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

1. Write a MongoDB query to 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:1});

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

1. 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.

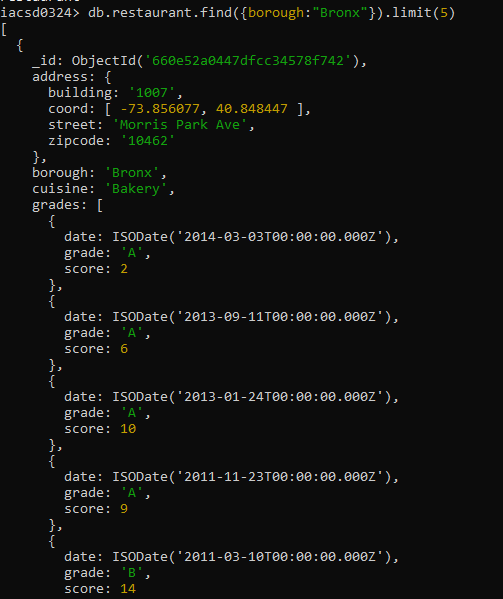
db.restaurant.find({},{restaurant\_id:1,name:1,borough:1,address:{zipcode:1},\_id:0});

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

db.restaurant.find({borough:"Bronx"});

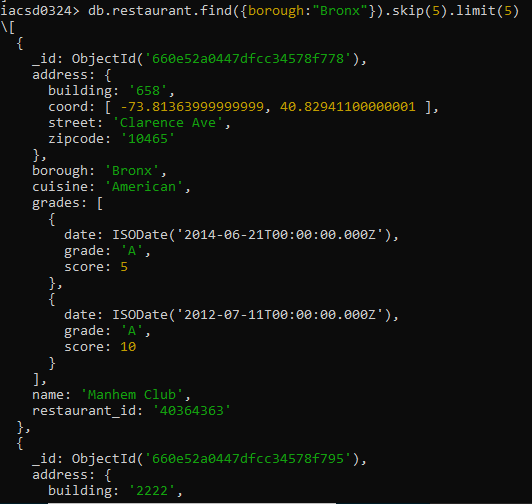
1. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

Ans:db.restaurant.find({borough:"Bronx"}).limit(5)



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

db.restaurant.find({borough:"Bronx"}).skip(5).limit(5)



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

db.restaurant.find({'grades.score':{$gt:90}}});

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

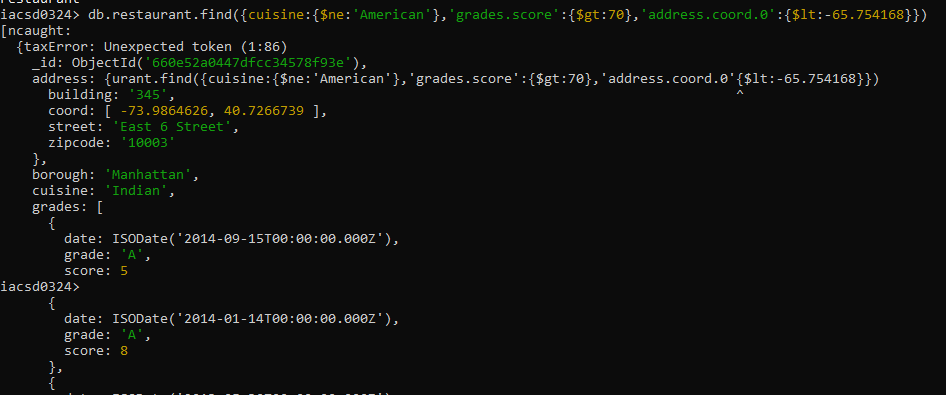
db.restaurant.find({'grades.score':{$gt:80,$lt:100}}});

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

95.754168.

db.restaurant.find({'address.coord.0':{$lt:-95.754168}});

1. 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.

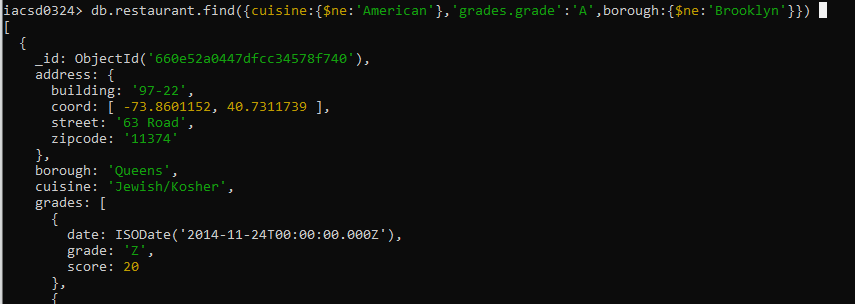
ANS= db.restaurant.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord.0'{$lt:-65.754168}})

1. 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.restaurant.find({cuisine:{$ne:'American'},'grades.score':{$gt:70},'address.coord.1’:{$lt:65.754168}})

1. 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.restaurant.find({cuisine:{$ne:'American'},'grades.grade':'A',borough:{$ne:'Brooklyn'}})



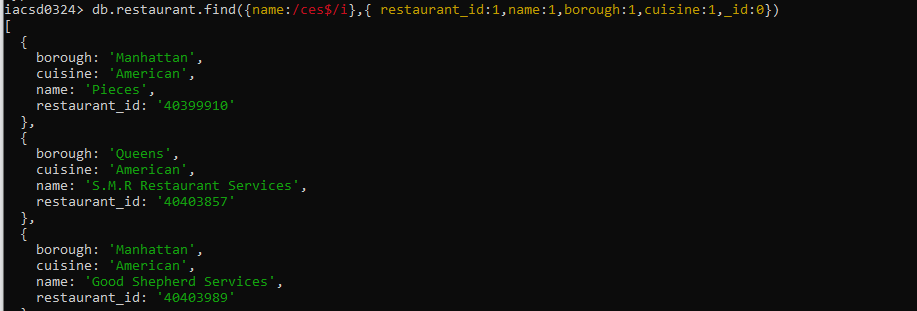
1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

db.restaurant.find({name:/^Wil/i},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



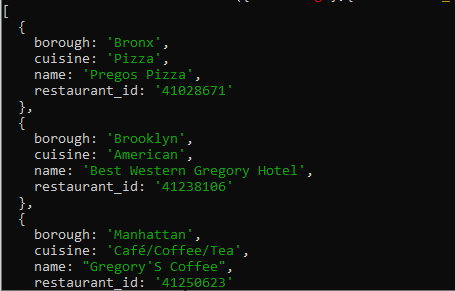
1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

db.restaurant.find({name:/ces$/i},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



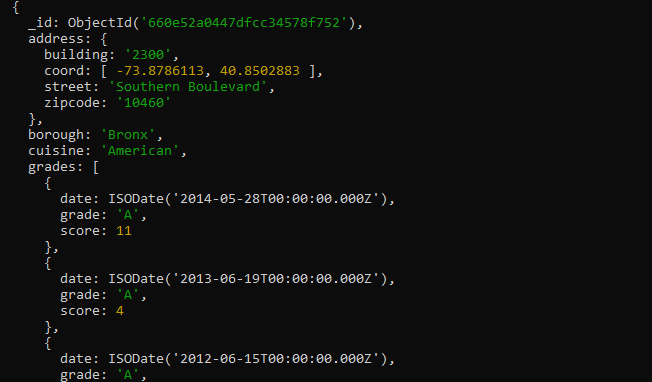
1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

iacsd0324> db.restaurant.find({name:/reg/},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



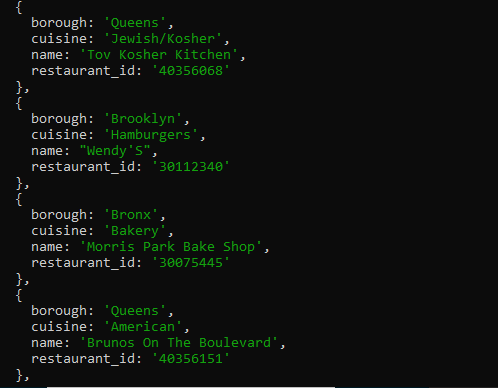
1. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

iacsd0324> db.restaurant.find({borough:'Bronx',cuisine:{$in:['American','Chinese']}})



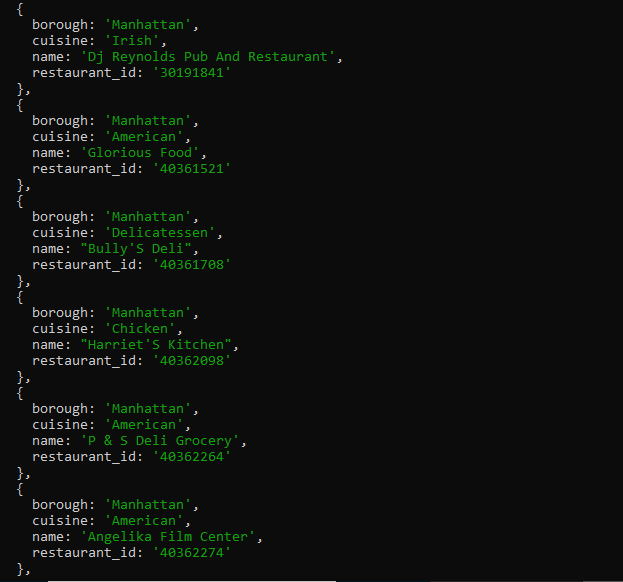
1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

db.restaurant.find({borough:{$in:['Staten Island','Queens','Bronx','Brooklyn']}},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



1. 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.

iacsd0324> db.restaurant.find({borough:{$nin:['Staten Island','Queens','Bronx','Brooklyn']}},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



1. 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.

iacsd0324> db.restaurant.find({'grades.score':{$lt:10}},{ restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



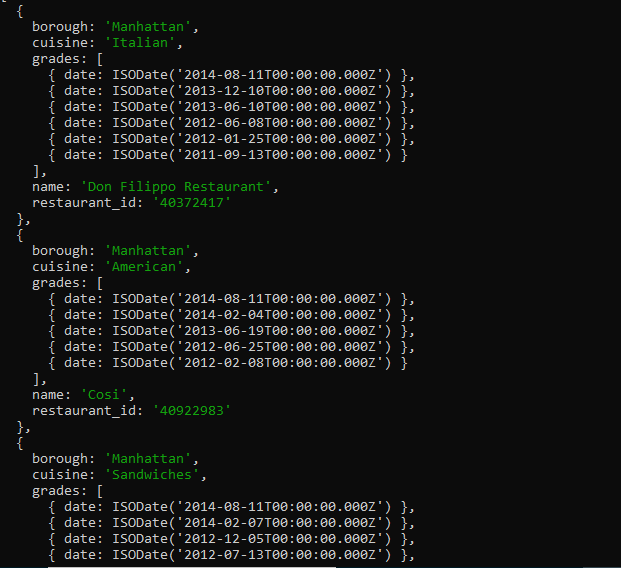
1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

db.restaurant.find({$or:[{name:/^Wil/i},{cuisine:{$nin:['American','Chinese']}}]},{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



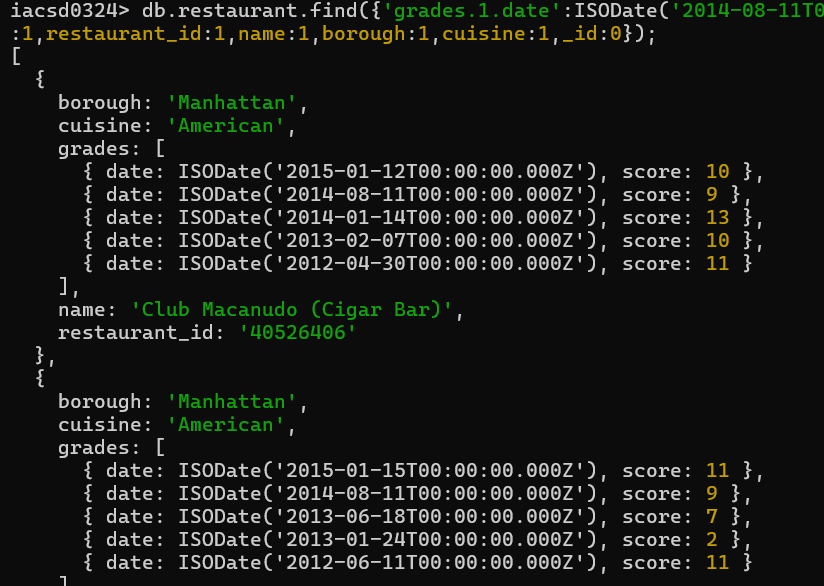
1. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates

db.restaurant.find({grades:{$elemMatch:{date:ISODate('2014-08-11T00:00:00.000Z'), grade:'A', score:11}}},{'grades.date':1,restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0})



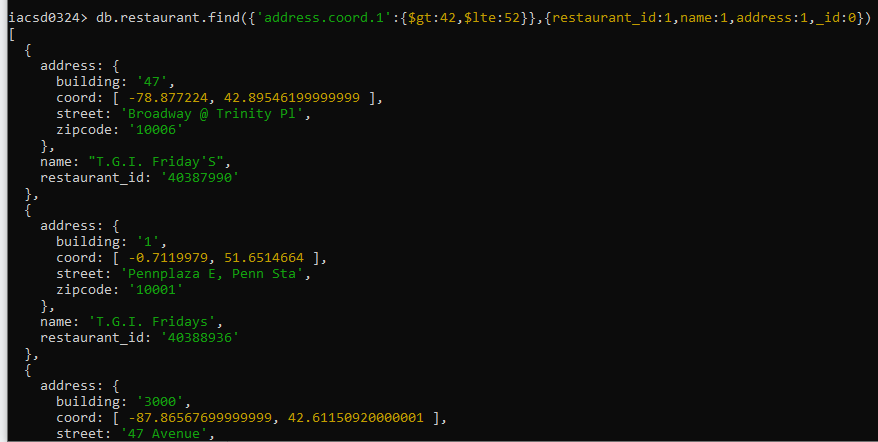
1. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

db.restaurant.find({'grades.1.date':ISODate('2014-08-11T00:00:00.000Z'), 'grades.1.grade':'A','grades.1.score':9},{'grades.date':1,'grades.score':1,restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0});



1. 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.restaurant.find({'address.coord.1':{$gt:42,$lte:52}},{restaurant\_id:1,name:1,address:1,\_id:0})



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

db.restaurant.find().sort({"name":1}).skip(800)



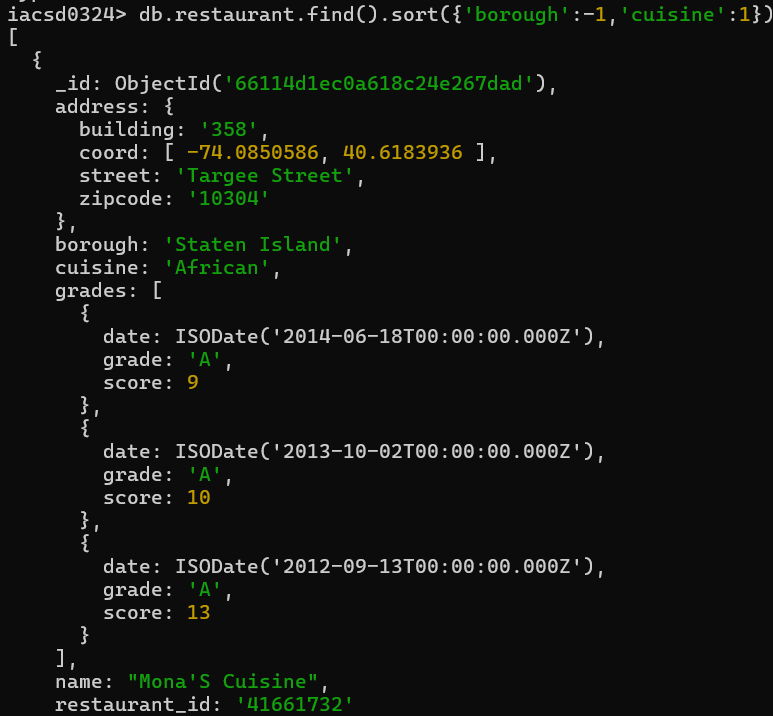
1. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

db.restaurant.find().sort({"name":-1})



1. 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.restaurant.find().sort({'borough':-1,'cuisine':1})



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

db.restaurant.find({'address.street':{$exists:true}});

1. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.
2. 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.
3. 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.
4. 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.