Ram Prakash **MongoDB – Complex Queries**

Note: As shown in below attached Screenshot. In Assignment sheet it was mentioned that we had to create database named as restaurants and collection named as addresses. But in Some question collection name was mentioned as restaurants that I think I was a mistake so I had solved all queries with collection named as addresses

4. Run the following command to import the json file provided. It will load the json file into the mongodb with database name - restaurants, collections name - addresses

ongoimport --db restaurants --collection addresses --file restaurants.json

5. Run mongo shell command
6. show databases
7. use restaurants
8. db.addresses.find() should print entire json data

Exercise Questions

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

```
db.addresses.find()
```


```
MongoDB Enterprise atlas-9itsca-shard-0:PRIMARY> db.addresses.find()
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea56"), "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 }, { "date
" : ISODate("2013-01-24T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2011-03-10T00:00:00Z"), "grade
" : "B", "score" : 14 } ], "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea57"), "address" : { "building" : "469", "coord" : [ -73.961704, 40.662942 ], "street" : "Flatbush Avenue", "zipcode" : "11225" }, "borough" : "Brookly
n", "cuisine" : "Hamburgers", "grades" : [ { "date" : ISODate("2014-12-30T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2014-07-01T00:00:00Z"), "grade" : "B", "score" : 23 },
{ "date" : ISODate("2013-04-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-05-08T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Wendy'S", "restaurant_id" : "3
0112340" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea58"), "address" : { "building" : "351", "coord" : [ -73.98513559999999, 40.7676919 ], "street" : "West   57 Street", "zipcode" : "10019" }, "borough"
: "Manhattan", "cuisine" : "Irish", "grades" : [ { "date" : ISODate("2014-09-06T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" :
11 }, { "date" : ISODate("2012-07-31T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-12-29T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Dj Reynolds Pub And Rest
aurant", "restaurant_id" : "30191841" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea59"), "address" : { "building" : "2780", "coord" : [ -73.98241999999999, 40.579505 ], "street" : "Stillwell Avenue", "zipcode" : "11224" }, "borough"
: "Brooklyn", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-06-10T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-06-05T00:00:00Z"), "grade" : "A", "score"
: 7 }, { "date" : ISODate("2012-04-13T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-10-12T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Riviera Caterer", "res
taurant_id" : "40356018" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea5a"), "address" : { "building" : "97-22", "coord" : [ -73.8601152, 40.7311739 ], "street" : "63 Road", "zipcode" : "11374" }, "borough" : "Queens", "c
uisine" : "Jewish/Kosher", "grades" : [ { "date" : ISODate("2014-11-24T00:00:00Z"), "grade" : "Z", "score" : 20 }, { "date" : ISODate("2013-01-17T00:00:00Z"), "grade" : "A", "score" : 13 }, {
"date" : ISODate("2012-08-02T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-12-15T00:00:00Z"), "grade" : "B", "score" : 25 } ], "name" : "Tov Kosher Kitchen", "restaurant
_id" : "40356068" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea5b"), "address" : { "building" : "8825", "coord" : [ -73.8803827, 40.7643124 ], "street" : "Astoria Boulevard", "zipcode" : "11369" }, "borough" : "Qu
eens", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-11-15T00:00:00Z"), "grade" : "Z", "score" : 38 }, { "date" : ISODate("2014-05-02T00:00:00Z"), "grade" : "A", "score" : 10
}, { "date" : ISODate("2013-03-02T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2012-02-10T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "Brunos On The Boulevard", "r
estaurant_id" : "40356151" }
{ "_id" : ObjectId("61efc6e5aeefed5c20a0ea5c"), "address" : { "building" : "2206", "coord" : [ -74.1377286, 40.6119572 ], "street" : "Victory Boulevard", "zipcode" : "10314" }, "borough" : "St
aten Island", "cuisine" : "Jewish/Kosher", "grades" : [ { "date" : ISODate("2014-10-06T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-05-20T00:00:00Z"), "grade" : "A", "sc
ore" : 12 }, { "date" : ISODate("2013-04-04T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-01-24T00:00:00Z"), "grade" : "A", "score" : 9 } ], "name" : "Kosher Island", "r
estaurant_id" : "40356442" }
```

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

Note: We can solve these query with simple find() function also and I solved some query with both method that you can see in some later queries

```
db.addresses.aggregate({$project:{restaurant_id:1,name:1,borough:1,cuisine:1}}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea56"),
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea57"),
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea58"),
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea59"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Riviera Caterer",
        "restaurant_id" : "40356018"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea5a"),
        "borough" : "Queens",
        "cuisine" : "Jewish/Kosher",
        "name" : "Tov Kosher Kitchen",
        "restaurant_id" : "40356068"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea5b"),
        "borough" : "Queens",
        "cuisine" : "American ",
        "name" : "Brunos On The Boulevard",
        "restaurant_id" : "40356151"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea5c"),
        "borough" : "Staten Island",
        "cuisine" : "Jewish/Kosher",
        "name" : "Kosher Island",
        "restaurant_id" : "40356442"
}
```

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.addresses.aggregate({$project:{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1}}).pretty()
```


```
{
        "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 Boulevard",
        "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"
```

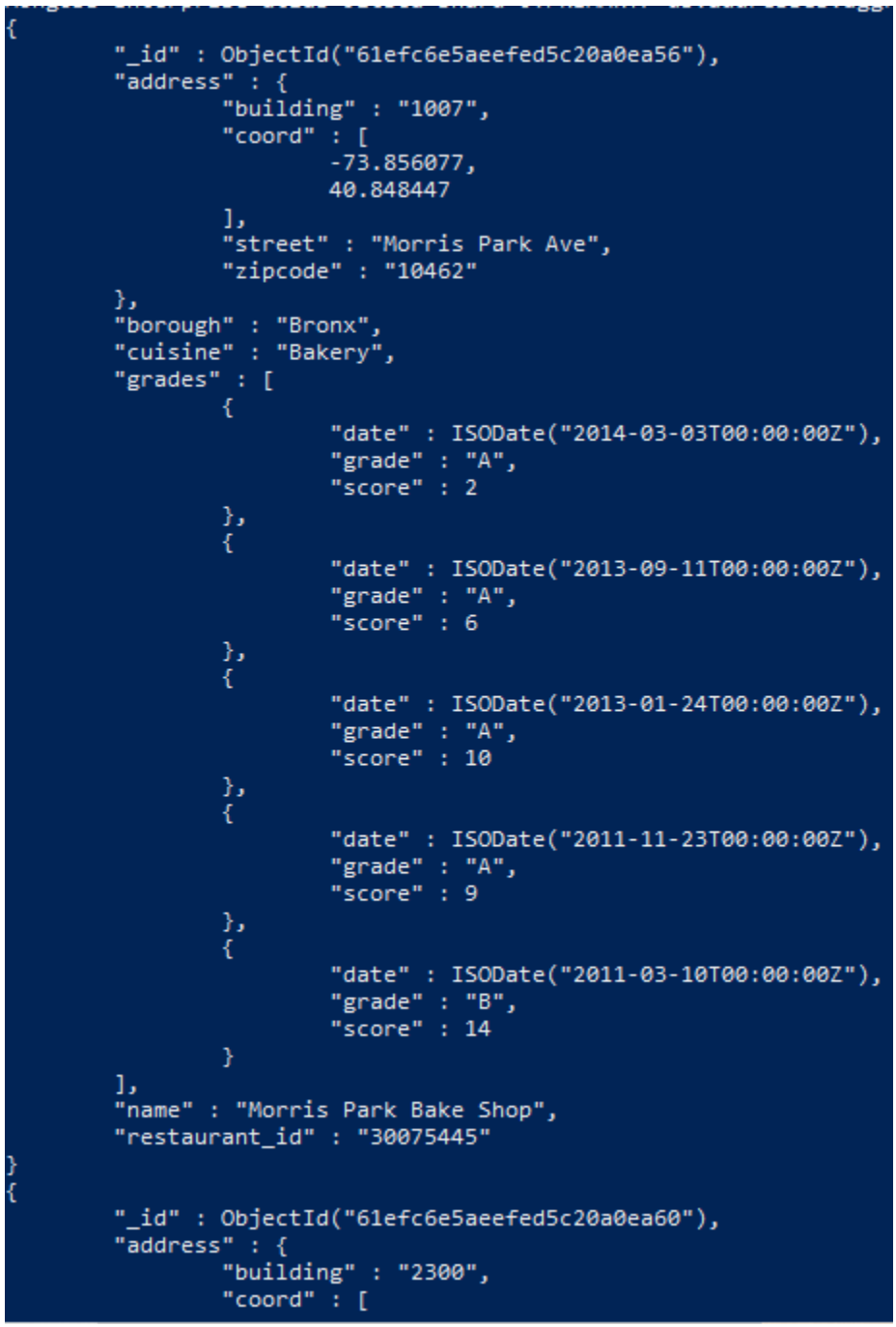
4. Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant.

```
db.addresses.aggregate({$project:{_id:0,restaurant_id:1,name:1,borough:1,address:{zipcode:1}}}).pretty()
```


```
{
        "address" : {
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "address" : {
                "zipcode" : "11225"
        },
        "borough" : "Brooklyn",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "address" : {
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
}
{
        "address" : {
                "zipcode" : "11224"
        },
        "borough" : "Brooklyn",
        "name" : "Riviera Caterer",
        "restaurant_id" : "40356018"
}
{
        "address" : {
                "zipcode" : "11374"
        },
        "borough" : "Queens",
        "name" : "Tov Kosher Kitchen",
        "restaurant_id" : "40356068"
}
{
        "address" : {
                "zipcode" : "11369"
```

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

```
db.addresses.aggregate([{$match:{borough:"Bronx"}},{$limit:5}]).pretty()
```




```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea56"),
        "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
                },
                {
                        "date" : ISODate("2013-01-24T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2011-11-23T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2011-03-10T00:00:00Z"),
                        "grade" : "B",
                        "score" : 14
                }
        ],
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea60"),
        "address" : {
                "building" : "2300",
                "coord" : [
```

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

```
db.addresses.aggregate({$match:{borough:"Bronx"}}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea56"),
        "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
                },
                {
                        "date" : ISODate("2013-01-24T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2011-11-23T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2011-03-10T00:00:00Z"),
                        "grade" : "B",
                        "score" : 14
                }
        ],
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
```

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

```
db.addresses.aggregate([{$match:{borough:"Bronx"}},{$skip:5},{$limit:5}]).pretty()
```


```
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea93"),
        "address" : {
                "building" : "658",
                "coord" : [
                        -73.81363999999999,
                        40.82941100000001
                ],
                "street" : "Clarence Ave",
                "zipcode" : "10465"
        },
        "borough" : "Bronx",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-06-21T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2012-07-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                }
        ],
        "name" : "Manhem Club",
        "restaurant_id" : "40364363"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eaab"),
        "address" : {
                "building" : "2222",
                "coord" : [
                        -73.84971759999999,
                        40.8304811
                ],
                "street" : "Haviland Avenue",
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-12-18T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2014-05-01T00:00:00Z"),
                        "grade" : "B",
                        "score" : 17
```

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

```
db.addresses.find({"grades.score":{$gt:90}}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ebb4"),
        "address" : {
                "building" : "65",
                "coord" : [
                        -73.9782725,
                        40.7624022
                ],
                "street" : "West   54 Street",
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-08-22T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2014-03-28T00:00:00Z"),
                        "grade" : "C",
                        "score" : 131
                },
                {
                        "date" : ISODate("2013-09-25T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2013-04-08T00:00:00Z"),
                        "grade" : "B",
                        "score" : 25
                },
                {
                        "date" : ISODate("2012-10-15T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2011-10-19T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                }
        ],
        "name" : "Murals On 54/Randolphs'S",
        "restaurant_id" : "40372466"
}
```

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

```
db.addresses.find({$and:[{"grades.score":{"$gt":80}},{"grades.score":{"$lt":100}}]}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0edb8"),
        "address" : {
                "building" : "130",
                "coord" : [
                        -73.984758,
                        40.7457939
                ],
                "street" : "Madison Avenue",
                "zipcode" : "10016"
        },
        "borough" : "Manhattan",
        "cuisine" : "Pizza/Italian",
        "grades" : [
                {
                        "date" : ISODate("2014-12-24T00:00:00Z"),
                        "grade" : "Z",
                        "score" : 31
                },
                {
                        "date" : ISODate("2014-06-17T00:00:00Z"),
                        "grade" : "C",
                        "score" : 98
                },
                {
                        "date" : ISODate("2013-12-12T00:00:00Z"),
                        "grade" : "C",
                        "score" : 32
                },
                {
                        "date" : ISODate("2013-05-22T00:00:00Z"),
                        "grade" : "B",
                        "score" : 21
                },
                {
                        "date" : ISODate("2012-05-02T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                }
        ],
        "name" : "Bella Napoli",
        "restaurant_id" : "40393488"
}
```

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

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

```
"_id" : ObjectId("61efc6eeaeefed5c20a0f09e"),
"address" : {
        "building" : "3707",
        "coord" : [
                -101.8945214,
                33.5197474
        ],
        "street" : "82 Street",
        "zipcode" : "11372"
},
```

```
"_id" : ObjectId("61efc6f9aeefed5c20a0f409"),
"address" : {
        "building" : "15259",
        "coord" : [
                -119.6368672,
                36.2504996
        ],
        "street" : "10 Avenue",
        "zipcode" : "11357"
},
```

```
"_id" : ObjectId("61efc70faeefed5c20a0f8af"),
"address" : {
        "building" : "60",
        "coord" : [
                -111.9975205,
                42.0970258
        ],
        "street" : "West Side Highway",
        "zipcode" : "10006"
```

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.addresses.find({"cuisine":{$ne:"American "},"grades.score":{$gt: 70},"address.coord":{$lt:-65.754168}}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0edb8"),
        "address" : {
                "building" : "130",
                "coord" : [
                        -73.984758,
                        40.7457939
                ],
                "street" : "Madison Avenue",
                "zipcode" : "10016"
        },
        "borough" : "Manhattan",
        "cuisine" : "Pizza/Italian",
        "grades" : [
                {
                        "date" : ISODate("2014-12-24T00:00:00Z"),
                        "grade" : "Z",
                        "score" : 31
                },
                {
                        "date" : ISODate("2014-06-17T00:00:00Z"),
                        "grade" : "C",
                        "score" : 98
                },
                {
                        "date" : ISODate("2013-12-12T00:00:00Z"),
                        "grade" : "C",
                        "score" : 32
                },
                {
                        "date" : ISODate("2013-05-22T00:00:00Z"),
                        "grade" : "B",
                        "score" : 21
                },
                {
                        "date" : ISODate("2012-05-02T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                }
        ],
        "name" : "Bella Napoli",
        "restaurant_id" : "40393488"
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0edc3"),
        "address" : {
                "building" : "101",
                "coord" : [
```


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: I think both the questions 11 and 12 are same.

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.addresses.find({"borough":{$ne:"Brooklyn"},"cuisine":{$ne:"American "},"grades.grade":{$eq:"A"}}).sort({cuisine:-1}).pretty()
```


```
 "borough" : "Queens",
 "cuisine" : "Vietnamese/Cambodian/Malaysia",
 "grades" : [
        {
                "date" : ISODate("2014-06-12T00:00:00Z"),
                "grade" : "B",
                "score" : 21
        },
        {
                "date" : ISODate("2013-05-20T00:00:00Z"),
                "grade" : "A",
                "score" : 13
        },
        {
                "date" : ISODate("2012-12-26T00:00:00Z"),
                "grade" : "A",
                "score" : 10
        },
        {
                "date" : ISODate("2012-12-03T00:00:00Z"),
                "grade" : "P",
                "score" : 5
        },
        {
                "date" : ISODate("2012-05-04T00:00:00Z"),
                "grade" : "B",
                "score" : 27
        }
```

```
 "borough" : "Manhattan",
 "cuisine" : "Vegetarian",
 "grades" : [
        {
                "date" : ISODate("2014-02-05T00:00:00Z"),
                "grade" : "A",
                "score" : 13
        },
        {
                "date" : ISODate("2013-04-24T00:00:00Z"),
                "grade" : "B",
                "score" : 19
        },
        {
                "date" : ISODate("2012-01-17T00:00:00Z"),
                "grade" : "A",
                "score" : 12
        },
        {
                "date" : ISODate("2011-07-25T00:00:00Z"),
                "grade" : "A",
                "score" : 10
        }
 ],
 "name" : "Angelica Kitchen",
 "restaurant_id" : "40388281"
```

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.addresses.find({name:{$regex:"^Wil"}},{_id:0,restaurant_id:1,name:1, borough:1, cuisine:1}).pretty()
```


```
{
        "borough" : "Bronx",
        "cuisine" : "American ",
        "name" : "Wild Asia",
        "restaurant_id" : "40357217"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Wilken'S Fine Food",
        "restaurant_id" : "40356483"
}
{
        "borough" : "Bronx",
        "cuisine" : "Pizza",
        "name" : "Wilbel Pizza",
        "restaurant_id" : "40871979"
}
MongoDB Enterprise atlas-9itsca-shard-0:PRIMARY>
```

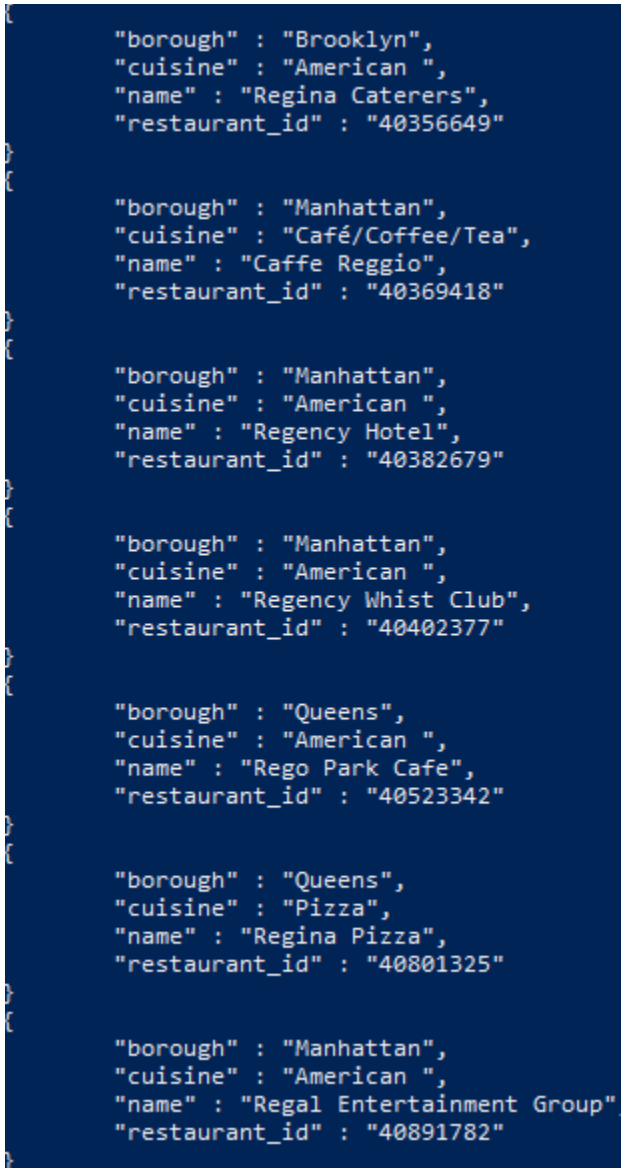

15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

```
db.addresses.find({name:{$regex:"ces$"}},{_id:0,restaurant_id:1,name:1, borough:1, cuisine:1}).pretty()
```


```
{
        "borough" : "Queens",
        "cuisine" : "American ",
        "name" : "S.M.R Restaurant Services",
        "restaurant_id" : "40403857"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Good Shepherd Services",
        "restaurant_id" : "40403989"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Pieces",
        "restaurant_id" : "40399910"
}
{
        "borough" : "Queens",
        "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
        "name" : "The Ice Box-Ralph'S Famous Italian Ices",
        "restaurant_id" : "40690899"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Alices",
        "restaurant_id" : "40782042"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Re: Sources",
        "restaurant_id" : "40876068"
}
```

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.addresses.find({name:{$regex:"Reg"}},{_id:0,restaurant_id:1,name:1, borough:1, 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.addresses.find({borough:"Bronx",$or:[{cuisine:"American "},{cuisine:"Chisene"}]},{borough:1,cuisine:1}).pretty()
```


```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea60"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb63"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb30"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb51"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb75"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ec43"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eac3"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb2f"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eb7f"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0eaef"),
        "borough" : "Bronx",
        "cuisine" : "American "
}
```

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.addresses.find({$or:[{borough:"Bronx"},{borough:"Queens"},{borough:"Brooklyn"},{borough:"Staten Island"}]},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1}).pret
```

```
        "restaurant_id" : "40366162"
}
{
        "borough" : "Queens",
        "cuisine" : "Jewish/Kosher",
        "name" : "Naomi Kosher Pizza",
        "restaurant_id" : "40366425"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Soul Food",
        "name" : "Mitchell'S Restaurant",
        "restaurant_id" : "40366961"
}
{
        "borough" : "Queens",
        "cuisine" : "Pizza/Italian",
        "name" : "Valentino'S Pizza",
        "restaurant_id" : "40369012"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "White Castle",
        "restaurant_id" : "40369667"
}
{
        "borough" : "Queens",
        "cuisine" : "Hamburgers",
        "name" : "Mcdonald'S",
        "restaurant_id" : "40369724"
}
Type "it" for more
MongoDB Enterprise > it
{
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "Burger King",
        "restaurant_id" : "40370917"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Kosher Delight",
        "restaurant_id" : "40371419"
}
{
        "borough" : "Queens",
        "cuisine" : "Italian",
        "name" : "Clinton Restaurant",
```

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.addresses.find({borough: {$nin: ["Staten Island","Queens","Bronx","Brooklyn"]}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()
```


```
{
        "borough" : "Manhattan",
        "cuisine" : "Continental",
        "name" : "Lorenzo & Maria'S",
        "restaurant_id" : "40363630"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Berkely",
        "restaurant_id" : "40363685"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Nyac Main Dining Room",
        "restaurant_id" : "40364467"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "The Princeton Club",
        "restaurant_id" : "40365361"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Mexican",
        "name" : "Mexico Lindo Restaurant",
        "restaurant_id" : "40367038"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Capitol Restaurant",
        "restaurant_id" : "40367677"
}
{
        "borough" : "Manhattan",
        "cuisine" : "French",
        "name" : "La Bonne Soupe Bistro",
        "restaurant_id" : "40367715"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Seafood",
        "name" : "Oyster Bar",
        "restaurant_id" : "40368223"
```

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.addresses.find({"grades.score": {$lte: 10}},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()
```

```
"Brooklyn",
```


```
{
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Sonny'S Heros",
        "restaurant_id" : "40363744"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Regina Caterers",
        "restaurant_id" : "40356649"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Seuda Foods",
        "restaurant_id" : "40360045"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "The Movable Feast",
        "restaurant_id" : "40361606"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Continental",
        "name" : "Lorenzo & Maria'S",
        "restaurant_id" : "40363630"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Berkely",
        "restaurant_id" : "40363685"
}
{
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Nyac Main Dining Room",
        "restaurant_id" : "40364467"
}
{
```

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.addresses.find({$or:[{cuisine:{$nin:["American ","Chisene"]}},{name:{$regex:"^Wil"}}]},{_id:0,restaurant_id:1,name:1,borough:1,cuisine:1}).pretty()
```


```
{
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Seuda Foods",
        "restaurant_id" : "40360045"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Continental",
        "name" : "Lorenzo & Maria'S",
        "restaurant_id" : "40363630"
}
{
        "borough" : "Queens",
        "cuisine" : "Pizza",
        "name" : "Rizzo'S Fine Pizza",
        "restaurant_id" : "40364920"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Mexican",
        "name" : "Mexico Lindo Restaurant",
        "restaurant_id" : "40367038"
}
{
        "borough" : "Manhattan",
        "cuisine" : "French",
        "name" : "La Bonne Soupe Bistro",
        "restaurant_id" : "40367715"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Seafood",
        "name" : "Oyster Bar",
        "restaurant_id" : "40368223"
}
{
        "borough" : "Staten Island",
        "cuisine" : "Hamburgers",
        "name" : "Mcdonald'S",
        "restaurant_id" : "40370356"
}
{
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Dairy Luncheonette",
        "restaurant_id" : "40371684"
}
{
```

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.addresses.find({"grades":{$elemMatch:{"date":ISODate("2014-08-11T00:00:00Z"),"grade":"A","score":11}}},{_id:0, restaurant_id:1, name:1, grades:1}).pretty()
```

```
MongoDB Enterprise > db.addresses.find({"grades" : {$elemMatch: {"d
{
        "grades" : [
                {
                        "date" : ISODate("2014-08-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2013-12-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2013-06-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2012-06-08T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2012-01-25T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2011-09-13T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                }
        ],
        "name" : "Don Filippo Restaurant",
        "restaurant_id" : "40372417"
```

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.addresses.find({$and:[{"grades.1.grade":"A"},{"grades.1.score": 9},{"grades.1.date":ISODate("2014-08-11T00:00:00Z")}]},{_id:0,restaurant_id:1,name:1,grades:1}).pretty()
```


```
{
        "grades" : [
                {
                        "date" : ISODate("2015-01-12T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2014-08-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2014-01-14T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2013-02-07T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2012-04-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                }
        ],
        "name" : "Club Macanudo (Cigar Bar)",
        "restaurant_id" : "40526406"
}
```

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.addresses.find({$and:[{"address.coord.1":{$gt:42}},{"address.coord.1":{$lte:52}}]},{_id:0,restaurant_id:1,name:1,address:1}).pretty()
```


```
{
        "address" : {
                "building" : "1",
                "coord" : [
                        -0.7119979,
                        51.6514664
                ],
                "street" : "Pennplaza E, Penn Sta",
                "zipcode" : "10001"
        },
        "name" : "T.G.I. Fridays",
        "restaurant_id" : "40388936"
}
{
        "address" : {
                "building" : "47",
                "coord" : [
                        -78.877224,
                        42.89546199999999
                ],
                "street" : "Broadway @ Trinity Pl",
                "zipcode" : "10006"
        },
        "name" : "T.G.I. Friday'S",
        "restaurant_id" : "40387990"
}
{
        "address" : {
                "building" : "3000",
                "coord" : [
                        -87.86567699999999,
                        42.61150920000001
                ],
                "street" : "47 Avenue",
                "zipcode" : "11101"
        },
        "name" : "Di Luvio'S Deli",
        "restaurant_id" : "40402284"
}
{
        "address" : {
                "building" : "21972199",
                "coord" : [
                        -78.589606,
                        42.8912372
                ],
                "street" : "Broadway",
                "zipcode" : "10024"
        },
```

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

Find simple find() Funtion

```
MongoDB Enterprise > db.addresses.find({},{_id:0, name:1}).sort({name: 1}).pretty()
{ "name" : "(Lewis Drug Store) Locanda Vini E Olii" }
{ "name" : "1 East 66Th Street Kitchen" }
{ "name" : "101 Deli" }
{ "name" : "101 Restaurant And Bar" }
{ "name" : "1020 Bar" }
{ "name" : "104-01 Foster Avenue Coffee Shop(Ups)" }
{ "name" : "10Th Avenue Pizza & Cafe" }
{ "name" : "111 Restaurant" }
{ "name" : "15 East Restaurant" }
{ "name" : "200 Fifth Avenue Restaurant & Sports Bar" }
{ "name" : "21 Club" }
{ "name" : "2A" }
{ "name" : "3 Deli & Grill" }
{ "name" : "3 Guys" }
{ "name" : "3 Guys Resturant" }
{ "name" : "42Nd Street Pizza Diner" }
{ "name" : "44 & X Hell'S Kitchen" }
{ "name" : "44 Sw Ristorante & Bar" }
{ "name" : "5 Burro Cafe" }
{ "name" : "525 Lex Restaurant & Bar" }
Type "it" for more
```

With aggregate function projection

```
MongoDB Enterprise > db.addresses.aggregate([{$sort:{name:1}},{$project:{_id:0,name:1}}]).pre
{ "name" : "(Lewis Drug Store) Locanda Vini E Olii" }
{ "name" : "1 East 66Th Street Kitchen" }
{ "name" : "101 Deli" }
{ "name" : "101 Restaurant And Bar" }
{ "name" : "1020 Bar" }
{ "name" : "104-01 Foster Avenue Coffee Shop(Ups)" }
{ "name" : "10Th Avenue Pizza & Cafe" }
{ "name" : "111 Restaurant" }
{ "name" : "15 East Restaurant" }
{ "name" : "200 Fifth Avenue Restaurant & Sports Bar" }
{ "name" : "21 Club" }
{ "name" : "2A" }
{ "name" : "3 Deli & Grill" }
{ "name" : "3 Guys" }
{ "name" : "3 Guys Resturant" }
{ "name" : "42Nd Street Pizza Diner" }
{ "name" : "44 & X Hell'S Kitchen" }
{ "name" : "44 Sw Ristorante & Bar" }
{ "name" : "5 Burro Cafe" }
{ "name" : "525 Lex Restaurant & Bar" }
Type "it" for more
```

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

With Simple find () function

```
MongoDB Enterprise > db.addresses.find({},{_id:0, name:1}).sort({name: -1}).p
{ "name" : "Zum Stammtisch" }
{ "name" : "Zum Schneider" }
{ "name" : "Zorba'S" }
{ "name" : "Zebu Grill" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zafi'S Luncheonette" }
{ "name" : "Yvonne Yvonne Restaurant" }
{ "name" : "Yura & Company On Madison" }
{ "name" : "Yummy Kitchen" }
{ "name" : "Your Bakery" }
{ "name" : "Yonah Shimmels Knishes" }
{ "name" : "Yolanda Pizzeria Restaurant" }
{ "name" : "Yip'S" }
{ "name" : "Yen Yen Restaurant" }
Type "it" for more
```

With aggregate function project

```
MongoDB Enterprise > db.addresses.aggregate([{$sort:{name:-1}},{$project:{_id:0,name:1}}]).pr
{ "name" : "Zum Stammtisch" }
{ "name" : "Zum Schneider" }
{ "name" : "Zorba'S" }
{ "name" : "Zebu Grill" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bread Basket" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zaro'S Bakery" }
{ "name" : "Zafi'S Luncheonette" }
{ "name" : "Yvonne Yvonne Restaurant" }
{ "name" : "Yura & Company On Madison" }
{ "name" : "Yummy Kitchen" }
{ "name" : "Your Bakery" }
{ "name" : "Yonah Shimmels Knishes" }
{ "name" : "Yolanda Pizzeria Restaurant" }
{ "name" : "Yip'S" }
{ "name" : "Yen Yen Restaurant" }
Type "it" for more
MongoDB Enterprise >
```

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.

With simple find() function

```
MongoDB Enterprise > db.addresses.find({},{_id:0,cuisine:1,borough:1}).sort({cuisine:1,borough: -1}).p
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Queens", "cuisine" : "African" }
{ "borough" : "Brooklyn", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
Type "it" for more
```

With aggregate project function

```
MongoDB Enterprise > db.addresses.aggregate([{$sort:{cuisine:1,borough:-1}},{$project:{_id:0,cuisine:1,borough:1}}]).pr
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Manhattan", "cuisine" : "Afghan" }
{ "borough" : "Queens", "cuisine" : "African" }
{ "borough" : "Brooklyn", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Bronx", "cuisine" : "African" }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
{ "borough" : "Staten Island", "cuisine" : "American " }
Type "it" for more
```

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

```
db.addresses.find({"address.street":{$ne:{$regex:"Street"}}},{address:1}).pretty()
```

```
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea83"),
        "address" : {
                "building" : "1031",
                "coord" : [
                        -73.9075537,
                        40.6438684
                ],
                "street" : "East   92 Street",
                "zipcode" : "11236"
        }
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea5e"),
        "address" : {
                "building" : "6409",
                "coord" : [
                        -74.00528899999999,
                        40.628886
                ],
                "street" : "11 Avenue",
                "zipcode" : "11219"
        }
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea64"),
        "address" : {
                "building" : "705",
                "coord" : [
                        -73.9653967,
                        40.6064339
                ],
                "street" : "Kings Highway",
                "zipcode" : "11223"
        }
}
{
        "_id" : ObjectId("61efc6e5aeefed5c20a0ea69"),
        "address" : {
```

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

```
db.addresses.find({"address.coord": {$type:"double"}}, {_id:0,address:1}).pretty()
```


```
{
        "address" : {
                "building" : "1031",
                "coord" : [
                        -73.9075537,
                        40.6438684
                ],
                "street" : "East   92 Street",
                "zipcode" : "11236"
        }
}
{
        "address" : {
                "building" : "6409",
                "coord" : [
                        -74.00528899999999,
                        40.628886
                ],
                "street" : "11 Avenue",
                "zipcode" : "11219"
        }
}
{
        "address" : {
                "building" : "705",
                "coord" : [
                        -73.9653967,
                        40.6064339
                ],
                "street" : "Kings Highway",
                "zipcode" : "11223"
        }
}
{
        "address" : {
                "building" : "284",
                "coord" : [
                        -73.9829239,
                        40.6580753
                ],
                "street" : "Prospect Park West",
                "zipcode" : "11215"
        }
}
{
        "address" : {
```

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.addresses.find({"grades":{$elemMatch:{"score":{$mod:[7,0]}}}},{_id:0,restaurant_id:1,name:1,grades:1}).pretty()
```


```
{
        "grades" : [
                {
                        "date" : ISODate("2014-02-05T00:00:00Z"),
                        "grade" : "A",
                        "score" : 0
                },
                {
                        "date" : ISODate("2013-01-29T00:00:00Z"),
                        "grade" : "A",
                        "score" : 3
                },
                {
                        "date" : ISODate("2011-12-08T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                }
        ],
        "name" : "Sonny'S Heros",
        "restaurant_id" : "40363744"
}
{
        "grades" : [
                {
                        "date" : ISODate("2014-11-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2013-10-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2012-10-04T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2012-05-21T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2011-12-30T00:00:00Z"),
                        "grade" : "B",
                        "score" : 19
                }
        ],
        "name" : "Seuda Foods",
```

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.addresses.find({name:{$regex:"mon"}},{_id:0,name:1,borough:1,"address.coord":1,cuisine:1}).pretty()
```


```
{
        "address" : {
                "coord" : [
                        -74.10465599999999,
                        40.58834
                ]
        },
        "borough" : "Staten Island",
        "cuisine" : "American ",
        "name" : "Richmond County Country Club"
}
{
        "address" : {
                "coord" : [
                        -73.9812843,
                        40.5947365
                ]
        },
        "borough" : "Brooklyn",
        "cuisine" : "Pizza/Italian",
        "name" : "Lb Spumoni Gardens"
}
{
        "address" : {
                "coord" : [
                        -73.9901605,
                        40.7526176
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Delmonico'S Kitchen"
}
{
        "address" : {
                "coord" : [
                        -73.951199,
                        40.7166026
                ]
        },
        "borough" : "Brooklyn",
        "cuisine" : "Italian",
        "name" : "Bamonte'S Restaurant"
}
{
```

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.addresses.find({name:{$regex:"^Mad"}},{_id:0,name:1,borough:1,"address.coord":1,cuisine:1}).pretty()
```


```
        "cuisine" : "American ",
        "name" : "Madison Square"
}
{
        "address" : {
                "coord" : [
                        -73.98302199999999,
                        40.742313
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "Indian",
        "name" : "Madras Mahal"
}
{
        "address" : {
                "coord" : [
                        -73.98171959999999,
                        40.7499406
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "French",
        "name" : "Madison Bistro"
}
{
        "address" : {
                "coord" : [
                        -74.000002,
                        40.72735
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Madame X"
}
{
        "address" : {
                "coord" : [
                        -73.9717845,
                        40.6897199
                ]
        },
        "borough" : "Brooklyn",
        "cuisine" : "African",
        "name" : "Madiba"
}
{
        "address" : {
                "coord" : [
```