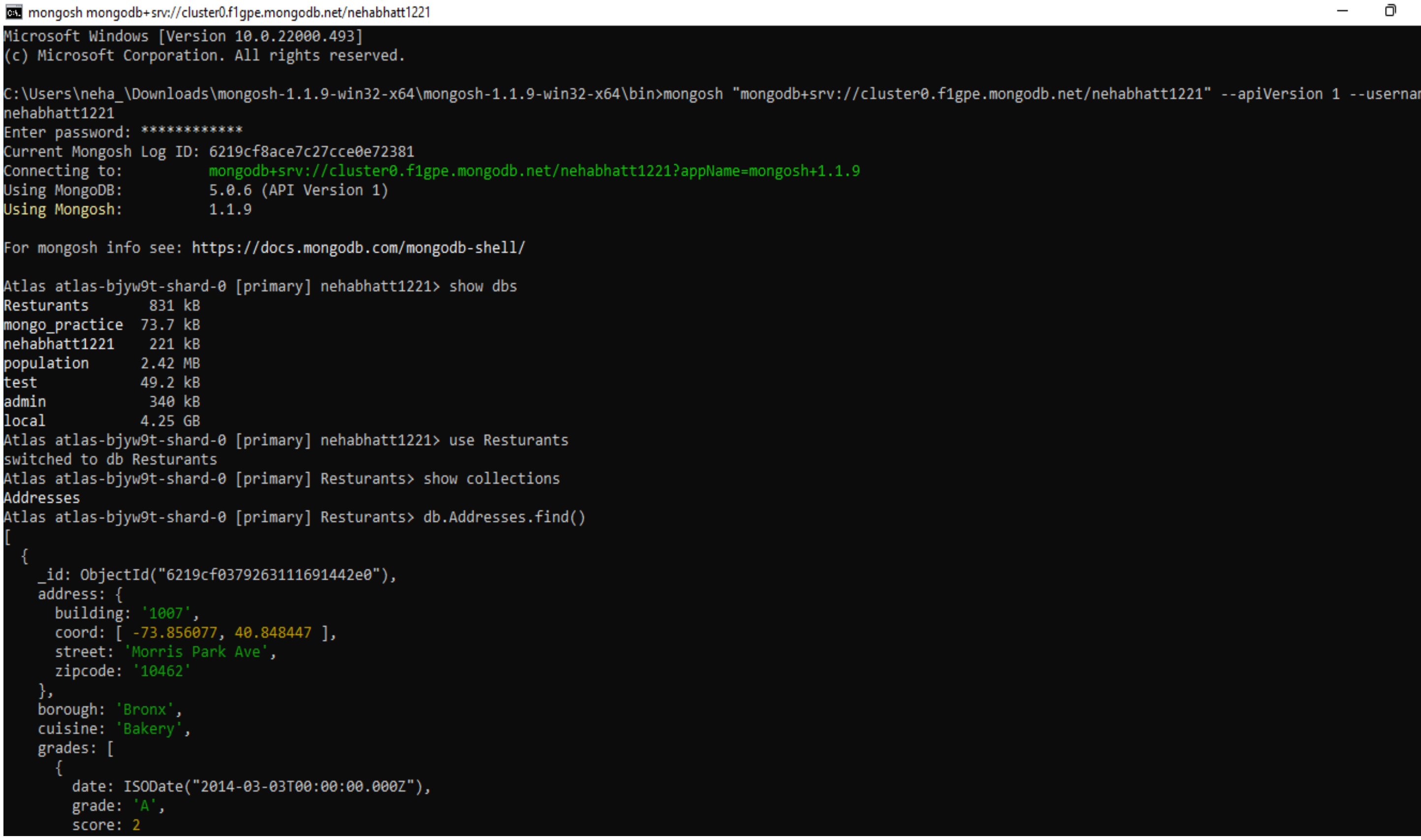
6. show databases 7. use restaurants

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

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



```
  _id: ObjectId("6219cf0379263111691442e0")
> address: Object
  borough: "Bronx"
  cuisine: "Bakery"
> grades: Array
  name: "Morris Park Bake Shop"
  restaurant_id: "30075445"

  _id: ObjectId("6219cf0379263111691442e1")
> address: Object
  borough: "Brooklyn"
  cuisine: "Hamburgers"
> grades: Array
  name: "Wendy'S"
  restaurant_id: "30112340"

  _id: ObjectId("6219cf0379263111691442e2")
> address: Object
  borough: "Manhattan"
  cuisine: "Irish"
> grades: Array
  name: "Dj Reynolds Pub And Restaurant"
```

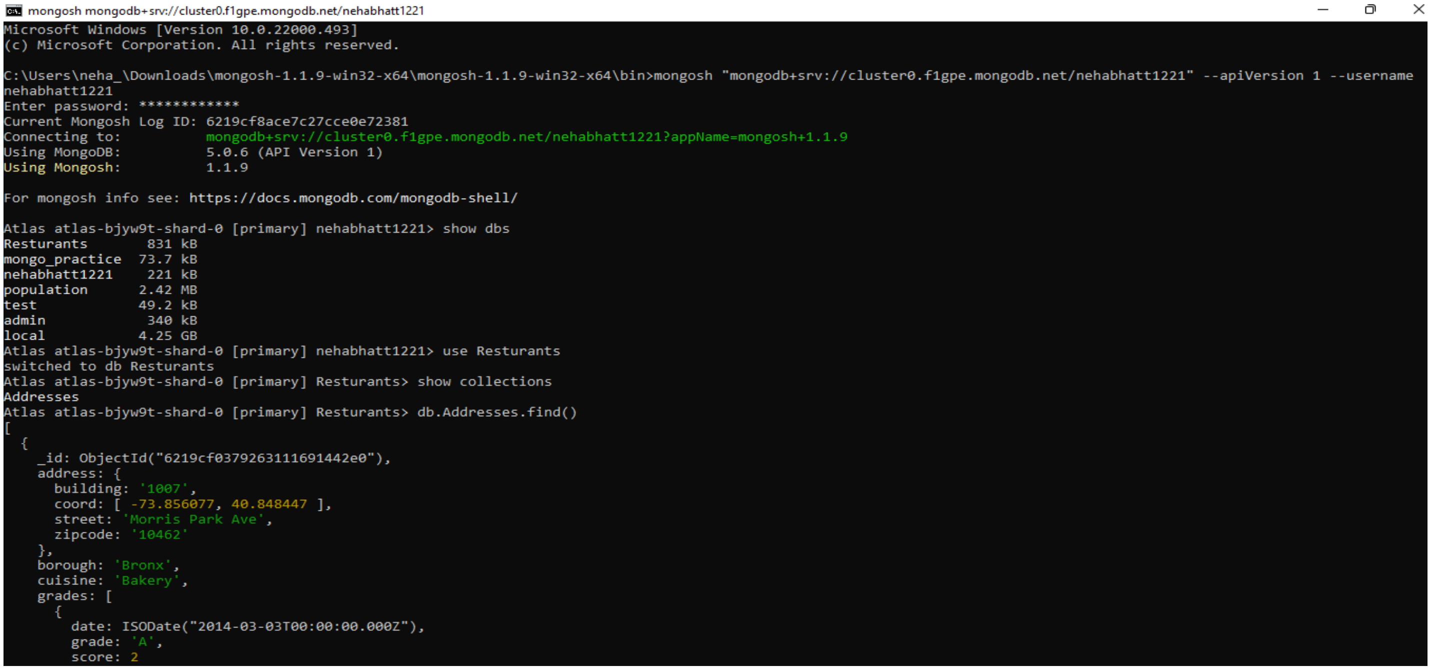


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\neha_\Downloads\mongosh-1.1.9-win32-x64\mongosh-1.1.9-win32-x64\bin>mongosh "mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221" --apiVersion 1 --usernar
nehabhatt1221
Enter password: ************
Current Mongosh Log ID: 6219cf8ace7c27cce0e72381
Connecting to:          mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221?appName=mongosh+1.1.9
Using MongoDB:          5.0.6 (API Version 1)
Using Mongosh:          1.1.9

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

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> show dbs
Resturants       831 kB
mongo_practice  73.7 kB
nehabhatt1221    221 kB
population      2.42 MB
test            49.2 kB
admin            340 kB
local           4.25 GB
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> use Resturants
switched to db Resturants
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> show collections
Addresses
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find()
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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
```

Exercise Questions

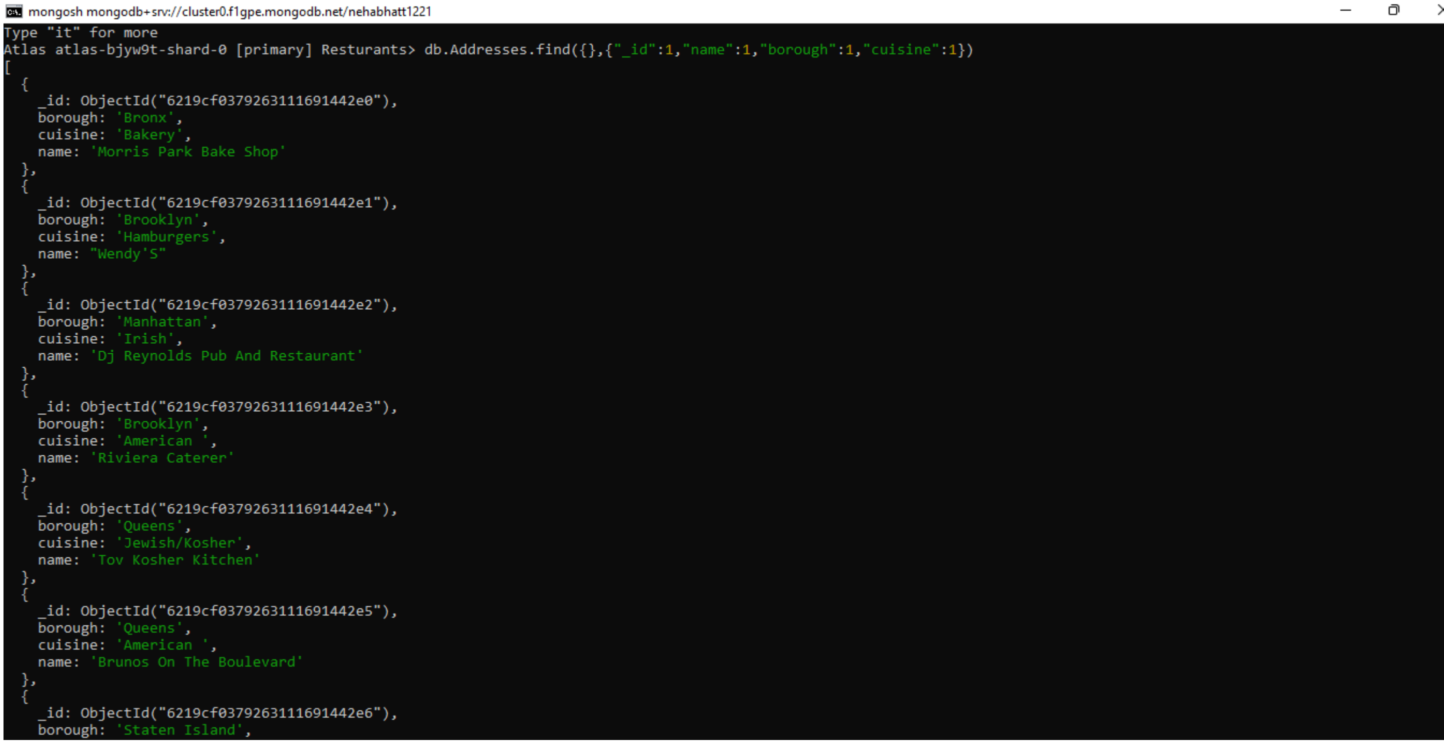
1. Write a MongoDB query to display all the documents in the collection restaurants



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\neha_\Downloads\mongosh-1.1.9-win32-x64\mongosh-1.1.9-win32-x64\bin>mongosh "mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221" --apiVersion 1 --username
nehabhatt1221
Enter password: ************
Current Mongosh Log ID: 6219cf8ace7c27cce0e72381
Connecting to:          mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221?appName=mongosh+1.1.9
Using MongoDB:          5.0.6 (API Version 1)
Using Mongosh:          1.1.9

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

Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> show dbs
Resturants       831 kB
mongo_practice  73.7 kB
nehabhatt1221    221 kB
population      2.42 MB
test            49.2 kB
admin            340 kB
local           4.25 GB
Atlas atlas-bjyw9t-shard-0 [primary] nehabhatt1221> use Resturants
switched to db Resturants
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> show collections
Addresses
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find()
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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
```

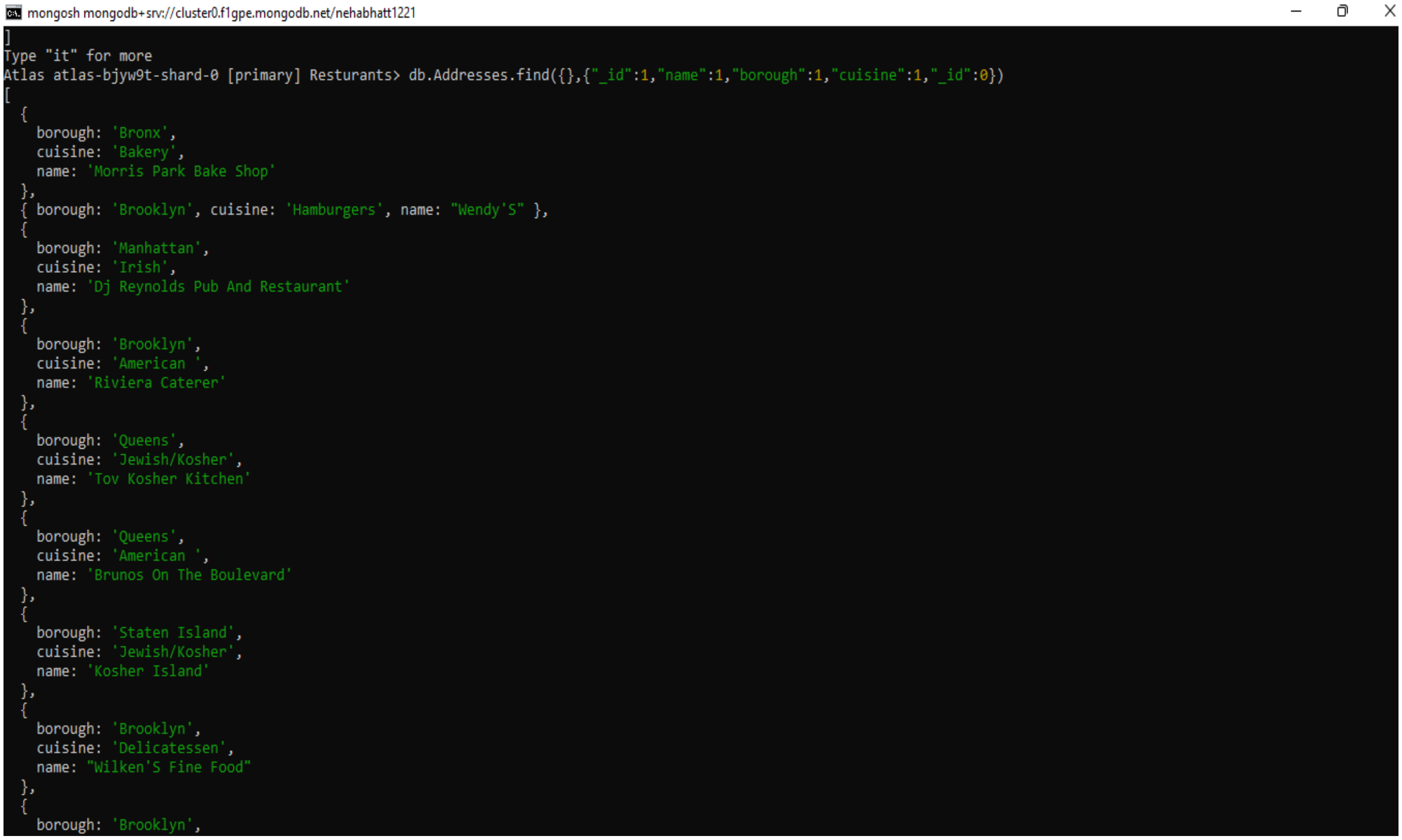
2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({},{"_id":1,"name":1,"borough":1,"cuisine":1})
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },
  {
    _id: ObjectId("6219cf0379263111691442e1"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S"
  },
  {
    _id: ObjectId("6219cf0379263111691442e2"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    _id: ObjectId("6219cf0379263111691442e3"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer'
  },
  {
    _id: ObjectId("6219cf0379263111691442e4"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    _id: ObjectId("6219cf0379263111691442e5"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard'
  },
  {
    _id: ObjectId("6219cf0379263111691442e6"),
    borough: 'Staten Island',
```

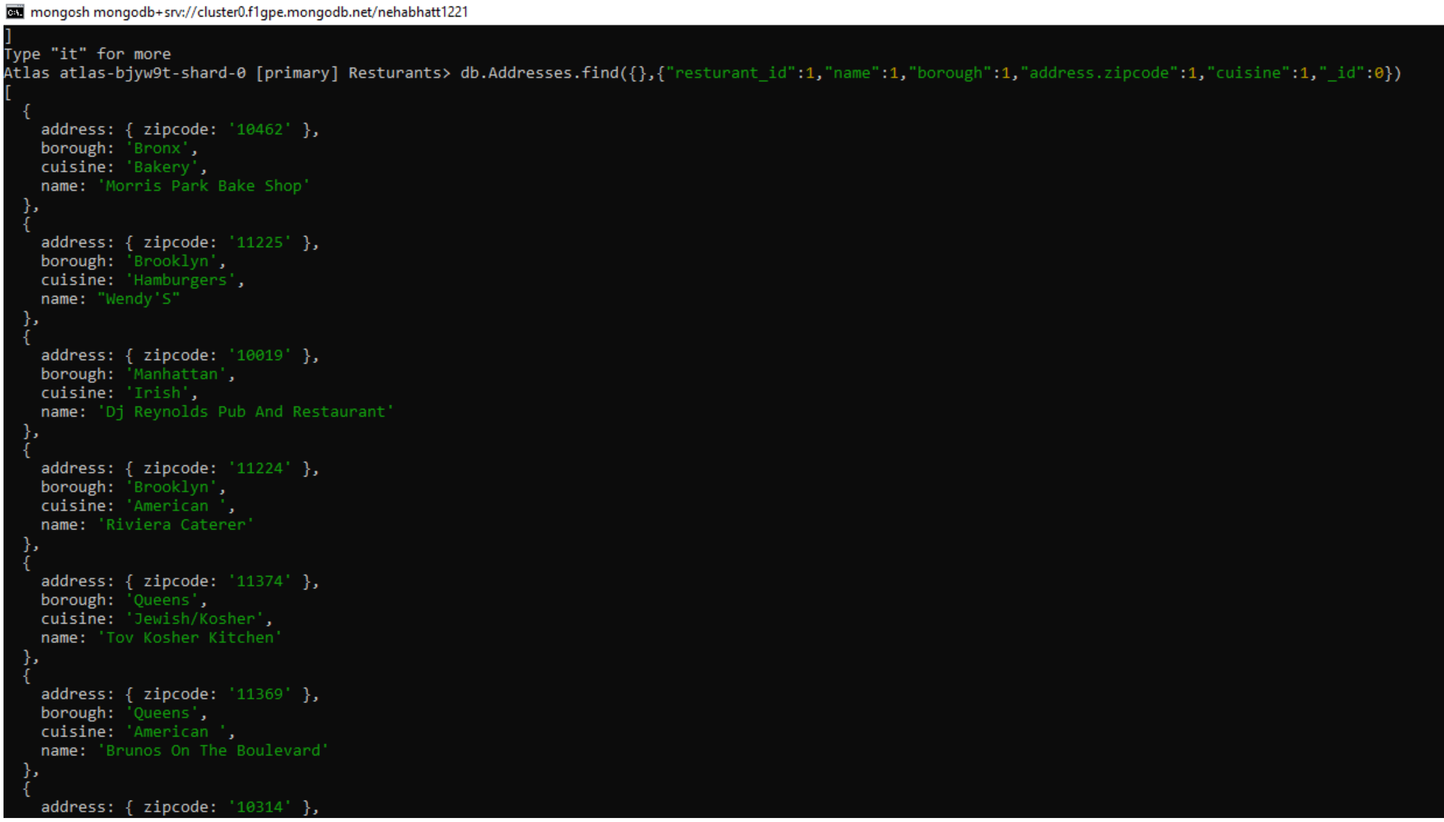
3. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

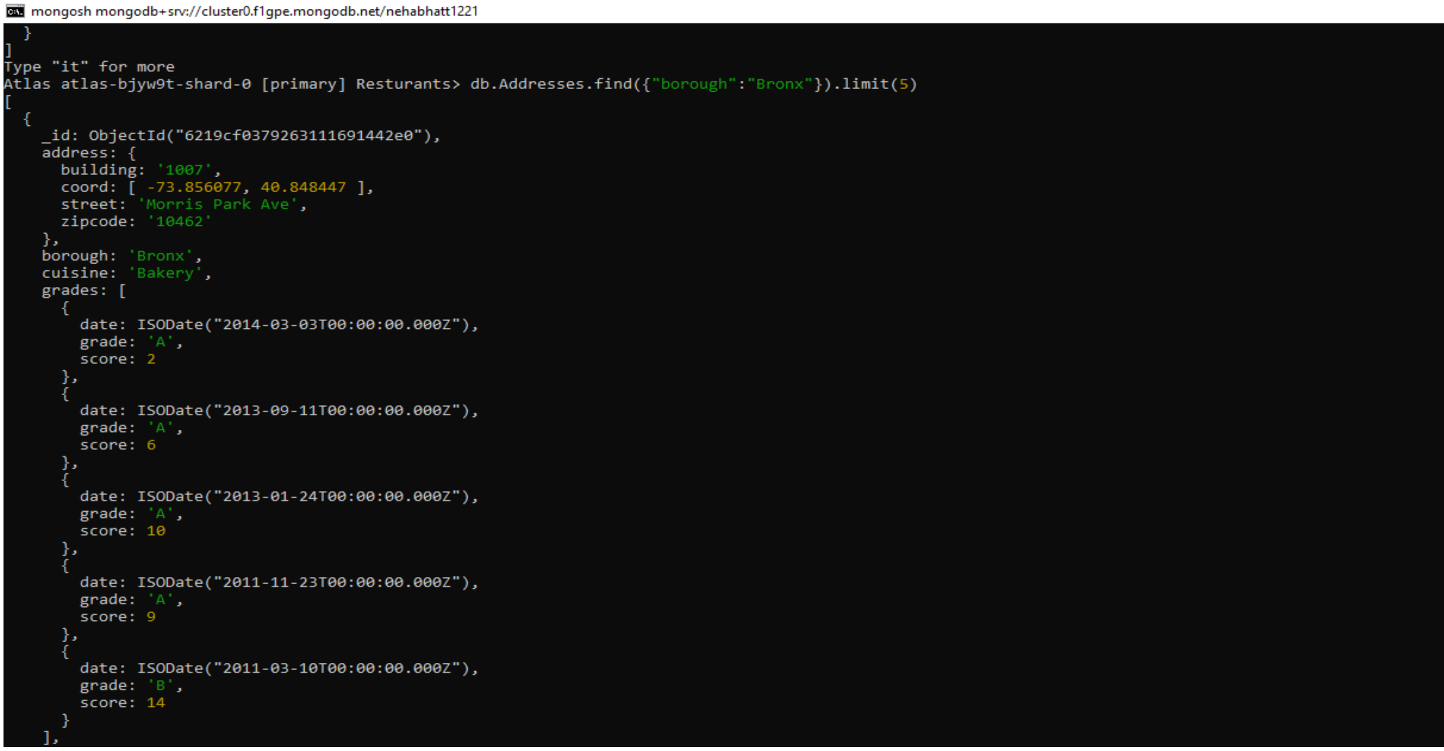
```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({},{"_id":1,"name":1,"borough":1,"cuisine":1,"_id":0})
[
  {
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },
  { borough: 'Brooklyn', cuisine: 'Hamburgers', name: "Wendy'S" },
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer'
  },
  {
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard'
  },
  {
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food"
  },
  {
    borough: 'Brooklyn',
```

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.



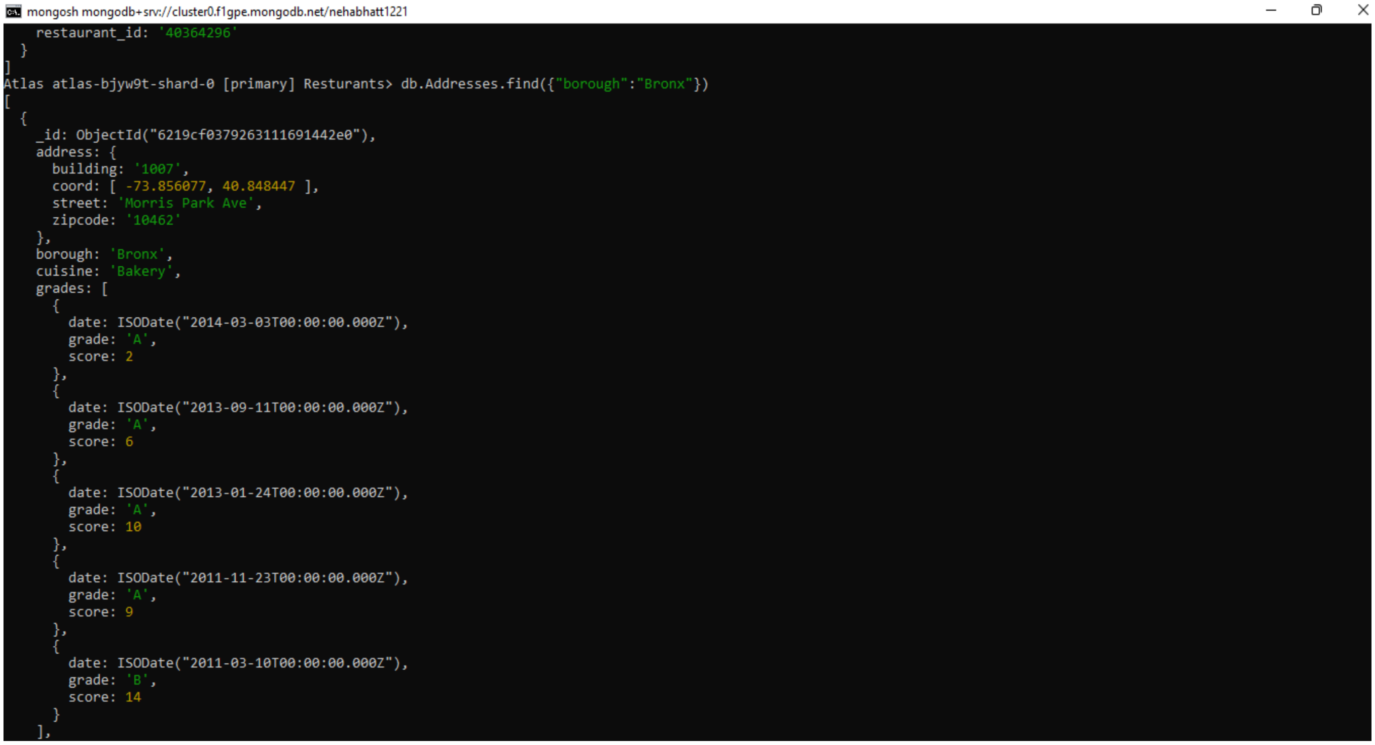
```
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({},{"resturant_id":1,"name":1,"borough":1,"address.zipcode":1,"cuisine":1,"_id":0})
[
  {
    address: { zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },
  {
    address: { zipcode: '11225' },
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S"
  },
  {
    address: { zipcode: '10019' },
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    address: { zipcode: '11224' },
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer'
  },
  {
    address: { zipcode: '11374' },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    address: { zipcode: '11369' },
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard'
  },
  {
    address: { zipcode: '10314' },
```

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



```
mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221
  }
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":"Bronx"}).limit(5)
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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
      }
    ],
```

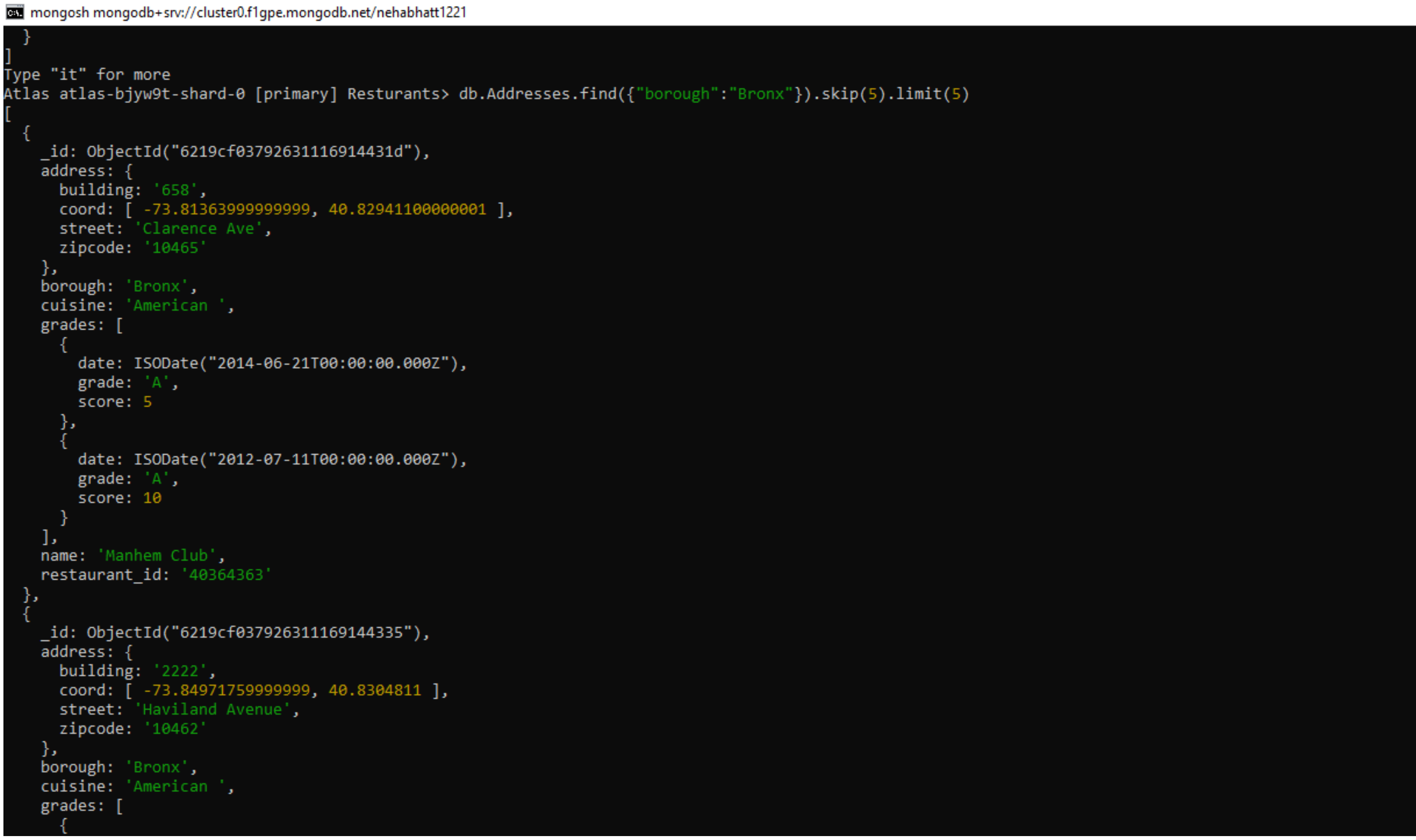
6. Write a MongoDB query to display all the restaurant which is in the borough Bronx.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
    restaurant_id: '40364296'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":"Bronx"})
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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
      }
    ],
```

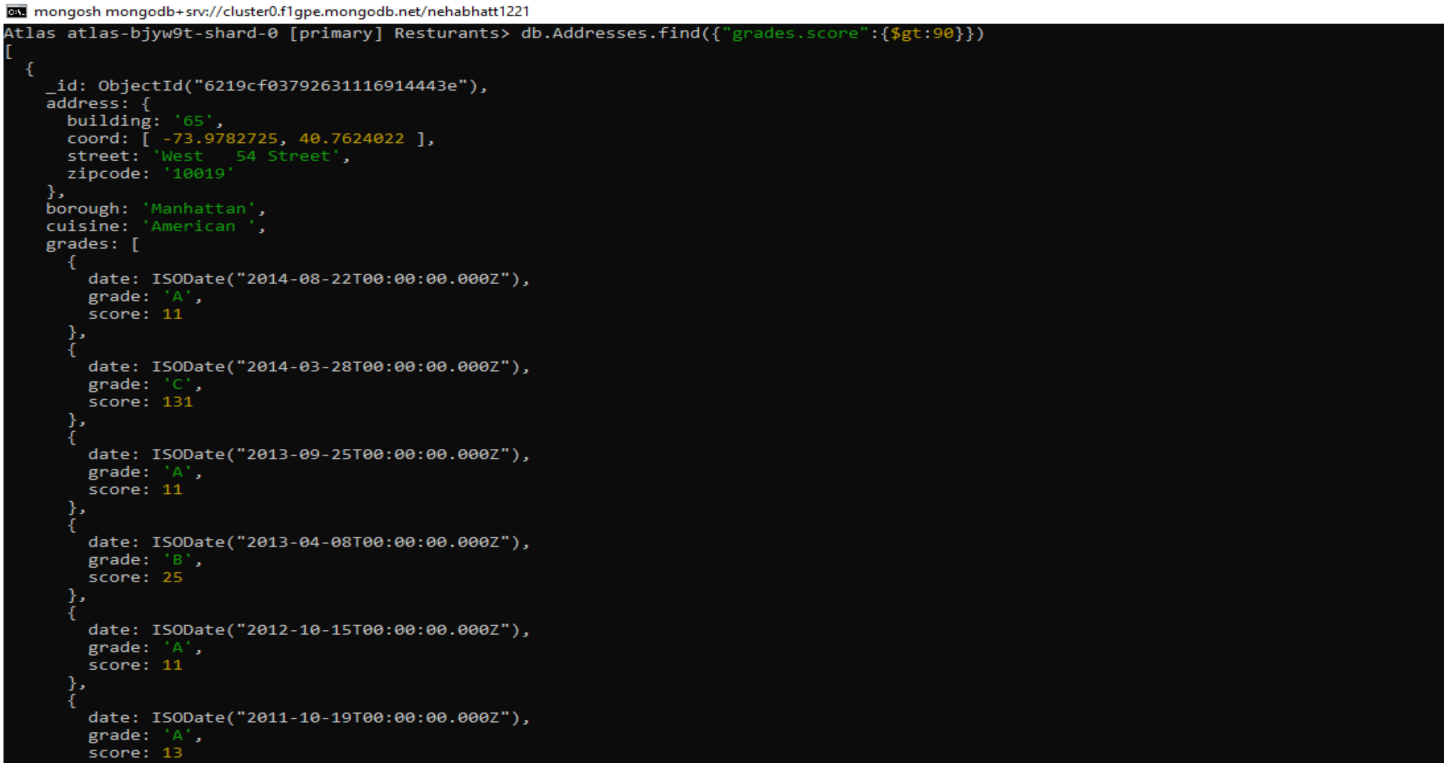
7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  }
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":"Bronx"}).skip(5).limit(5)
[
  {
    _id: ObjectId("6219cf037926311169144331d"),
    address: {
      building: '658',
      coord: [ -73.81363999999999, 40.829411000000001 ],
      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("6219cf0379263111169144335"),
    address: {
      building: '2222',
      coord: [ -73.84971759999999, 40.8304811 ],
      street: 'Haviland Avenue',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
```

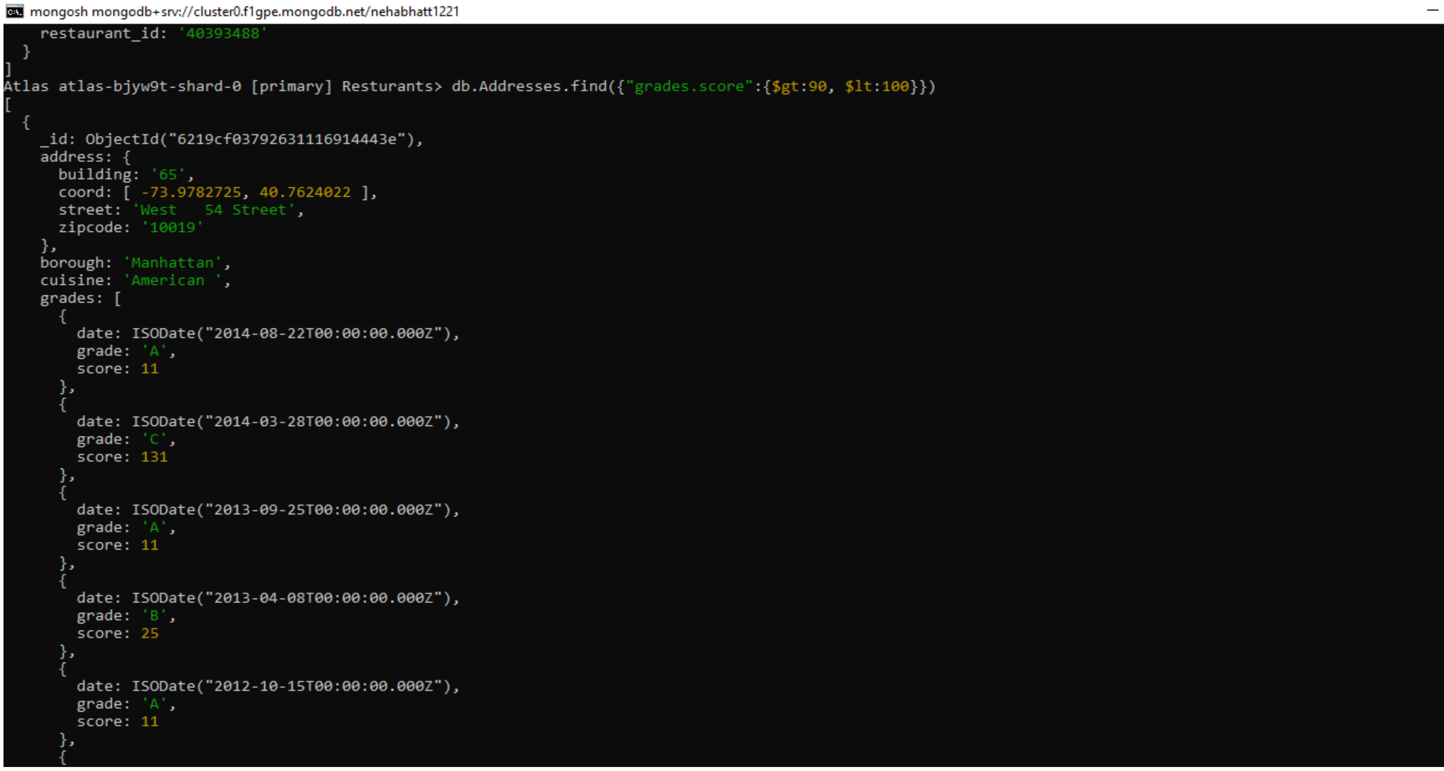
8. Write a MongoDB query to find the restaurants who achieved a score more than 90.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.score":{$gt:90}})
[
  {
    _id: ObjectId("6219cf03792631116914443e"),
    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
```

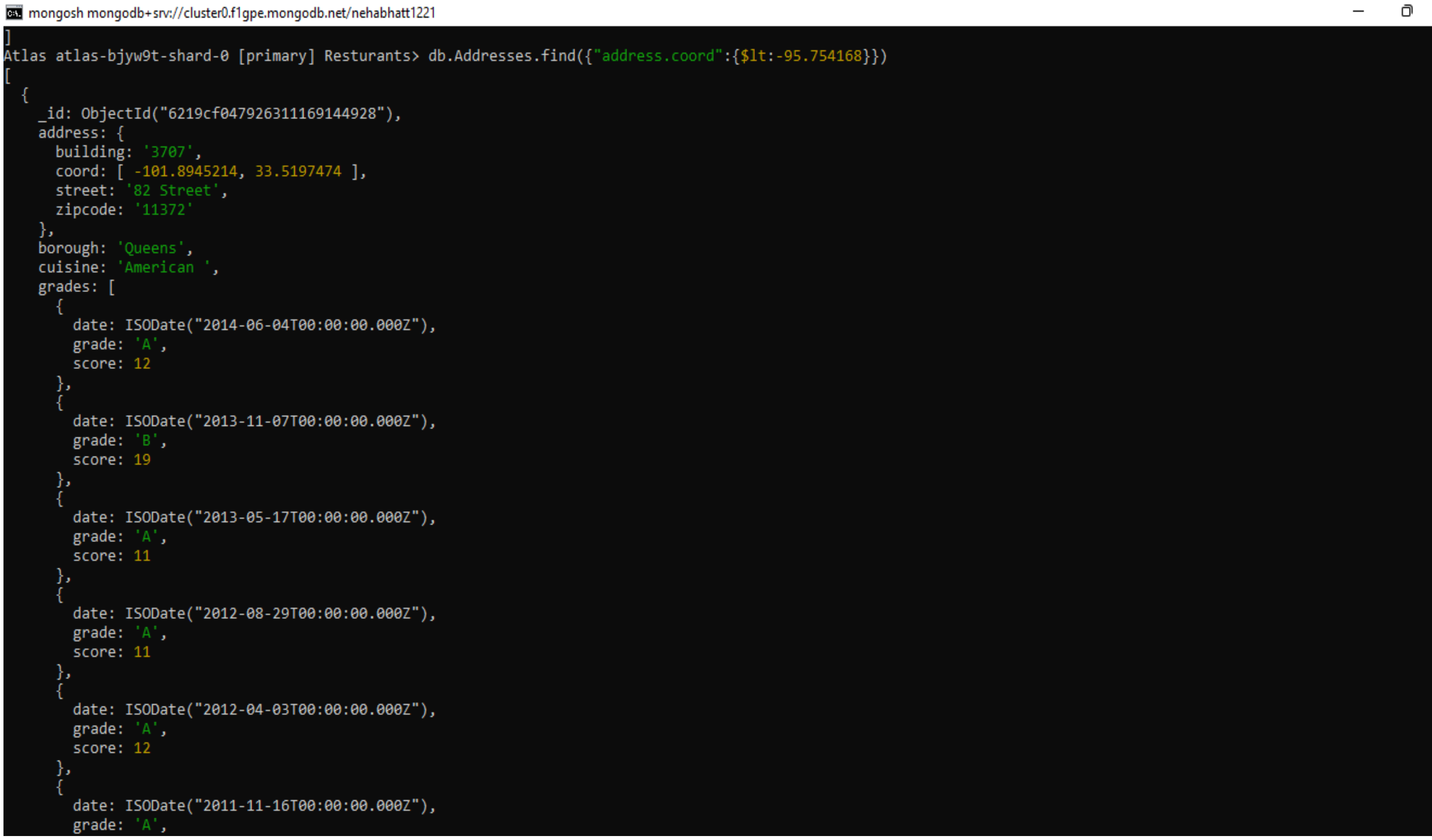
9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
    restaurant_id: '40393488'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.score":{$gt:90, $lt:100}})
[
  {
    _id: ObjectId("6219cf03792631116914443e"),
    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
      },
      {
```

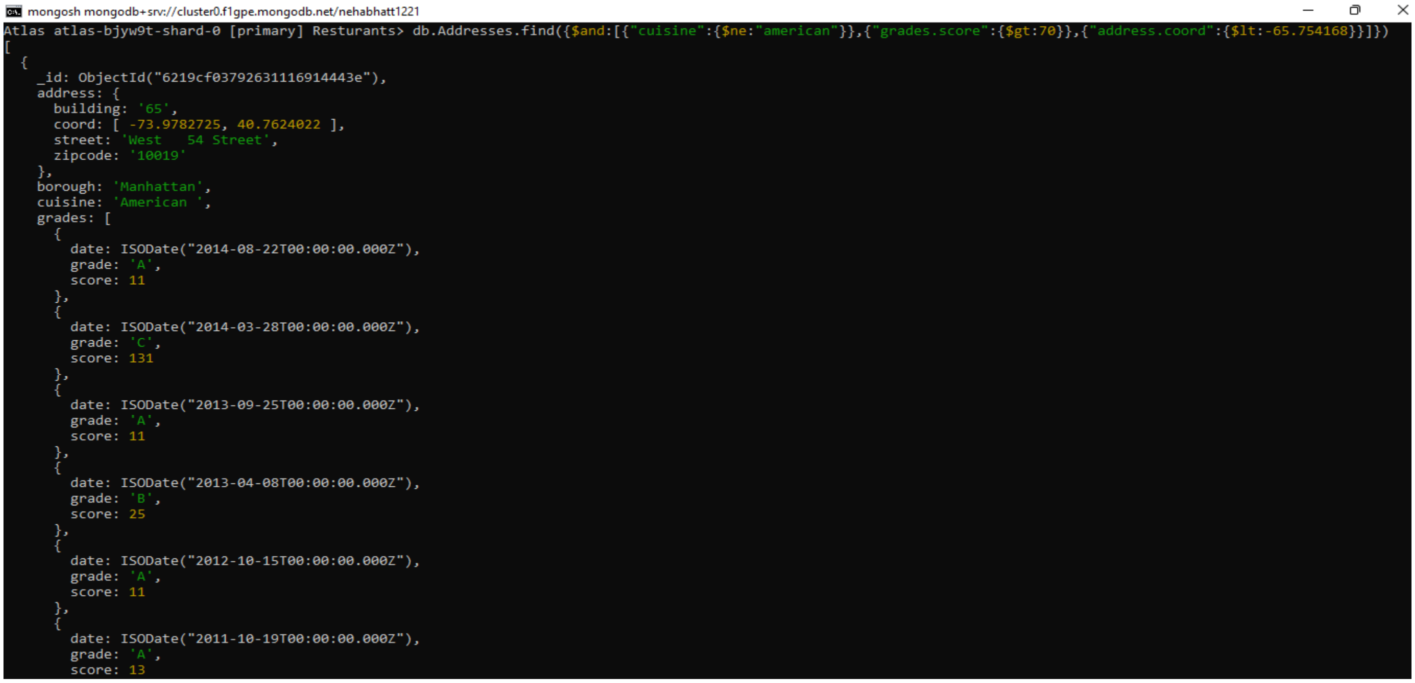
10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"address.coord":{$lt:-95.754168}})
[
  {
    _id: ObjectId("6219cf047926311169144928"),
    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',
```

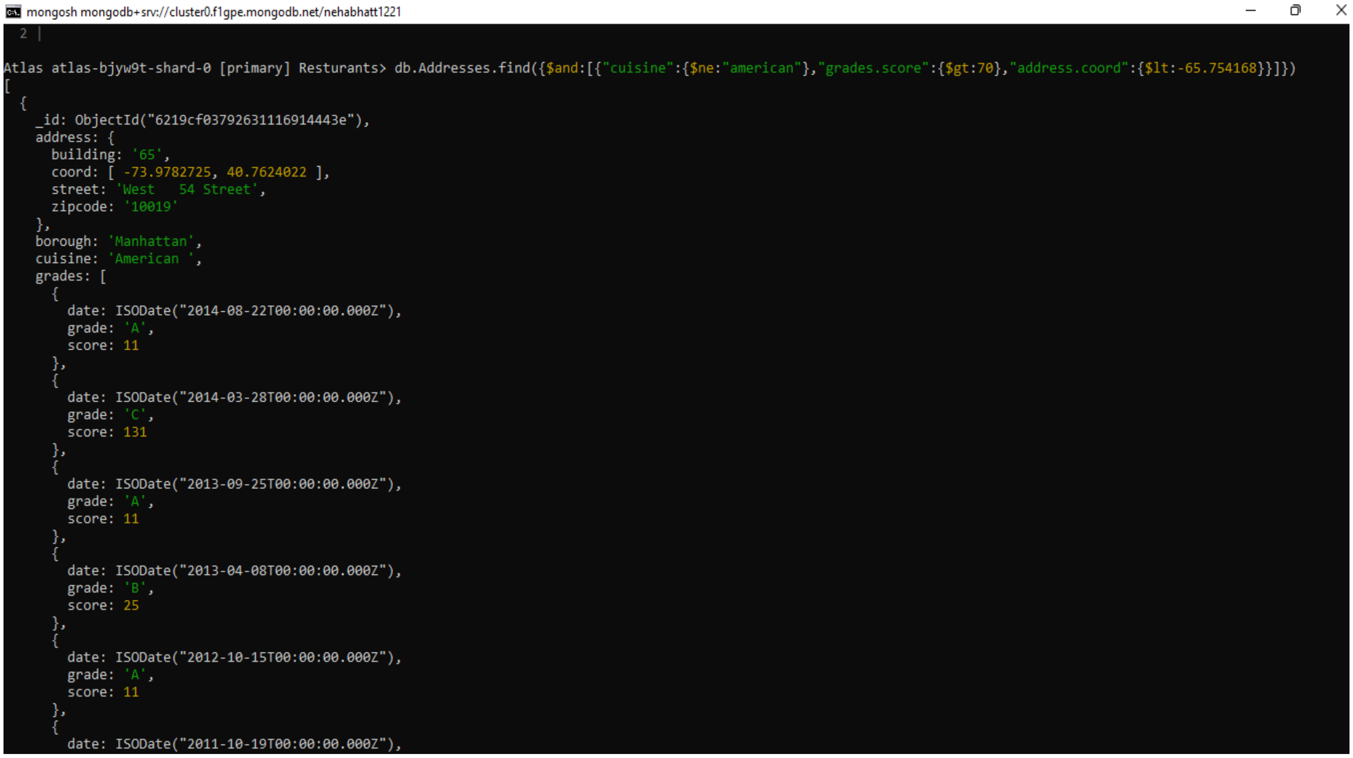
11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({$and:[{"cuisine":{$ne:"american"}},{"grades.score":{$gt:70}},{"address.coord":{$lt:-65.754168}}]})
[
  {
    _id: ObjectId("6219cf03792631116914443e"),
    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
```

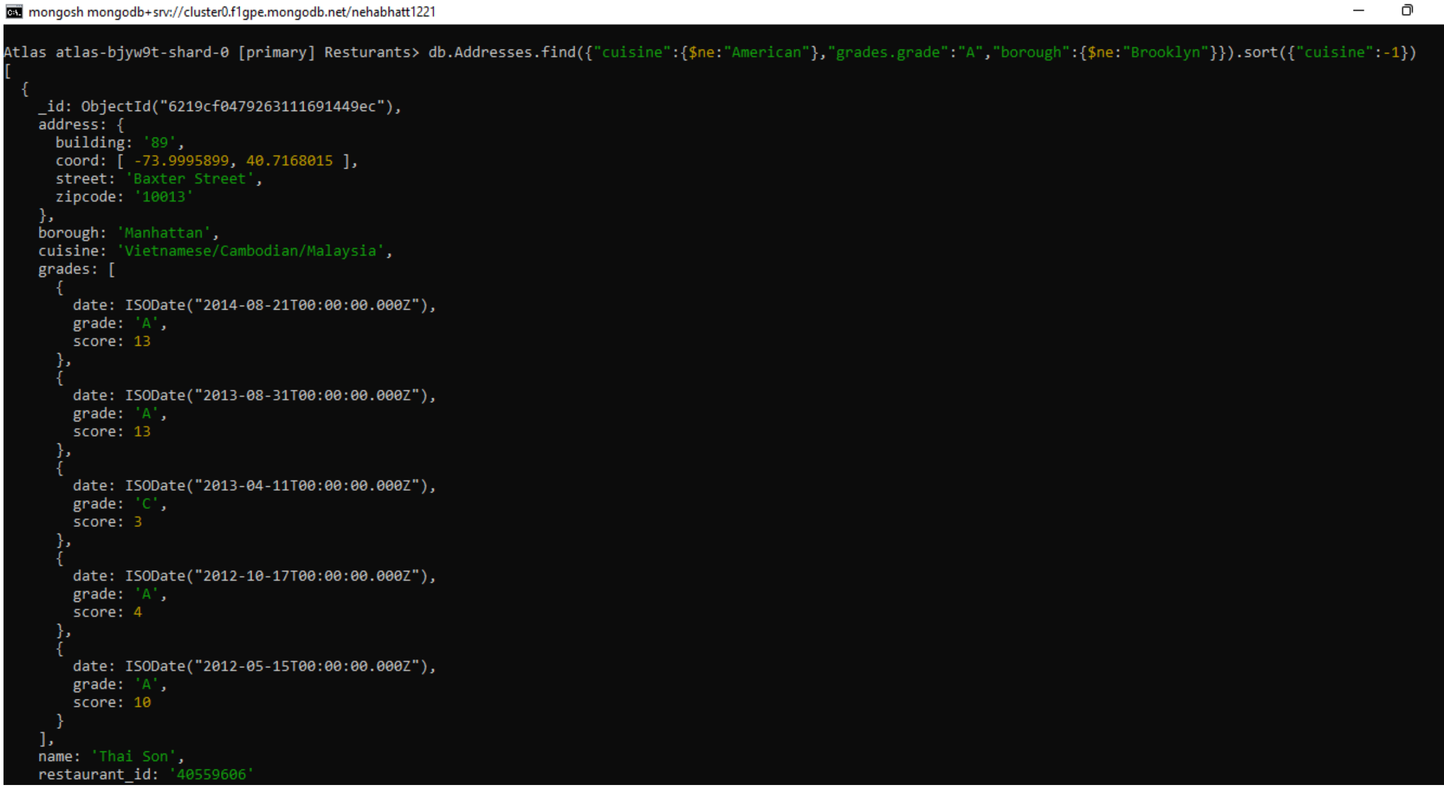
12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({$and:[{"cuisine":{$ne:"american"},"grades.score":{$gt:70},"address.coord":{$lt:-65.754168}}]})
[
  {
    _id: ObjectId("6219cf03792631116914443e"),
    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"),
```

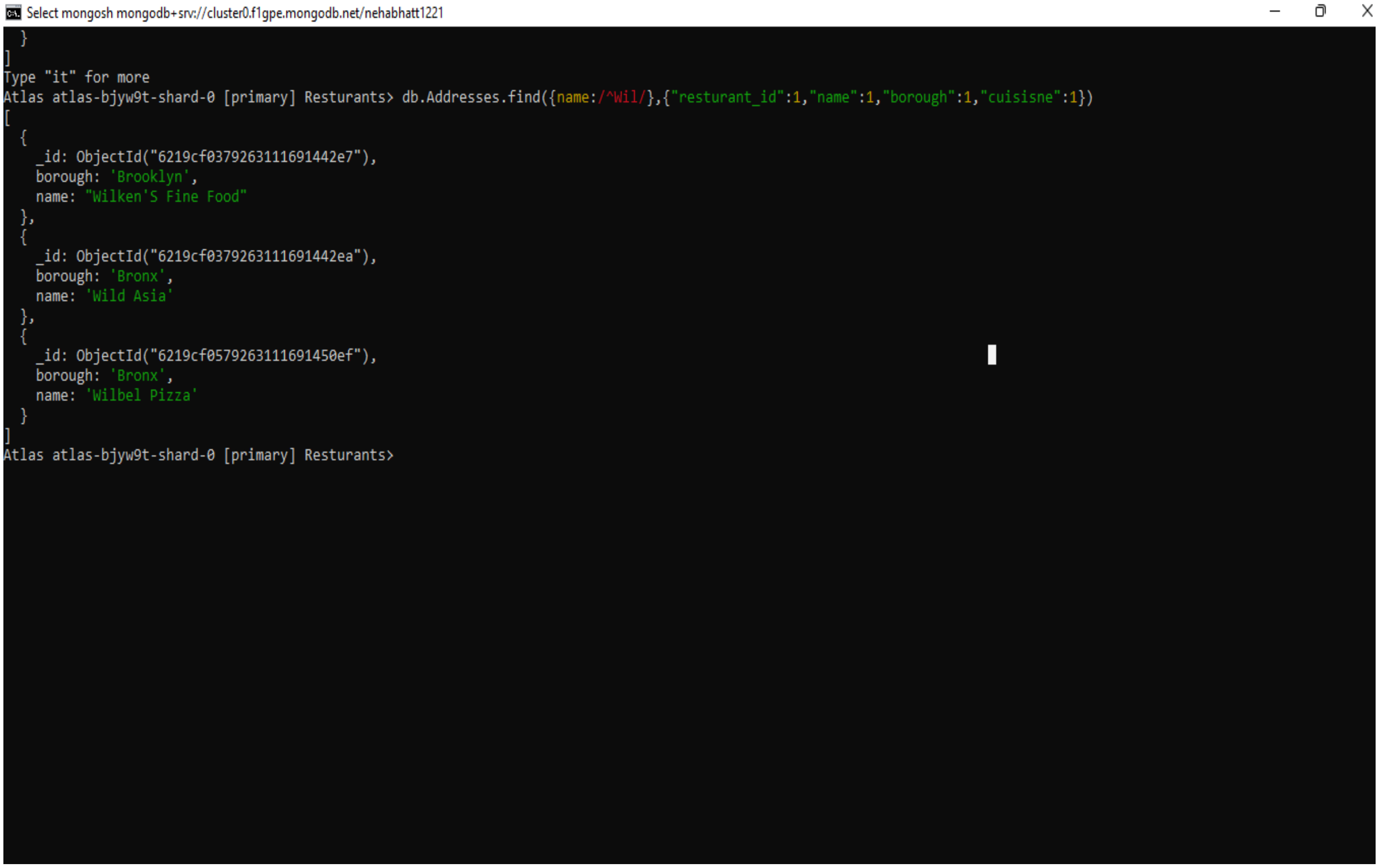
13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

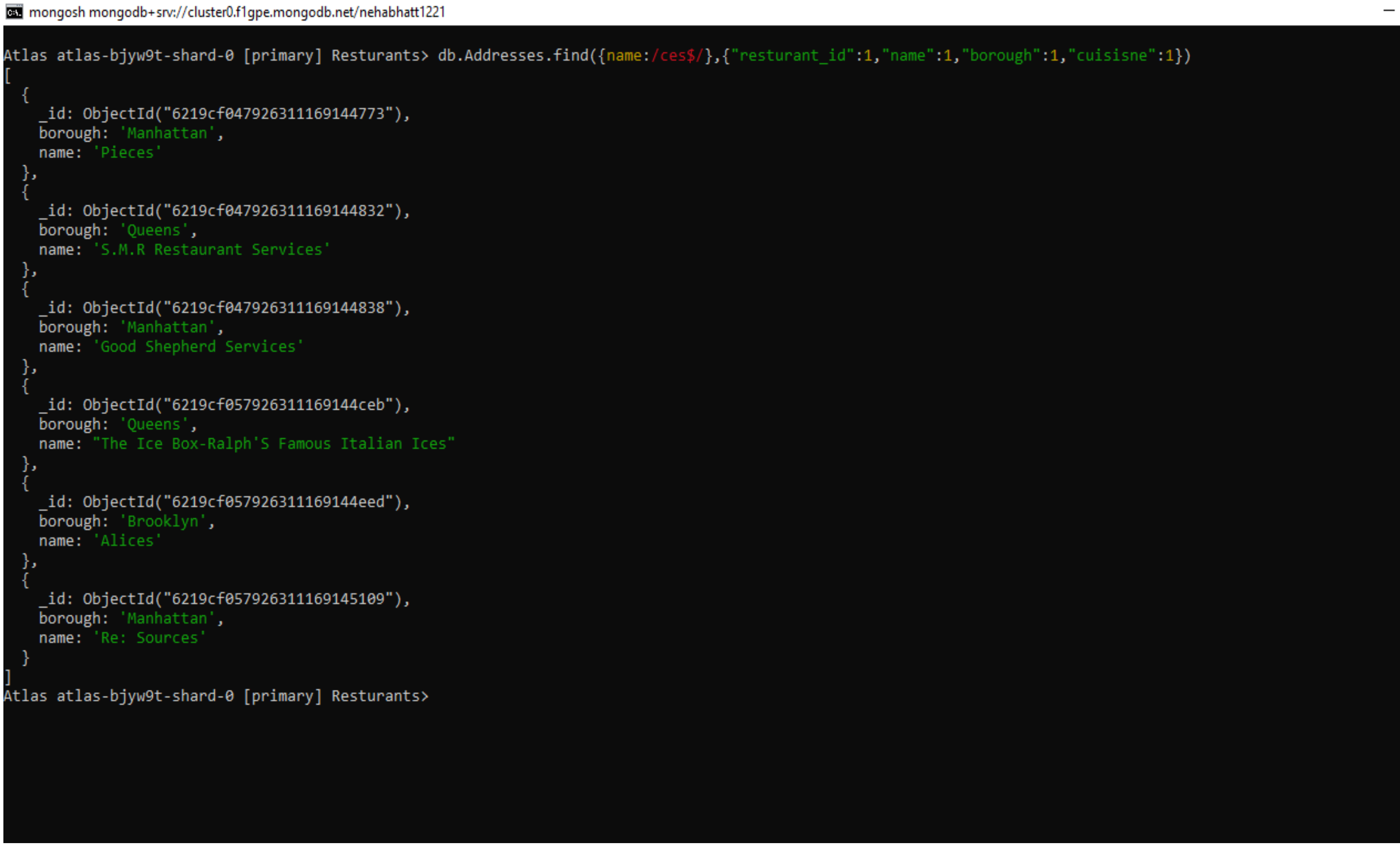
```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"cuisine":{$ne:"American"},"grades.grade":"A","borough":{$ne:"Brooklyn"}}).sort({"cuisine":-1})
[
  {
    _id: ObjectId("6219cf0479263111691449ec"),
    address: {
      building: '89',
      coord: [ -73.9995899, 40.7168015 ],
      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.


Select mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  }
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({name:/^Wil/},{"resturant_id":1,"name":1,"borough":1,"cuisisne":1})
[
  {
    _id: ObjectId("6219cf0379263111691442e7"),
    borough: 'Brooklyn',
    name: "Wilken'S Fine Food"
  },
  {
    _id: ObjectId("6219cf0379263111691442ea"),
    borough: 'Bronx',
    name: 'Wild Asia'
  },
  {
    _id: ObjectId("6219cf0579263111691450ef"),
    borough: 'Bronx',
    name: 'Wilbel Pizza'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants>
```

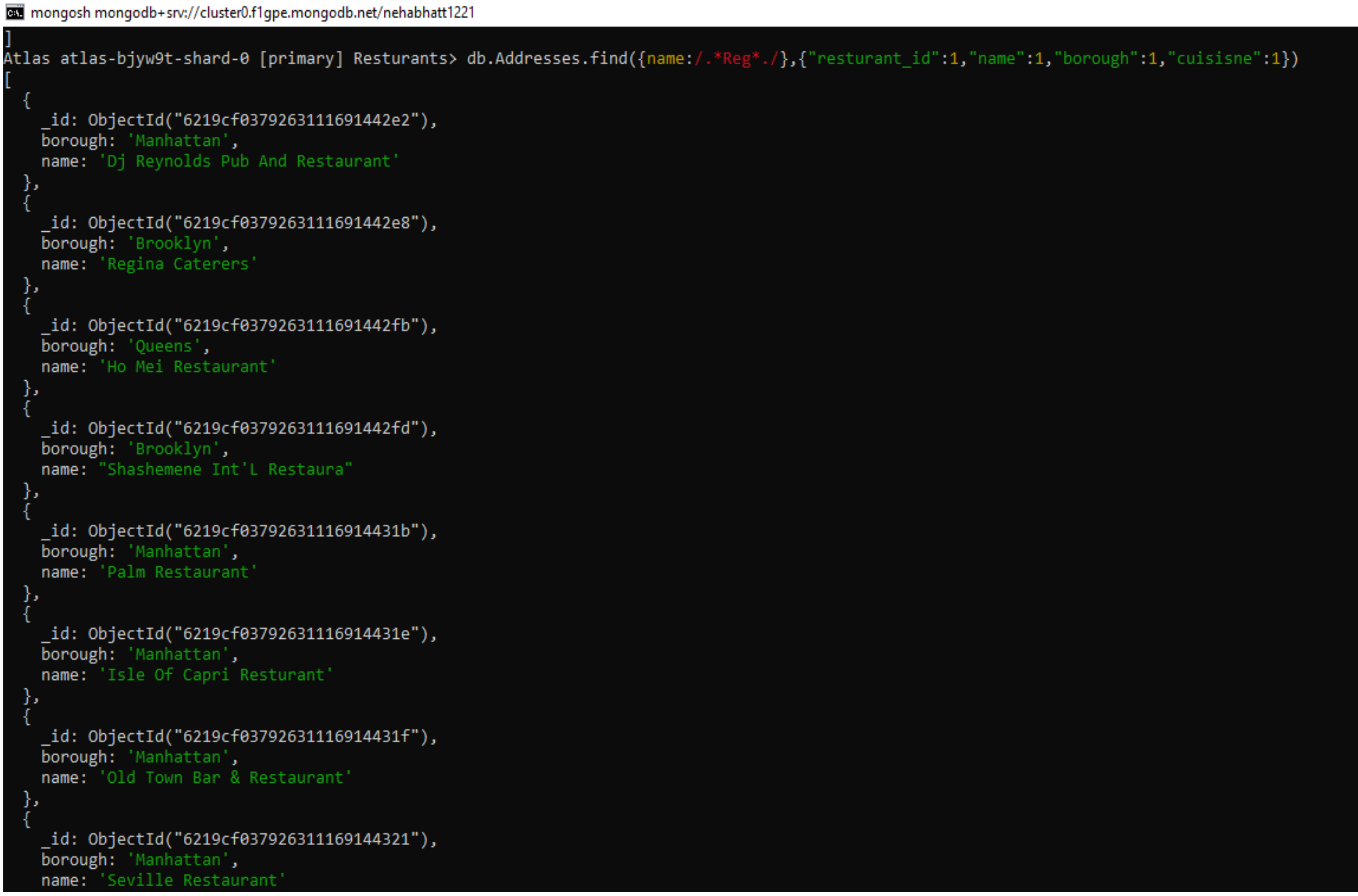
15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({name:/ces$/},{"resturant_id":1,"name":1,"borough":1,"cuisisne":1})
[
  {
    _id: ObjectId("6219cf047926311169144773"),
    borough: 'Manhattan',
    name: 'Pieces'
  },
  {
    _id: ObjectId("6219cf047926311169144832"),
    borough: 'Queens',
    name: 'S.M.R Restaurant Services'
  },
  {
    _id: ObjectId("6219cf047926311169144838"),
    borough: 'Manhattan',
    name: 'Good Shepherd Services'
  },
  {
    _id: ObjectId("6219cf057926311169144ceb"),
    borough: 'Queens',
    name: "The Ice Box-Ralph'S Famous Italian Ices"
  },
  {
    _id: ObjectId("6219cf057926311169144eed"),
    borough: 'Brooklyn',
    name: 'Alices'
  },
  {
    _id: ObjectId("6219cf057926311169145109"),
    borough: 'Manhattan',
    name: 'Re: Sources'
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants>
```

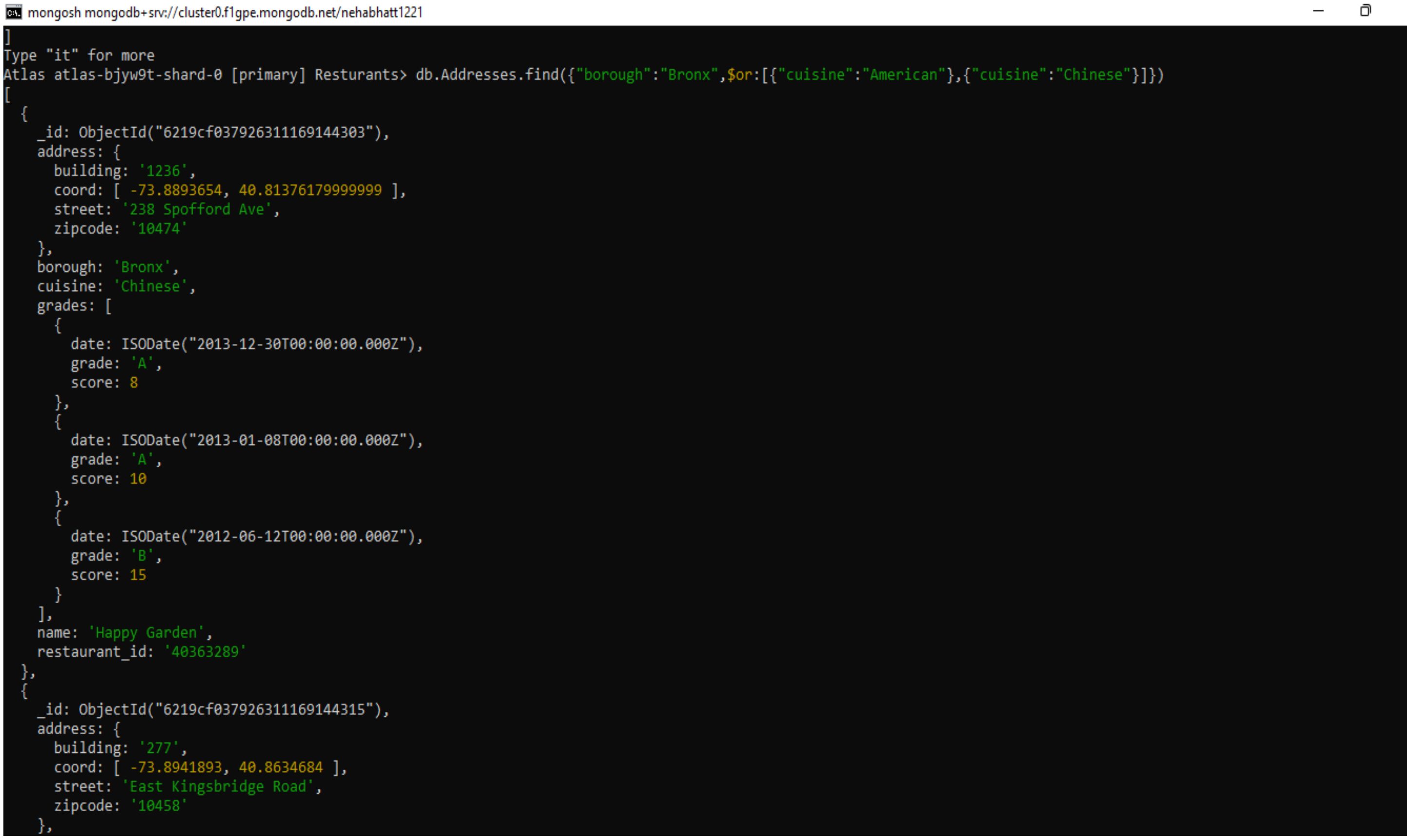
16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({name:/.*Reg*./},{"resturant_id":1,"name":1,"borough":1,"cuisisne":1})
[
  {
    _id: ObjectId("6219cf0379263111691442e2"),
    borough: 'Manhattan',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    _id: ObjectId("6219cf0379263111691442e8"),
    borough: 'Brooklyn',
    name: 'Regina Caterers'
  },
  {
    _id: ObjectId("6219cf0379263111691442fb"),
    borough: 'Queens',
    name: 'Ho Mei Restaurant'
  },
  {
    _id: ObjectId("6219cf0379263111691442fd"),
    borough: 'Brooklyn',
    name: "Shashemene Int'L Restaura"
  },
  {
    _id: ObjectId("6219cf03792631116914431b"),
    borough: 'Manhattan',
    name: 'Palm Restaurant'
  },
  {
    _id: ObjectId("6219cf03792631116914431e"),
    borough: 'Manhattan',
    name: 'Isle Of Capri Resturant'
  },
  {
    _id: ObjectId("6219cf03792631116914431f"),
    borough: 'Manhattan',
    name: 'Old Town Bar & Restaurant'
  },
  {
    _id: ObjectId("6219cf0379263111169144321"),
    borough: 'Manhattan',
    name: 'Seville Restaurant'
```

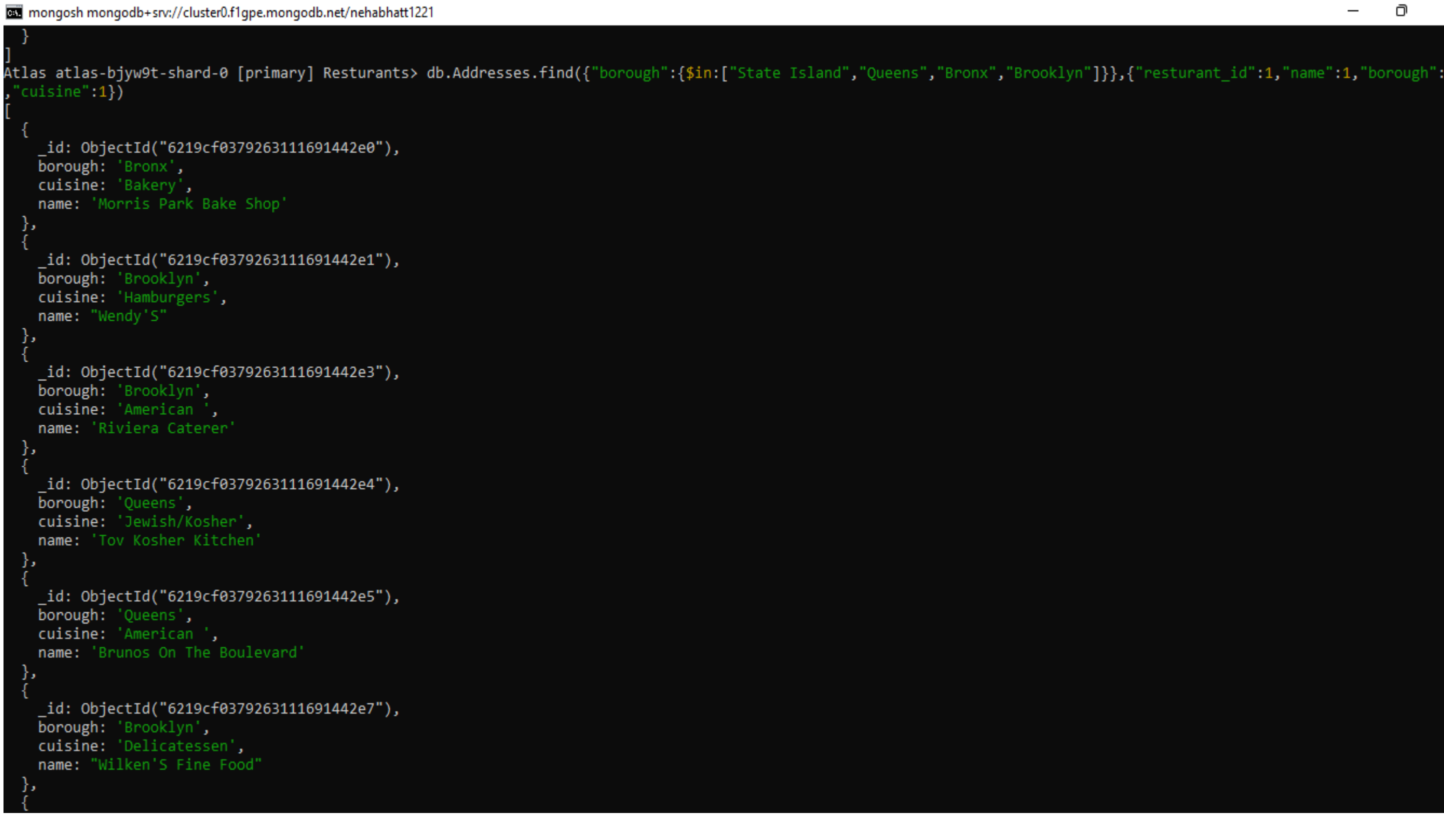
17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":"Bronx",$or:[{"cuisine":"American"},{"cuisine":"Chinese"}]})
[
  {
    _id: ObjectId("6219cf037926311169144303"),
    address: {
      building: '1236',
      coord: [ -73.8893654, 40.81376179999999 ],
      street: '238 Spofford Ave',
      zipcode: '10474'
    },
    borough: 'Bronx',
    cuisine: 'Chinese',
    grades: [
      {
        date: ISODate("2013-12-30T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-01-08T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-06-12T00:00:00.000Z"),
        grade: 'B',
        score: 15
      }
    ],
    name: 'Happy Garden',
    restaurant_id: '40363289'
  },
  {
    _id: ObjectId("6219cf037926311169144315"),
    address: {
      building: '277',
      coord: [ -73.8941893, 40.8634684 ],
      street: 'East Kingsbridge Road',
      zipcode: '10458'
    },
```

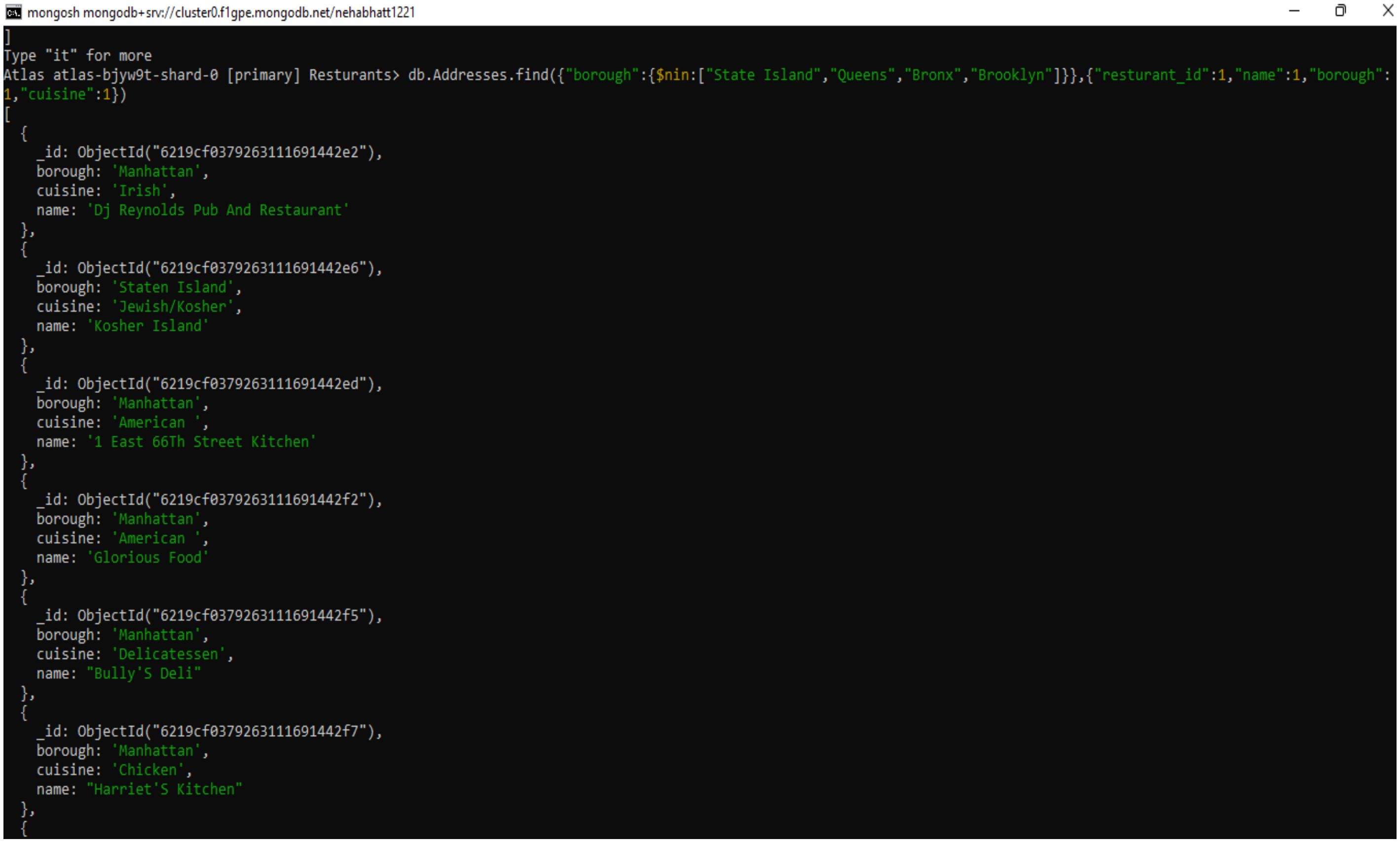
18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":{$in:["State Island","Queens","Bronx","Brooklyn"]}},{"resturant_id":1,"name":1,"borough":
,"cuisine":1})
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },
  {
    _id: ObjectId("6219cf0379263111691442e1"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S"
  },
  {
    _id: ObjectId("6219cf0379263111691442e3"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer'
  },
  {
    _id: ObjectId("6219cf0379263111691442e4"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    _id: ObjectId("6219cf0379263111691442e5"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard'
  },
  {
    _id: ObjectId("6219cf0379263111691442e7"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food"
  },
  {
```

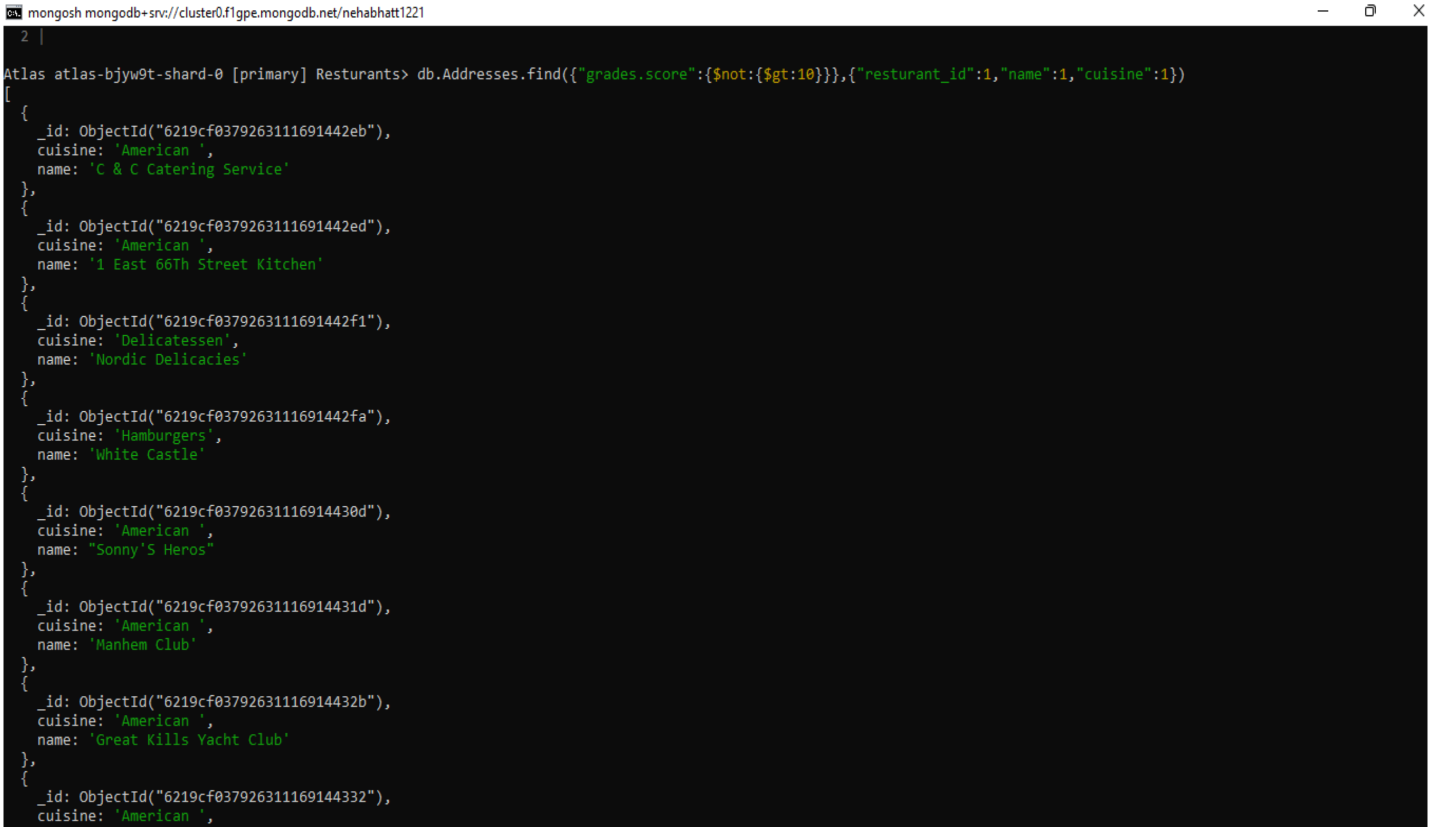
19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"borough":{$nin:["State Island","Queens","Bronx","Brooklyn"]}},{"resturant_id":1,"name":1,"borough":
1,"cuisine":1})
[
  {
    _id: ObjectId("6219cf0379263111691442e2"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    _id: ObjectId("6219cf0379263111691442e6"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island'
  },
  {
    _id: ObjectId("6219cf0379263111691442ed"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen'
  },
  {
    _id: ObjectId("6219cf0379263111691442f2"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Glorious Food'
  },
  {
    _id: ObjectId("6219cf0379263111691442f5"),
    borough: 'Manhattan',
    cuisine: 'Delicatessen',
    name: "Bully'S Deli"
  },
  {
    _id: ObjectId("6219cf0379263111691442f7"),
    borough: 'Manhattan',
    cuisine: 'Chicken',
    name: "Harriet'S Kitchen"
  },
  {
```

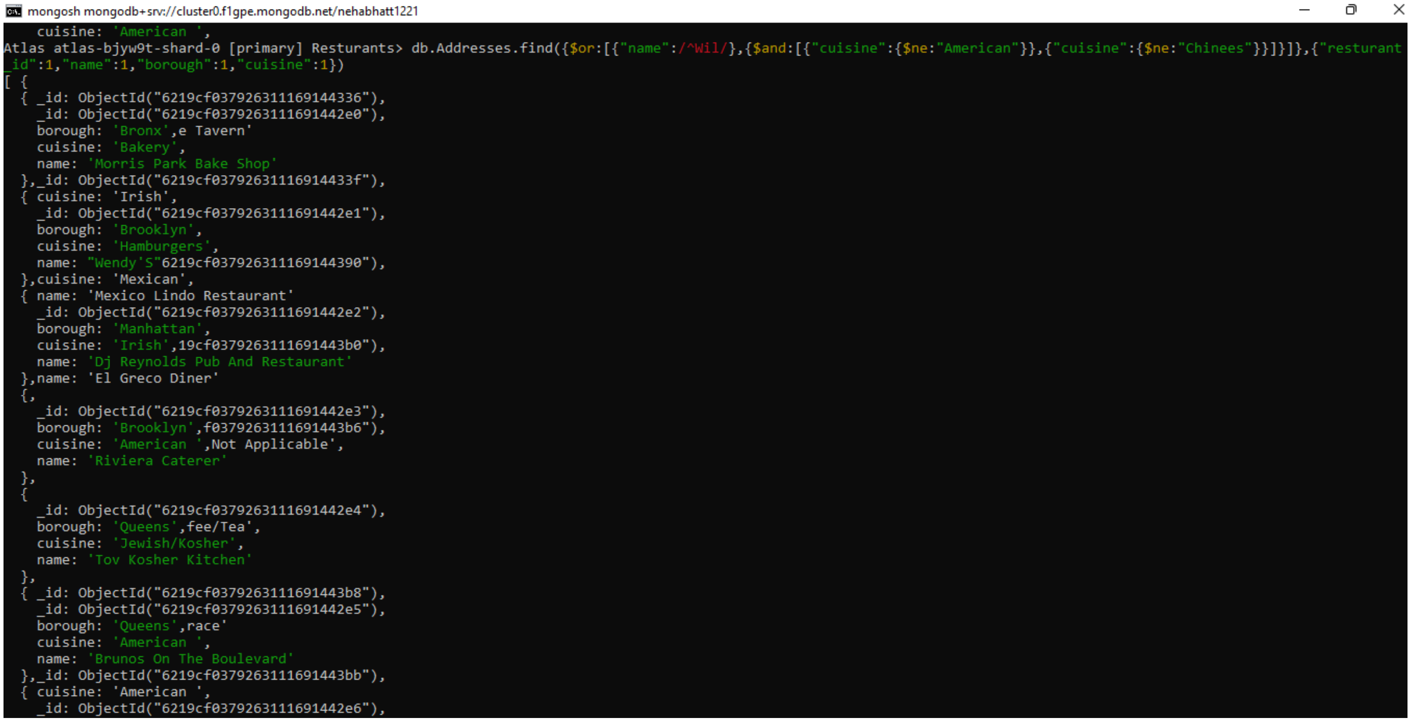
20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
  2 |

Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.score":{$not:{$gt:10}}},{"resturant_id":1,"name":1,"cuisine":1})
[
  {
    _id: ObjectId("6219cf0379263111691442eb"),
    cuisine: 'American ',
    name: 'C & C Catering Service'
  },
  {
    _id: ObjectId("6219cf0379263111691442ed"),
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen'
  },
  {
    _id: ObjectId("6219cf0379263111691442f1"),
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies'
  },
  {
    _id: ObjectId("6219cf0379263111691442fa"),
    cuisine: 'Hamburgers',
    name: 'White Castle'
  },
  {
    _id: ObjectId("6219cf037926311169144430d"),
    cuisine: 'American ',
    name: "Sonny'S Heros"
  },
  {
    _id: ObjectId("6219cf037926311169144431d"),
    cuisine: 'American ',
    name: 'Manhem Club'
  },
  {
    _id: ObjectId("6219cf037926311169144432b"),
    cuisine: 'American ',
    name: 'Great Kills Yacht Club'
  },
  {
    _id: ObjectId("6219cf0379263111691444332"),
    cuisine: 'American ',
```

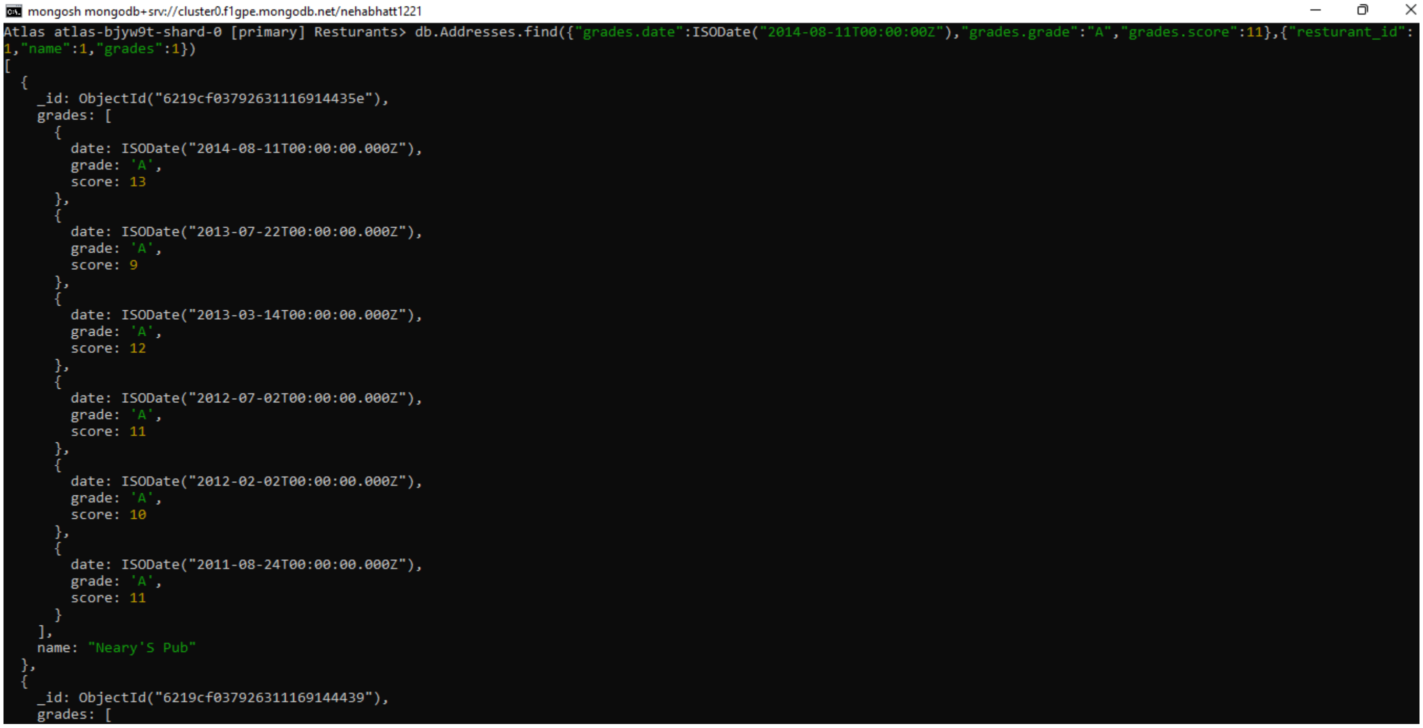
21. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

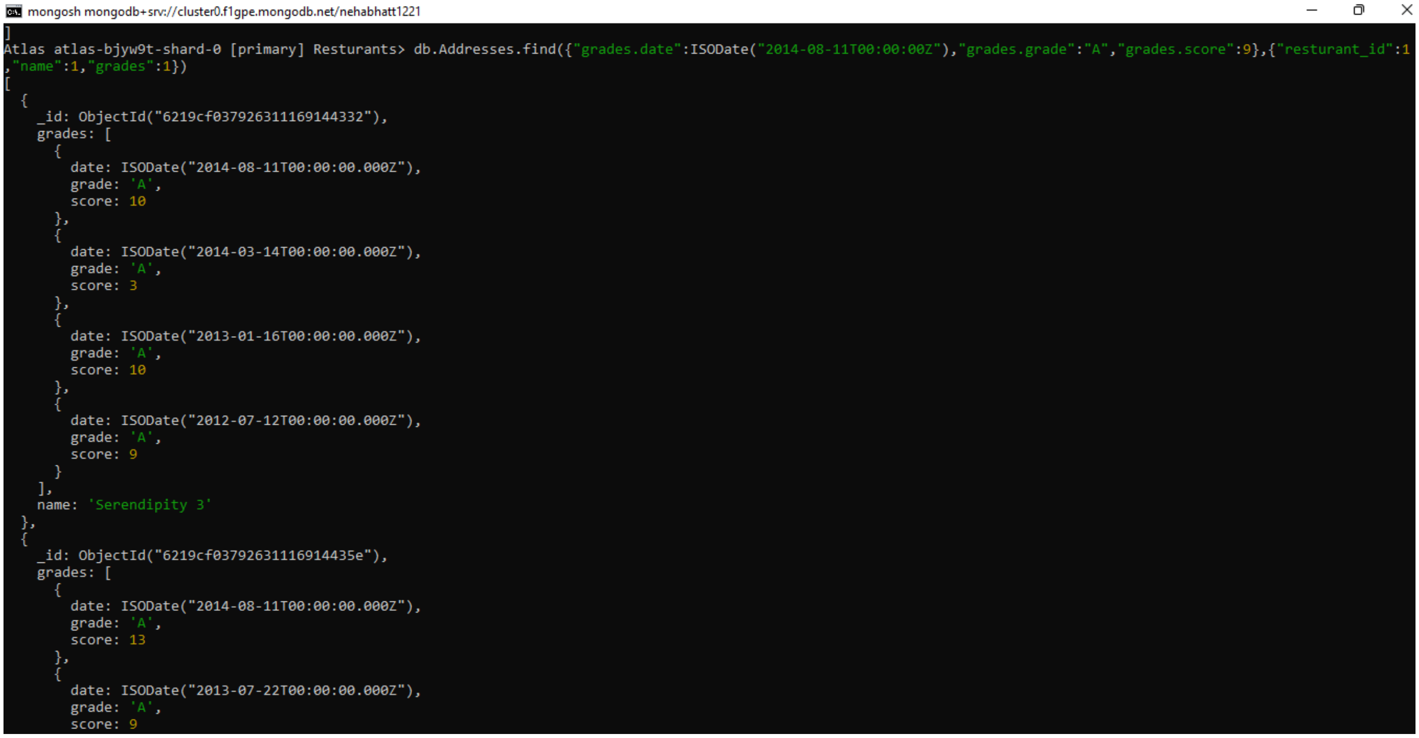
```
    cuisine: 'American ',
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({$or:[{"name":/^Wil/},{$and:[{"cuisine":{$ne:"American"}},{"cuisine":{$ne:"Chinees"}}]}]},{"resturant
_id":1,"name":1,"borough":1,"cuisine":1})
[ {
  { _id: ObjectId("6219cf037926311169144336"),
    _id: ObjectId("6219cf0379263111691442e0"),
    borough: 'Bronx',e Tavern'
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },_id: ObjectId("6219cf037926311169144433f"),
  { cuisine: 'Irish',
    _id: ObjectId("6219cf0379263111691442e1"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S"6219cf037926311169144390"),
  },cuisine: 'Mexican',
  { name: 'Mexico Lindo Restaurant'
    _id: ObjectId("6219cf0379263111691442e2"),
    borough: 'Manhattan',
    cuisine: 'Irish',19cf0379263111691443b0"),
    name: 'Dj Reynolds Pub And Restaurant'
  },name: 'El Greco Diner'
  {,
    _id: ObjectId("6219cf0379263111691442e3"),
    borough: 'Brooklyn',f0379263111691443b6"),
    cuisine: 'American ',Not Applicable',
    name: 'Riviera Caterer'
  },
  {
    _id: ObjectId("6219cf0379263111691442e4"),
    borough: 'Queens',fee/Tea',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  { _id: ObjectId("6219cf0379263111691443b8"),
    _id: ObjectId("6219cf0379263111691442e5"),
    borough: 'Queens',race'
    cuisine: 'American ',
    name: 'Brunos On The Boulevard'
  },_id: ObjectId("6219cf0379263111691443bb"),
  { cuisine: 'American ',
    _id: ObjectId("6219cf0379263111691442e6"),
```

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..


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

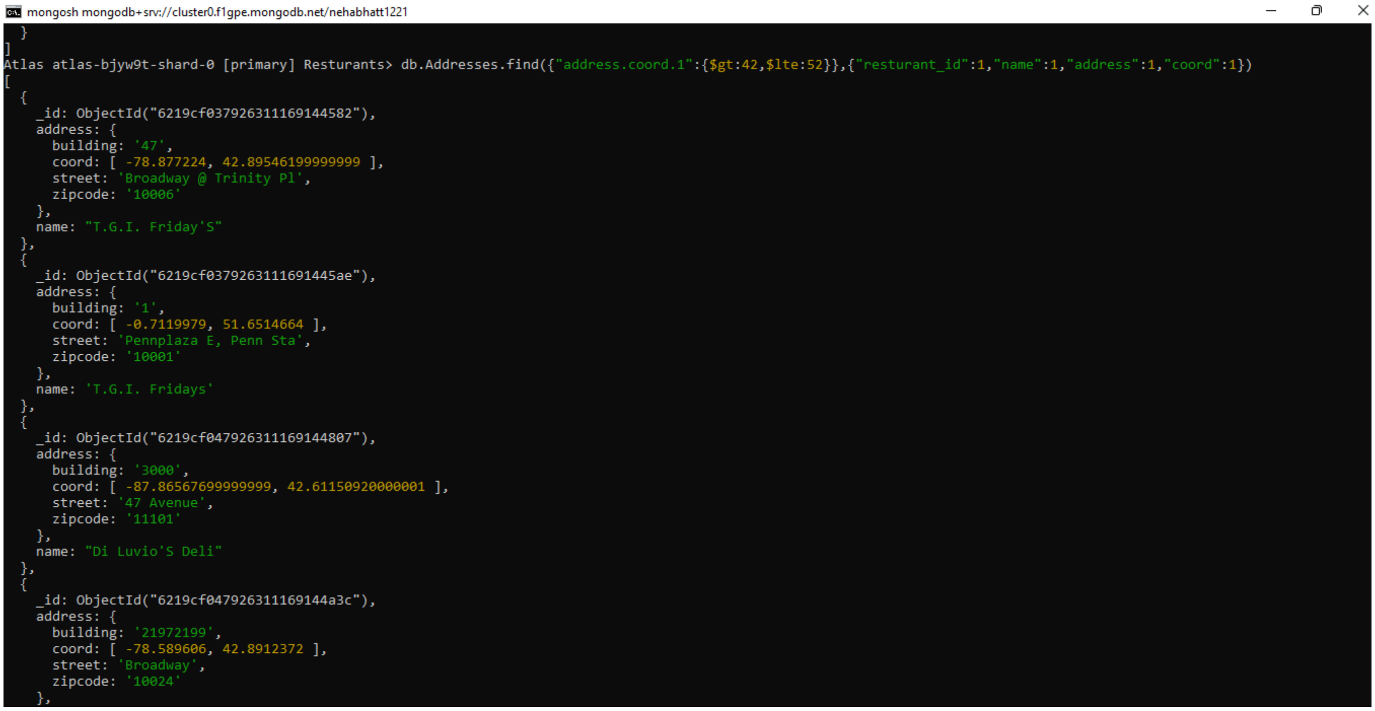
```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11},{"resturant_id":
1,"name":1,"grades":1})
[
  {
    _id: ObjectId("6219cf03792631116914435e"),
    grades: [
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        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"
  },
  {
    _id: ObjectId("6219cf037926311169144439"),
    grades: [
```

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"


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":9},{"resturant_id":1
,"name":1,"grades":1})
[
  {
    _id: ObjectId("6219cf037926311169144332"),
    grades: [
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2014-03-14T00:00:00.000Z"),
        grade: 'A',
        score: 3
      },
      {
        date: ISODate("2013-01-16T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-07-12T00:00:00.000Z"),
        grade: 'A',
        score: 9
      }
    ],
    name: 'Serendipity 3'
  },
  {
    _id: ObjectId("6219cf03792631116914435e"),
    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
```

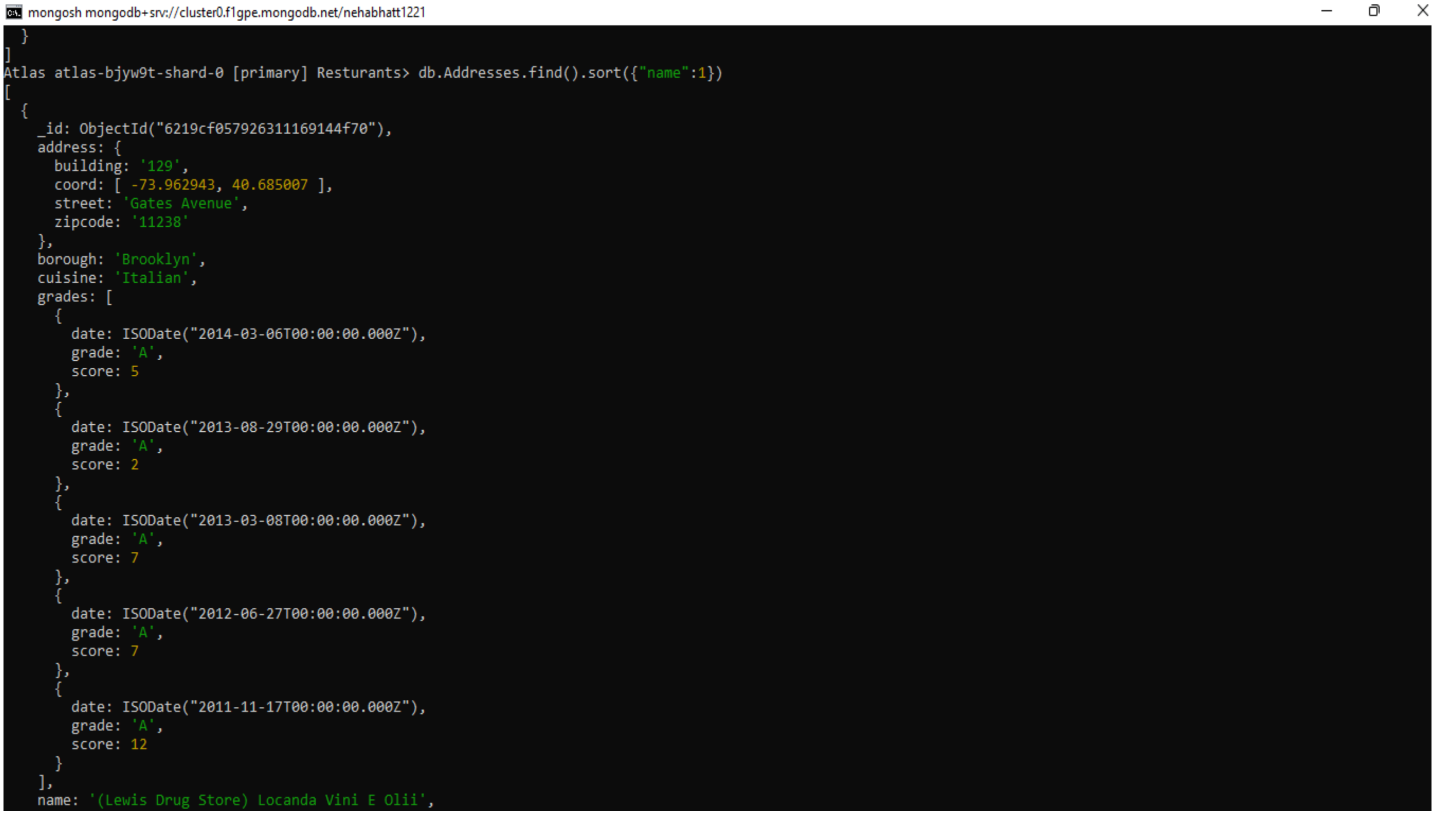
24. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52..



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

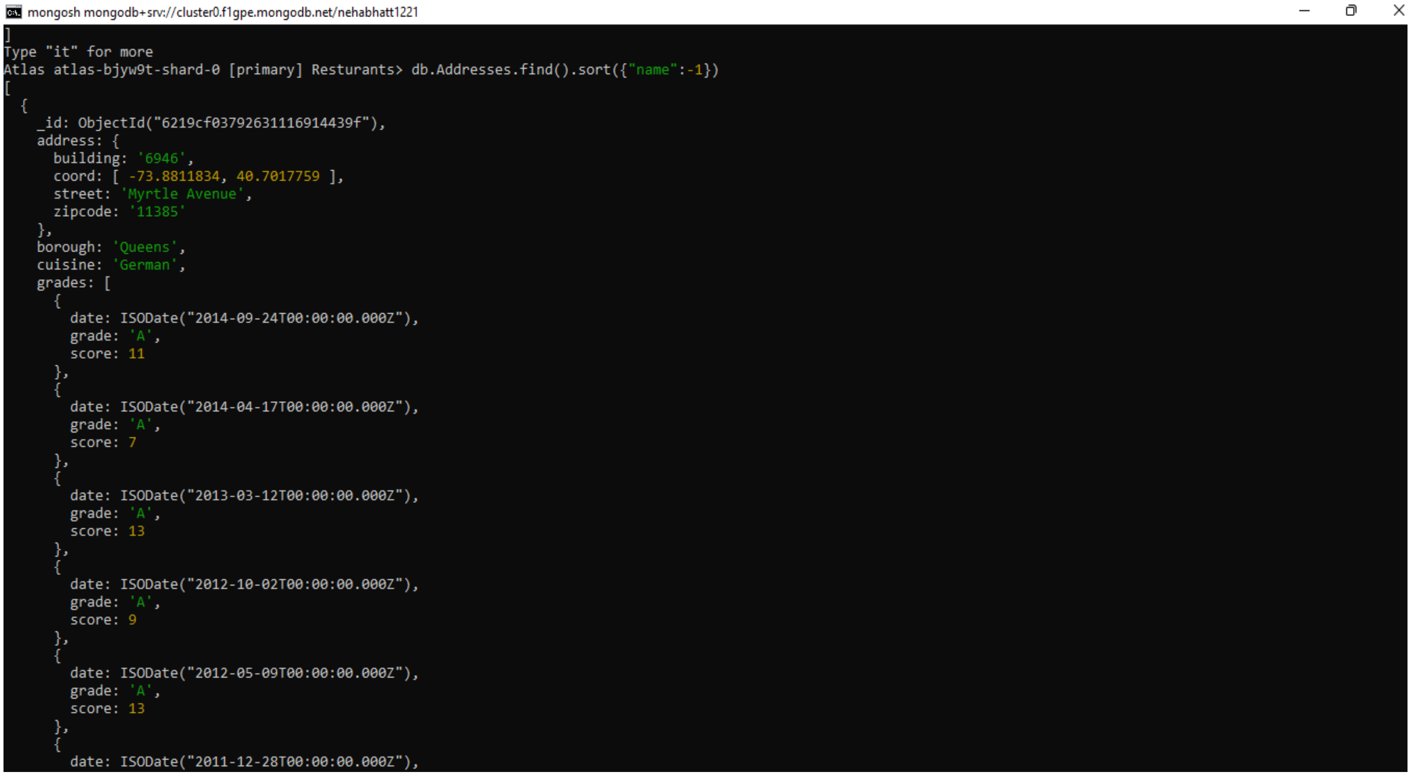
```
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"address.coord.1":{$gt:42,$lte:52}},{"resturant_id":1,"name":1,"address":1,"coord":1})
[
  {
    _id: ObjectId("6219cf037926311169144582"),
    address: {
      building: '47',
      coord: [ -78.877224, 42.89546199999999 ],
      street: 'Broadway @ Trinity Pl',
      zipcode: '10006'
    },
    name: "T.G.I. Friday'S"
  },
  {
    _id: ObjectId("6219cf0379263111691445ae"),
    address: {
      building: '1',
      coord: [ -0.7119979, 51.6514664 ],
      street: 'Pennplaza E, Penn Sta',
      zipcode: '10001'
    },
    name: 'T.G.I. Fridays'
  },
  {
    _id: ObjectId("6219cf047926311169144807"),
    address: {
      building: '3000',
      coord: [ -87.86567699999999, 42.61150920000001 ],
      street: '47 Avenue',
      zipcode: '11101'
    },
    name: "Di Luvio'S Deli"
  },
  {
    _id: ObjectId("6219cf047926311169144a3c"),
    address: {
      building: '21972199',
      coord: [ -78.589606, 42.8912372 ],
      street: 'Broadway',
      zipcode: '10024'
    },
```

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


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

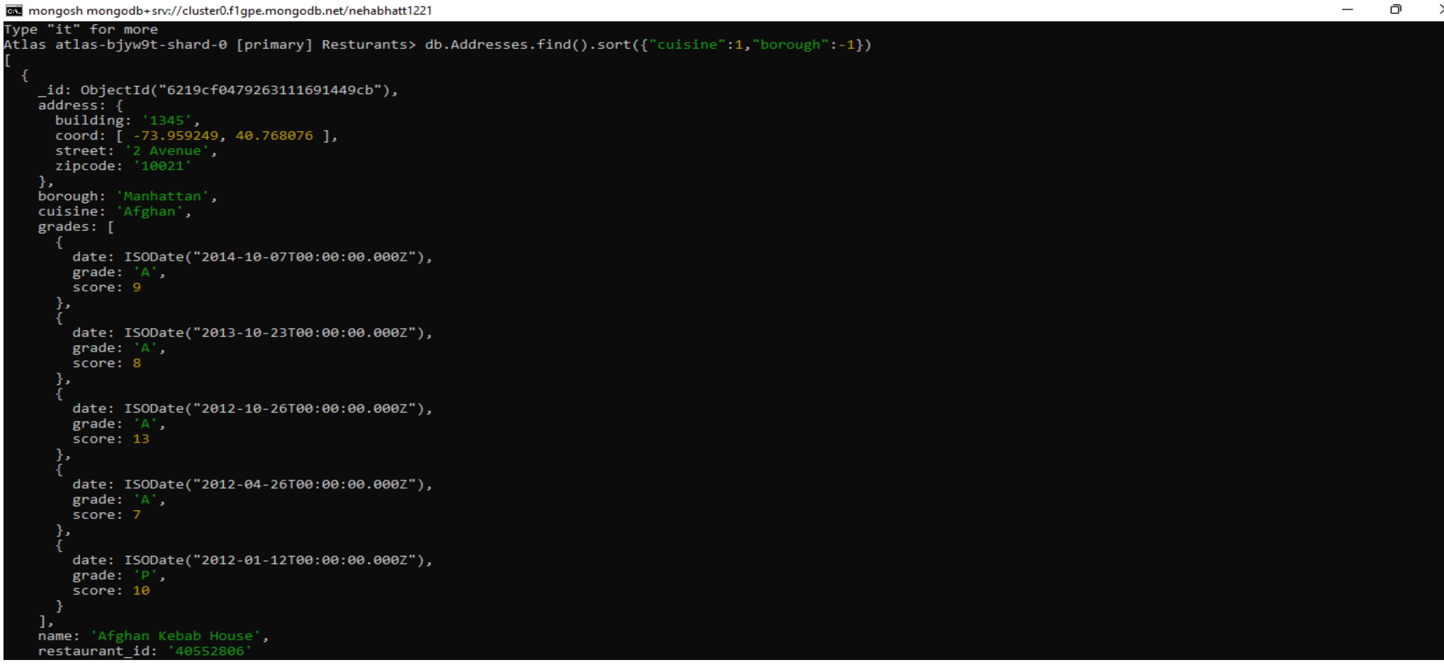
```
  }
]
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find().sort({"name":1})
[
  {
    _id: ObjectId("6219cf057926311169144f70"),
    address: {
      building: '129',
      coord: [ -73.962943, 40.685007 ],
      street: 'Gates Avenue',
      zipcode: '11238'
    },
    borough: 'Brooklyn',
    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',
```

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


mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find().sort({"name":-1})
[
  {
    _id: ObjectId("6219cf03792631116914439f"),
    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"),
```

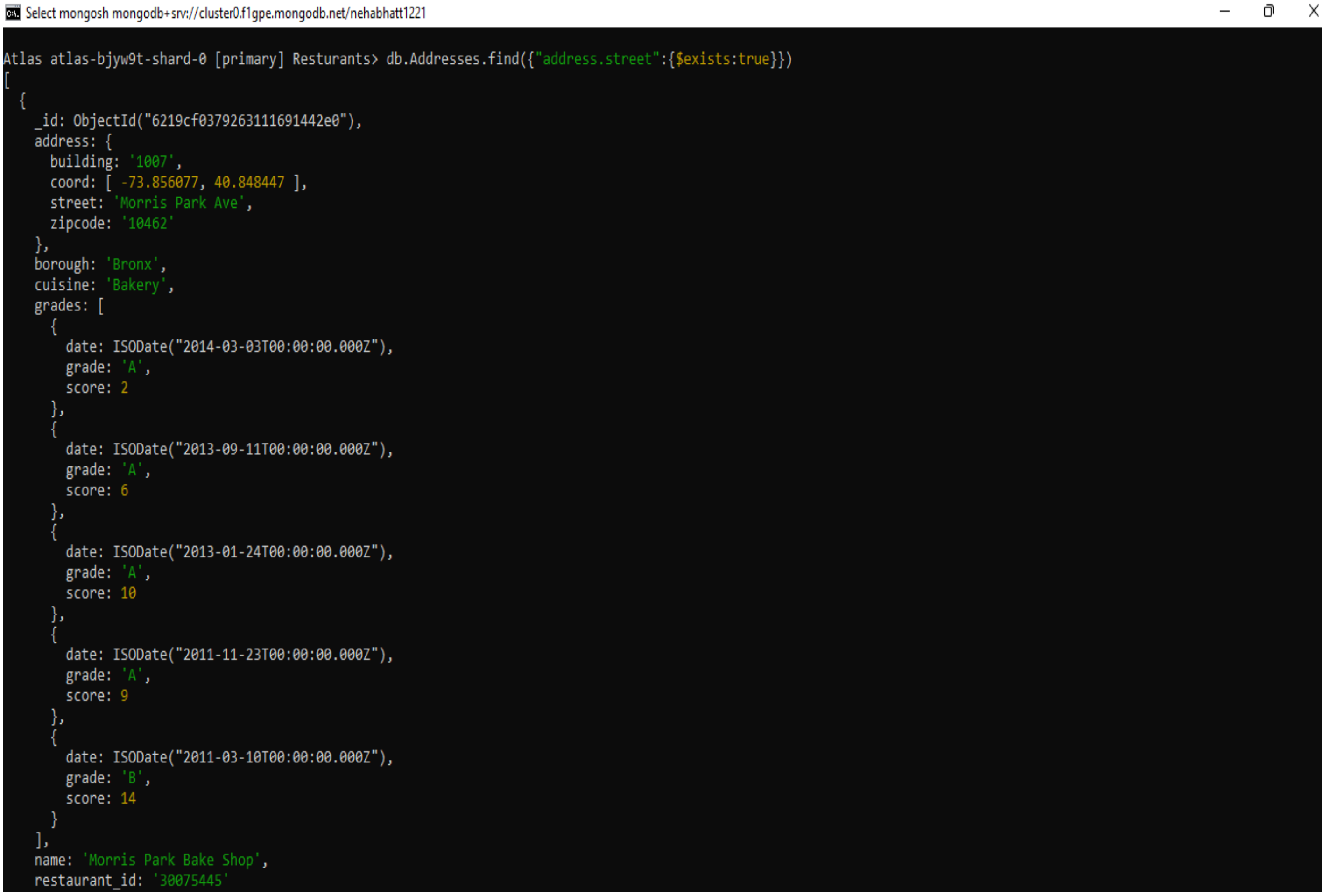
27. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

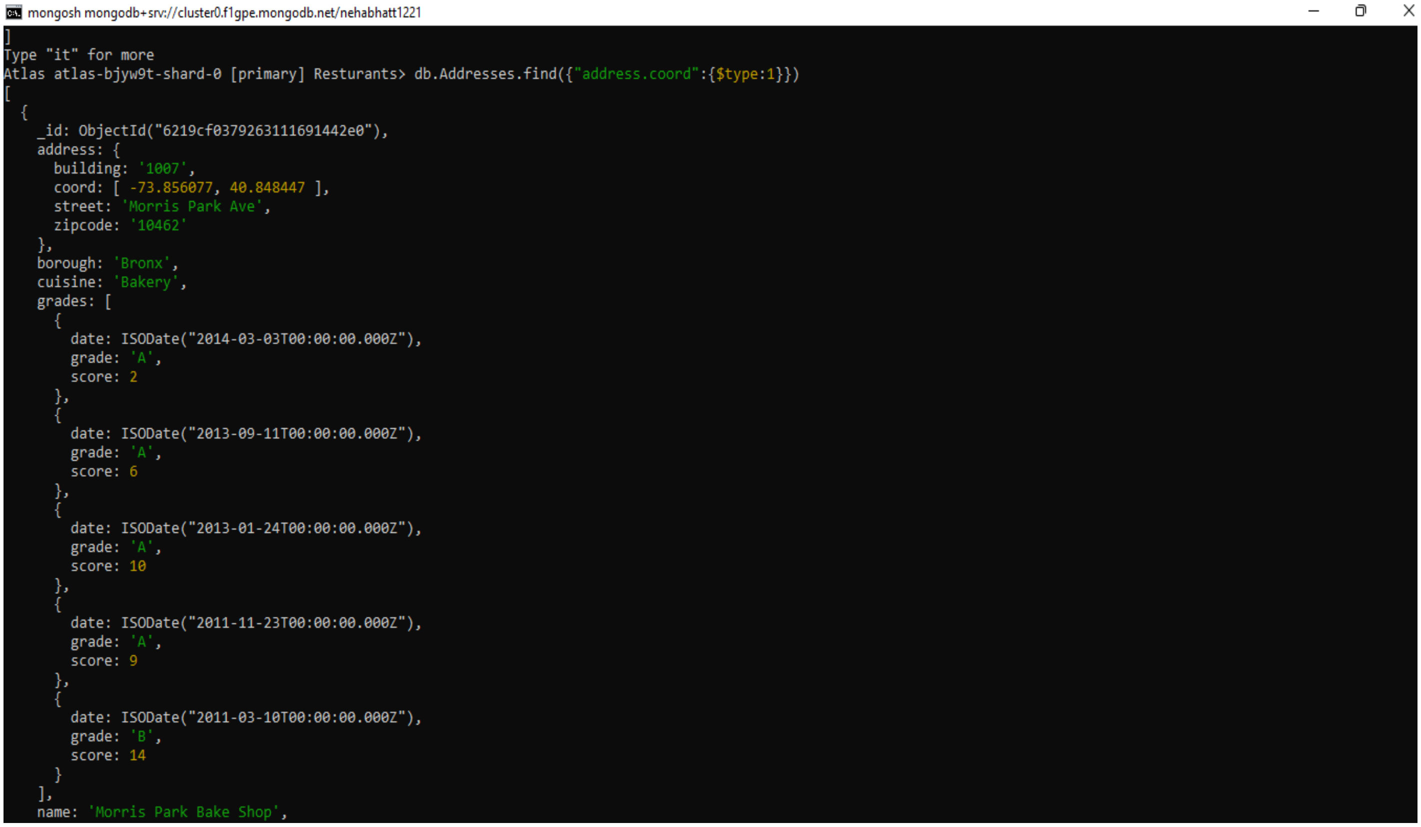
```
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find().sort({"cuisine":1,"borough":-1})
[
  {
    _id: ObjectId("6219cf0479263111691449cb"),
    address: {
      building: '1345',
      coord: [ -73.959249, 40.768076 ],
      street: '2 Avenue',
      zipcode: '10021'
    },
    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',
    restaurant_id: '40552806'
```

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


Select mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"address.street":{$exists:true}})
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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'
```

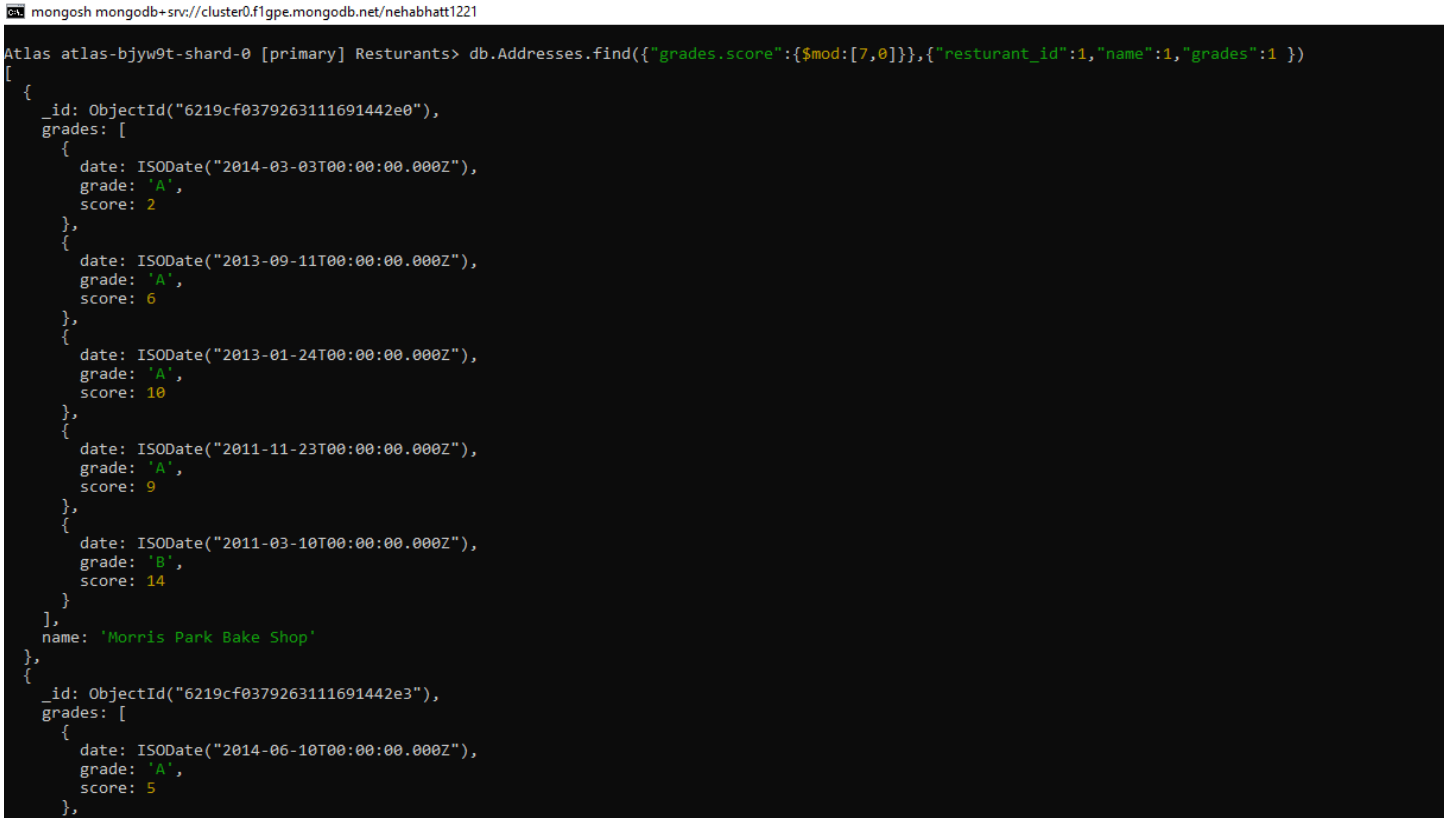
29. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"address.coord":{$type:1}})
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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',
```

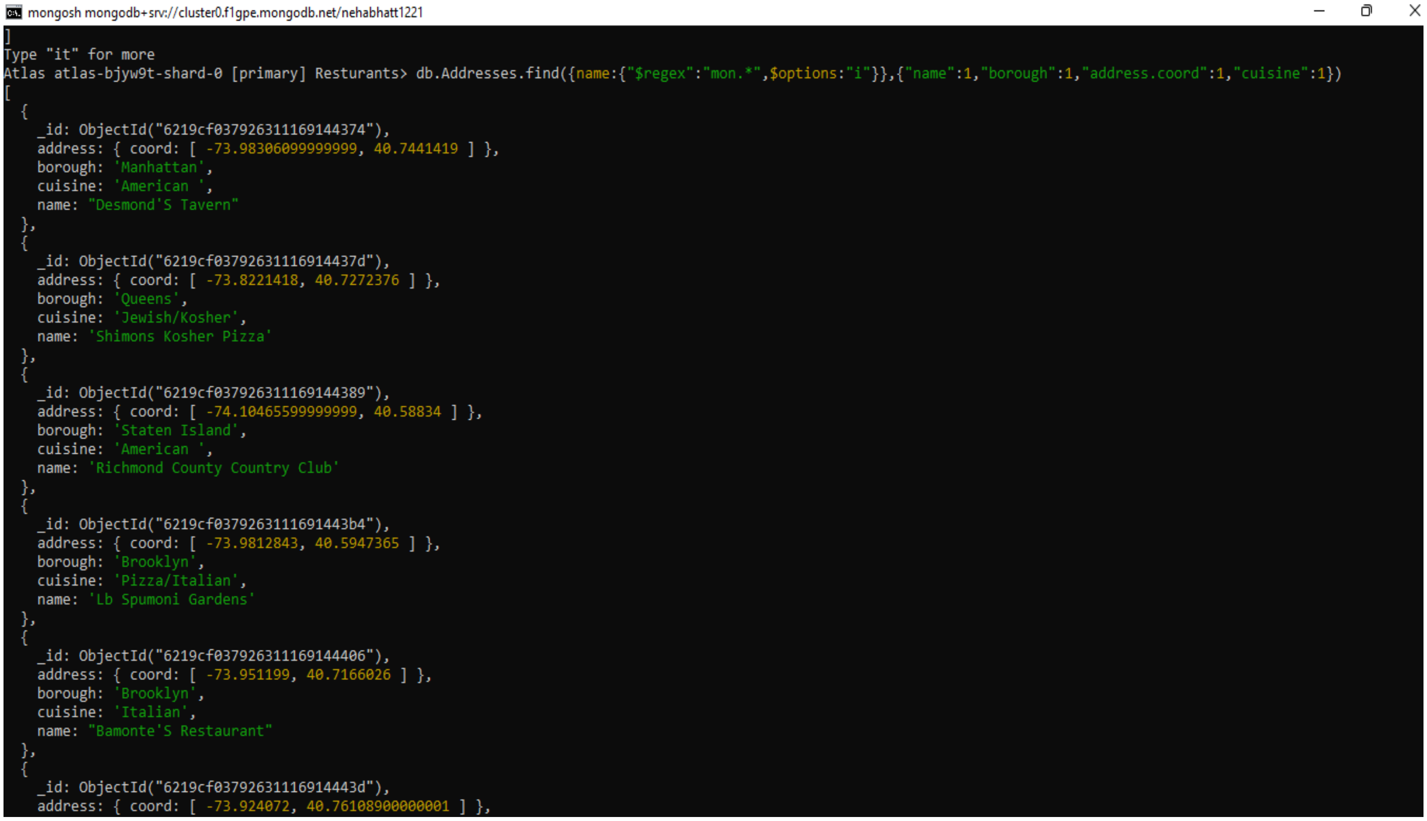
30. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({"grades.score":{$mod:[7,0]}},{"resturant_id":1,"name":1,"grades":1 })
[
  {
    _id: ObjectId("6219cf0379263111691442e0"),
    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'
  },
  {
    _id: ObjectId("6219cf0379263111691442e3"),
    grades: [
      {
        date: ISODate("2014-06-10T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
```

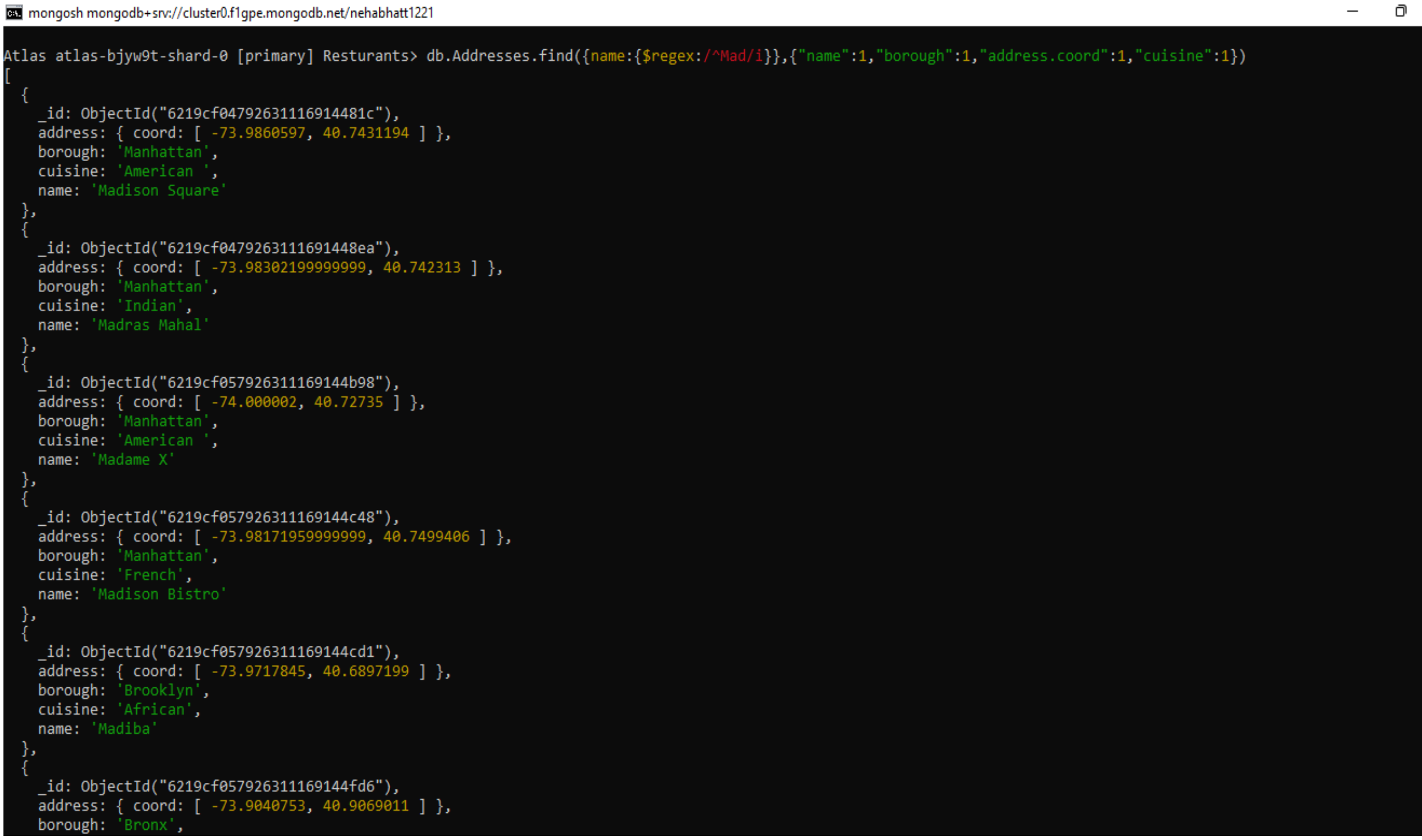
31. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
]
Type "it" for more
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({name:{"$regex":"mon.*",$options:"i"}},{"name":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("6219cf037926311169144374"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    _id: ObjectId("6219cf03792631116914437d"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("6219cf037926311169144389"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
  {
    _id: ObjectId("6219cf0379263111691443b4"),
    address: { coord: [ -73.9812843, 40.5947365 ] },
    borough: 'Brooklyn',
    cuisine: 'Pizza/Italian',
    name: 'Lb Spumoni Gardens'
  },
  {
    _id: ObjectId("6219cf037926311169144406"),
    address: { coord: [ -73.951199, 40.7166026 ] },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    name: "Bamonte'S Restaurant"
  },
  {
    _id: ObjectId("6219cf03792631116914443d"),
    address: { coord: [ -73.924072, 40.76108900000001 ] },
```

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



mongosh mongodb+srv://cluster0.f1gpe.mongodb.net/nehabhatt1221

```
Atlas atlas-bjyw9t-shard-0 [primary] Resturants> db.Addresses.find({name:{$regex:/^Mad/i}},{"name":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("6219cf04792631116914481c"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("6219cf0479263111691448ea"),
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
    _id: ObjectId("6219cf057926311169144b98"),
    address: { coord: [ -74.000002, 40.72735 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madame X'
  },
  {
    _id: ObjectId("6219cf057926311169144c48"),
    address: { coord: [ -73.98171959999999, 40.7499406 ] },
    borough: 'Manhattan',
    cuisine: 'French',
    name: 'Madison Bistro'
  },
  {
    _id: ObjectId("6219cf057926311169144cd1"),
    address: { coord: [ -73.9717845, 40.6897199 ] },
    borough: 'Brooklyn',
    cuisine: 'African',
    name: 'Madiba'
  },
  {
    _id: ObjectId("6219cf057926311169144fd6"),
    address: { coord: [ -73.9040753, 40.9069011 ] },
    borough: 'Bronx',
```