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

db.restaurants.find().pretty()

{

"\_id" : ObjectId("625e99b24f4a151c0ab16db4"),

"address" : {

"building" : "8825",

"coord" : [

-73.8803827,

40.7643124

],

"street" : "Astoria Boulevard",

"zipcode" : "11369"

},

"borough" : "Queens",

"cuisine" : "American",

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

db.restaurants.find({restaurants\_id:1,name:1,borough:1,cuisine:1}).pretty()

>

> db.restaurants.find({},{restaurants\_id:1,name:1,borough:1,cuisine:1}).pretty()

{

"\_id" : ObjectId("625e99b24f4a151c0ab16db4"),

"borough" : "Queens",

"cuisine" : "American",

"name" : "Brunos On The Boulevard"

}

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.restaurants.find({},{\_id:0,restaurants\_id:1,name:1,borough:1,cuisine:1}).pretty()

{

"borough" : "Queens",

"cuisine" : "American",

"name" : "Brunos On The Boulevard"

}

>

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.restaurants.find({},{\_id:0,restaurants\_id:1,name:1,borough:1,cuisine:1,"address.zipcode":1}).pretty()

{

"address" : {

"zipcode" : "11369"

},

"borough" : "Queens",

"cuisine" : "American",

"name" : "Brunos On The Boulevard"

}

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

db.restaurants.find({borough:"Bronx"}).pretty()

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

db.restaurants.find({borough:"Bronx"}).limit(5).pretty()

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

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

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

db.restaurants.find({"grades.score":{$gt:90}})

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

db.restaurants.find({"grades.score":{$gt:80,$lt:90}})

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

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

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.

db.restaurants.find({$and:[{"cuisine":{$ne:"American"}},{"grades.score":{$gt:70}},{"address.coord":{$lt:-65.754168}}]});

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.

Note : Do this query without using $and operator.

db.restaurants.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 achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

db.restaurants.find({cuisine:{$ne:"American"},"grades.grade":"A","borough":{$ne:"Brooklyn"}}).sort({"cuisine":-1}).pretty()

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.

db.restaurants.find({name:/^Wil/},{restaurants\_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 contain 'ces' as last three letters for its name.

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

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.

db.restaurants.find({name:/.\*Reg.\*/},{restaurants\_id:1,name:1,borough:1,cuisine:1})

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

db.restaurants.find({borough:"Bronx",$or:[{cuisine:"American"},{cuisine:"Chinese"}]}).pretty()

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.

db.restaurants.find({borough:{$in:["Staten Island","Queens","Bronx","Brooklyn"]}},{restaurants\_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 Bronxor Brooklyn.

db.restaurants.find({borough:{$nin:["Staten Island","Queens","Bronx","Brooklyn"]}},{restaurants\_id:1,name:1,borough:1,cuisine:1}).pretty()

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.

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'.

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.,