Tutorial-02

(MongoDB)

Import restaurant.json file using below command.
mongoimport --db databasename --collection res --file D:\restaurants.json
Note: Don't write above command in mongo shell. Directly execute it from the command prompt.

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

```
db.restaurants.find()
```

2. 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({},{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.restaurants.find({},{restaurant_id: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.restaurants.find({},{restaurant_id:1,borough:1,cuisine:1,_id:0,"address.zipcode":1)
```

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 5 which are in the borough Bronx

```
db.restaurants.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.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:100}}).pretty()
```

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})
```

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/},{"restaurantId":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$/},{"restaurant 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.*/},{"restaurant 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"}]})
```

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"]}},
{
"restaurant_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"]}},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
);
```

20. 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.restaurants.find(
{"grades.score" :
{ $not:
{$gt : 10}
}
},
{
"restaurant_id" : 1,
"name":1,"borough":1,
"cuisine" :1
}
);
```

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

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

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

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

  );
```

23. Write a MongoDB query to arrange the name of the cuisine in ascending order and for that same cuisine borough should be in descending order

```
db.restaurants.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 out how many times each cuisine is offered at various restaurants.

```
db.restaurants.find(

          {

           "address.coord.1": {$gt : 42, $lte : 52}

          },

            {"restaurant_id" : 1,"name":1,"address":1,"coord":1}

         );
```

25. Find out how many times each cuisine is offered at various restaurants in descending order.

```
db.restaurants.find().sort({"name":1});
```

26. Which cuisine is highly offered among all restaurants?.

```
db.restaurants.find().sort(

            {"name":-1}

            );
```

27. Find out the top 5 highly offered cuisines among all restaurants?

```
db.restaurants.find().sort(
                   {"cuisine":1,"borough" : -1,}
                  );
```