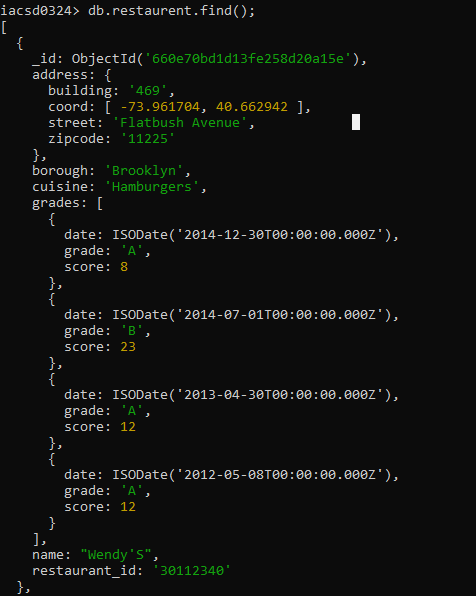
1.Write a MongoDB query to display all the documents in the collection restaurant

🡪 db.restaurent.find();





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

all the documents in the collection restaurant.

🡺

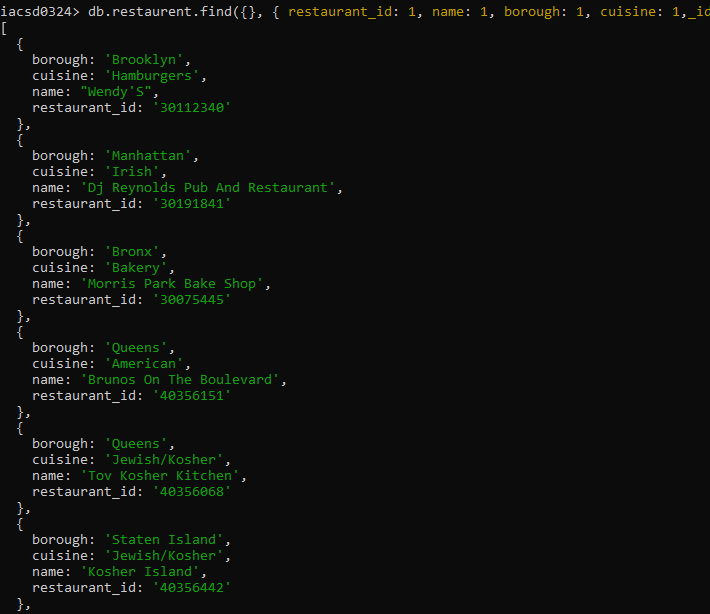
iacsd0324> db.restaurent.find({},{restaurant\_id:1,name:1,borough:1,cuisine:1});



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.restaurent.find({}, { restaurant\_id: 1, name: 1, borough: 1, cuisine: 1,\_id:0 });

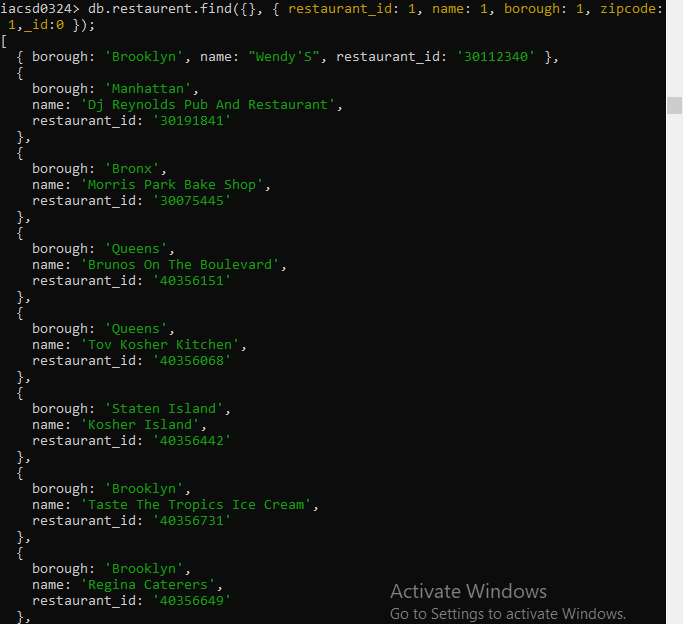


4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code,

but exclude the field \_id for all the documents in the collection restaurant.

🡺

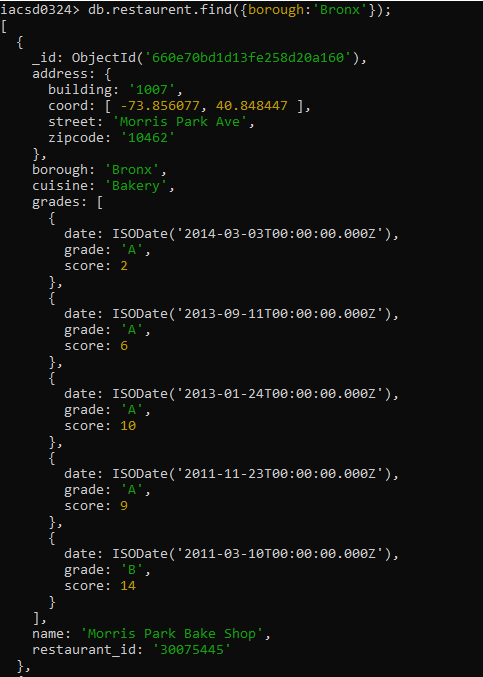
db.restaurent.find({}, { restaurant\_id: 1, name: 1, borough: 1, zipcode: 1,\_id:0 });



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

🡺

db.restaurent.find({borough:'Bronx'});



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

🡺

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

the borough Bronx.

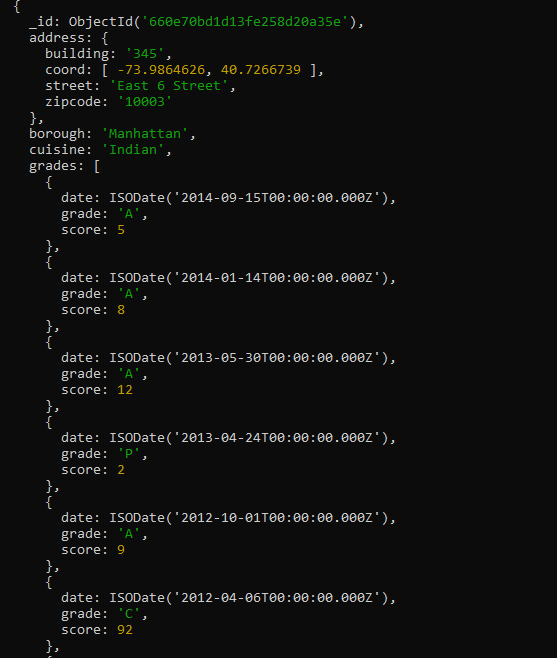
🡺

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

🡺

db.restaurent.find({grades:{$elemMatch:{score:{$gt:90}}}});





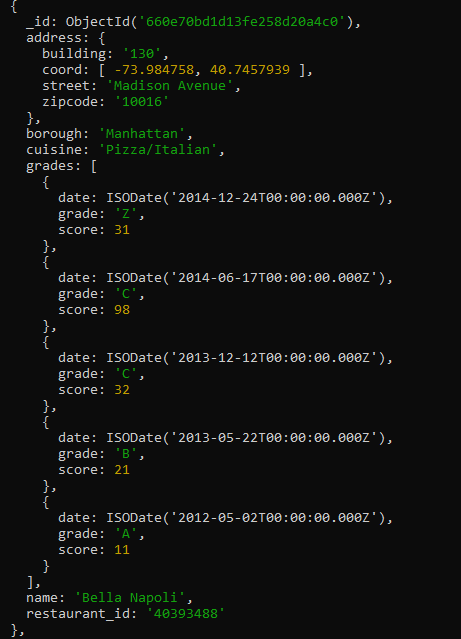
9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but

less than 100.

🡺

iiacsd0324> db.restaurent.find({grades:{$elemMatch:{score:{$gt:80,$lt:100}}}});





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

95.754168.

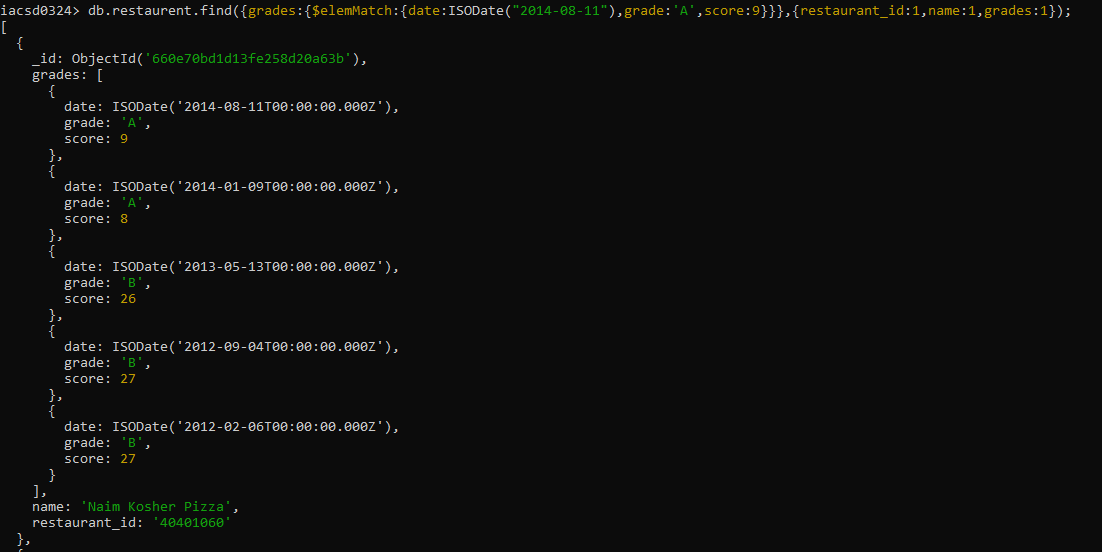
🡺

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.restaurent.find({grades:{$elemMatch:{date:ISODate("2014-08-11"),grade:'A',score:9}}},{restaurant\_id:1,name:1,grades:1});

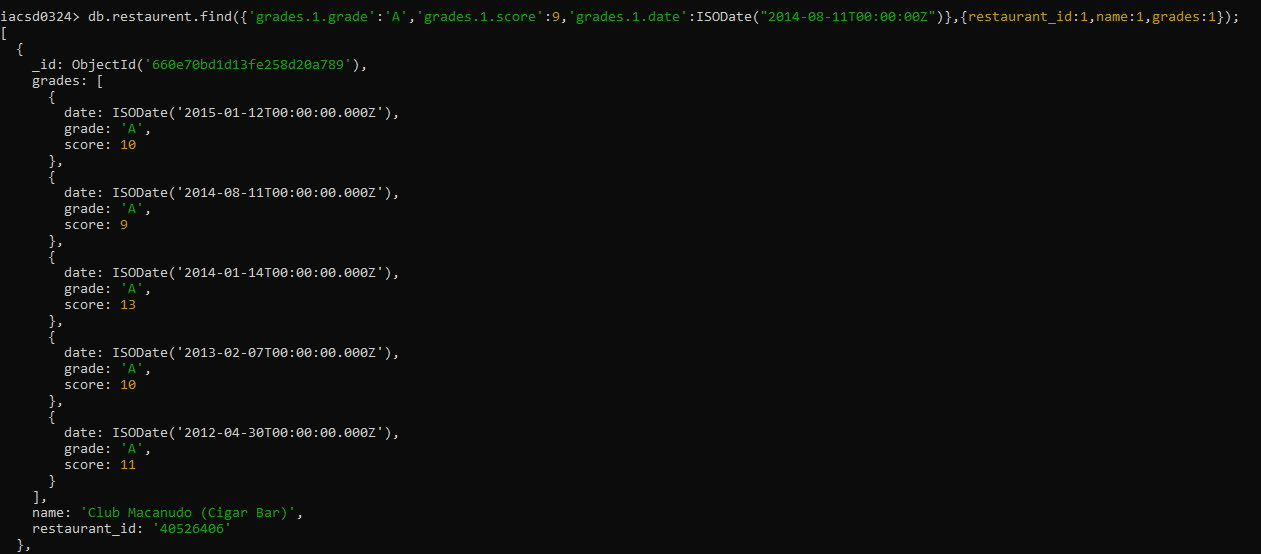


23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants

where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate

"2014-08-11T00:00:00Z".

🡺db.restaurent.find({'grades.1.grade':'A','grades.1.score':9,'grades.1.date':ISODate("2014-08-11T00:00:00Z")},{restaurant\_id:1,name:1,grades:1});



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

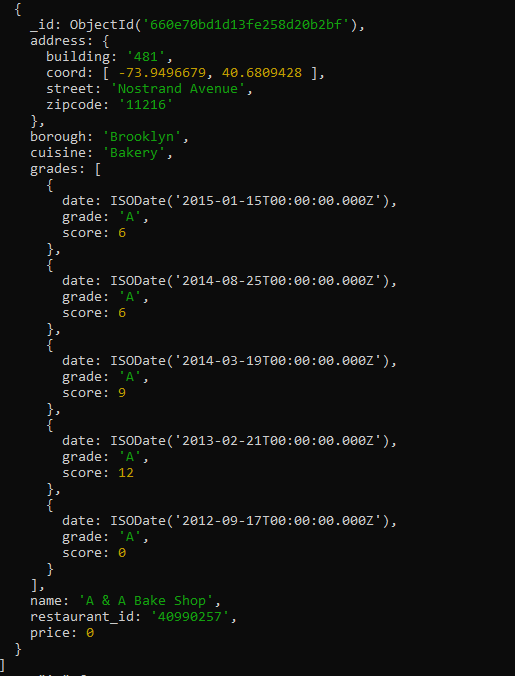
🡺 db.restaurent.find({'address.coord.1':{$gt:42,$lt:52}},{ restaurant\_id:1,name:1,address:1})



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

with all the columns.

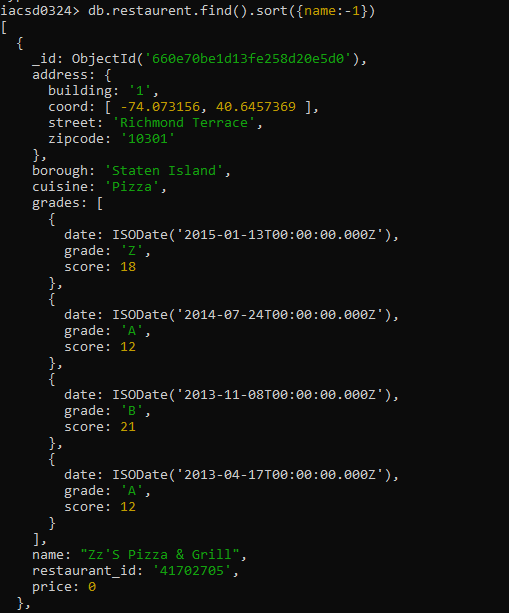
🡺db.restaurent.find().sort({name:1})



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

all the columns.

🡺 db.restaurent.find().sort({name:-1})



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.

🡺