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

**>db.restaurant.find().pretty()**

2. 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})**

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

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

**> db.restaurant.find({"borough":"Bronx"}).pretty()**

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

**> db.restaurant.find({"borough":"Bronx"}).limit(5).pretty()**

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).pretty()**

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

**>db.restaurant.find({"grades.0.score":{$gt :90 }}).pretty()**

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

**> db.restaurant.find({"grades.0.score":{$gt :80,$lt:100}}).pretty()**

**>db.restaurant.find({"grades.0.score":{$lt:50}}).pretty().count()**

**3764**

10. 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}} ).pretty()**

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.restaurant.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 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"} }).sort({"cuisine":-1})**

13. 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/},**

**{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1})**

14. 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.

**>db.restaurant.find({"grades.0.score":**

**{$not {$gt:10}},**

**{ “ restaurant\_id”:1,**

**“name”:1,”borough”:1,”cuisine”:1**

**}**

**).pretty()**

15. 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({name: /^Wil/},**

**$or:[{ "cuisine" : "American "},{ "cuisine" : "Chinese" }]**

**{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1}).pretty()**

16. 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.date": ISODate("2014-08-11T00:00:00Z"),**

**"grades.grade":"A" ,**

**"grades.score" :9**

**},**

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

**).pretty()**

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

**>db.restaurants.find().sort({"name":1}).pretty()**

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

>**db.restaurant.find({"address.street" : { $exists : true } }**

**).pretty()**

19. 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.restaurant.find({ “name”:**

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

**},**

**{**

**"name":1,**

**"borough":1,**

**"address.coord":1,**

**"cuisine" :1**

**}**

**).pretty()**

20. 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.

**db.restaurant.find(**

**{ “name” :**

**{ $regex : /^Mad/i, }**

**},**

**{**

**"name":1,**

**"borough":1,**

**"address.coord":1,**

**"cuisine" :1**

**}**

**).pretty()**