

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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({'$and': [{'grades.score' : {'$gt' : 80}}, {'grades.score' : {'$lt' : 100}}]}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee6595e"),
  "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" : 6
    }
  ]
}
```

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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({'address.coord.0' : {'$lt' : -95.754168}}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65e46"),
  "address" : {
    "building" : "3707",
    "coord" : [
      -101.8945214,
      33.5197474
    ],
    "street" : "82 Street",
    "zipcode" : "11372"
  },
  "borough" : "Queens",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-06-04T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2013-11-07T00:00:00Z"),
      "grade" : "B",
      "score" : 19
    },
    {
      "date" : ISODate("2013-05-17T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-08-29T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-04-03T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    }
  ]
}
```

3. 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. Note: Do this without using \$and operator

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"cuisine" : {$ne : "American "}, "grades.score" : {$gt : 70}, "address.coord.0" : {$lt : -65.754168}}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65a03"),
  "address" : {
    "building" : "345",
    "coord" : [
      -73.9864626,
      40.7266739
    ],
    "street" : "East 6 Street",
    "zipcode" : "10003"
  },
  "borough" : "Manhattan",
  "cuisine" : "Indian",
  "grades" : [
    {
      "date" : ISODate("2014-09-15T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2014-01-14T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2013-05-30T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2013-04-24T00:00:00Z"),
      "grade" : "P",
      "score" : 2
    },
    {
      "date" : ISODate("2012-10-01T00:00:00Z"),
      "grade" : "A",
      "score" : 76
    }
  ]
}
```

4. Write a MongoDB query for the above question using \$and operator

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"and" : [{"cuisine" : {$ne : "American "}}, {"grades.score" : {$gt : 70}}, {"address.coord.0" : {$lt : -65.754168}}]}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65a03"),
  "address" : {
    "building" : "345",
    "coord" : [
      -73.9864626,
      40.7266739
    ],
    "street" : "East 6 Street",
    "zipcode" : "10003"
  },
  "borough" : "Manhattan",
  "cuisine" : "Indian",
  "grades" : [
    {
      "date" : ISODate("2014-09-15T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2014-01-14T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2013-05-30T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2013-04-24T00:00:00Z"),
      "grade" : "P",
      "score" : 2
    },
    {
      "date" : ISODate("2012-10-01T00:00:00Z"),
      "grade" : "A",
      "score" : 76
    }
  ]
}
```

5. 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 Bronx or Brooklyn

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"borough":{"$in":["Staten Island","Queens","Bronx","Brooklyn"]}}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65800"),
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "name" : "Riviera Caterer",
  "restaurant_id" : "40356018"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65801"),
  "borough" : "Brooklyn",
  "cuisine" : "Hamburgers",
  "name" : "Wendy'S",
  "restaurant_id" : "30112340"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65802"),
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "name" : "Morris Park Bake Shop",
  "restaurant_id" : "30075445"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65803"),
  "borough" : "Queens",
  "cuisine" : "American ",
  "name" : "Brunos On The Boulevard",
  "restaurant_id" : "40356151"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65804"),
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "name" : "Tov Kosher Kitchen",
  "restaurant_id" : "40356068"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65805"),
```

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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find().sort({"name":1}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee6648e"),
  "address" : {
    "building" : "129",
    "coord" : [
      -73.962043,
      40.685007
    ],
    "street" : "Gates Avenue",
    "zipcode" : "11238"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Italian",
  "grades" : [
    {
      "date" : ISODate("2014-03-06T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-08-29T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-03-08T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-06-27T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2011-11-17T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "(Lewis Drug Store) Locanda Vini E Olii",
  "restaurant_id" : "40804423"
}
```

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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({ "address.street": { $exists: true } }).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee657ff"),
  "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 Restaurant",
  "restaurant_id" : "30191841"
}
```

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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"borough":"Bronx", $or:[{"cuisine":"American"}, {"cuisine":"Chinese"}]}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65809"),
  "address" : {
    "building" : "2300",
    "coord" : [
      -73.8786113,
      40.8502883
    ],
    "street" : "Southern Boulevard",
    "zipcode" : "10460"
  },
  "borough" : "Bronx",
  "cuisine" : "American",
  "grades" : [
    {
      "date" : ISODate("2014-05-28T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-06-19T00:00:00Z"),
      "grade" : "A",
      "score" : 4
    },
    {
      "date" : ISODate("2012-06-15T00:00:00Z"),
      "grade" : "A",
      "score" : 3
    }
  ],
  "name" : "Wild Asia",
  "restaurant_id" : "40357217"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65820"),
  "address" : {
    "building" : "1236",

```

9. 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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"address.coord.1": {"$gt": 42, "$lte": 52}}, {"restaurant_id": 1, "name": 1, "address": 1, "coord": 1}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65aa0"),
  "address" : {
    "building" : "47",
    "coord" : [
      -78.877224,
      42.89546199999999
    ],
    "street" : "Broadway @ Trinity Pl",
    "zipcode" : "10006"
  },
  "name" : "T.G.I. Friday'S",
  "restaurant_id" : "40387990"
},
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65acc"),
  "address" : {
    "building" : "1",
    "coord" : [
      -0.7119979,
      51.6514664
    ],
    "street" : "Pennplaza E, Penn Sta",
    "zipcode" : "10001"
  },
  "name" : "T.G.I. Fridays",
  "restaurant_id" : "40388936"
},
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65d26"),
  "address" : {
    "building" : "3000",
    "coord" : [
      -87.86567699999999,
      42.611509200000004
    ],
    "street" : "3000 N",
    "zipcode" : "60641"
  },
  "name" : "T.G.I. Fridays",
  "restaurant_id" : "40388936"
}
```

10. 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.

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"$and": [{"cuisine": {"$ne": "American"}}, {"grades.grade": "A"}, {"borough": {"$ne": "Brooklyn"}}]).sort({"cuisine": -1}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65f0a"),
  "address" : {
    "building" : "89",
    "coord" : [
      -73.9995899,
      40.7168015
    ],
    "street" : "Baxter Street",
    "zipcode" : "10013"
  },
  "borough" : "Manhattan",
  "cuisine" : "Vietnamese/Cambodian/Malaysia",
  "grades" : [
    {
      "date" : ISODate("2014-08-21T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2013-08-31T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2013-04-11T00:00:00Z"),
      "grade" : "C",
      "score" : 3
    },
    {
      "date" : ISODate("2012-10-17T00:00:00Z"),
      "grade" : "A",
      "score" : 4
    },
    {
      "date" : ISODate("2012-05-15T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ]
}
```

11. 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"

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade": "A", "grades.score": 11}, {"restaurant_id": 1, "name": 1, "grades": 1}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee6587d"),
  "grades" : [
    {
      "date" : ISODate("2014-08-11T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2013-07-22T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2013-03-14T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2012-07-02T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-02-02T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-08-24T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    }
  ],
  "name" : "Neary's Pub",
  "restaurant_id" : "40365871"
}
```

12. 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

```
C:\Windows\System32\cmd.exe - mongo
> db.restaurants.find({name: /^Wil/}, {"restaurant_id": 1, "name": 1, "borough": 1, "cuisine": 1}).pretty()
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65806"),
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken's Fine Food",
  "restaurant_id" : "40356483"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee65809"),
  "borough" : "Bronx",
  "cuisine" : "American",
  "name" : "Wild Asia",
  "restaurant_id" : "40357217"
}
{
  "_id" : ObjectId("626a2b3e8b39dd0c3ee660f"),
  "borough" : "Bronx",
  "cuisine" : "Pizza",
  "name" : "Wilbel Pizza",
  "restaurant_id" : "40871979"
}
```