

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 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" }
{ "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" : "40356731" }
{ "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" }
{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Souls Foods", "restaurant_id" : "40360015" }
```

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

```

db.restaurant.find({"borough": "Bronx"}).limit(5).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
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ],

```

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

```

db.restaurant.find({"borough": "Bronx"}).limit(5).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
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ],

```

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" } ] } );

```

```

    "_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
      },
      {
        "date" : ISODate("2013-01-24T00:00:00Z"),
        "grade" : "A",
        "score" : 10
      },
      {
        "date" : ISODate("2011-11-23T00:00:00Z"),

```

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

```

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

```

```

{
  "_id" : ObjectId("6374aafd684c6082d53e96b2"),
  "address" : {
    "building" : "6909",
    "coord" : [
      -74.0259567,
      40.6353674
    ],
    "street" : "3 Avenue",
    "zipcode" : "11209"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "grades" : [
    {
      "date" : ISODate("2014-08-21T00:00:00Z"),
      "grade" : "A",
      "score" : 4
    },
    {
      "date" : ISODate("2014-03-05T00:00:00Z"),
      "grade" : "A",
      "score" : 3
    },
    {
      "date" : ISODate("2013-01-10T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ],
  "name" : "Nordic Delicacies",

```

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("6374aafe684c6082d53e96af"),
  "borough" : "Brooklyn",
  "cuisine" : "Jewish/Kosher",
  "name" : "Seuda Foods",
  "restaurant_id" : "40360045"
},
{
  "_id" : ObjectId("6374aafe684c6082d53e96b0"),
  "borough" : "Brooklyn",
  "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
  "name" : "Carvel Ice Cream",
  "restaurant_id" : "40360076"
},
{
  "_id" : ObjectId("6374aafe684c6082d53e96b1"),
  "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);
```

```

    "id": ObjectId("6374aaf6d84c6082d53e96de"), "address": { "building": "658", "coord": [ -73.81363999999999, 40.829100000000002 ], "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-06-21T00:00:00Z"), "grade": "B", "score": 10 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 19 } ], "name": "Manhem Club", "restaurant_id": "40364363" },
    { "_id": ObjectId("6374aaf6d84c6082d53e96fe"), "address": { "building": "2222", "coord": [ -73.84971759999999, 40.829100000000002 ], "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-12-18T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2013-03-14T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "C", "score": 37 } ], "name": "Yanke Tavern", "restaurant_id": "40364364" },
    { "_id": ObjectId("6374aaf6d84c6082d53e970e"), "address": { "building": "72", "coord": [ -73.92506, 40.8275556 ], "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-04-15T00:00:00Z"), "grade": "A", "score": 15 }, { "date": ISODate("2012-05-30T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "A", "score": 4 }, { "date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "C", "score": 37 } ], "name": "Yanke Tavern", "restaurant_id": "40364365" },
    { "_id": ObjectId("6374aaf6d84c6082d53e9721"), "address": { "building": "331", "coord": [ -73.87786539999999, 40.872100000000002 ], "borough": "Bronx", "cuisine": "Irish", "grades": [ { "date": ISODate("2014-08-26T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2013-09-11T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "B", "score": 23 }, { "date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 19 }, { "date": ISODate("2011-10-20T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "C", "score": 37 } ], "name": "Mcdwyer's", "restaurant_id": "40364366" },
    { "_id": ObjectId("6374aaf6d84c6082d53e973a"), "address": { "building": "5820", "coord": [ -73.9002615, 40.885186 ], "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-02-26T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2013-05-15T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "B", "score": 19 }, { "date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2011-08-15T00:00:00Z"), "grade": "C", "score": 37 } ], "name": "Mcdwyer's", "restaurant_id": "40364367" }
  ]
}

```

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.

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

{
  "_id" : ObjectId("6374aafd684c6082d53e96a1"),
  "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("6374aafd684c6082d53e96a4"),
  "grades" : [
    {

```