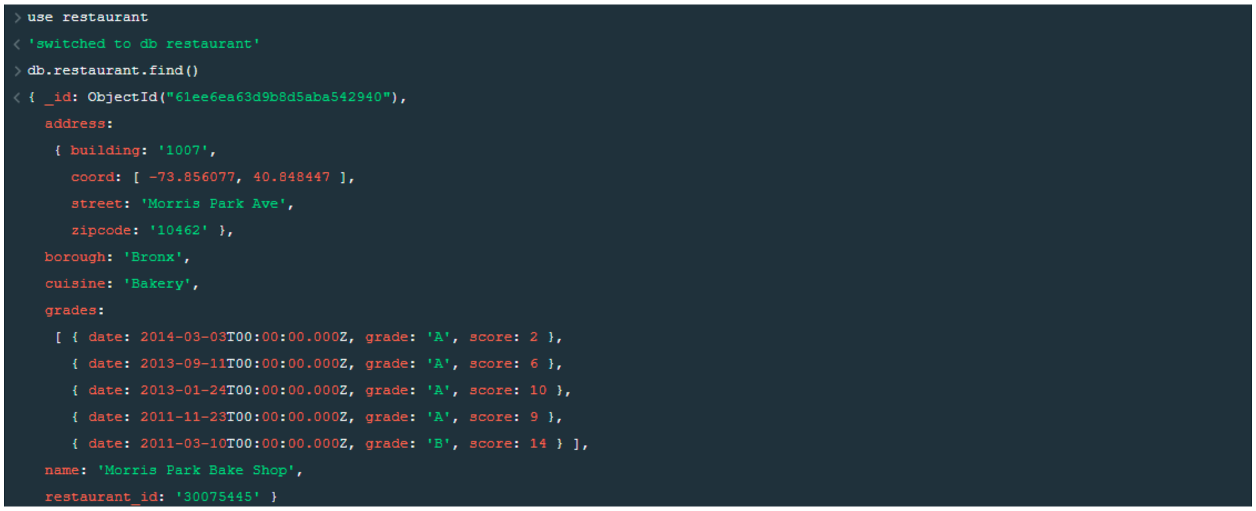
Srikanth COLLATY / Sylvain BUI / Thassadhith AIT LARBI

TP1 NoSQL

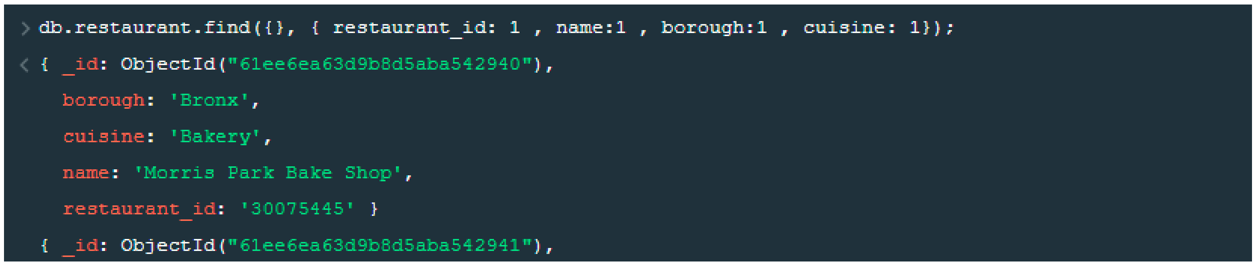
Write a MongoDB query to:

1. Display all the documents in the collection restaurants.



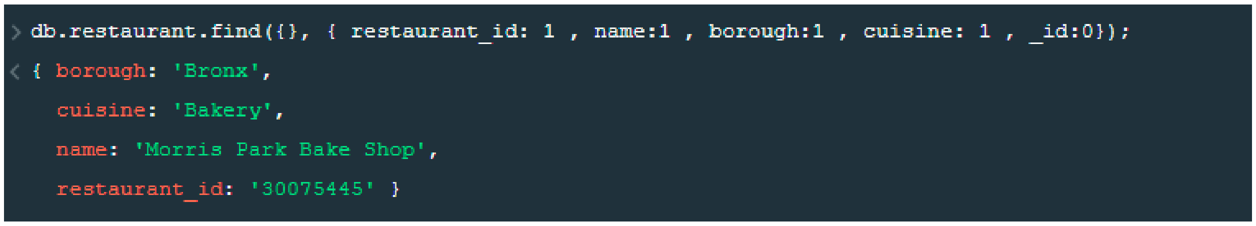
```
> use restaurant
< 'switched to db restaurant'
> db.restaurant.find()
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    address:
     { building: '1007',
       coord: [ -73.856077, 40.848447 ],
       street: 'Morris Park Ave',
       zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades:
     [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
       { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
       { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
       { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },
       { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
```

2. Display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.



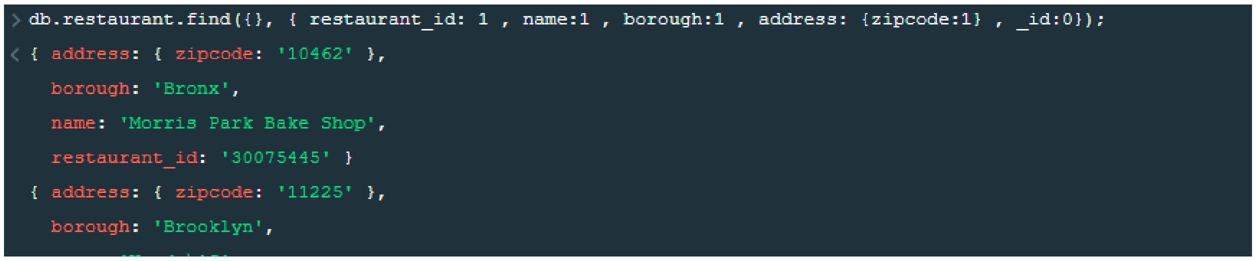
```
> db.restaurant.find({}, { restaurant_id: 1 , name:1 , borough:1 , cuisine: 1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542941"),
```

3. Display the fields restaurant_id, name, borough and cuisine, but exclude the field_id for all the documents in the collection restaurant.



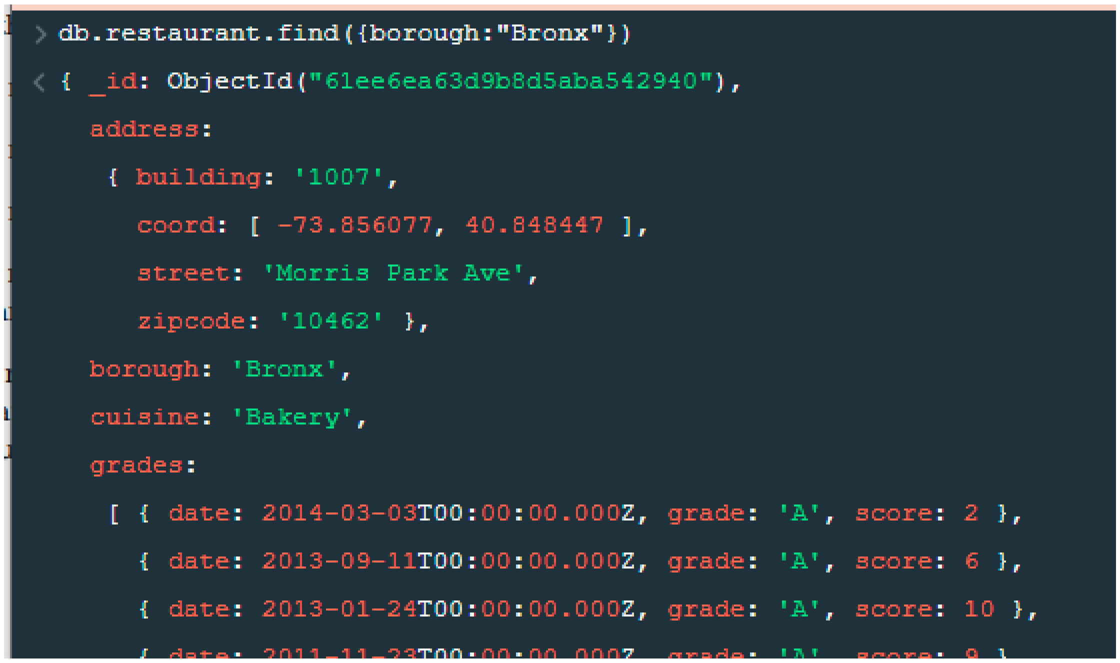
```
> db.restaurant.find({}, { restaurant_id: 1 , name:1 , borough:1 , cuisine: 1 , _id:0});
< { borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
```

4. Display the fields restaurant_id, name, borough and zip code, but exclude the field_id for all the documents in the collection restaurant.



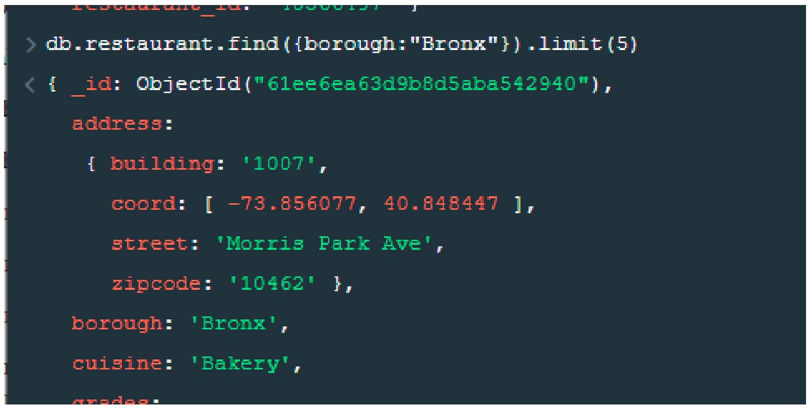
```
> db.restaurant.find({}, { restaurant_id: 1 , name:1 , borough:1 , address: {zipcode:1} , _id:0});
< { address: { zipcode: '10462' },
    borough: 'Bronx',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
  { address: { zipcode: '11225' },
    borough: 'Brooklyn',
```

5. Display all the restaurant which is in the borough Bronx.



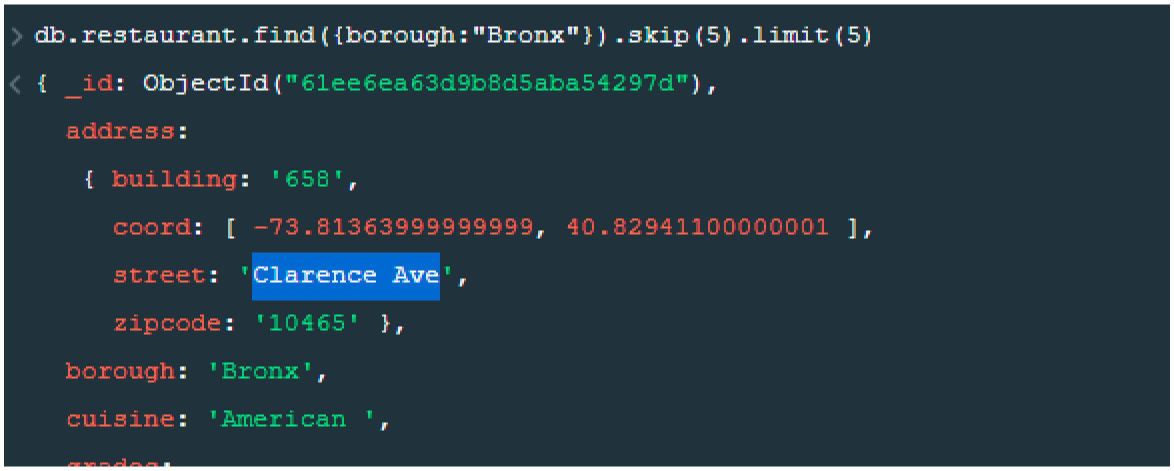
```
> db.restaurant.find({borough:"Bronx"})
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    address:
     { building: '1007',
       coord: [ -73.856077, 40.848447 ],
       street: 'Morris Park Ave',
       zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades:
     [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
       { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
       { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
```

6. Display the first 5 restaurant which is in the borough Bronx



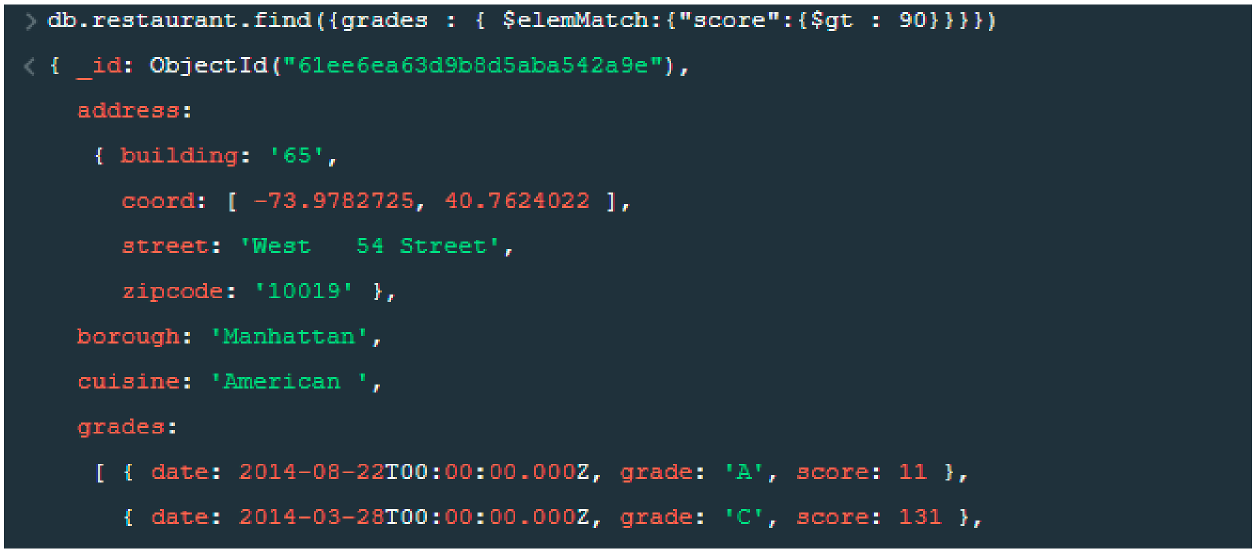
```
> db.restaurant.find({borough:"Bronx"}).limit(5)
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    address:
     { building: '1007',
       coord: [ -73.856077, 40.848447 ],
       street: 'Morris Park Ave',
       zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
```

7. Display the next 5 restaurants after skipping first 5 which are in the borough Bronx.



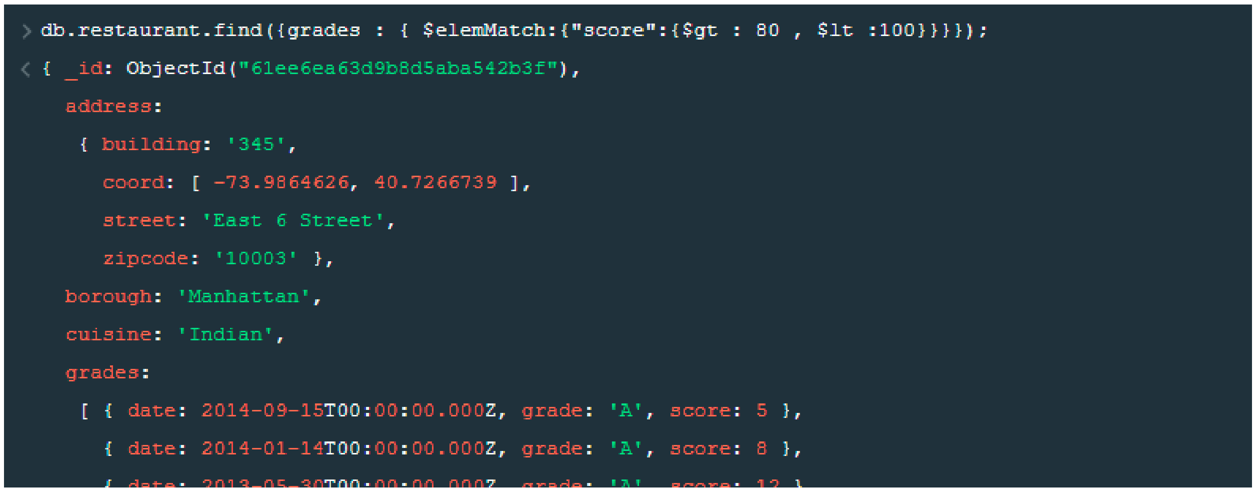
```
> db.restaurant.find({borough:"Bronx"}).skip(5).limit(5)
< { _id: ObjectId("61ee6ea63d9b8d5aba54297d"),
    address:
     { building: '658',
       coord: [ -73.81363999999999, 40.82941100000001 ],
       street: 'Clarence Ave',
       zipcode: '10465' },
    borough: 'Bronx',
    cuisine: 'American ',
```

8. Find the restaurants who achieved a score more than 90.



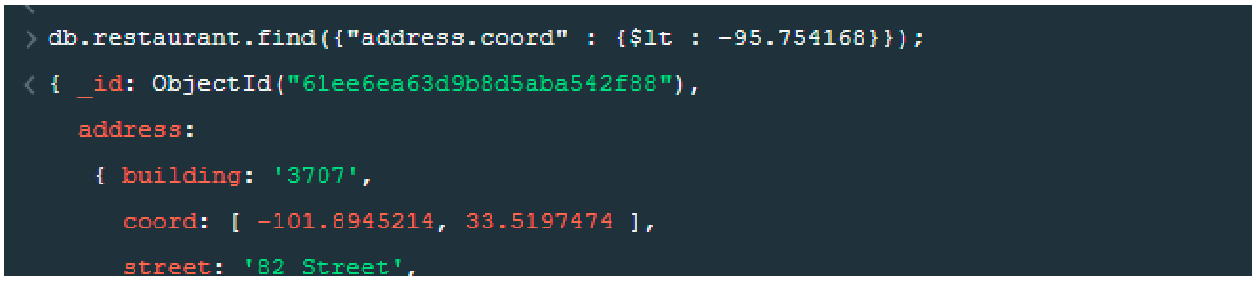
```
> db.restaurant.find({grades : { $elemMatch:{"score":{$gt : 90}}}})
< { _id: ObjectId("61ee6ea63d9b8d5aba542a9e"),
    address:
     { building: '65',
       coord: [ -73.9782725, 40.7624022 ],
       street: 'West   54 Street',
       zipcode: '10019' },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades:
     [ { date: 2014-08-22T00:00:00.000Z, grade: 'A', score: 11 },
       { date: 2014-03-28T00:00:00.000Z, grade: 'C', score: 131 },
```

9. Find the restaurants that achieved a score, more than 80 but less than 100.



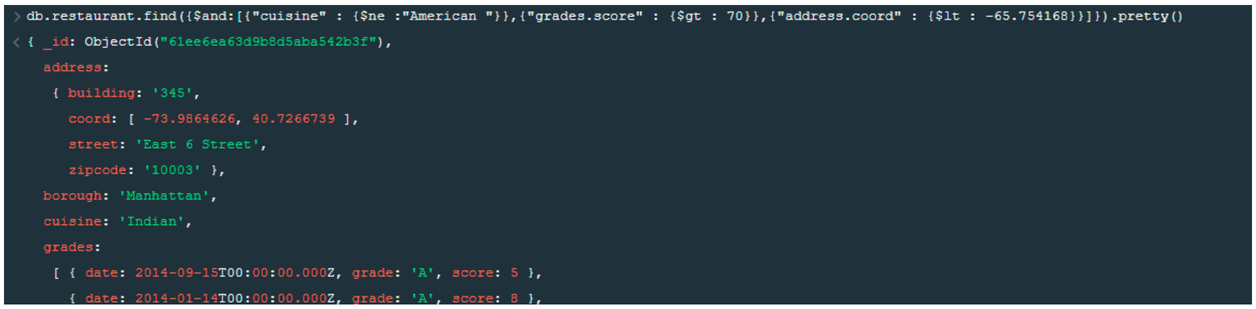
```
> db.restaurant.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});
< { _id: ObjectId("61ee6ea63d9b8d5aba542b3f"),
    address:
     { building: '345',
       coord: [ -73.9864626, 40.7266739 ],
       street: 'East 6 Street',
       zipcode: '10003' },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades:
     [ { date: 2014-09-15T00:00:00.000Z, grade: 'A', score: 5 },
       { date: 2014-01-14T00:00:00.000Z, grade: 'A', score: 8 },
       { date: 2013-05-30T00:00:00.000Z, grade: 'A', score: 12 },
```

10. Find the restaurants which locate in latitude value less than -95.754168.



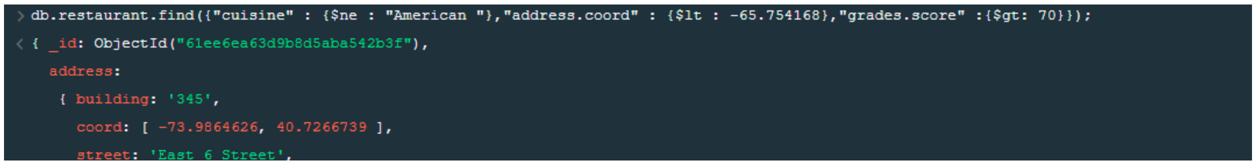
```
> db.restaurant.find({"address.coord" : {$lt : -95.754168}});
< { _id: ObjectId("61ee6ea63d9b8d5aba542f88"),
    address:
     { building: '3707',
       coord: [ -101.8945214, 33.5197474 ],
       street: '82 Street',
```

11. Find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.



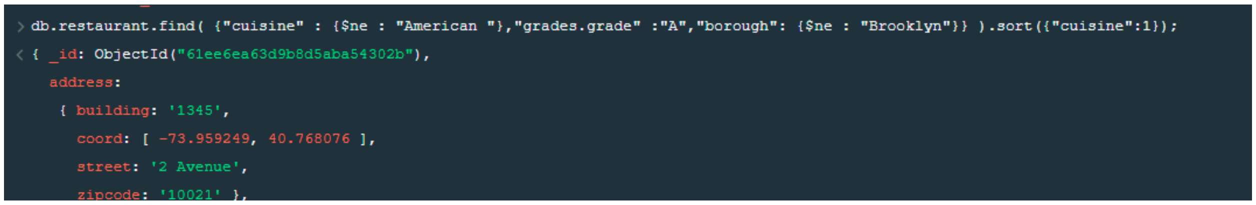
```
> db.restaurant.find({$and:[{"cuisine" : {$ne :"American "}},{"grades.score" : {$gt : 70}},{"address.coord" : {$lt : -65.754168}}]}).pretty()
< { _id: ObjectId("61ee6ea63d9b8d5aba542b3f"),
    address:
     { building: '345',
       coord: [ -73.9864626, 40.7266739 ],
       street: 'East 6 Street',
       zipcode: '10003' },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades:
     [ { date: 2014-09-15T00:00:00.000Z, grade: 'A', score: 5 },
       { date: 2014-01-14T00:00:00.000Z, grade: 'A', score: 8 },
```

12. 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. Note : Do this query without using $and operator.



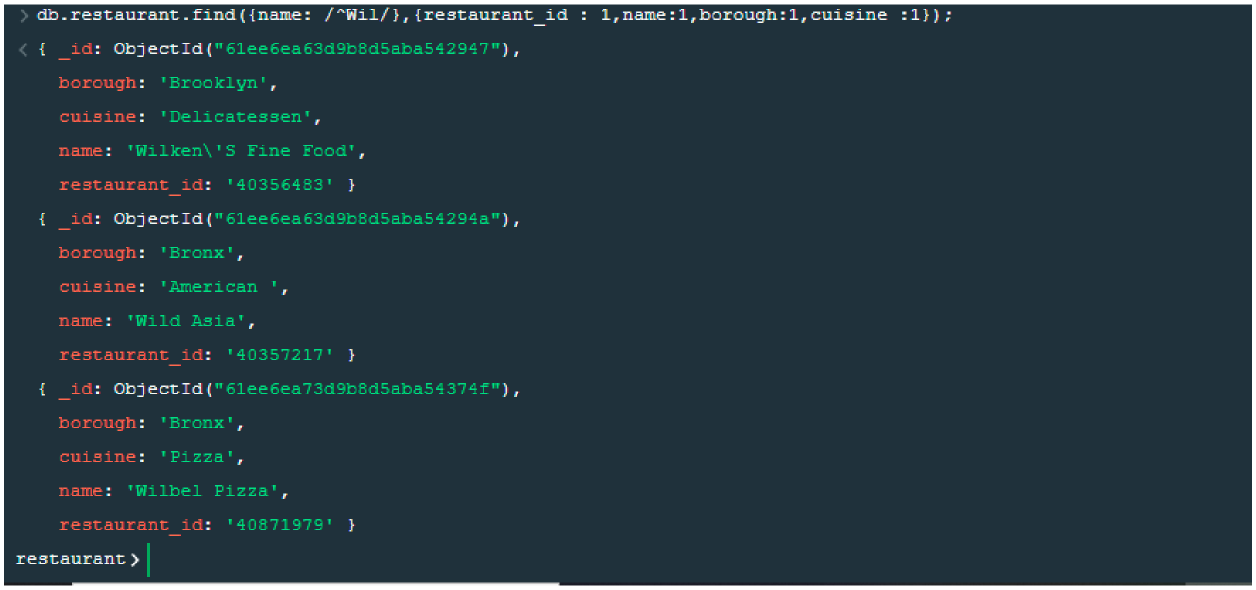
```
> db.restaurant.find({"cuisine" : {$ne : "American "},"address.coord" : {$lt : -65.754168},"grades.score" :{$gt: 70}});
< { _id: ObjectId("61ee6ea63d9b8d5aba542b3f"),
    address:
     { building: '345',
       coord: [ -73.9864626, 40.7266739 ],
       street: 'East 6 Street',
```

13. 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.



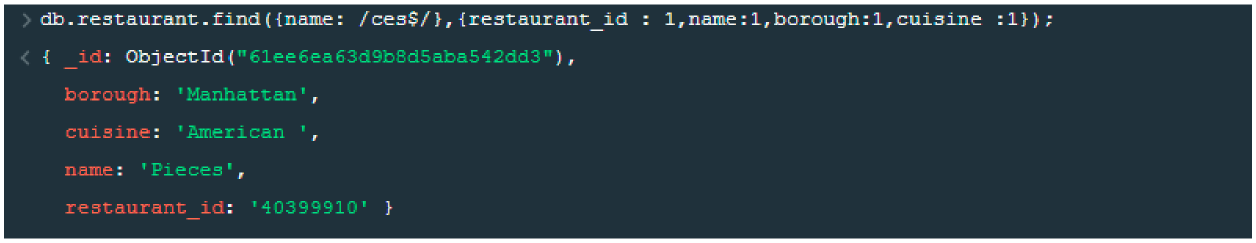
```
> db.restaurant.find( {"cuisine" : {$ne : "American "},"grades.grade" :"A","borough": {$ne : "Brooklyn"}} ).sort({"cuisine":1});
< { _id: ObjectId("61ee6ea63d9b8d5aba54302b"),
    address:
     { building: '1345',
       coord: [ -73.959249, 40.768076 ],
       street: '2 Avenue',
       zipcode: '10021' },
```

14. Find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.



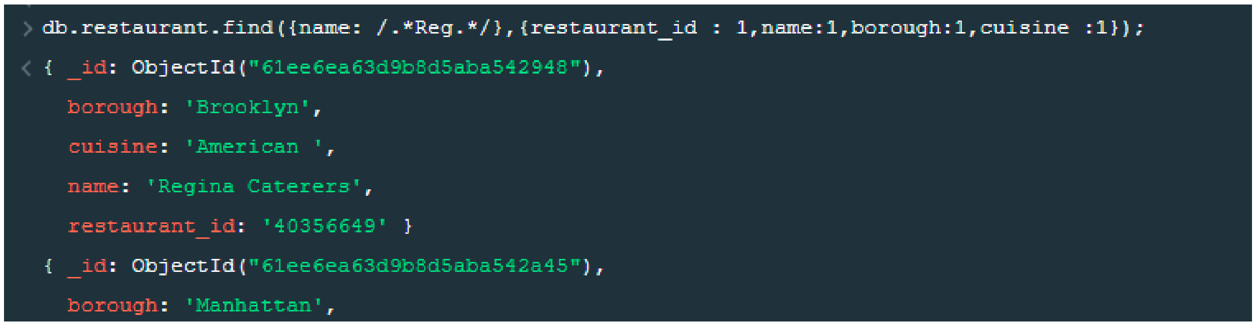
```
> db.restaurant.find({name: /^Wil/},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542947"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Wilken\'S Fine Food',
    restaurant_id: '40356483' }
  { _id: ObjectId("61ee6ea63d9b8d5aba54294a"),
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357217' }
  { _id: ObjectId("61ee6ea73d9b8d5aba54374f"),
    borough: 'Bronx',
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979' }
restaurant>
```

15. Find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.



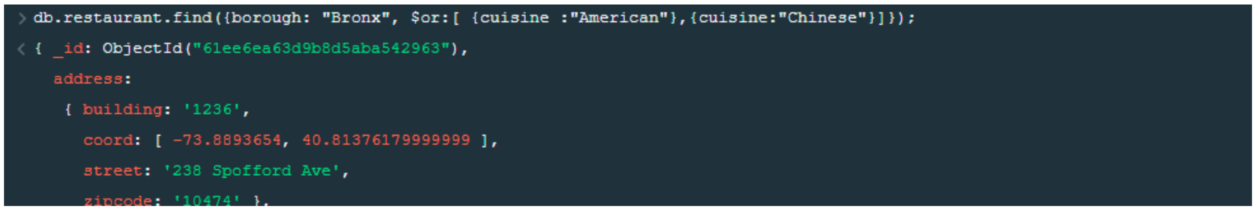
```
> db.restaurant.find({name: /ces$/},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542dd3"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Pieces',
    restaurant_id: '40399910' }
```

16. Find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.



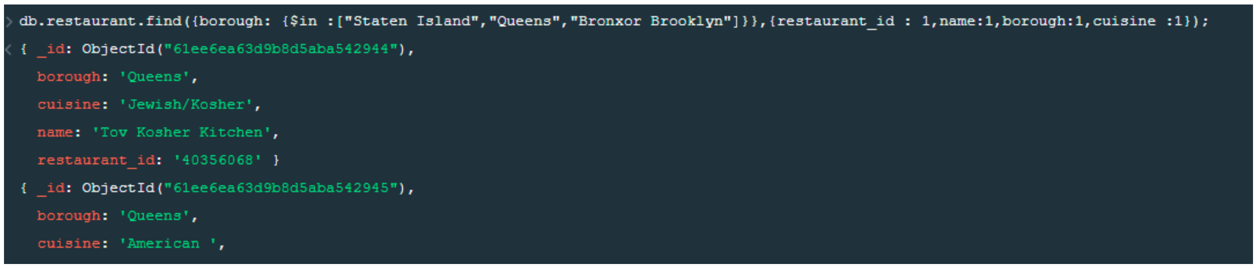
```
> db.restaurant.find({name: /.*Reg.*/},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542948"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542a45"),
    borough: 'Manhattan',
```

17. Find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.



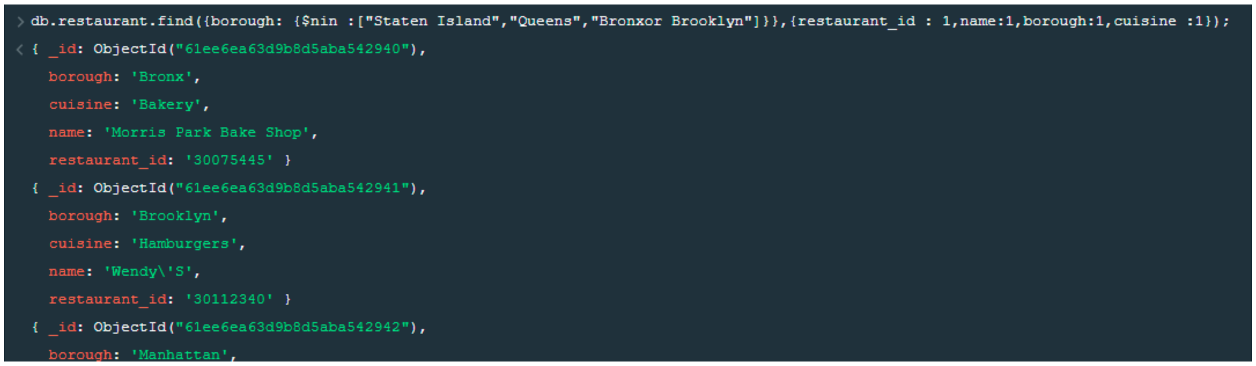
```
> db.restaurant.find({borough: "Bronx", $or:[ {cuisine :"American"},{cuisine:"Chinese"}]});
< { _id: ObjectId("61ee6ea63d9b8d5aba542963"),
    address:
     { building: '1236',
       coord: [ -73.8893654, 40.81376179999999 ],
       street: '238 Spofford Ave',
       zipcode: '10474' },
```

18. Find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.



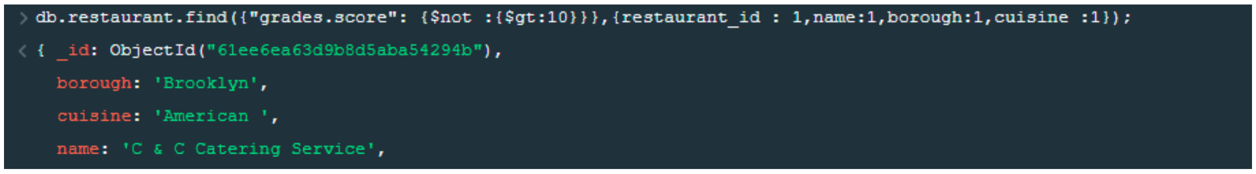
```
> db.restaurant.find({borough: {$in :["Staten Island","Queens","Bronxor Brooklyn"]}},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542944"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356068' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542945"),
    borough: 'Queens',
    cuisine: 'American ',
```

19. 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.



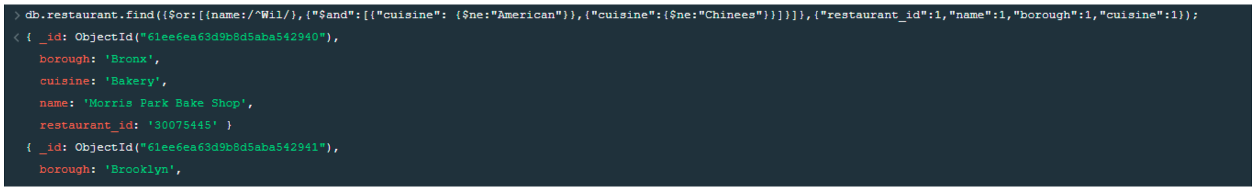
```
> db.restaurant.find({borough: {$nin :["Staten Island","Queens","Bronxor Brooklyn"]}},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542941"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'Wendy\'S',
    restaurant_id: '30112340' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542942"),
    borough: 'Manhattan',
```

20. Find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.



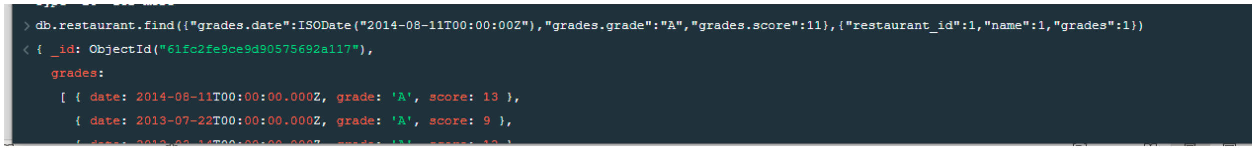
```
> db.restaurant.find({"grades.score": {$not :{$gt:10}}},{restaurant_id : 1,name:1,borough:1,cuisine :1});
< { _id: ObjectId("61ee6ea63d9b8d5aba54294b"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'C & C Catering Service',
```

21. 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'.



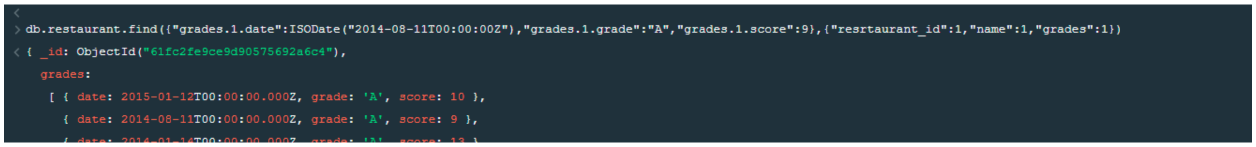
```
> db.restaurant.find({$or:[{name:/^Wil/},{"$and":[{"cuisine": {$ne:"American"}},{"cuisine":{$ne:"Chinees"}}]}]},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1});
< { _id: ObjectId("61ee6ea63d9b8d5aba542940"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
  { _id: ObjectId("61ee6ea63d9b8d5aba542941"),
    borough: 'Brooklyn',
```

22. 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..



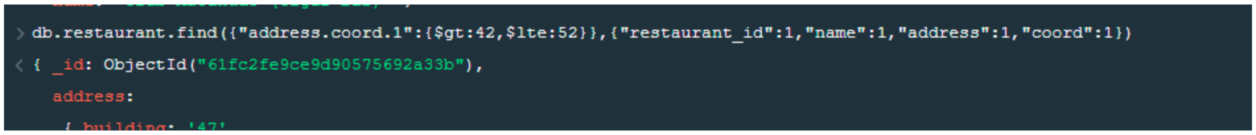
```
> db.restaurant.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11},{"restaurant_id":1,"name":1,"grades":1})
< { _id: ObjectId("61fc2fe9ce9d90575692a117"),
    grades:
     [ { date: 2014-08-11T00:00:00.000Z, grade: 'A', score: 13 },
       { date: 2013-07-22T00:00:00.000Z, grade: 'A', score: 9 },
```

23. 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".



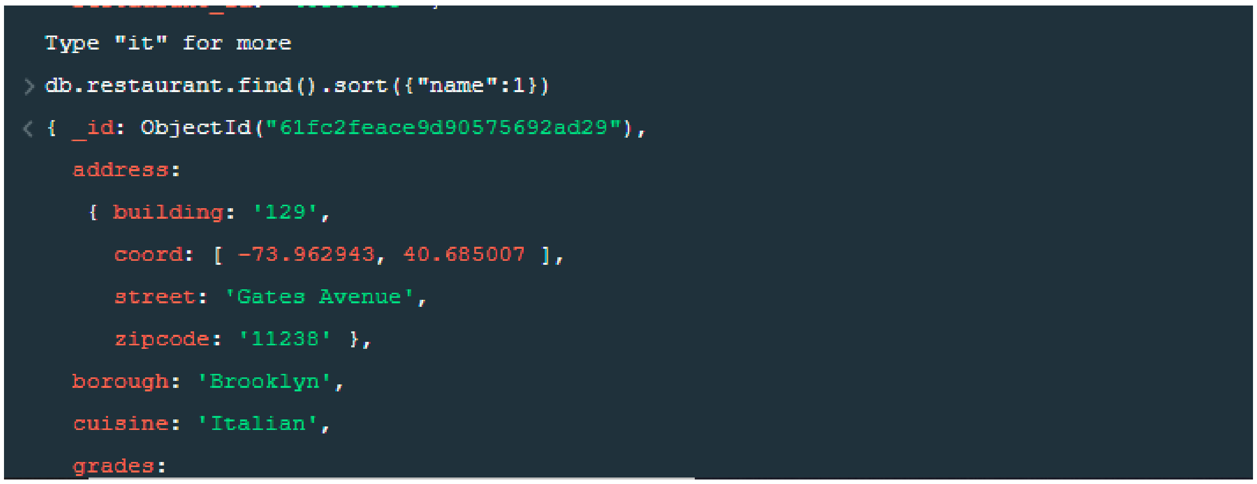
```
<
> db.restaurant.find({"grades.1.date":ISODate("2014-08-11T00:00:00Z"),"grades.1.grade":"A","grades.1.score":9},{"resrtaurant_id":1,"name":1,"grades":1})
< { _id: ObjectId("61fc2fe9ce9d90575692a6c4"),
    grades:
     [ { date: 2015-01-12T00:00:00.000Z, grade: 'A', score: 10 },
       { date: 2014-08-11T00:00:00.000Z, grade: 'A', score: 9 },
```

24. 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..



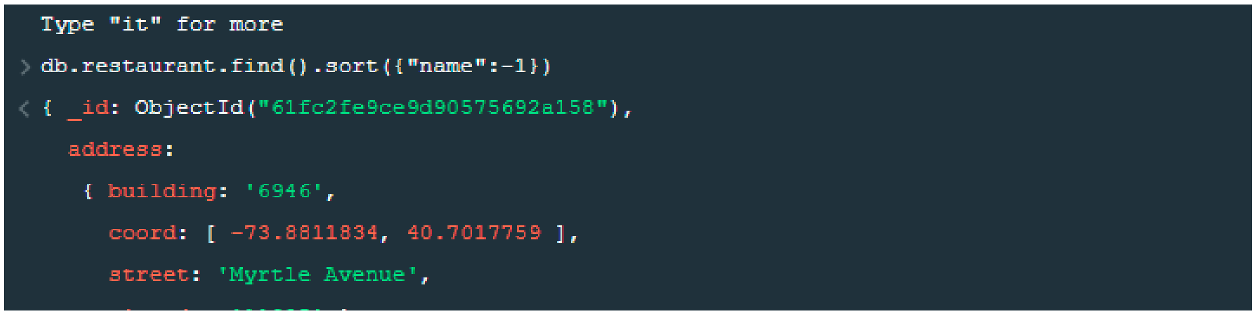
```
> db.restaurant.find({"address.coord.1":{$gt:42,$lte:52}},{"restaurant_id":1,"name":1,"address":1,"coord":1})
< { _id: ObjectId("61fc2fe9ce9d90575692a33b"),
    address:
     { building: '47'
```

25. Arrange the name of the restaurants in ascending order along with all the columns



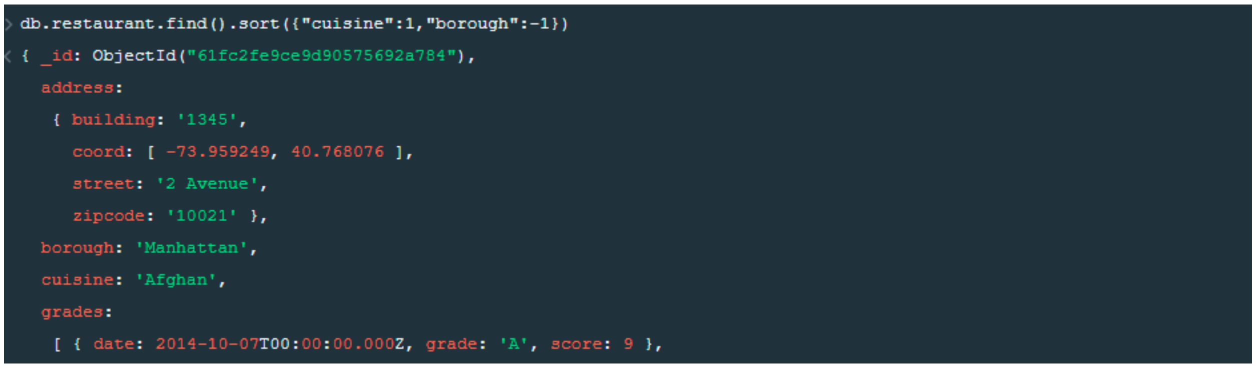
```
  Type "it" for more
> db.restaurant.find().sort({"name":1})
< { _id: ObjectId("61fc2feace9d90575692ad29"),
    address:
     { building: '129',
       coord: [ -73.962943, 40.685007 ],
       street: 'Gates Avenue',
       zipcode: '11238' },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    grades:
```

26. Arrange the name of the restaurants in descending along with all the columns.



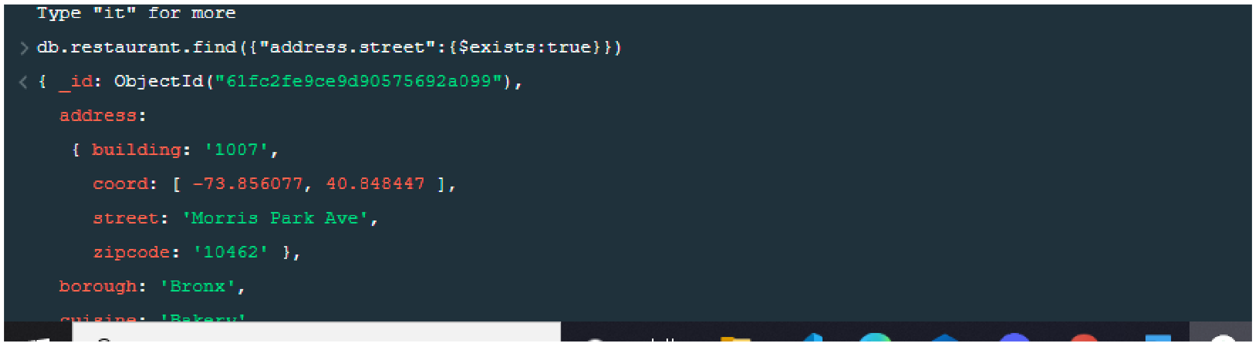
```
  Type "it" for more
> db.restaurant.find().sort({"name":-1})
< { _id: ObjectId("61fc2fe9ce9d90575692a158"),
    address:
     { building: '6946',
       coord: [ -73.8811834, 40.7017759 ],
       street: 'Myrtle Avenue',
```

27. Arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.



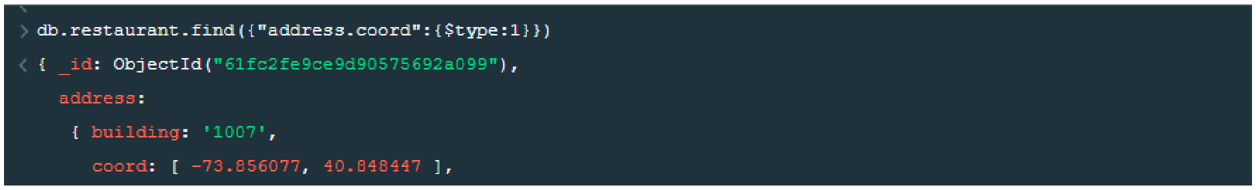
```
> db.restaurant.find().sort({"cuisine":1,"borough":-1})
< { _id: ObjectId("61fc2fe9ce9d90575692a784"),
    address:
     { building: '1345',
       coord: [ -73.959249, 40.768076 ],
       street: '2 Avenue',
       zipcode: '10021' },
    borough: 'Manhattan',
    cuisine: 'Afghan',
    grades:
     [ { date: 2014-10-07T00:00:00.000Z, grade: 'A', score: 9 },
```

28. Know whether all the addresses contains the street or not.



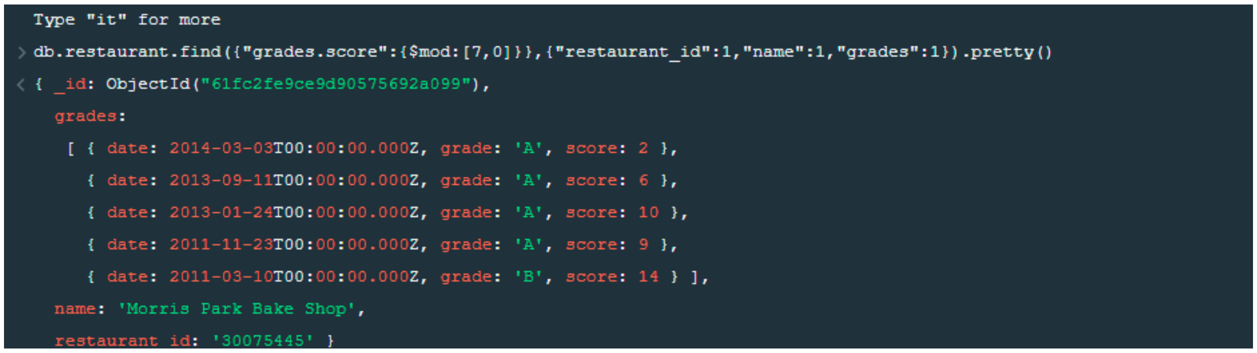
```
 Type "it" for more
> db.restaurant.find({"address.street":{$exists:true}})
< { _id: ObjectId("61fc2fe9ce9d90575692a099"),
    address:
     { building: '1007',
       coord: [ -73.856077, 40.848447 ],
       street: 'Morris Park Ave',
       zipcode: '10462' },
    borough: 'Bronx',
```

29. Select all documents in the restaurants collection where the coord field value is Double



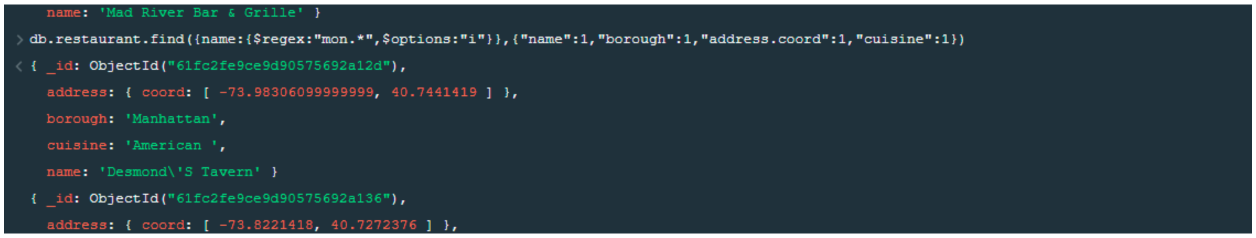
```
> db.restaurant.find({"address.coord":{$type:1}})
< { _id: ObjectId("61fc2fe9ce9d90575692a099"),
    address:
     { building: '1007',
       coord: [ -73.856077, 40.848447 ],
```

30. Select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.



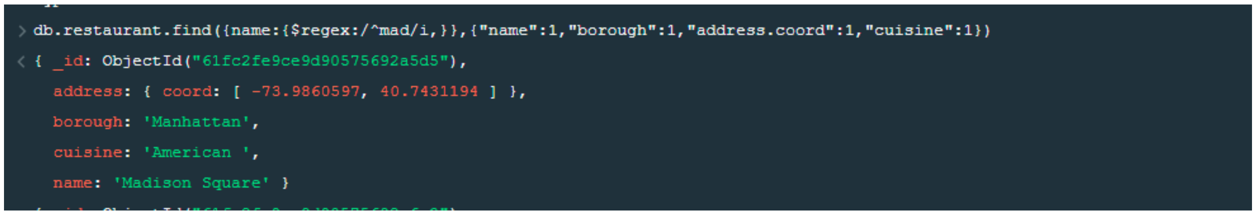
```
 Type "it" for more
> db.restaurant.find({"grades.score":{$mod:[7,0]}},{"restaurant_id":1,"name":1,"grades":1}).pretty()
< { _id: ObjectId("61fc2fe9ce9d90575692a099"),
    grades:
     [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
       { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
       { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
       { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },
       { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445' }
```

31. Find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains ’mon’ as three letters somewhere in its name.



```
    name: 'Mad River Bar & Grille' }
> db.restaurant.find({name:{$regex:"mon.*",$options:"i"}},{"name":1,"borough":1,"address.coord":1,"cuisine":1})
< { _id: ObjectId("61fc2fe9ce9d90575692a12d"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Desmond\'S Tavern' }
  { _id: ObjectId("61fc2fe9ce9d90575692a136"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
```

32. Find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.



```
> db.restaurant.find({name:{$regex:/^mad/i,}},{"name":1,"borough":1,"address.coord":1,"cuisine":1})
< { _id: ObjectId("61fc2fe9ce9d90575692a5d5"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square' }
```