

Mongodb Assignment3:

Complex

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

```
mongo mongo://restaurants.4x6t6.mongodb.net/myFirstDatabase
Atlas atlas-ruuw8-shard-0 [primary] myFirstDatabase> show dbs
restaurant 881 kB
admin 336 kB
local 3.83 GB
Atlas atlas-ruuw8-shard-0 [primary] myFirstDatabase> use restaurant
switched to db restaurant
Atlas atlas-ruuw8-shard-0 [primary] restaurant> db.address.find()
Atlas atlas-ruuw8-shard-0 [primary] restaurant> db.address.find()
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ]
  },
  {
    name: 'Morris Park Bake Shop',
  }
]
```

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

```
Atlas atlas-ruuw8-shard-0 [primary] restaurant> db.address.find({}, {_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop'
  },
  {
    _id: ObjectId("6206230b624bceb260229f70"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'Wendy'S'
  },
  {
    _id: ObjectId("6206230b624bceb260229f71"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant'
  },
  {
    _id: ObjectId("6206230b624bceb260229f72"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Riviera Caterer'
  },
  {
    _id: ObjectId("6206230b624bceb260229f73"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    _id: ObjectId("6206230b624bceb260229f74"),
    borough: 'Queens',
    cuisine: 'American',
    name: 'Brunos On The Boulevard'
  },
  {
    _id: ObjectId("6206230b624bceb260229f75"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island'
  },
  {
    _id: ObjectId("6206230b624bceb260229f76"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
  }
]
```

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.

```
mongosh mongodb+srv://restaurants.4x6t6.mongodb.net/myFirstDatabase
Type "it" for more
Atlas atlas-ruuwu8-shard-0 [primary] restaurant> db.address.find({},(restaurant_id:1,_id:0,name:1,borough:1,cuisine:1))
[
  {
    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',
  }
]
```

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.

```
mongosh mongodb+srv://restaurants.4x6t6.mongodb.net/myFirstDatabase
Type "it" for more
Atlas atlas-ruuwu8-shard-0 [primary] restaurant> db.address.find({},(restaurant_id:1,_id:0,name:1,borough:1,"address.zipcode":1))
[
  {
    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' },
    borough: 'Queens',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    address: { zipcode: '10314' },
    borough: 'Staten Island',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    address: { zipcode: '11234' },
    borough: 'Brooklyn',
    name: 'Wilken'S Fine Food',
  }
]
```

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

```
mongosh mongodb+srv://restaurants.4xht6.mongodb.net/myFirstDatabase
Atlas atlas-ruuwu8-shard-0 [primary] restaurant> db.address.find({"borough":"Bronx"}).limit(5)
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("6206230b624bceb260229f79"),
    address: {
      building: '2300',
      coord: [ -73.8786113, 40.8502883 ],
      street: 'Southern Boulevard',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: 'Murals On 54/Randolphs'S',
    restaurant_id: '40372466'
  }
]
```

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

```
mongosh mongodb+srv://restaurants.4xht6.mongodb.net/myFirstDatabase
Atlas atlas-ruuwu8-shard-0 [primary] restaurant> db.address.find({"grades.score":{$gt:90}})
[
  {
    _id: ObjectId("6206230b624bceb26022a0cd"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: 'Murals On 54/Randolphs'S',
    restaurant_id: '40372466'
  }
]
```

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

```
Atlas atlas-ruwuw8-shard-0 [primary] restaurant> db.address.find({"address.coord":{"$lt":-95.754168}})
[
  {
    _id: ObjectId("6206230d624bceb26022a5b7"),
    address: {
      building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372'
    },
    borough: 'Queens',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-04T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2013-11-07T00:00:00.000Z"),
        grade: 'B',
        score: 19
      },
      {
        date: ISODate("2013-05-17T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-04-03T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2011-11-16T00:00:00.000Z"),
        grade: 'A',
        score: 7
      }
    ],
    name: 'Burger King',
    restaurant_id: '40534067'
  },
  {
```

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.

```
Atlas atlas-ruwuw8-shard-0 [primary] restaurant> db.address.find({"$and":[{"cuisine":{"$ne":"American"}},{"grades.score":{"$gt":70}},{"address.coord":{"$lt":-65.754168}}])
[
  {
    _id: ObjectId("6206230b624bceb26022a0cd"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: "Murals On 54/Randolphs'S",
    restaurant_id: '40372466'
  },
  {
```

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.

```
Atlas atlas-rumuw8-shard-0 [primary] restaurant> db.address.find({"$and":[{"cuisine":{"$ne":"American"}},{"!Atlas atlas-rumuw8-shard-0 [primary] restaurant> db.address.find({name:"Wil"},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})
[
  {
    _id: ObjectId("6206230b624bceb260229f76"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Wilken'S Fine Food',
    restaurant_id: '40356483'
  },
  {
    _id: ObjectId("6206230b624bceb260229f79"),
    borough: 'Bronx',
    cuisine: 'American ',
    name: 'Wild Asia',
    restaurant_id: '40357213'
  },
  {
    _id: ObjectId("62062311624bceb26022ad7e"),
    borough: 'Bronx',
    cuisine: 'Pizza',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
Atlas atlas-rumuw8-shard-0 [primary] restaurant>
```

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

```

Atlas atlas-ruuuw8-shard-0 [primary] restaurant> db.address.find({'name':/.*Reg.*/},{ 'restaurant_id':1,'name':1,'borough':1,'cuisine':1})
[
  {
    _id: ObjectId("6206230b624bceb260229f77"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    _id: ObjectId("6206230b624bceb26022a074"),
    borough: 'Manhattan',
    cuisine: 'Café/Coffee/Tea',
    name: 'Caffe Reggio',
    restaurant_id: '40369418'
  },
  {
    _id: ObjectId("6206230b624bceb26022a183"),
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
  {
    _id: ObjectId("6206230d624bceb26022a4a0"),
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Regency Whist Club',
    restaurant_id: '40402377'
  },
  {
    _id: ObjectId("6206230d624bceb26022a583"),
    borough: 'Queens',
    cuisine: 'American',
    name: 'Rego Park Cafe',
    restaurant_id: '40523342'
  },
  {
    _id: ObjectId("62062311624bceb26022abf1"),
    borough: 'Queens',
    cuisine: 'Pizza',
    name: 'Regina Pizza',
    restaurant_id: '40801325'
  },
  {
    _id: ObjectId("62062311624bceb26022ae08"),
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Regal Entertainment Group',

```

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.

```
mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-rwuuu8-shard-0 [primary] restaurant> db.address.find({"borough":{"$in":["Staten Island","Queens","Bronx","Brooklyn"]}},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("6206230b624bceb260229f70"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'Wendy S',
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("6206230b624bceb260229f72"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    _id: ObjectId("6206230b624bceb260229f73"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356008'
  },
  {
    _id: ObjectId("6206230b624bceb260229f74"),
    borough: 'Queens',
    cuisine: 'American',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    _id: ObjectId("6206230b624bceb260229f75"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    _id: ObjectId("6206230b624bceb260229f76"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    restaurant_id: '40356483'
  }
]
```

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.

```
Select mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-rwuuu8-shard-0 [primary] restaurant> db.address.find({"grades.score":{"$not":{"$gt":10}}},{"restaurant_id":1,"name":1,"borough":1,"cuisine":1})
[
  {
    _id: ObjectId("6206230b624bceb260229f7a"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'C & C Catering Service',
    restaurant_id: '40357437',
    restaurant_id: '40356649'
  },
  {
    _id: ObjectId("6206230b624bceb260229f7c"),
    borough: 'Manhattan',
    cuisine: 'American',
    name: '1 East 66th Street Kitchen',
    restaurant_id: '40359480',
    restaurant_id: '40356731'
  },
  {
    _id: ObjectId("6206230b624bceb260229f80"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies',
    restaurant_id: '40361390',
    restaurant_id: '40357217'
  },
  {
    _id: ObjectId("6206230b624bceb260229f89"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'White Castle',
    restaurant_id: '40362344',
    restaurant_id: '40357437'
  },
  {
    _id: ObjectId("6206230b624bceb260229f9c"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Sonny's Heroes',
    restaurant_id: '40363744',
    restaurant_id: '40358429'
  },
  {
    _id: ObjectId("6206230b624bceb260229fac"),
    borough: 'Bronx',
    cuisine: 'American',
    name: 'Manhem Club',
    restaurant_id: '40364363',
    restaurant_id: '40360045'
  },
  {
    _id: ObjectId("6206230b624bceb260229fba"),
    borough: 'Staten Island',
    cuisine: 'American',
    name: 'Great Kills Yacht Club',
    restaurant_id: '40360045'
  }
]
```

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.

```

mongosh mongodb+srv://restaurants:4xbt6.mongodb.net/myFirstDatabase
cuisine: 'American ',
Atlas atlas-runuw8-shard-0 [primary] restaurant> db.address.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11},{"restaurant_id":1,"name":1,"grades":1})
[
  {
    restaurant_id: '48368076'
  },
  {
    _id: ObjectId("6206230b624bceb260229fed"),
    grades: [ 'Manhattan',0b624bceb260229f7f'],
    (sine: 'American ',
      date: ISODate("2014-08-11T00:00:00.000Z"),
      grade: 'A', '48364863'
      score: 13: '48361322'
    ),
    { ObjectID("6206230b624bceb260229fc5"),
      date: ISODate("2013-07-22T00:00:00.000Z"),
      grade: 'A',can ',
      score: 9 Horse Tavern',
    },aurant_id: '48364958',
    (taurant_id: '48361398'
      date: ISODate("2013-03-14T00:00:00.000Z"),
      grade: 'A',6206230b624bceb260229f82'),
      score: 12hattan',0b624bceb260229f82'),
    ),ine: 'Irish",n',
    (e: "Dorrian'S Red Hand Restaurant",
      date: ISODate("2012-07-02T00:00:00.000Z"),
      grade: 'A', '48361606'
      score: 11
    ), ObjectID("6206230b624bceb26022a81f"),
    ough: 'Manhattan',0b624bceb260229f83"),
      date: ISODate("2012-02-02T00:00:00.000Z"),
      grade: 'A',indo Restaurant',
      score: 10: '48367838'
    ),aurant_id: '48361618'
    (
      date: ISODate("2011-08-24T00:00:00.000Z"),
      grade: 'A',lyn',30b624bceb260229f85"),
      score: 11ek',,,
    ),e: 'El Greco Diner',,,
    ),staurant_id: '48367795' Grocery",
      name: "Mary'S Pub",1998'
      restaurant_id: '48365871'
    ),_id: ObjectId("6206230b624bceb26022a845"),
    { borough: 'Bronx',06230b624bceb260229f89'),
      _id: ObjectId("6206230b624bceb26022a8c8"),
      grades: [ e Lark'S Nest",
        (taurant_id: '48367946'
          grades: 'A',
          score: 11("6206230b624bceb26022a846"),
          ),ugh: 'Bronx',
          (sine: 'Café/Coffee/Tea',mary] restaurant>

```

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.

```

}
Atlas atlas-runuw8-shard-0 [primary] restaurant> db.address.find({"address.coord.1":{">42,$lt:52}},{"restaurant_id":1,"name":1,"address":1,"coord":1})
[
  {
    _id: ObjectId("6206230b624bceb26022a211"),
    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("6206230b624bceb26022a23d"),
    address: {
      building: '1',
      coord: [ -0.7119979, 51.6514664 ],
      street: 'Pennplaza E, Penn Sta',
      zipcode: '10001'
    },
    name: "T.G.I. Friday'S",
    restaurant_id: '40388936'
  },
  {
    _id: ObjectId("6206230d624bceb26022a496"),
    address: {
      building: '3000',
      coord: [ -87.86567699999999, 42.61150920000001 ],
      street: '47 Avenue',
      zipcode: '11101'
    },
    name: "Di Luvio'S Deli",
    restaurant_id: '40402284'
  },
  {
    _id: ObjectId("6206230d624bceb26022a6cb"),
    address: {
      building: '21972199',
      coord: [ -78.589606, 42.8912372 ],
      street: 'Broadway',
      zipcode: '10024'
    },
    name: 'La Caridad 78',
    restaurant_id: '40568285'
  },
]

```

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

```
mongosh mongodb+srv://restaurants.4xht6.mongodb.net/myFirstDatabase
]
restaurant_id: '40882356'
}
Atlas atlas-ruuw8-shard-0 [primary] restaurant> db.address.find().sort({"name":-1})
[
  {
    _id: ObjectId("6206230b624bceb26022a02e"),
    address: {
      building: '6946',
      coord: [ -73.8811834, 40.7017759 ],
      street: 'Myrtle Avenue',
      zipcode: '11385'
    },
    borough: 'Queens',
    cuisine: 'German',
    grades: [
      {
        date: ISODate("2014-09-24T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-04-17T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2013-03-12T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-10-02T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-05-09T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2011-12-28T00:00:00.000Z"),
        grade: 'B',
        score: 13
      }
    ]
  },
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    address: {
      building: '1087',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ]
  },
  {
    _id: ObjectId("6206230b624bceb260229f70"),
    address: {
      building: '469',
      coord: [ -73.961704, 40.662942 ],
      street: '14th St',
      zipcode: '10013'
    },
    borough: 'Manhattan',
    cuisine: 'Italian',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ]
  }
]
```

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

```
mongosh mongodb+srv://restaurants.4xht6.mongodb.net/myFirstDatabase
]
Type "it" for more
Atlas atlas-ruuw8-shard-0 [primary] restaurant> db.address.find({"address.street":{"exists:true"}})
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    address: {
      building: '1087',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ]
  },
  {
    _id: ObjectId("6206230b624bceb260229f70"),
    address: {
      building: '469',
      coord: [ -73.961704, 40.662942 ],
      street: '14th St',
      zipcode: '10013'
    },
    borough: 'Manhattan',
    cuisine: 'Italian',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ]
  }
]
```


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.

```
mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-ruwuw8-shard-0 [primary] restaurant> db.address.find({"grades.score":{$mod:[7,0]}},{"restaurant_id":1,"name":1,"grades":1})
[
  {
    _id: ObjectId("6206230b624bceb260229f6f"),
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("6206230b624bceb260229f72"),
    grades: [
      {
        date: ISODate("2014-06-10T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2013-06-05T00:00:00.000Z"),
        grade: 'A',
        score: 7
      }
    ],
  }
]
```

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

```
mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-ruwuw8-shard-0 [primary] restaurant> db.address.find().sort({"name":1})
[
  {
    name: 'Manhattan Monster',
    _id: ObjectId("62062311624bceb26022abff"),
    address: {
      building: '129',
      coord: [ -73.9622943, 40.685007 ],
      street: 'Gates Avenue',
      zipcode: '11238',
    },
    me: 'Monte'S',
    borough: 'Brooklyn',
    cuisine: 'Italian',
    grades: [
      {
        _id: ObjectId("6206230b624bceb26022a14f"),
        {
          coord: [ -73.9901605, 40.7526176 ]
        },
        date: ISODate("2014-03-06T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2013-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2013-03-08T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-06-27T00:00:00.000Z"),
        grade: 'A',
        score: 7
      }
    ],
    name: '(Lewis Drug Store) Locanda Vini E Olii',
    restaurant_id: '40804423'
  },
  {
    address: {
      coord: [ -73.9705633, 40.7605759 ]
    },
    borough: 'Manhattan',
    _id: ObjectId("6206230b624bceb260229f7c"),
    address: {
      building: '1',
      coord: [ -73.96926909999999, 40.7685235 ],
      street: 'East 66 Street',
    },
    name: 'Delmonico Gourmet'
  }
]
```

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.

```
mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-runuw8-shard-0 [primary] restaurant> db.address.find((name:{$regex:/^Mad/i,}),{"name":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("6206230d624bceb26022a4ab"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("6206230d624bceb26022a579"),
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
    _id: ObjectId("6206230f624bceb26022a827"),
    address: { coord: [ -74.000002, 40.72735 ] },
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Madame X'
  },
  {
    _id: ObjectId("6206230f624bceb26022a8d7"),
    address: { coord: [ -73.98171959999999, 40.7499406 ] },
    borough: 'Manhattan',
    cuisine: 'French',
    name: 'Madison Bistro'
  },
  {
    _id: ObjectId("6206230f624bceb26022a960"),
    address: { coord: [ -73.9717845, 40.6897199 ] },
    borough: 'Brooklyn',
    cuisine: 'African',
    name: 'Madiba'
  },
  {
    _id: ObjectId("62062311624bceb26022ac65"),
    address: { coord: [ -73.9040753, 40.9069011 ] },
    borough: 'Bronx',
    cuisine: 'Italian',
    name: 'Madison'S'
  },
  {
    _id: ObjectId("62062311624bceb26022ace3"),
    address: { coord: [ -73.9886598, 40.7565811 ] },
    borough: 'Manhattan',
  },
]
```

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

```
mongosh mongodb+srv://restaurants.4xbt6.mongodb.net/myFirstDatabase
Atlas atlas-runuw8-shard-0 [primary] restaurant> db.address.find((name:{$regex:"mon.*",options:"i"}),{"name":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("6206230b624bceb26022a003"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American',
    name: 'Desmond'S Tavern'
  },
  {
    _id: ObjectId("6206230b624bceb26022a00c"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("6206230b624bceb26022a018"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American',
    name: 'Richmond County Country Club'
  },
  {
    _id: ObjectId("6206230b624bceb26022a043"),
    address: { coord: [ -73.9812843, 40.5947365 ] },
    borough: 'Brooklyn',
    cuisine: 'Pizza/Italian',
    name: 'Ib Spumoni Gardens'
  },
  {
    _id: ObjectId("6206230b624bceb26022a095"),
    address: { coord: [ -73.951199, 40.7166026 ] },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    name: 'Bamonte'S Restaurant'
  },
  {
    _id: ObjectId("6206230b624bceb26022a0cc"),
    address: { coord: [ -73.924072, 40.76108900000001 ] },
    borough: 'Queens',
    cuisine: 'Greek',
    name: 'Omonia Cafe'
  },
  {
    _id: ObjectId("6206230b624bceb26022a11c"),
    address: { coord: [ -74.0023353, 40.7333573 ] },
    borough: 'Manhattan',
  },
]
```