1. Write a mongo DB query to display all the documents in the collection restaurants.
2. Write a mongo DB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.
3. db.Addresses.find({},{restaurant\_id:1,name:1,borough:1,cuisine:1})

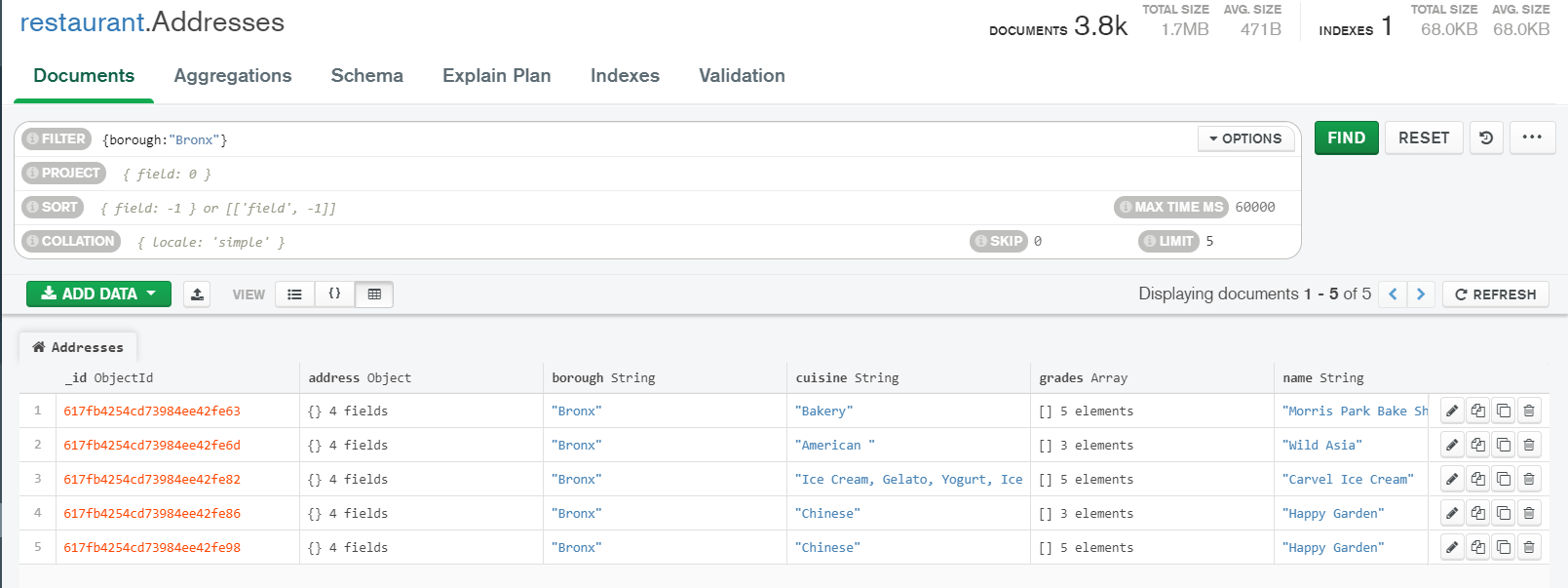
3. WRITE A MONGODB QUERY TO DISPLAY THE FIELDS RESTAURANT\_ID,NAME,BOROUGH,AND CUSINE,BUT EXCLUDE THE FIELD\_IS FOR ALL THE DOCUMTS IN THE COLLECTION RESTAURANT.

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

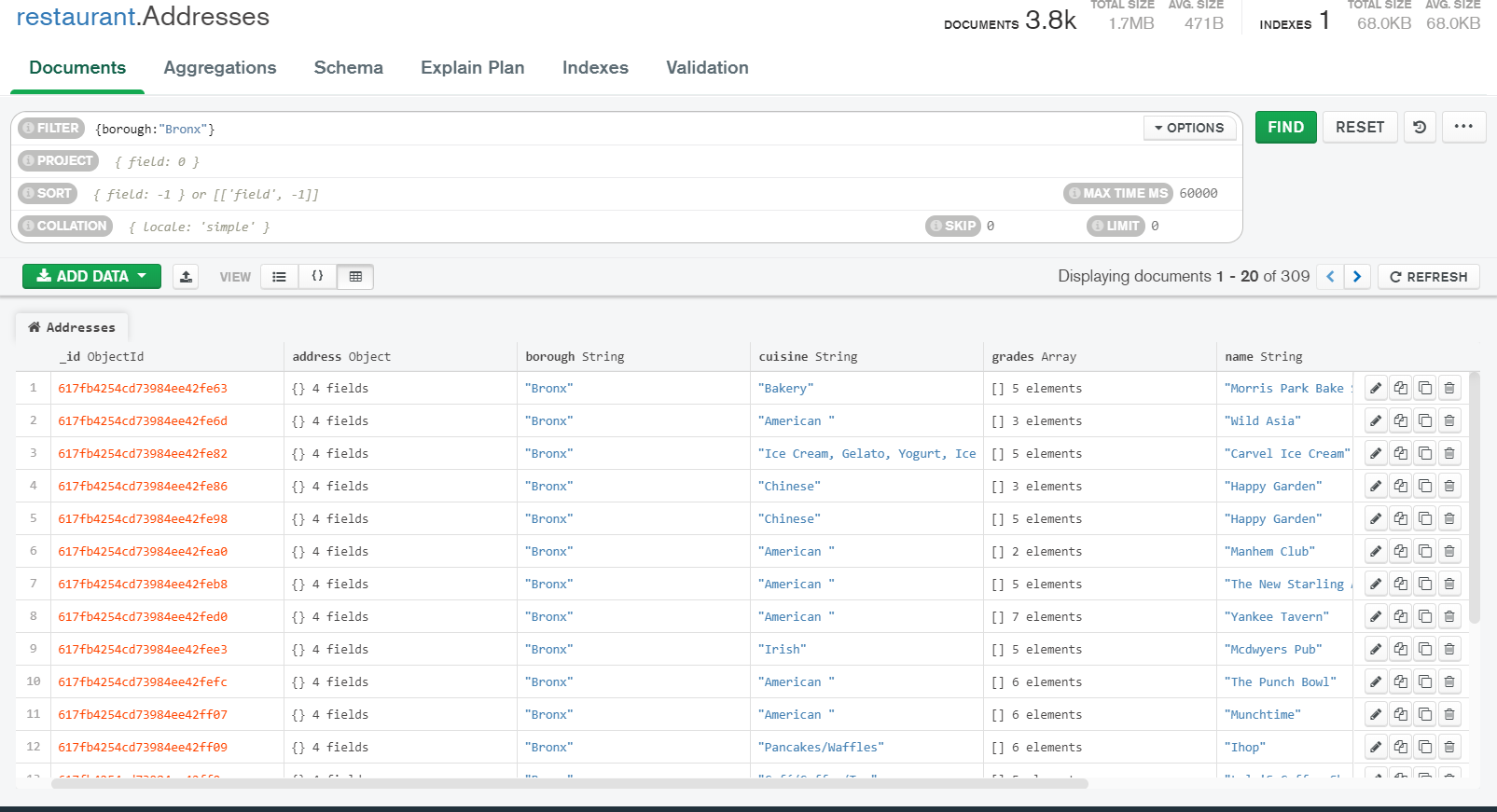
4. WRITE A MONGODB QUERY TO DISPLAY THE FIELDS RESTAURANT\_ID,NAME,BOROUGH,AND ZIP CODE,BUT EXCLUDE THE FIELD\_IS FOR ALL THE DOCUMTS IN THE COLLECTION RESTAURANT.

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

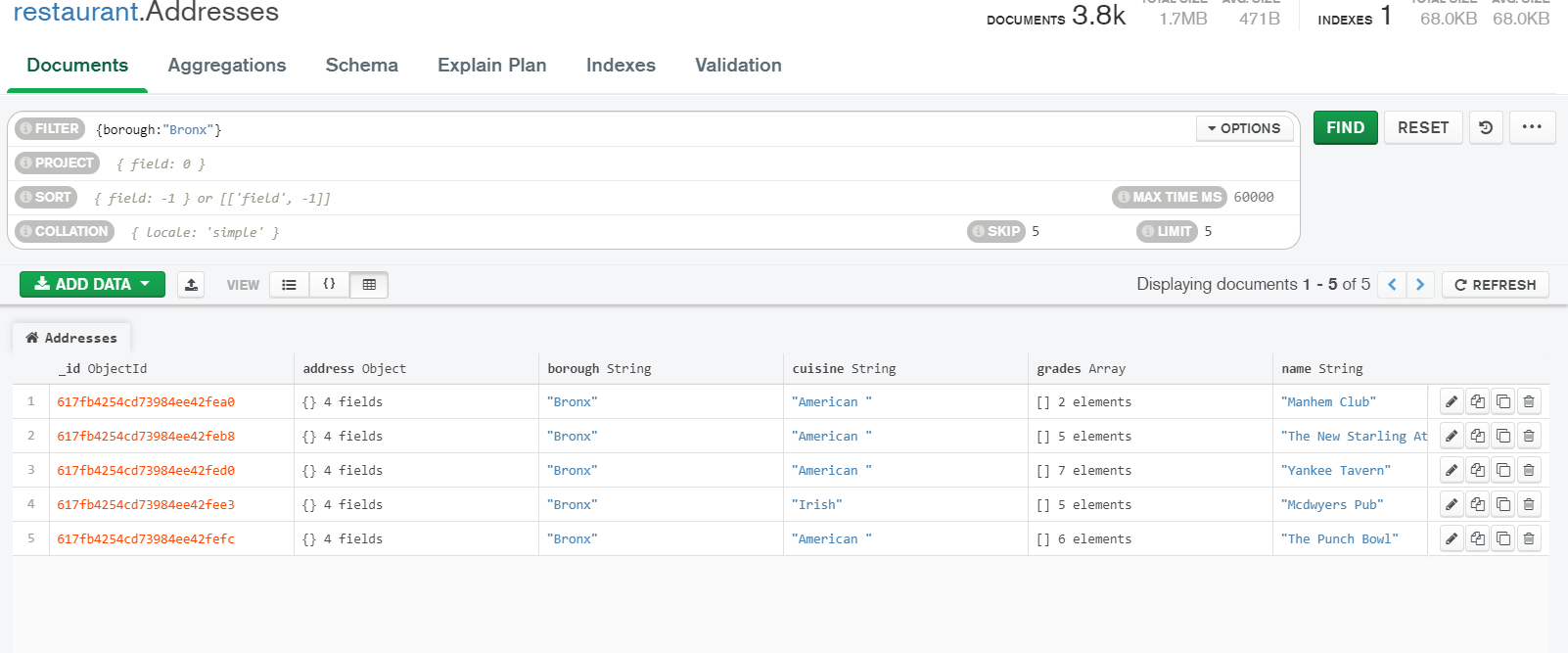
5. WRITE A MONGODB QUERY TO DISPLAY THE first 5 restaurant which is in the borough bronx.



6.write a mongodb query to display all the restaurant which is in the borough Bronx.



7.write a mongo dB 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 restaurant who achived a score more that 90.

db.Addresses.find({"grades.score":{$gte:90}})

9.Write a mongodb query to find the restaurant who achived a score more that 80 less than 100.

db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});

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

db.Addresses.find({"address.coord" : {$lt : -95.754168}});

11.Write a mongodb query to find the restaurant which do not prepare any cuisine of ‘american’and achived a score more than 70 and located in the longitude less than -65.754168

db.Addresses.find(

{

"cuisine" : {$ne : "American "},

"grades.score" :{$gt: 70},

"address.coord" : {$lt : -65.754168}

}

);

12. Write a mongodb query to find the restaurant which do not prepare any cuisine of ‘american’and achived a score more than 70 and located in the longitude less than -65.754168

db.Addresses.find(

{

"cuisine" : {$ne : "American "},

"grades.score" :{$gt: 70},

"address.coord" : {$lt : -65.754168}

}

);

13. write a mongodb query to find the restaurants which do not prepare any cuisine of American and achived a grad point ‘a’ not belongs to the borough Brooklyn the documents must be display according to the cuisine in descending order

db.Addresses.find( {

"cuisine" : {$ne : "American "},

"grades.grade" :"A",

"borough": {$ne : "Brooklyn"}

}

).sort({"cuisine":-1});

14. write a mongodb query to find the restaurant id,name ,brough and cuisine for those restaurants which contains ‘wil’ as first three letters for its name .

db.Addresses.find(

{name: /^Wil/},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

15.write a mongodb query to find the restaurant id,name,borough and cuisine for those restaurants which contains ‘ces’ as last three letters for its name

db.Addresses.find(

{name: /ces$/},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

16.write a mongodb query to find the restaurant id,name,borough and cuisine for those restaurents contain ‘reg’ as three letters somewhere in its name.

db.Addresses.find(

{"name": /.\*Reg.\*/},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

17.write a mongodb query to find the restaurents which belong to the borough Bronx and prepared either American or chinse dish

db.Addresses.find(

{

"borough": "Bronx" ,

$or : [

{ "cuisine" : "American " },

{ "cuisine" : "Chinese" }

]

}

);

18.write a mongodb query to find the restaurant id,name borough and cusine for those restaurents which belong to the borough staten island or queen or Bronxor Brooklyn.

db.Addresses.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

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 bronoxor Brooklyn.

db.restaurants.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

20.write a mongodb query to find the restaurant id,name,borough and cuisine for those restaurents which achieved a score which is not more than 10

db.Addresses.find(

{"grades.score" :

{ $not:

{$gt : 10}

}

},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

21.write a mongodb query to find the restaurant id,name,borough and cuisine for those restaurants which prepared dish except ‘American’ and ‘chinses’ or restaurent’s name begins with letter ‘wil’.

db.Addresses.find(

{$or: [

{name: /^Wil/},

{"$and": [

{"cuisine" : {$ne :"American "}},

{"cuisine" : {$ne :"Chinees"}}

]}

]}

);

22.write a mongdb query to find the restaurant id,name,and grades for those restaurents which achieved a grade of’a’ and scored 11 on an ISODate “2014-08-11T00.00.00z” among many of survey dates.

23.write a mongodb query to find the restauren id,name and grades for those restaurents where the 2nd element of grades array contains a grad of ‘A’ AND SCORE 9 ON AN ISOate “2014-08-11T00:00:00Z”

db.Addresses.find(

{ "grades.1.date": ISODate("2014-08-11T00:00:00Z"),

"grades.1.grade":"A" ,

"grades.1.score" : 9

},

);

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

db.Addresses.find(

{

"address.coord.1": {$gt : 42, $lte : 52}

},

);

25.write a mongodb query to arrange the name of the restaurents in ascending order along with all the columns.

db.Addresses.find().sort({"name":1});

26.write a mongodb query to arrange the name of the restaurents in s=descending along with all the columns.

db.Addresses.find().sort(

{"name":-1}

);

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

db.Addresses.find().sort(

{"name":-1}

);

28.write a mongdb query to know whether all the addresses in the restaurents the street or not

db.Addresses.find(

{"address.street" :

{ $exists : true }

}

);

29 write a mongodb query which will select all documents in the restaurents collection share the coord field value in double.

db.Addresses.find(

{"address.coord" :

{$type : 1}

}

);

30.write a mongodb query which will select the restaurant id,name and grades foe those restaurents which return 0 as a remainder after dividing the score by 7.

db.Addresses.find(

{"grades.score" :

{$mod : [7,0]}

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);

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.Addresses.find(

{ name :

{ $regex : "mon.\*", $options: "i" }

},

);

32. write a mongodb query to find the restaurant name,borough,longitude,and attitude and cuisine for those restaurants which contains ‘mad’as three letters somewhere in its name.

db.Addresses.find(

{ name :

{ $regex : /^Mad/i, }

},

);