Assignmnet_1

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

Ans: **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.

Ans: **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.

Ans: **db.restaurant.find({},{_id: 0,restaurant_id:1,name:1, _id:0,borough:1,cuisine:1})**

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.

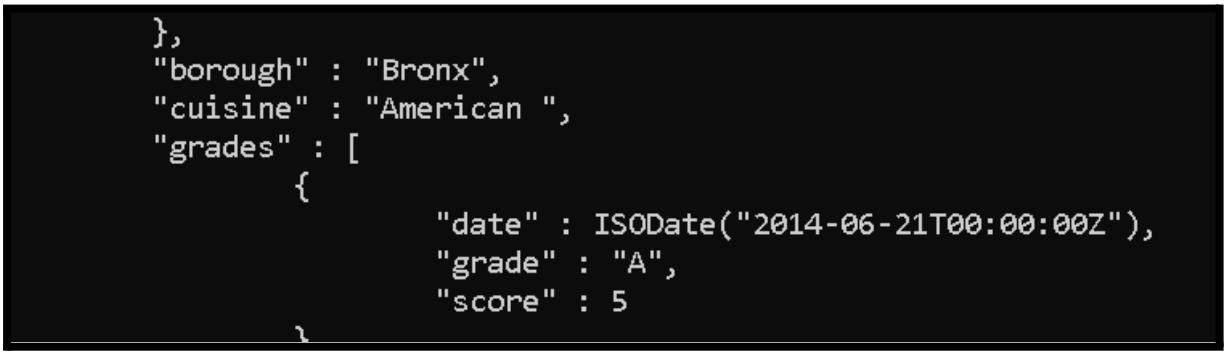
Ans: **db.restaurant.find({},{"address.zipcode":1,restaurant_id:1,name:1,bourough:1,_id:0}) .pretty()**

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

Ans: **db.restaurant.find({borough:"Bronx"},{_id:0}).pretty()**

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

Ans: db.restaurant.find({borough:"Bronx"},{_id:0}).pretty().limit(5)



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

Ans: **db.restaurant.find({borough:'Bronx'}).limit(5).skip(5)**

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

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

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

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

Assignmnet_2

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

Ans: db.restaurant.find({"address.coord.0":{$lt : -95.754168}}).pretty();

2. 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.

Ans: db.restaurant.find({

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

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

{"address.coord.0": {$lt : -65.754168}},

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

]

});

3. 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.

Ans:

```
db.restaurant.aggregate([{
        $match:{
                'cuisine':{$ne : 'American '},
                'address.coord.0':{$lt :-65.754168},
                'grades.score' : {$gt : 70}

    }

  }]);
```

4. 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
}
).pretty();
```

5. 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.

Ans:

```
db.restaurant.find({name: /ces$/},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1});
```

6. 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.

Ans:

```
db.restaurant.find(
{name: /.*Reg.*/},
{
"restaurant_id" : 1,
"name" : 1,
"bourough" : 1,
"cuisine" : 1
}
).pretty();
```

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

Ans:

```
db.restaurant.find({
                    cuisine : {$in : ["Chinese","American "]},
                    borough : "Bronx"

                  }).pretty();
```

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

Ans:

```
db.restaurant.aggregate([
                          {$sort : {name : 1}}
                         ]).pretty();
```

9. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

Ans:

```
db.restaurant.aggregate([
                        {$sort : {name : -1}}
                        ]).pretty();
```

10. 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.

Ans:

```
db.restaurant.aggregate([
                        {$sort : {name : 1, borough:-1}}
                ]).pretty();
```

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

Ans:

```
db.restaurant.find(
                        {'address.street': {$exists:true}}
                        ).pretty();
```

---

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.restaurant.aggregate([
{$match : {
        cuisine : {$ne : "American "},
      'grades.grade' :  'A',
        borough : {$ne : 'Brooklyn'}
        }
},
{$sort : {cuisine : -1}}
                    ]).pretty()
```

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

```
db.restaurant.aggregate([
            {$match : {
                borough:'Bronx',
                cuisine : {$in : ['American ','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.restaurant.find(
                { $or :
                  [{borough : 'Staten Island'},
                  {borough : 'Queens'},
                {borugh  : 'Bronxor Brooklyn'}]
                  }
            ).pretty()
```

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.restaurant.find(
            {borough : {$in : ['Staten Island','Queens','Bronxor Brooklyn']}},
            {restaurant_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.

```
db.restaurant.find(
                {'grades.score' : {$lte : 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 'Chinees' or restaurant's name begins with letter 'Wil'.

```
db.restaurant.find(
    {cuisine : {$nin : ['American ','Chinese']},
    name : /^Wil/},
    {restaurant_id : 1, name : 1, borough : 1, cuisine : 1}
        ).pretty()
```

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..

```
db.restaurant.find(
   {'grades.grade': 'A',
     'grades.score' : 11,
     'grades.date' : ISODate('2014-08-11T00:00:00Z')
    }
      ).pretty()
```

23. 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.1.grade' : 'A',
     'grades.1.score' : 9,
     'grades.1.date' : ISODate("2014-08-11T00:00:00Z")},
    {restaurant_id : 1,name:1 , grades : 1}
      ).pretty()
```

24. 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(
   {$and :[{'address.coord.1' : {$gt : 42}},
           {'address.coord.1' : {$lt : 52}}
            ]
   },
    { restaurant_id : 1 , name : 1 , address : 1 , 'address.coord' : 1}
      ).pretty()
```

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

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

db.restaurant.aggregate([

{$sort : {name : 1}}

])

26. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

db.restaurant.find().sort({name : -1});

27. 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})

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

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

29. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

30. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

```
db.restaurant.find(
        {"grades.score" :  {$mod : [7,0]}
        },
          {"restaurant_id" : 1,"name":1,"grades":1}
      ).pretty();
```

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.restaurant.find(
        {name : /.*mon.*/},
        {name:1, borough:1, "address.coord" : 1 , cuisine : 1}
              ).pretty()
```

32. 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 : /^Mad/},
        {name:1, borough:1, "address.coord" : 1 , cuisine : 1}
              ).pretty()
```