BDA

Practical 8

19BCE248

AIM: To perform MongoDB queries for Restaurant Dataset

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

```
db.restaurant.find().pretty()

        "_id" : ObjectId("6374aafd684c6082d53e96a1"),
        "address" : {
                "building" : "1007",
                "coord" : [
                        -73.856077,
                        40.848447
                ],
                "street" : "Morris Park Ave",
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "grades" : [
                {
                        "date" : ISODate("2014-03-03T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
                        "date" : ISODate("2013-09-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 6
                },
                {
```

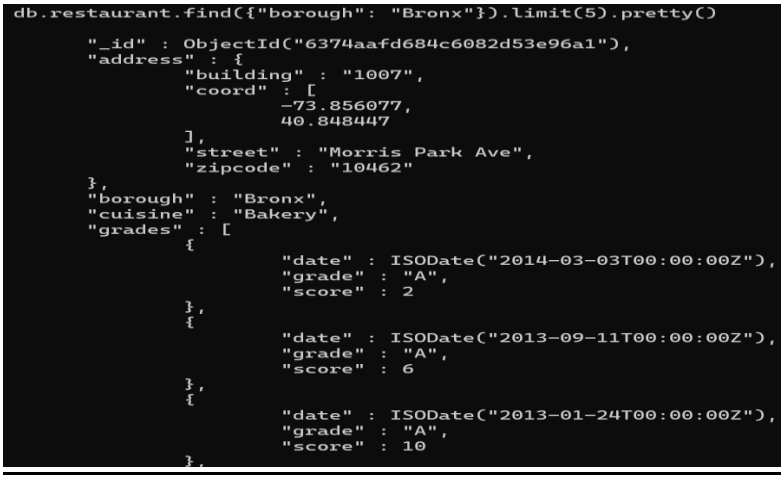
2. **Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.**

   db.restaurants.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0});

```
> db.restaurant.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1,"_id":0});
{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Bőulevard", "restaurant_id" : "40356151" }
{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "4
0356731" }
{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
```

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

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



5. **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" }] } );
```

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

```
db.restaurants.find(
        {"address.street" :
          { $exists : true }
        }
      );
```

7. **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.**

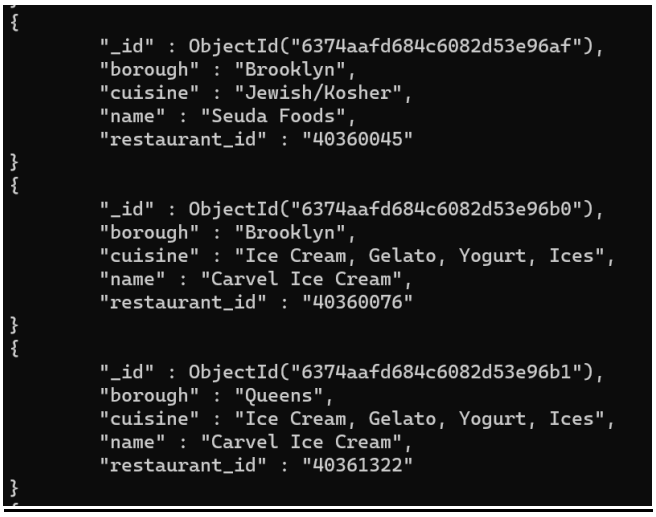
db.restaurants.find(

```
{
  "address.coord.1": {$gt : 42, $lte : 52}
},
 {"restaurant_id" : 1,"name":1,"address":1,"coord":1});
```

```
{
        "_id" : ObjectId("6374aafe684c6082d53ea4e4"),
        "address" : {
                "building" : "0",
                "coord" : [
                        -88.0778799,
                        42.4154769
                ],
                "street" : "& Grand Central",
                "zipcode" : "10017"
        },
        "name" : "Hyatt, Ny Central/Room Service",
        "restaurant_id" : "40879243"
}
```

8. **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});



```
{
        "_id" : ObjectId("6374aafd684c6082d53e96af"),
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Seuda Foods",
        "restaurant_id" : "40360045"
}
{
        "_id" : ObjectId("6374aafd684c6082d53e96b0"),
        "borough" : "Brooklyn",
        "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
        "name" : "Carvel Ice Cream",
        "restaurant_id" : "40360076"
}
{
        "_id" : ObjectId("6374aafd684c6082d53e96b1"),
        "borough" : "Queens",
        "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
        "name" : "Carvel Ice Cream",
        "restaurant_id" : "40361322"
}
```

9. **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);

10. **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.**