**Using restaurants.json write the following queries**

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

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

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

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

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

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

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

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

**db.restaurant.find({"address.coord":{$lt:-95.754168}}).pretty()**

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 Bronx or Brooklyn.

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

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

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

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

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({"cuisine":1},{"borough":-1}).pretty()**

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":{"$gt":42,"$lt":52}},{"restaurant\_id":1,"address":1,"name":1}).pretty()**

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}).pretty()**

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.

**db.restaurant.find({"grades.score":{"$not":{"$gt":10}}},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()**

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({"cuisine":1},{"borough":-1}).pretty()**

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

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.date":ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A","grades.score":9},{"restaurant\_id":1,"name":1,"grades":1,"\_id":0}).pretty()**

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":{$lt:-65.754168}}).pretty()**