MongoDB Assignment3

1.Display all the documents in the collection restaurants.

db.addresses.find({}).pretty()

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

db.addresses.find({}, {restaurant_id: 1 ,name: 1 ,borough: 1, cuisine: 1}).pretty()

3.Display the fields restaurant_id, name, borough and cuisine but exclude the field _id .

db.addresses.find({},{_id:0 , restaurant_id: 1 ,name: 1 ,borough: 1, cuisine: 1}).pretty()

4.Display the fields restaurant_id, name, borough and zip code, but exclude field _id.

db.addresses.find({},{_id : 0 , restaurant_id: 1 ,name: 1 ,borough: 1, "address.zipcode" : 1}).pretty()

5.Display the first 5 restaurants which is the borough Bronx.

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

6.Display all the restaurants which is in the borough Bronx.

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

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

db.addresses.find({borough: "Bronx"},{name: 1,borough: 1}).limit(5).skip(5)

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

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

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

db.addresses.find({$and:[{"grades.score" : {$gt:80}},{"grades.score": {$lt: 100}}]})

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

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

11.Find restaurants that do not prepare any cuisine of ‘American’ and their grade score more than 70 and latitude less than –65.75418

db.addresses.find({cuisine: {$ne: "American "}, "grades.score": {$gt: 70}, "address.coord": {$lt: -65.754168}})

12. Find restaurants that do not prepare any cuisine of ‘American’ and their grade score more than 70 and located in the longitude less than –65.75418.

db.addresses.find({cuisine:{$ne: "American "}, "grades.score": {$gt: 70}, "address.coord": {$lt: -65.754168}})

13. Find restaurants that do not prepare any cuisine of ‘American’ and and achieved grade point ‘A’ not belongs to the borough Brooklyn. Display in descending order.

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

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

db.addresses.find({name: {$regex:/^Wil/}},{_id:1,name: 1,borough: 1, cuisine: 1})

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

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

16. Find the restaurant id, name, borough and cuisine for those restaurants which contain ‘Reg’ as last three letters somewhere in its name.

db.addresses.find({name: {$regex:/Reg/i}},{_id:1,name: 1,borough: 1, cuisine: 1})

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

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

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.addresses.find({"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},{_id:0,name: 1,borough: 1, cuisine: 1,restaurant_id: 1})

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.addresses.find({"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

20.Find the restaurant id , name, borough and cuisine for those restaurants 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. Find the restaurant id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinese' or restaurant's name begins with letter 'Wil".

db.addresses.find({$or: [{name: /^Wil/}, {"$and": [{"cuisine" : {$ne :"American "}},{"cuisine" : {$ne :"Chinese"}}]}]}, {"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1})

22.Find the restaurant id, name, and grade for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00:00Z" among many of survey dates.

db.addresses.find({"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A" , "grades.score" : 11}, {"restaurant_id" : 1,"name":1,"grades":1})

23. Find the restaurant id, name, and grade for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08- 11T00:00:00:00Z" among many of survey dates.

db.addresses.find({ "grades.1.date": ISODate("2014-08-11T00:00:00Z"),"grades.1.grade":"A" ,"grades.1.score" : 9 },{"restaurant_id" : 1,"name":1,"grades":1})

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 up to 52.

db.addresses.find({"address.coord.1": {$gt : 42, $lte : 52}},{"restaurant_id" : 1,"name":1,"address":1,"coord":1})

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

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

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

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

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

db.addresses.find().sort({"cuisine":1,"borough" : -1,})

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

db.addresses.find({"address.street" : { $exists : true } })

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

db.addresses.find({"address.coord" : {$type : 1}})

30.Select the restaurant id, name, and grades for those restaurants which returns 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. Find the restaurant id, name, borough, longitude and attitude and cuisine for those restaurants which contain ‘mon’ as last three letters somewhere in its name.

db.addresses.find({name : { $regex : "mon.*", $options: "i" } },{"name":1,"borough":1,"address.coord":1,"cuisine" :1})

32. Find the restaurant id, name, borough, longitude and attitude and cuisine for those restaurants which contain ‘Mad’ as first three letters somewhere in its name.

db.addresses.find({ name :{ $regex : /^Mad/i, } },{"name":1,"borough":1,"address.coord":1,"cuisine":1})