

MongoDB – Complex Queries

Mongo DB Exercises - With the Restaurants Data Set

1. Download the restaurants.zip file
2. Unzip the file, you will see restaurants.json file

```
C:\Users\LENOVO>mongoimport --db restaurants --collection addresses --drop --file D:/restaurants.json
2022-02-10T18:57:18.376+0530 connected to: mongodb://localhost/
2022-02-10T18:57:18.381+0530 dropping: restaurants.addresses
2022-02-10T18:57:18.580+0530 3772 document(s) imported successfully. 0 document(s) failed to import.
```

3. Run the mongod server
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

mongoimport --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

```
C:\Users\LENOVO>use restaurants
switched to db restaurants
> show dbs
admin          0.000GB
config        0.000GB
local          0.000GB
population    0.002GB
restaurants   0.001GB
test           0.002GB
youtube        0.000GB
> db.addresses.find()
{
  "_id" : ObjectId("620512b1bed51f4a97fb2732"),
  "address" : {
    "building" : "97-22",
    "coord" : [
      -73.8601152,
      40.7311739
    ],
    "street" : "63 Road",
    "zipcode" : "11374"
  },
  "borough" : "Queens",
  "cuisine" : "lewi h/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" : "B",
      "score" : 25
    }
  ],
  "name" : "Tov Kosher Kitchen",
  "restaurantId" : "40356068"
}
{
  "_id" : ObjectId("620512b1bed51f4a97fb2733"),
  "address" : {
    "building" : "469",
    "coord" : [
      -73.9617046,
      40.662942
    ],
    "street" : "Flatbush Avenue",
    "zipcode" : "11228"
  },
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "grades" : [
    {
      "date" : ISODate("2011-12-15T00:00:00Z"),
      "grade" : "B",
      "score" : 8
    },
    {
      "date" : ISODate("2014-07-01T00:00:00Z"),
      "grade" : "B",
      "score" : 23
    },
    {
      "date" : ISODate("2013-04-08T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2012-05-05T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Mandy's",
  "restaurantId" : "30112340"
}
{
  "_id" : ObjectId("620512b1bed51f4a97fb2734"),
  "address" : {
    "building" : "206",
    "coord" : [
      -74.1377286,
      40.6119578
    ],
    "street" : "Victory Boulevard",
    "zipcode" : "10314"
  },
  "borough" : "Staten 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",
      "score" : 12
    },
    {
      "date" : ISODate("2013-04-04T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Kosher Deli",
  "restaurantId" : "40356104"
}
{
  "_id" : ObjectId("620512b1bed51f4a97fb2735"),
  "address" : {
    "building" : "8325",
    "coord" : [
      -73.8803827,
      40.7643124
    ],
    "street" : "Astoria Boulevard",
    "zipcode" : "11360"
  },
  "borough" : "Queens",
  "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" : "Bruno On The Boulevard",
  "restaurantId" : "40356151"
}
{
  "_id" : ObjectId("620512b1bed51f4a97fb2736"),
  "address" : {
    "building" : "711A",
    "coord" : [
      -73.9068566,
      40.6199031
    ],
    "street" : "Avenue U",
    "zipcode" : "11234"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "grades" : [
    {
      "date" : ISODate("2014-05-01T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2014-01-10T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2013-08-01T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2012-03-09T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2011-10-14T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    }
  ],
  "name" : "Wilken's Fine Food",
  "restaurantId" : "40356483"
}
{
  "_id" : ObjectId("620512b1bed51f4a97fb2737"),
  "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" : 9
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
      "grade" : "B",
      "score" : 14
    }
  ],
  "name" : "Morris Park
```

9. Then start working on the following exercises and submit your queries as the answers to the questions

Query Reference Links and Cheat sheets

1. <https://docs.mongodb.com/manual/crud/>

Exercise Questions

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

```
Type "it" for more
> db.addresses.find().pretty()
{
  "_id" : ObjectId("620512b61ed51f4497fb2732"),
  "address" : {
    "building" : "97-22",
    "coord" : [
      -73.8601152,
      40.7311739
    ],
    "street" : "63 Road",
    "zipcode" : "11374"
  },
  "borough" : "Queens",
  "cuisine" : "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("620512b61ed51f4497fb2733"),
  "address" : {
    "building" : "469",
    "coord" : [
      -73.961704,
      40.662942
    ],
    "street" : "Flatbush Avenue",
  }
}
```

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

```
> db.addresses.find({},{restaurant_id:1,borough:1,cuisine:1})
{
  "_id" : ObjectId("620512b61ed51f4497fb2732"),
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "name" : "Tov Kosher Kitchen",
  "restaurant_id" : "40356068"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2733"),
  "borough" : "Brooklyn",
  "cuisine" : "Hamburgers",
  "name" : "Kosher's Deli",
  "restaurant_id" : "40356442"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2734"),
  "borough" : "Staten Island",
  "cuisine" : "Jewish/Kosher",
  "name" : "Kosher Island Deli",
  "restaurant_id" : "40356442"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2735"),
  "borough" : "Queens",
  "cuisine" : "American",
  "name" : "Brunos On The Boulevard",
  "restaurant_id" : "40356151"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2736"),
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken's Fine Food",
  "restaurant_id" : "40356483"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2737"),
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "name" : "Norris Park Bake Shop",
  "restaurant_id" : "30675445"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2738"),
  "borough" : "Bronx",
  "cuisine" : "American",
  "name" : "Wild Asia Restaurant",
  "restaurant_id" : "40356727"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2739"),
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "name" : "Regina Caterers",
  "restaurant_id" : "40356649"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2735"),
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "name" : "Riviera Caterers",
  "restaurant_id" : "40356018"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2736"),
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "name" : "C & C Catering Service",
  "restaurant_id" : "40357437"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb273c"),
  "borough" : "Brooklyn",
  "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
  "name" : "Taste The Tropics Ice Cream",
  "restaurant_id" : "40356731"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb273d"),
  "borough" : "Brooklyn",
  "cuisine" : "Chinese",
  "name" : "Mao May Kitchen",
  "restaurant_id" : "40358423"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb273e"),
  "borough" : "Manhattan",
  "cuisine" : "American",
  "name" : "1 East 66th Street Kitchen",
  "restaurant_id" : "40359480"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb273f"),
  "borough" : "Queens",
  "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
  "name" : "Nordic Delicacies",
  "restaurant_id" : "40361322"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2740"),
  "borough" : "Brooklyn",
  "cuisine" : "Jewish/Kosher",
  "name" : "Seud Foods",
  "restaurant_id" : "40360645"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2741"),
  "borough" : "Brooklyn",
  "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
  "name" : "Carrie's Deli",
  "restaurant_id" : "40360076"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2742"),
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Nordic Delicacies",
  "restaurant_id" : "40361390"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2743"),
  "borough" : "Queens",
  "cuisine" : "Delicatessen",
  "name" : "Steve Chu's Deli & Grocery",
  "restaurant_id" : "40361998"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2744"),
  "borough" : "Manhattan",
  "cuisine" : "American",
  "name" : "Glorious Food",
  "restaurant_id" : "40361521"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2745"),
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "name" : "The Movable Feast",
  "restaurant_id" : "40361606"
}
(type "it" for more)
{
  "_id" : ObjectId("620512b61ed51f4497fb2746"),
  "borough" : "Manhattan",
  "cuisine" : "American",
  "name" : "P & S Deli Grocery",
  "restaurant_id" : "40362647"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2747"),
  "borough" : "Manhattan",
  "cuisine" : "Chicken",
  "name" : "Harriet's Kitchen",
  "restaurant_id" : "40362098"
}
{
  "_id" : ObjectId("620512b61ed51f4497fb2748"),
  "borough" : "Queens",
  "cuisine" : "Delicatessen",
  "name" : "Sal's Deli",
  "restaurant_id" : "40361618"
}
```

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.

```
Command Prompt - mongo
> db.addresses.find({ "restaurant_id":1, "name":1, "borough":1, "cuisine":1 })
{
  "_id": ObjectId("620512b1ed51f4a97fb2732"),
  "borough": "Queens",
  "cuisine": "Jewish/Kosher",
  "name": "Tov Kosher Kitchen",
  "restaurant_id": "40356068"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2733"),
  "borough": "Brooklyn",
  "cuisine": "Hamburgers",
  "name": "Wendy's",
  "restaurant_id": "30112340"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2734"),
  "borough": "Staten Island",
  "cuisine": "Jewish/Kosher",
  "name": "Kosher Island",
  "restaurant_id": "40356442"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2735"),
  "borough": "Queens",
  "cuisine": "American",
  "name": "Brunos On The Boulevard",
  "restaurant_id": "40356151"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2736"),
  "borough": "Brooklyn",
  "cuisine": "Delicatessen",
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2737"),
  "borough": "Bronx",
  "cuisine": "Bakery",
  "name": "Wild Asia",
  "restaurant_id": "40357217"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2738"),
  "borough": "Bronx",
  "cuisine": "American",
  "name": "Regina Caterers",
  "restaurant_id": "40356649"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2739"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "Riviera Caterer",
  "restaurant_id": "40356018"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2740"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "C & C Catering Service",
  "restaurant_id": "40357431"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb273c"),
  "borough": "Brooklyn",
  "cuisine": "Ice Cream, Gelato, Yogurt, Ices",
  "name": "Taste The Tropics Ice Cream",
  "restaurant_id": "40356731"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb273d"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "May May Kitchen",
  "restaurant_id": "40358429"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb273e"),
  "borough": "Queens",
  "cuisine": "Ice Cream, Gelato, Yogurt, Ices",
  "name": "Carvel Ice Cream",
  "restaurant_id": "40361322"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2740"),
  "borough": "Brooklyn",
  "cuisine": "Jewish/Kosher",
  "name": "Seuda Foods",
  "restaurant_id": "40362069"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2741"),
  "borough": "Brooklyn",
  "cuisine": "Ice Cream, Gelato, Yogurt, Ices",
  "name": "Carvel Ice Cream",
  "restaurant_id": "40360076"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2742"),
  "borough": "Brooklyn",
  "cuisine": "Delicatessen",
  "name": "Nordic Delicacies",
  "restaurant_id": "40361390"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2743"),
  "borough": "Queens",
  "cuisine": "Delicatessen",
  "name": "Steve Chu's Deli & Grocery",
  "restaurant_id": "40361998"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2744"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Glorious Food",
  "restaurant_id": "40361221"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2745"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "The Movable Feast",
  "restaurant_id": "40361606"
}
type "it" for more
> it
{
  "_id": ObjectId("620512b1ed51f4a97fb2746"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "P & S Deli Grocery",
  "restaurant_id": "40362264"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2747"),
  "borough": "Manhattan",
  "cuisine": "Chicken",
  "name": "Harriet's Kitchen",
  "restaurant_id": "40362098"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2748"),
  "borough": "Queens",
  "cuisine": "Delicatessen",
  "name": "Salli's Deli",
  "restaurant_id": "40361018"
}

```

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.

```
Command Prompt - mongo
> db.addresses.find({ "restaurant_id":1, "name":1, "borough":1, "address.zipcode":1, "id":0 })
{
  "address": { "zipcode": "11374" },
  "borough": "Queens",
  "name": "Tov Kosher Kitchen",
  "restaurant_id": "40356068"
},
{
  "address": { "zipcode": "11225" },
  "borough": "Brooklyn",
  "name": "Wendy's",
  "restaurant_id": "30112340"
},
{
  "address": { "zipcode": "10314" },
  "borough": "Staten Island",
  "name": "Kosher Island",
  "restaurant_id": "40356442"
},
{
  "address": { "zipcode": "11369" },
  "borough": "Queens",
  "name": "Brunos On The Boulevard",
  "restaurant_id": "40356151"
},
{
  "address": { "zipcode": "11234" },
  "borough": "Brooklyn",
  "name": "Wilken's Fine Food",
  "restaurant_id": "40356649"
},
{
  "address": { "zipcode": "10462" },
  "borough": "Bronx",
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
},
{
  "address": { "zipcode": "10468" },
  "borough": "Bronx",
  "name": "Wild Asia",
  "restaurant_id": "40357217"
},
{
  "address": { "zipcode": "11219" },
  "borough": "Brooklyn",
  "name": "Regina Caterers",
  "restaurant_id": "40356649"
},
{
  "address": { "zipcode": "11224" },
  "borough": "Brooklyn",
  "name": "Riviera Caterer",
  "restaurant_id": "40356018"
},
{
  "address": { "zipcode": "11214" },
  "borough": "Brooklyn",
  "name": "C & C Catering Service",
  "restaurant_id": "40357431"
},
{
  "address": { "zipcode": "11226" },
  "borough": "Brooklyn",
  "name": "Taste The Tropics Ice Cream",
  "restaurant_id": "40356731"
},
{
  "address": { "zipcode": "11208" },
  "borough": "Brooklyn",
  "name": "May May Kitchen",
  "restaurant_id": "40358429"
},
{
  "address": { "zipcode": "10005" },
  "borough": "Manhattan",
  "name": "1 East 66th Street Kitchen",
  "restaurant_id": "40359480"
},
{
  "address": { "zipcode": "11004" },
  "borough": "Queens",
  "name": "Carvel Ice Cream",
  "restaurant_id": "40361322"
},
{
  "address": { "zipcode": "11223" },
  "borough": "Brooklyn",
  "name": "Seuda Foods",
  "restaurant_id": "40360045"
},
{
  "address": { "zipcode": "11118" },
  "borough": "Brooklyn",
  "name": "Carvel Ice Cream",
  "restaurant_id": "40360076"
},
{
  "address": { "zipcode": "11209" },
  "borough": "Brooklyn",
  "name": "Nordic Delicacies",
  "restaurant_id": "40361390"
},
{
  "address": { "zipcode": "11116" },
  "borough": "Queens",
  "name": "Steve Chu's Deli & Grocery",
  "restaurant_id": "40361998"
},
{
  "address": { "zipcode": "10021" },
  "borough": "Manhattan",
  "name": "Glorious Food",
  "restaurant_id": "40361221"
},
{
  "address": { "zipcode": "11215" },
  "borough": "Brooklyn",
  "name": "The Movable Feast",
  "restaurant_id": "40361606"
}
type "it" for more

```

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

```
Command Prompt - mongo
> db.addresses.find({ "borough": "Bronx" }).limit(5)
{
  "_id": ObjectId("620512b1ed51f4a97fb2737"),
  "address": { "building": "1007", "coord": [ -73.856071, 40.848447 ],
  "street": "Morris Park Ave.", "zipcode": "10462" },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    { "date": ISODate("2014-03-03T00:00:00Z"), "grade": "A", "score": 6 },
    { "date": ISODate("2013-09-11T00:00:00Z"), "grade": "A", "score": 6 },
    { "date": ISODate("2011-11-23T00:00:00Z"), "grade": "A", "score": 9 },
    { "date": ISODate("2010-11-10T00:00:00Z"), "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2738"),
  "address": { "building": "2309", "coord": [ -72.878611, 40.850288 ],
  "street": "Bronx Boulevard", "zipcode": "10460" },
  "borough": "Bronx",
  "cuisine": "Ice Cream, Gelato, Yogurt, Ices",
  "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("620512b1ed51f4a97fb2750"),
  "address": { "building": "1006", "coord": [ -73.84856870000002, 40.8903781 ],
  "street": "East 233 Street", "zipcode": "10466" },
  "borough": "Bronx",
  "cuisine": "Carvel Ice Cream",
  "grades": [
    { "date": ISODate("2014-04-24T00:00:00Z"), "grade": "B", "score": 15 },
    { "date": ISODate("2013-09-17T00:00:00Z"), "grade": "B", "score": 14 }
  ],
  "name": "Carvel Ice Cream",
  "restaurant_id": "40361391"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb2752"),
  "address": { "building": "1226", "coord": [ -73.8893654, 40.8126179999999 ],
  "street": "238 Spofford Ave.", "zipcode": "10474" },
  "borough": "Bronx",
  "cuisine": "Chinese",
  "grades": [
    { "date": ISODate("2013-12-30T00:00:00Z"), "grade": "A", "score": 10 },
    { "date": ISODate("2012-08-29T00:00:00Z"), "grade": "A", "score": 8 },
    { "date": ISODate("2011-07-11T00:00:00Z"), "grade": "A", "score": 11 },
    { "date": ISODate("2010-09-02T00:00:00Z"), "grade": "A", "score": 10 }
  ],
  "name": "Happy Garden",
  "restaurant_id": "40361229"
},
{
  "_id": ObjectId("620512b1ed51f4a97fb761"),
  "address": { "building": "277", "coord": [ -73.894189, 40.8634684 ],
  "street": "East 19th Bridge Road", "zipcode": "10458" },
  "borough": "Bronx",
  "cuisine": "Chinese",
  "grades": [
    { "date": ISODate("2014-03-03T00:00:00Z"), "grade": "A", "score": 10 },
    { "date": ISODate("2013-09-26T00:00:00Z"), "grade": "A", "score": 10 }
  ],
  "name": "Happy Garden",
  "restaurant_id": "40364296"
}
```

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

```
Command Prompt - mongo

> db.Command.addresses.find({ "borough": "Bronx" })
{ "_id": ObjectId("620512bd16d51f4497fb2737"), "address": { "building": "1097", "coord": [ -73.856071, 40.848847 ], "street": "110th Street", "zipCode": "10465" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2013-03-03T00:00:00Z"), "grade": "A", "score": 6 }, { "date": ISODate("2013-09-11T00:00:00Z"), "grade": "A", "score": 6 }, { "date": ISODate("2013-01-24T00:00:00Z"), "grade": "B+", "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": "30875445" }
{ "_id": ObjectId("620512bd16d51f4497fb2738"), "address": { "building": "2300", "coord": [ -73.878613, 40.850288 ], "street": "Southern Boulevard", "zipCode": "10460" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-05-26T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2013-06-15T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2012-06-15T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Wild Asia", "restaurant_id": "40357232" }
{ "_id": ObjectId("620512bd16d51f4497fb2750"), "address": { "building": "1006", "coord": [ -73.84865870000002, 40.8930781 ], "street": "East 233 Street", "zipCode": "10466" }, "borough": "Bronx", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "grades": [ { "date": ISODate("2014-04-24T00:00:00Z"), "grade": "A", "score": 18 }, { "date": ISODate("2013-09-05T00:00:00Z"), "grade": "A", "score": 18 }, { "date": ISODate("2012-03-21T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2011-07-11T00:00:00Z"), "grade": "A", "score": 9 } ], "name": "Carvel Ice Cream & Frozen Custard", "restaurant_id": "40363091" }
{ "_id": ObjectId("620512bd16d51f4497fb2761"), "address": { "building": "1236", "coord": [ -73.889624, 40.817691 ], "street": "238 Spofford Ave", "zipCode": "10474" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2013-12-30T00:00:00Z"), "grade": "A", "score": 8 }, { "date": ISODate("2013-01-08T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-02-10T00:00:00Z"), "grade": "B", "score": 15 } ], "name": "Happy Garden", "restaurant_id": "40363289" }
{ "_id": ObjectId("620512bd16d51f4497fb2761"), "address": { "building": "277", "coord": [ -73.894169, 40.8634684 ], "street": "East Kingsbridge Road", "zipCode": "10458" }, "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-03-03T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2014-06-26T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2013-09-05T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-07-11T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Manho Club", "restaurant_id": "40364367" }
{ "_id": ObjectId("620512bd16d51f4497fb2764"), "address": { "building": "2222", "coord": [ -73.84975999999999, 40.881818 ], "street": "21st Avenue", "zipCode": "10465" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-01-15T00:00:00Z"), "grade": "B", "score": 17 }, { "date": ISODate("2013-01-14T00:00:00Z"), "grade": "A", "score": 12 } ], "name": "The New Starling Athletic Club Of The Bronx", "restaurant_id": "40364956" }
{ "_id": ObjectId("620512bd16d51f4497fb27a7"), "address": { "building": "72", "coord": [ -73.92506, 40.8275556 ], "street": "East 161 Street", "zipCode": "10451" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-05-15T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2012-02-08T00:00:00Z"), "grade": "B", "score": 19 } ], "name": "The New Starling Athletic Club Of The Bronx", "restaurant_id": "40364956" }
{ "_id": ObjectId("620512bd16d51f4497fb27a7"), "address": { "building": "72", "coord": [ -73.92506, 40.8275556 ], "street": "East 161 Street", "zipCode": "10451" }, "borough": "Bronx", "cuisine": "American", "grades": [ { "date": ISODate("2014-05-15T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2013-11-14T00:00:00Z"), "grade": "A", "score": 4 } ], "name": "The New Starling Athletic Club Of The Bronx", "restaurant_id": "40364956" }
```

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

```
Command Prompt - mongo

> db.addresses.find("borough": "Bronx").skip(5).limit(5)
{
  "_id": ObjectId("620512b61ed5f449fb72d70e"), "address": {"building": "658", "coord": [-73.81369999999999, 40.823411000000005], "street": "Clarence Ave", "zipcode": "10465"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-06-21T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2012-07-11T00:00:00Z"), "grade": "A", "score": 10}], "name": "Manhattan Club Restaurant", "restaurant_id": "40363631"}
{
  "_id": ObjectId("620512b61ed5f449fb72784"), "address": {"building": "2222", "coord": [-73.84971999999999, 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("2013-09-14T00:00:00Z"), "grade": "A", "score": 12}, {"date": ISODate("2012-05-20T00:00:00Z"), "grade": "B", "score": 19}], "name": "The New Starling Athletic Club Of The Bronx", "restaurant_id": "40364956"}
{
  "_id": ObjectId("620512b61ed5f449fb727ac"), "address": {"building": "1334", "coord": [-73.925068275356, 40.8451611], "street": "Bronx", "zipcode": "10451"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-04-15T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2013-11-14T00:00:00Z"), "grade": "A", "score": 4}, {"date": ISODate("2013-07-29T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2012-05-30T00:00:00Z"), "grade": "A", "score": 13}, {"date": ISODate("2012-01-09T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2010-11-15T00:00:00Z"), "grade": "C", "score": 37}], "name": "Yankee Tavern", "restaurant_id": "40365499"}
{
  "_id": ObjectId("620512b61ed5f449fb727ac"), "address": {"building": "1334", "coord": [-73.87786599999999, 40.8743771], "street": "East 204 Street", "zipcode": "10467"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-03-27T00:00:00Z"), "grade": "B", "score": 13}, {"date": ISODate("2013-11-10T00:00:00Z"), "grade": "A", "score": 13}, {"date": ISODate("2013-07-29T00:00:00Z"), "grade": "B", "score": 13}, {"date": ISODate("2013-03-03T00:00:00Z"), "grade": "A", "score": 13}], "name": "Irish Pub", "restaurant_id": "40365500"}
{
  "_id": ObjectId("620512b61ed5f449fb72d3"), "address": {"building": "21", "coord": [-73.91684240, 40.8481362], "street": "East 170 Street", "zipcode": "10452"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-12-16T00:00:00Z"), "grade": "B", "score": 22}, {"date": ISODate("2013-08-29T00:00:00Z"), "grade": "B", "score": 22}, {"date": ISODate("2012-08-29T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2012-02-13T00:00:00Z"), "grade": "B", "score": 16}], "name": "McWayers Pub", "restaurant_id": "40365592"}
{
  "_id": ObjectId("620512b61ed5f449fb72d3"), "address": {"building": "21", "coord": [-73.91684240, 40.8481362], "street": "East 170 Street", "zipcode": "10452"}, "borough": "Bronx", "cuisine": "American", "grades": [{"date": ISODate("2014-12-16T00:00:00Z"), "grade": "B", "score": 22}, {"date": ISODate("2013-08-29T00:00:00Z"), "grade": "B", "score": 22}, {"date": ISODate("2012-08-29T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2012-02-13T00:00:00Z"), "grade": "B", "score": 16}], "name": "Hunchtime", "restaurant_id": "40365748"}
```

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

```
Command Prompt - mongo

> db.addresses.find({grades:[{"score":{$gt:90}}]})

{ "_id" : ObjectId("6052b161edsf14af07fb2930"), "address" : { "building" : "65", "coord" : [ -73.978272540, 47.624022 ], "street" : "West 54 Street", "zipcode" : "10019", "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 95 }, { "date" : ISODate("2013-04-15T00:00:00Z"), "grade" : "A", "score" : 91 }, { "date" : ISODate("2013-05-01T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-09T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "Murals On 54/Randolph's", "restaurant_id" : "40372466" },
{ "_id" : ObjectId("6052b161edsf14af07fb2931"), "address" : { "building" : "345", "coord" : [ -73.98646240, 46.7226593 ], "street" : "East 6 Street", "zipcode" : "10003", "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 85 }, { "date" : ISODate("2014-04-15T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2011-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 } ], "name" : "Gandhi", "restaurant_id" : "40381295" },
{ "_id" : ObjectId("6052b161edsf14af07fb2931"), "address" : { "building" : "130", "coord" : [ -73.98475840, 47.4579393 ], "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" }
```

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

```
Command Prompt - mongo
> db.addresses.find({grades:{$elemMatch:{score:{$gt:90}}}})
{
  "_id": ObjectId("620512b1ed51f4a97fb2800"),
  "address": { "building": "65", "coord": [-73.982725, 40.762402], "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("2012-04-10T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Murals On 54/Randolph's", "restaurant_id": "40372465" },
  "_id": ObjectId("620512b1ed51f4a97fb2931"),
  "address": { "building": "345", "coord": [-73.986462, 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": "C", "score": 92 }, { "date": ISODate("2011-11-03T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Gandhi", "restaurant_id": "40381295" },
  "_id": ObjectId("620512b1ed51f4a97fb2a0d"),
  "address": { "building": "130", "coord": [-73.984758, 40.7457939], "street": "Madison Avenue", "zipcode": "10015", "borough": "Manhattan", "cuisine": "PizzaItalian", "grades": [ { "date": ISODate("2014-12-24T00:00:00Z"), "grade": "Z", "score": 31 } ], "name": "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.

```
Command Prompt - mongo
> db.addresses.find({address.coord:{$lt:-95.754168}})
{
  "_id": ObjectId("620512b1ed51f4a97fb3054"),
  "address": { "building": "3707", "coord": [-101.8945214, 33.51974074], "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": 13 } ], "name": "Address 3707", "restaurant_id": "40381295" },
  "_id": ObjectId("620512b1ed51f4a97fb30e5"),
  "address": { "building": "15259", "coord": [-119.6368672, 36.2504998], "street": "10 Avenue", "zipcode": "11357", "borough": "Queens", "cuisine": "Italian", "grades": [ { "date": ISODate("2014-09-04T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2014-03-26T00:00:00Z"), "grade": "A", "score": 8 }, { "date": ISODate("2013-03-04T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-09-27T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2012-04-20T00:00:00Z"), "grade": "A", "score": 7 }, { "date": ISODate("2011-11-23T00:00:00Z"), "grade": "C", "score": 34 } ], "name": "Cascarino'S", "restaurant_id": "40668683" },
  "_id": ObjectId("620512b1ed51f4a97fb3589"),
  "address": { "building": "60", "coord": [-111.9975285, 42.0970258], "street": "West Side Highway", "zipcode": "10006", "borough": "Manhattan", "cuisine": "Japanese", "grades": [ { "date": ISODate("2014-03-20T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2013-06-28T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2012-07-27T00:00:00Z"), "grade": "A", "score": 2 } ], "name": "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant_id": "40882356" }
}
```

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.

```
Command Prompt - mongo
> db.addresses.find({$and:[{cuisine:{$ne:"American"}}, {"grades.score":{$gt:70}}, {"address.coord":{$lt:-65.754168}}]})
uncaught exception: SyntaxError: expected property name, got '['
@she11:1:51
> db.addresses.find({$and:[
... {cuisine:{$ne:"American"}},
... {"grades.score":{$gt:70}},
... {"address.coord":{$lt:-65.75168}}
... ]})
{
  "_id": ObjectId("620512b1ed51f4a97fb2727240.764022"),
  "address": { "building": "65", "coord": [-73.9782727240.764022], "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("2012-10-15T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2011-10-19T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Murals On 54/Randolph's", "restaurant_id": "40372465" },
  "_id": ObjectId("620512b1ed51f4a97fb2f50a51e"),
  "address": { "building": "345", "coord": [-73.986462, 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": "C", "score": 92 }, { "date": ISODate("2011-11-03T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Gandhi", "restaurant_id": "40381295" },
  "_id": ObjectId("620512b1ed51f4a97fb2f50a5ac"),
  "address": { "building": "14", "coord": [-73.983909, 40.740735],
  "street": "Eas 23 Street", "zipcode": "10010", "borough": "Manhattan", "cuisine": "American", "grades": [ { "date": ISODate("2014-10-28T00:00:00Z"), "grade": "A", "score": 9 }, { "date": ISODate("2014-09-04T00:00:00Z"), "grade": "A", "score": 8 }, { "date": ISODate("2013-08-06T00:00:00Z"), "grade": "C", "score": 58 }, { "date": ISODate("2013-02-14T00:00:00Z"), "grade": "A", "score": 13 } ], "name": "Li ve Bait & Restaurant", "restaurant_id": "40387237" },
  "_id": ObjectId("620512b1ed51f4a97fb2f50a67e"),
  "address": { "building": "130", "coord": [-73.984758, 40.7457939], "street": "Madison Avenue", "zipcode": "10016", "borough": "Manhattan", "cuisine": "PizzaItalian", "grades": [ { "date": ISODate("2014-12-24T00:00:00Z"), "grade": "Z", "score": 31 } ], "name": "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": "El Molino Rojo Restaurant", "restaurant_id": "40393488" }
}
```

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.

```
Command Prompt - mongo

> db.addresses.find({$and: [{cuisine:'American'}], "grades.score":{$gt:70}}, {"address.coord":0,$
  _id:0,taxi_id:0}).pretty()
{
  "_id": "0250510e26231bd2bfef0487",
  "address": {
    "building": "65",
    "coord": [-73.92827225
  40.763407],
    "street": "West 54 Street",
    "zipcode": "10019",
    "borough": "Manhattan",
    "cuisine": "American"
  },
  "grades": [
    {
      "date": ISODate("2014-08-22T00:00:00Z"),
      "grade": "C",
      "score": 131
    },
    {
      "date": ISODate("2013-09-25T00:00:00Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2014-03-08T00:00:00Z"),
      "grade": "A",
      "score": 11
    },
    {
      "date": ISODate("2010-11-19T00:00:00Z"),
      "grade": "A",
      "score": 11
    }
  ],
  "score": 25,
  "taxi_id": "0250510e26231bd2bfef0487"
},
{
  "_id": "0250510e26231bd2bfef051e",
  "address": {
    "building": "345",
    "coord": [-73.986462
  40.7266739],
    "street": "East 6 Street",
    "zipcode": "10003",
    "borough": "Manhattan",
    "cuisine": "Indian"
  },
  "grades": [
    {
      "date": ISODate("2014-08-22T00:00:00Z"),
      "grade": "P",
      "score": 5
    },
    {
      "date": ISODate("2014-01-14T00:00:00Z"),
      "grade": "A",
      "score": 8
    },
    {
      "date": ISODate("2013-04-24T00:00:00Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2012-10-18T00:00:00Z"),
      "grade": "P",
      "score": 41
    }
  ],
  "score": 17,
  "taxi_id": "0250510e26231bd2bfef051e"
},
{
  "_id": "0250510e26231bd2bfef05ac",
  "address": {
    "building": "14",
    "coord": [-73.9838909
  40.740735],
    "street": "East 23 Street",
    "zipcode": "10016",
    "borough": "Manhattan",
    "cuisine": "American"
  },
  "grades": [
    {
      "date": ISODate("2014-10-28T00:00:00Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2013-12-01T00:00:00Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-06-15T00:00:00Z"),
      "grade": "C",
      "score": 41
    }
  ],
  "score": 25,
  "taxi_id": "0250510e26231bd2bfef05ac"
},
{
  "_id": "0250510e26231bd2bfef05a67e",
  "address": {
    "building": "13B",
    "coord": [-73.984758
  40.745795],
    "street": "Madison Avenue",
    "zipcode": "10016",
    "borough": "Manhattan",
    "cuisine": "Pizza Italian"
  },
  "grades": [
    {
      "date": ISODate("2014-12-24T00:00:00Z"),
      "grade": "Z",
      "score": 1
    },
    {
      "date": ISODate("2014-06-17T00:00:00Z"),
      "grade": "C",
      "score": 98
    },
    {
      "date": ISODate("2013-05-22T00:00:00Z"),
      "grade": "B",
      "score": 21
    },
    {
      "date": ISODate("2012-05-02T00:00:00Z"),
      "grade": "A",
      "score": 11
    }
  ],
  "score": 98,
  "taxi_id": "0250510e26231bd2bfef05a67e"
},
{
  "_id": "0250510e26231bd2bfef05a67f",
  "address": {
    "building": "101",
    "coord": [-73.9242306
  40.873679],
    "street": "East 101 Street",
    "zipcode": "10016",
    "borough": "Bronx",
    "cuisine": "Latin (Cuban, Dominican, Puerto Rican, South & Central American)"
  },
  "grades": [
    {
      "date": ISODate("2013-10-01T00:00:00Z"),
      "grade": "A",
      "score": 10
    },
    {
      "date": ISODate("2013-03-10T00:00:00Z"),
      "grade": "B",
      "score": 25
    },
    {
      "date": ISODate("2012-10-25T00:00:00Z"),
      "grade": "A",
      "score": 12
    },
    {
      "date": ISODate("2012-05-25T00:00:00Z"),
      "grade": "B",
      "score": 18
    },
    {
      "date": ISODate("2011-09-14T00:00:00Z"),
      "grade": "B",
      "score": 19
    },
    {
      "date": ISODate("2011-04-25T00:00:00Z"),
      "grade": "C",
      "score": 76
    }
  ],
  "score": 99.8868,
  "taxi_id": "0250510e26231bd2bfef05a67f"
},
{
  "_id": "0250510e26231bd2bfef05a67g",
  "address": {
    "building": "293",
    "coord": [-73.946023
  40.7999999],
    "street": "Brooklyn Avenue",
    "zipcode": "11211",
    "borough": "Brooklyn",
    "cuisine": "Bakery"
  },
  "grades": [
    {
      "date": ISODate("2014-03-19T00:00:00Z"),
      "grade": "A",
      "score": 18
    },
    {
      "date": ISODate("2013-10-19T00:00:00Z"),
      "grade": "A",
      "score": 13
    },
    {
      "date": ISODate("2012-11-19T00:00:00Z"),
      "grade": "A",
      "score": 9
    },
    {
      "date": ISODate("2011-09-27T00:00:00Z"),
      "grade": "A",
      "score": 8
    }
  ],
  "score": 18,
  "taxi_id": "0250510e26231bd2bfef05a67g"
}
```

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.

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.

```
 1 Command Prompt - mongo
 2
 3 > db.addresses.find({name:"Wil's"}, {restaurant_id:1, name:1, borough:1, cuisine:1})
 4   "_id" : ObjectId("4205310e02631db2ef50a326"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken's Fine Food"
 5   "_id" : ObjectId("4205310e02631db2ef50a329"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wil's"
 6   "_id" : ObjectId("d6205310e02631db2ef50b12e"), "borough" : "Bronx", "cuisine" : "Pizza", "name" : "Wilb
 7 el Pizza"
 8
```

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.

```
PS C:\Users\user\Documents\GitHub\Node.js\RestaurantApp> mongo
MongoDB shell version: 4.2.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 4.2.0
Welcome to the MongoDB shell.
For usage details type "help".
For built-in documentation type "docs".
For more complete documentation see https://www.mongodb.com/docs/manual/
Type "quit" to exit.
> db.addresses.find({name:"Wil's"}, {restaurant_id:1, name:1, borough:1, cuisine:1})
[{"_id": ObjectId("6205310e02631db2ef50a326"), "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Wil's - Wilens Food Fund"}, {"_id": ObjectId("6205310e02631db2ef50a329"), "borough": "Bronx", "cuisine": "American", "name": "Wild Asia"}, {"_id": ObjectId("6205310e02631db2ef50b12e"), "borough": "Bronx", "cuisine": "Pizza", "name": "Wilbri Pizza"}]
```

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.Command Prompt - mongo
> db.addresses.find({ "name": { "$regex": "/Regina/"}}, {restaurant_id:1,name:1,borough:1,cuisine:1})
[{"_id": ObjectId("6205310e2631bd2fe50a32b"), "borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356649"}, {"_id": ObjectId("6205310e2631bd2fe50a426"), "borough": "Manhattan", "cuisine": "Café/Coffee/Tea", "name": "Caffè Reggio", "restaurant_id": "40364818"}, {"_id": ObjectId("6205310e2631bd2fe50a536"), "borough": "Manhattan", "cuisine": "American", "name": "Regency Hotel", "restaurant_id": "40382679"}, {"_id": ObjectId("6205310e2631bd2fe50a934"), "borough": "Manhattan", "cuisine": "American", "name": "Regency Whist Club", "restaurant_id": "40402377"}, {"_id": ObjectId("6205310e2631bd2fe50a934"), "borough": "Queens", "cuisine": "American", "name": "Rego Park Café", "restaurant_id": "40523342"}, {"_id": ObjectId("6205310e2631bd2fe50a98f"), "borough": "Queens", "cuisine": "Pizza", "name": "Regina Pizza", "restaurant_id": "40801325"}, {"_id": ObjectId("6205310e2631bd2fe50b1ba"), "borough": "Manhattan", "cuisine": "American", "name": "Regal Entertainment Group", "restaurant_id": "40891782"}]
```

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

```
Command Prompt - mongo
> db.addresses.find({borough:"Bronx"},{son:[{"cuisine":"American"}, {"cuisine":"Chinese"}]})

{
  "_id": ObjectId("6205310e02631db2e5fa344"),
  "address": { "building": "1236", "coord": [ -73.88933654, 40.8137617999999 ] },
  "street": "238 Spofford Ave.", "zipcode": "10474",
  "cuisine": "Chinese", "grades": [ { "date": ISODate("2013-12-30T00:00:00Z"), "grade": "A", "score": 8 },
    { "date": ISODate("2013-01-08T00:00:00Z"), "grade": "A", "score": 10 },
    { "date": ISODate("2012-06-12T00:00:00Z"), "grade": "A", "score": 15 } ],
  "name": "Happy Garden", "restaurant_id": "40363289"
}

{
  "_id": ObjectId("6205310e02631db2e5fa354"),
  "address": { "building": "277", "coord": [ -73.8941893, 40.8634684 ] },
  "street": "East Kingsbridge Road", "zipcode": "10458",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-03-03T00:00:00Z"), "grade": "A", "score": 10 },
    { "date": ISODate("2011-09-26T00:00:00Z"), "grade": "A", "score": 15 } ],
  "name": "Happy Garden", "restaurant_id": "40364296"
}

{
  "_id": ObjectId("6205310e02631db2e5fa80d2"),
  "address": { "building": "2459", "coord": [ -73.89299629, 40.8601686 ] },
  "street": "Webster Avenue", "zipcode": "10458",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-01-28T00:00:00Z"), "grade": "A", "score": 9 },
    { "date": ISODate("2014-01-07T00:00:00Z"), "grade": "A", "score": 9 } ],
  "name": "Wok II", "restaurant_id": "40510328"
}

{
  "_id": ObjectId("6205310e02631db2e5fa954"),
  "address": { "building": "2102", "coord": [ -73.8266969999999, 40.8687177 ] },
  "street": "Barlow Ave.", "zipcode": "10475",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-08-04T00:00:00Z"), "grade": "A", "score": 7 },
    { "date": ISODate("2012-04-13T00:00:00Z"), "grade": "B", "score": 17 } ],
  "name": "Dragon City", "restaurant_id": "40529205"
}

{
  "_id": ObjectId("73-9100876_40.886044"),
  "address": { "building": "5511", "coord": [ -73.9100876, 40.886044 ] },
  "street": "Westchester Avenue", "zipcode": "10461",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-07-27T00:00:00Z"), "grade": "A", "score": 5 },
    { "date": ISODate("2013-11-06T00:00:00Z"), "grade": "A", "score": 8 } ],
  "name": "Hunan Balcony", "restaurant_id": "40519967"
}

{
  "_id": ObjectId("6205310e02631db2e5fa0aa8"),
  "address": { "building": "332", "coord": [ -73.88312840, 40.85533 ] },
  "street": "East 187 Street", "zipcode": "10458",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-11-07T00:00:00Z"), "grade": "A", "score": 9 },
    { "date": ISODate("2014-05-15T00:00:00Z"), "grade": "A", "score": 9 } ],
  "name": "Great Wall Restaurant", "restaurant_id": "4052226"
}

{
  "_id": ObjectId("6205310e02631db2e5fa097"),
  "address": { "building": "202", "coord": [ -73.9164584, 40.8341845 ] },
  "street": "East 167 Street", "zipcode": "10456",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2013-12-12T00:00:00Z"), "grade": "A", "score": 9 },
    { "date": ISODate("2012-06-20T00:00:00Z"), "grade": "B", "score": 17 } ],
  "name": "Lucky Ho Restaurant", "restaurant_id": "4051226"
}

{
  "_id": ObjectId("6205310e02631db2e5fa0af"),
  "address": { "building": "1474", "coord": [ -73.8812272999999, 40.8281391 ] },
  "street": "Westchester Avenue", "zipcode": "10472",
  "borough": "Bronx", "cuisine": "Chinese", "grades": [ { "date": ISODate("2014-12-24T00:00:00Z"), "grade": "A", "score": 19 } ],
  "name": "Taste The Exotics"
}
```

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 Bronx or Brooklyn.

```
Command Prompt - mongo
> db.addresses.find({borough:$in:["Staten Island","Queens","Bronx","Brooklyn"]},{restaurant_id:1,name:1,borough:1,cuisine:1})
{
  "_id": ObjectId("6205310e02631db2e5fa322"),
  "borough": "Queens", "cuisine": "American", "name": "Brunos On The Boulevard", "restaurant_id": "40356151"
}
{
  "_id": ObjectId("6205310e02631db2e5fa323"),
  "borough": "Brooklyn", "cuisine": "Hamburgers", "name": "Mendy's", "restaurant_id": "50112340"
}
{
  "_id": ObjectId("6205310e02631db2e5fa324"),
  "borough": "Brooklyn", "cuisine": "American", "name": "Riviera Caffere", "restaurant_id": "40356048"
}
{
  "_id": ObjectId("6205310e02631db2e5fa325"),
  "borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop", "restaurant_id": "40357445"
}
{
  "_id": ObjectId("6205310e02631db2e5fa326"),
  "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Wilken's Fine Food", "restaurant_id": "40356483"
}
{
  "_id": ObjectId("6205310e02631db2e5fa328"),
  "borough": "Staten Island", "cuisine": "Jewish/Kosher", "name": "Kosher Island", "restaurant_id": "40356442"
}
{
  "_id": ObjectId("6205310e02631db2e5fa329"),
  "borough": "Bronx", "cuisine": "American", "name": "Old Asia", "restaurant_id": "40357213"
}
{
  "_id": ObjectId("6205310e02631db2e5fa32a"),
  "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Taste The Exotics Ice Cream", "restaurant_id": "40356731"
}
{
  "_id": ObjectId("6205310e02631db2e5fa32b"),
  "borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356640"
}
{
  "_id": ObjectId("6205310e02631db2e5fa32c"),
  "borough": "Brooklyn", "cuisine": "American", "name": "C & Catering Service", "restaurant_id": "40357437"
}
{
  "_id": ObjectId("6205310e02631db2e5fa32d"),
  "borough": "Brooklyn", "cuisine": "Jewish/Kosher", "name": "Seoul Foods", "restaurant_id": "40360454"
}
{
  "_id": ObjectId("6205310e02631db2e5fa32f"),
  "borough": "Brooklyn", "cuisine": "Chinese", "name": "May Bay Kitchen", "restaurant_id": "40356842"
}
{
  "_id": ObjectId("6205310e02631db2e5fa330"),
  "borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Joy Kosher Kitchen", "restaurant_id": "40356869"
}
{
  "_id": ObjectId("6205310e02631db2e5fa331"),
  "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40360076"
}
{
  "_id": ObjectId("6205310e02631db2e5fa332"),
  "borough": "Queens", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40361322"
}
{
  "_id": ObjectId("6205310e02631db2e5fa333"),
  "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Mordis Delicacies", "restaurant_id": "40361390"
}
{
  "_id": ObjectId("6205310e02631db2e5fa335"),
  "borough": "Brooklyn", "cuisine": "American", "name": "The Movida Feijoada", "restaurant_id": "40361566"
}
{
  "_id": ObjectId("6205310e02631db2e5fa336"),
  "borough": "Queens", "cuisine": "Delicatessen", "name": "Sal's Deli", "restaurant_id": "40361618"
}
{
  "_id": ObjectId("6205310e02631db2e5fa337"),
  "borough": "Queens", "cuisine": "Delicatessen", "name": "Steve Chu's Deli & Grocery", "restaurant_id": "40361998"
}
{
  "_id": ObjectId("6205310e02631db2e5fa33c"),
  "borough": "Brooklyn", "cuisine": "Hamburgers", "name": "White Castle", "restaurant_id": "40362344"
}
Type "it" for more
>
```

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 Bronx or Brooklyn.

```
[-] Command Prompt - mongo
> db.addresses.find({borough:["Staten Island","Queens","Bronx","Brooklyn"]},{restaurant_id:1,name:1,borough:1,cuisine:1})
{
  "_id": ObjectId("6205310e02631db2ef50a327"),
  "borough": "Manhattan",
  "cuisine": "Irish",
  "name": "Dj Reynolds Pub And Restaurant",
  "restaurant_id": "30191841"
},
{
  "_id": ObjectId("6205310e02631db2ef50a32d"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "1 East 66th Street Kitchen",
  "restaurant_id": "40359480"
},
{
  "_id": ObjectId("6205310e02631db2ef50a330"),
  "borough": "Manhattan",
  "cuisine": "Delicatessen",
  "name": "The Corner Deli",
  "restaurant_id": "40362036"
},
{
  "_id": ObjectId("6205310e02631db2ef50a338"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Glorious Food",
  "restaurant_id": "4036121"
},
{
  "_id": ObjectId("6205310e02631db2ef50a339"),
  "borough": "Manhattan",
  "cuisine": "Chicken",
  "name": "Harriet's Kitchen",
  "restaurant_id": "40362098"
},
{
  "_id": ObjectId("6205310e02631db2ef50a33a"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "P & S Deli Grocery",
  "restaurant_id": "40362264"
},
{
  "_id": ObjectId("6205310e02631db2ef50a33b"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Angelina Film Center",
  "restaurant_id": "40362274"
},
{
  "_id": ObjectId("6205310e02631db2ef50a33f"),
  "borough": "Manhattan",
  "cuisine": "Turkish",
  "name": "The Corner Cafe",
  "restaurant_id": "40362304"
},
{
  "_id": ObjectId("6205310e02631db2ef50a340"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Downtown Deli",
  "restaurant_id": "40363021"
},
{
  "_id": ObjectId("6205310e02631db2ef50a343"),
  "borough": "Manhattan",
  "cuisine": "Bakery",
  "name": "Olive'S",
  "restaurant_id": "40363151"
},
{
  "_id": ObjectId("6205310e02631db2ef50a346"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Cafe Metro",
  "restaurant_id": "40363208"
},
{
  "_id": ObjectId("6205310e02631db2ef50a348"),
  "borough": "Manhattan",
  "cuisine": "Sandwiches/Salads/Mixed Buffet",
  "name": "The TexMex Deli",
  "restaurant_id": "40363426"
},
{
  "_id": ObjectId("6205310e02631db2ef50a34c"),
  "borough": "Manhattan",
  "cuisine": "Continental",
  "name": "Lombardi & Maria",
  "restaurant_id": "40363630"
},
{
  "_id": ObjectId("6205310e02631db2ef50a34e"),
  "borough": "Manhattan",
  "cuisine": "Pizza",
  "name": "Domino'S Pizza",
  "restaurant_id": "40363644"
},
{
  "_id": ObjectId("6205310e02631db2ef50a34f"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Berkely",
  "restaurant_id": "40363685"
},
{
  "_id": ObjectId("6205310e02631db2ef50a351"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Spoon Bread Catering",
  "restaurant_id": "40364179"
},
{
  "_id": ObjectId("6205310e02631db2ef50a355"),
  "borough": "Manhattan",
  "cuisine": "Chicken",
  "name": "Texas Rotisserie",
  "restaurant_id": "40364304"
},
{
  "_id": ObjectId("6205310e02631db2ef50a357"),
  "borough": "Manhattan",
  "cuisine": "Pizza",
  "name": "Domino'S Pizza",
  "restaurant_id": "40363945"
},
{
  "_id": ObjectId("6205310e02631db2ef50a359"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "21 Club",
  "restaurant_id": "40364362"
},
{
  "_id": ObjectId("6205310e02631db2ef50a35b"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Palm Restaurant",
  "restaurant_id": "40364355"
}
}
Type "it" for more
>
```

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.

```
[-] Command Prompt - mongo
> db.addresses.find({grades.score:{$not:{$gt:10}}},{restaurant_id:1,name:1,borough:1,cuisine:1})
{
  "_id": ObjectId("6205310e02631db2ef50a32c"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "C & C Catering Service",
  "restaurant_id": "40357437"
},
{
  "_id": ObjectId("6205310e02631db2ef50a32d"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "1 East 66th Street Kitchen",
  "restaurant_id": "40359480"
},
{
  "_id": ObjectId("6205310e02631db2ef50a333"),
  "borough": "Brooklyn",
  "cuisine": "Delicatessen",
  "name": "Nordic Delicacies",
  "restaurant_id": "40361390"
},
{
  "_id": ObjectId("6205310e02631db2ef50a33c"),
  "borough": "Brooklyn",
  "cuisine": "Hamburgers",
  "name": "White Castle",
  "restaurant_id": "40362344"
},
{
  "_id": ObjectId("6205310e02631db2ef50a34f"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "Sonny'S Heroes",
  "restaurant_id": "40363744"
},
{
  "_id": ObjectId("6205310e02631db2ef50a35a"),
  "borough": "Bronx",
  "cuisine": "American",
  "name": "Manhem Club",
  "restaurant_id": "40364363"
},
{
  "_id": ObjectId("6205310e02631db2ef50a368"),
  "borough": "Staten Island",
  "cuisine": "American",
  "name": "Great Kill Yacht Club",
  "restaurant_id": "40364610"
},
{
  "_id": ObjectId("6205310e02631db2ef50a371"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "Peridropity 3",
  "restaurant_id": "40364640"
},
{
  "_id": ObjectId("6205310e02631db2ef50a372"),
  "borough": "Manhattan",
  "cuisine": "American",
  "name": "White Horse Tavern",
  "restaurant_id": "40364658"
},
{
  "_id": ObjectId("6205310e02631db2ef50a37e"),
  "borough": "Manhattan",
  "cuisine": "Irish",
  "name": "Dorrian'S Red Restaurant",
  "restaurant_id": "40365239"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3cc"),
  "borough": "Manhattan",
  "cuisine": "Mexican",
  "name": "Mexico Lindo Restaurant",
  "restaurant_id": "40367038"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3f4"),
  "borough": "Brooklyn",
  "cuisine": "Greek",
  "name": "E Grecos Diner",
  "restaurant_id": "40367795"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3f6"),
  "borough": "Bronx",
  "cuisine": "Not Listed/Not Applicable",
  "name": "The Latin Nest",
  "restaurant_id": "40367946"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3f7"),
  "borough": "Bronx",
  "cuisine": "African",
  "name": "African Tarragon",
  "restaurant_id": "40368011"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3f9"),
  "borough": "Bronx",
  "cuisine": "American",
  "name": "African Market (Baboon Cafe)",
  "restaurant_id": "40368026"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3fa"),
  "borough": "Staten Island",
  "cuisine": "Italian",
  "name": "Roadhouse Restaurant",
  "restaurant_id": "40368034"
},
{
  "_id": ObjectId("6205310e02631db2ef50a3fb"),
  "borough": "Bronx",
  "cuisine": "Cafe/Coffee/Tea",
  "name": "Terrace Cafe",
  "restaurant_id": "40368018"
},
{
  "_id": ObjectId("6205310e02631db2ef50a41a"),
  "borough": "Manhattan",
  "cuisine": "French",
  "name": "Pergola Des Arretes",
  "restaurant_id": "40368039"
},
{
  "_id": ObjectId("6205310e02631db2ef50a427"),
  "borough": "Brooklyn",
  "cuisine": "Hamburgers",
  "name": "McDonald'S",
  "restaurant_id": "40369357"
},
{
  "_id": ObjectId("6205310e02631db2ef50a42a"),
  "borough": "Brooklyn",
  "cuisine": "American",
  "name": "The River Cafe",
  "restaurant_id": "40369608"
}
}
Type "it" for more
>
```

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'.

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..

```
Command Prompt - mongo
db.addresses.find({ "grades.grade": { "$eq": "A" }, "grades.score": { "$gt": 11 } })
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2014-08-11T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2015-01-22T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Neary's Pub", "restaurant_id": "40365871" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2014-08-11T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2013-12-10T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Don Filippo Restaurant", "restaurant_id": "40372417" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2013-08-29T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2010-09-13T00:00:00Z"), "grade": "A", "score": 12 } ], "name": "Muslim Sausage", "restaurant_id": "40393707" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2015-01-12T00:00:00Z"), "grade": "A", "score": 10 }, { "date": ISODate("2014-08-11T00:00:00Z"), "grade": "A", "score": 10 } ], "name": "Claro Macanudo (Cigar Bar)", "restaurant_id": "40526046" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2014-01-14T00:00:00Z"), "grade": "A", "score": 13 }, { "date": ISODate("2012-04-16T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Marino's Pizza & Restaurant", "restaurant_id": "40569917" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2011-11-23T00:00:00Z"), "grade": "A", "score": 9 } ], "name": "Gene's Coffee Shop", "restaurant_id": "40614916" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2014-03-04T00:00:00Z"), "grade": "B", "score": 22 }, { "date": ISODate("2013-09-13T00:00:00Z"), "grade": "B", "score": 5 } ], "name": "Union Cafe Restaurant", "restaurant_id": "40698823" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2014-08-11T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2013-08-28T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "Jojo's Pizza", "restaurant_id": "40892915" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2013-07-22T00:00:00Z"), "grade": "A", "score": 12 }, { "date": ISODate("2012-02-07T00:00:00Z"), "grade": "A", "score": 11 } ], "name": "The Fisherman", "restaurant_id": "40900001" }
{ "_id": ObjectId("40365310e026216d2bf2e0475"), "grades": [ { "date": ISODate("2012-09-05T00:00:00Z"), "grade": "A", "score": 11 }, { "date": ISODate("2011-10-12T00:00:00Z"), "grade": "A", "score": 7 } ], "name": "The Fisherman", "restaurant_id": "40900001" }
```

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"

```
Command Prompt - mongo
db.addresses.find({ "grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade": "A", "grades.lscore": 9, "name": "Club Macanudo (Cigar Bar)", "restaurant_id": "40526406" })
[{"_id": ObjectId("6205310e2631db2f50a940"), "grades": [{"date": ISODate("2015-01-12T00:00:00Z"), "grade": "A", "score": 10}, {"date": ISODate("2014-08-11T00:00:00Z"), "grade": "A", "score": 9}], "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..

```

Command Prompt - mongo
> db.addresses.find({"address.coord":{$gt:42,$lt:52}}, {"restaurant_id":1,"name":1,"coord":1})

```

The output shows several documents, each containing the address, name, and coordinates of a restaurant. The names include "T.G.I. Friday's", "T.G.I. Fridays", "Diluvio's Deli", "La Caridad 78", "Bijan's", "Hyatt, NY Central/Room Service", and "Sports Center At Chelsea Piers (Sushi Bar)".

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

```

Command Prompt - mongo
> db.addresses.find().sort({name:1})

```

The output shows the same set of documents as the previous query, but they are now sorted by the 'name' field in ascending order. The names are: Bijan's, Hyatt, NY Central/Room Service, La Caridad 78, Sports Center At Chelsea Piers (Sushi Bar), T.G.I. Friday's, and T.G.I. Fridays.

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

```

Command Prompt - mongo
> db.addresses.find().sort({name:-1})

```

The output shows the same set of documents as the previous queries, but they are now sorted by the 'name' field in descending order. The names are: T.G.I. Friday's, T.G.I. Fridays, Diluvio's Deli, La Caridad 78, Bijan's, and Hyatt, NY Central/Room Service.

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.

```
Command Prompt - mongo
> db.addresses.find().sort({cuisine:1,borough:-1})
{
  "_id": ObjectId("6205310e02631db2ef50ab0c"),
  "address": {
    "building": "1345",
    "coord": [-73.959249, 40.768076],
    "street": "2 Avenue",
    "zipcode": "10021",
    "borough": "Manhattan",
    "cuisine": "Afghan",
    "grades": [
      {
        "date": ISODate("2014-10-07T00:00:00Z"),
        "grade": "A",
        "score": 8
      },
      {
        "date": ISODate("2012-10-26T00:00:00Z"),
        "grade": "A",
        "score": 7
      },
      {
        "date": ISODate("2013-10-23T00:00:00Z"),
        "grade": "A",
        "score": 13
      },
      {
        "date": ISODate("2012-04-26T00:00:00Z"),
        "grade": "A",
        "score": 7
      },
      {
        "date": ISODate("2012-01-12T00:00:00Z"),
        "grade": "P",
        "score": 10
      }
    ],
    "name": "Afghan Kebab House",
    "restaurant_id": "40552806"
  },
  ...
}
```

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

```
Command Prompt - mongo
> db.addresses.find({address.street:{$exists:true}})
{
  "_id": ObjectId("6205310e02631db2ef50ab22"),
  "address": {
    "building": "8825",
    "coord": [-73.880381, 40.764124],
    "street": "Astoria Boulevard",
    "zipcode": "11360",
    "borough": "Queens",
    "cuisine": "American",
    "grades": [
      {
        "date": ISODate("2014-11-15T00:00:00Z"),
        "grade": "A",
        "score": 3
      },
      {
        "date": ISODate("2014-05-02T00:00:00Z"),
        "grade": "A",
        "score": 3
      },
      {
        "date": ISODate("2014-05-02T00:00:00Z"),
        "grade": "A",
        "score": 13
      },
      {
        "date": ISODate("2012-04-17T00:00:00Z"),
        "grade": "A",
        "score": 7
      },
      {
        "date": ISODate("2012-08-01T00:00:00Z"),
        "grade": "A",
        "score": 10
      },
      {
        "date": ISODate("2012-03-12T00:00:00Z"),
        "grade": "A",
        "score": 12
      },
      {
        "date": ISODate("2011-11-01T00:00:00Z"),
        "grade": "A",
        "score": 10
      }
    ],
    "name": "Liberia On The Boulevard",
    "restaurant_id": "40617699"
  },
  ...
}
```

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

```
Command Prompt - mongo

> db.addresses.find({ "address.coord": { $type: 1 } })
{
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "882",
    "coord": [-73.880838, 40.764312],
    "street": "Astoria Boulevard",
    "zipcode": "11309",
    "borough": "Queens",
    "cuisine": "American",
    "grades": [
      {
        "date": ISODate("2014-11-15T00:00:00Z"),
        "grade": "A",
        "score": 38
      },
      {
        "date": ISODate("2014-05-07T00:00:00Z"),
        "grade": "B",
        "score": 37
      }
    ],
    "name": "Brunch On The Boulevard",
    "restaurant_id": "4035615"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "469",
    "coord": [-73.961704, 40.669242],
    "street": "Flatbush Avenue",
    "zipcode": "11225",
    "borough": "Brooklyn",
    "cuisine": "Hamburgers",
    "grades": [
      {
        "date": ISODate("2014-12-30T00:00:00Z"),
        "grade": "B",
        "score": 8
      }
    ],
    "name": "Wendy's",
    "restaurant_id": "30112340"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "7808",
    "coord": [-73.88241, 40.75956],
    "street": "Prospect Avenue",
    "zipcode": "11224",
    "borough": "Brooklyn",
    "cuisine": "American",
    "grades": [
      {
        "date": ISODate("2013-06-10T00:00:00Z"),
        "grade": "A",
        "score": 7
      }
    ],
    "name": "Riviera Caterer",
    "restaurant_id": "4035618"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "1007",
    "coord": [-73.85670, 40.744842],
    "street": "Morris Park Ave",
    "zipcode": "10462",
    "borough": "Bronx",
    "cuisine": "Bakery",
    "grades": [
      {
        "date": ISODate("2014-03-03T00:00:00Z"),
        "grade": "A",
        "score": 2
      }
    ],
    "name": "Morris Park Bakery",
    "restaurant_id": "30070802"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "114",
    "coord": [-73.90685, 40.6199834],
    "street": "Avenue U",
    "zipcode": "11243",
    "borough": "Brooklyn",
    "cuisine": "Delicatessen",
    "grades": [
      {
        "date": ISODate("2014-05-29T00:00:00Z"),
        "grade": "A",
        "score": 10
      }
    ],
    "name": "The Deli",
    "restaurant_id": "30070802"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "351",
    "coord": [-73.98515, 40.7676919],
    "street": "West 57 Street",
    "zipcode": "10019",
    "borough": "Manhattan",
    "cuisine": "Irish",
    "grades": [
      {
        "date": ISODate("2014-01-14T00:00:00Z"),
        "grade": "A",
        "score": 10
      }
    ],
    "name": "Reynold's Pub And Restaurant",
    "restaurant_id": "30191841"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "200",
    "coord": [-74.33727, 40.61913],
    "street": "Victory Boulevard",
    "zipcode": "10314",
    "borough": "Queens",
    "cuisine": "Italian",
    "grades": [
      {
        "date": ISODate("2014-09-06T00:00:00Z"),
        "grade": "A",
        "score": 9
      }
    ],
    "name": "Kosher Island",
    "restaurant_id": "4035642"
  },
  "_id": ObjectId("6205310e2621dab2fe50a32"),
  "address": {
    "building": "2300",
    "coord": [-73.87861, 40.8528833],
    "street": "Southern Boulevard",
    "zipcode": "10460",
    "borough": "Bronx",
    "cuisine": "American",
    "grades": [
      {
        "date": ISODate("2014-04-28T00:00:00Z"),
        "grade": "A",
        "score": 11
      }
    ],
    "name": "Sous Chef"
  }
}
```

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.

```
Commands Prompt - mongo

> db.addresses.find({ "grades.score":{$mod:[7,0]}}, {"restaurant_id":1,"name":1,"grades":1})
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-11-15T00:00:00Z"),
      "grade": "7",
      "score": 38,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2014-05-02T00:00:00Z"),
      "grade": "A",
      "score": 10,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Brunos On The Boulevard",
  "restaurant_id": "4035601840156151"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-06-19T00:00:00Z"),
      "grade": "A",
      "score": 5,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2013-06-05T00:00:00Z"),
      "grade": "A",
      "score": 7,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Riviera Caterer",
  "restaurant_id": "4035601840156151"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-01-03T00:00:00Z"),
      "grade": "A",
      "score": 2,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2013-09-11T00:00:00Z"),
      "grade": "A",
      "score": 6,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Mona's Diner Take Away",
  "restaurant_id": "3007544540156151"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-05-07T00:00:00Z"),
      "grade": "A",
      "score": 4,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2012-04-30T00:00:00Z"),
      "grade": "A",
      "score": 6,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "1 East 66th Street Kitchen",
  "restaurant_id": "4035601840156151"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-11-10T00:00:00Z"),
      "grade": "A",
      "score": 11,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2013-10-10T00:00:00Z"),
      "grade": "A",
      "score": 10,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Soda Foods",
  "restaurant_id": "40360045"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-09-15T00:00:00Z"),
      "grade": "B",
      "score": 21,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2013-08-28T00:00:00Z"),
      "grade": "C",
      "score": 56,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Nay May Kitchen",
  "restaurant_id": "40358429"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a32f"),
  "grades": [
    {
      "date": ISODate("2014-08-16T00:00:00Z"),
      "grade": "A",
      "score": 9,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    },
    {
      "date": ISODate("2013-07-27T00:00:00Z"),
      "grade": "A",
      "score": 9,
      "restaurant_id": "6205310e0261d2bfef9a32f"
    }
  ],
  "name": "Sals Deli's Deli & Grocery",
  "restaurant_id": "40361618"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a337"),
  "grades": [
    {
      "date": ISODate("2014-03-19T00:00:00Z"),
      "grade": "A",
      "score": 12,
      "restaurant_id": "6205310e0261d2bfef9a337"
    },
    {
      "date": ISODate("2012-03-27T00:00:00Z"),
      "grade": "A",
      "score": 8,
      "restaurant_id": "6205310e0261d2bfef9a337"
    }
  ],
  "name": "Steve Chu's Deli & Grocery",
  "restaurant_id": "40361998"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a339"),
  "grades": [
    {
      "date": ISODate("2014-09-15T00:00:00Z"),
      "grade": "A",
      "score": 10,
      "restaurant_id": "6205310e0261d2bfef9a339"
    },
    {
      "date": ISODate("2014-03-04T00:00:00Z"),
      "grade": "A",
      "score": 13,
      "restaurant_id": "6205310e0261d2bfef9a339"
    }
  ],
  "name": "Harriet's Kitchen",
  "restaurant_id": "40362098"
},
{
  "_id": ObjectId("6205310e0261d2bfef9a33c"),
  "grades": [
    {
      "date": ISODate("2014-01-18T00:00:00Z"),
      "grade": "A",
      "score": 8,
      "restaurant_id": "6205310e0261d2bfef9a33c"
    },
    {
      "date": ISODate("2013-03-18T00:00:00Z"),
      "grade": "A",
      "score": 8,
      "restaurant_id": "6205310e0261d2bfef9a33c"
    }
  ],
  "name": "Lil's Deli"
}
```

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.

```
Command Prompt - mongo
> db.addresses.find({
...   {name:{$regex:"mon", $options:"i"}}, 
...   {borough:1}, {address.coord:1}, {cuisine:1})
{
  "_id": ObjectId("6205310e02631db2ef50a8bc"), "address": { "coord": [ -73.8221418, 40.7272376 ] }, "borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Shimons Kosher Pizza"
  "_id": ObjectId("6205310e02631db2ef50a8bf"), "address": { "coord": [ -73.98306999999999, 40.7441419 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Desmond'S Tavern"
  "_id": ObjectId("6205310e02631db2ef50a8c7"), "address": { "coord": [ -74.10465599999999, 40.58834 ] }, "borough": "Staten Island", "cuisine": "American", "name": "Richmond County Country Club"
  "_id": ObjectId("6205310e02631db2ef50a8d4"), "address": { "coord": [ -73.9812843, 40.5947365 ] }, "borough": "Queens", "cuisine": "Italian", "name": "Monte'S Italian Garden"
  "_id": ObjectId("6205310e02631db2ef50a8d9"), "address": { "coord": [ -73.951199, 40.7166026 ] }, "borough": "Brooklyn", "cuisine": "Italian", "name": "Bamonte'S Restaurant"
  "_id": ObjectId("6205310e02631db2ef50a8f1"), "address": { "coord": [ -73.924072, 40.761089000000001 ] }, "borough": "Queens", "cuisine": "Greek", "name": "Omonia Cafe"
  "_id": ObjectId("6205310e02631db2ef50a8ce"), "address": { "coord": [ -74.0023353, 40.7333573 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Manhattan Monster"
  "_id": ObjectId("6205310e02631db2ef50a8fc"), "address": { "coord": [ -74.001094, 40.729583 ] }, "borough": "Manhattan", "cuisine": "Italian", "name": "Monte'S"
  "_id": ObjectId("6205310e02631db2ef50a8f5"), "address": { "coord": [ -73.9901605, 40.7526176 ] }, "borough": "Queens", "cuisine": "American", "name": "Delamico'S Kitchen"
  "_id": ObjectId("6205310e02631db2ef50a8f8"), "address": { "coord": [ -73.9979536, 40.6914824 ] }, "borough": "Brooklyn", "cuisine": "Bottled beverages, including water, sodas, juices, etc.", "name": "Monte'S Bar & Grill"
  "_id": ObjectId("6205310e02631db2ef50a6a0"), "address": { "coord": [ -73.9707505, 40.7035651 ] }, "borough": "Manhattan", "cuisine": "Delicatessen", "name": "Delamico Gourmet"
  "_id": ObjectId("6205310e02631db2ef50a712"), "address": { "coord": [ -73.9766037, 40.7508686 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Delamico Gourmet"
  "_id": ObjectId("6205310e02631db2ef50a724"), "address": { "coord": [ -73.9780633, 40.7605759 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Montebello Restaurant"
  "_id": ObjectId("6205310e02631db2ef50a76d"), "address": { "coord": [ -74.1914659, 40.552743600000001 ] }, "borough": "Staten Island", "cuisine": "Italian", "name": "Villa Monte Pizzeria"
  "_id": ObjectId("6205310e02631db2ef50a867"), "address": { "coord": [ -73.97198299999999, 40.764464 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Harmonie Club"
  "_id": ObjectId("6205310e02631db2ef50a929"), "address": { "coord": [ -73.795719000000001, 40.7095637 ] }, "borough": "Queens", "cuisine": "Bakery", "name": "Ramon's Bakery"
  "_id": ObjectId("6205310e02631db2ef50a95f"), "address": { "coord": [ -73.978521000000001, 40.729192 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Mona'S"
  "_id": ObjectId("6205310e02631db2ef50a989"), "address": { "coord": [ -73.984634800000001, 40.7630755 ] }, "borough": "Brooklyn", "cuisine": "American", "name": "Casino Simon Theatre"
  "_id": ObjectId("6205310e02631db2ef50a9bb"), "address": { "coord": [ -73.9938361, 40.6991317 ] }, "borough": "Brooklyn", "cuisine": "American", "name": "Mondial Bakery"
  "_id": ObjectId("6205310e02631db2ef50a9e3"), "address": { "coord": [ -74.028486, 40.630438 ] }, "borough": "Brooklyn", "cuisine": "Mediterranean", "name": "Omonia Cafe"
}
Type "it" for more
```

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.

```
Command Prompt - mongo
> db.addresses.find({name:{$regex:"Mad|i_"},name:1,borough:1,address.coord:1,cuisine:1})
{
  "_id": ObjectId("6205310e02631db2ef50a879"), "address": { "coord": [ -73.9860597, 40.7431194 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Madison Square"
  "_id": ObjectId("6205310e02631db2ef50a881"), "address": { "coord": [ -73.98302199999999, 40.742313 ] }, "borough": "Manhattan", "cuisine": "Indian", "name": "Madras Mahal"
  "_id": ObjectId("6205310e02631db2ef50a8d8"), "address": { "coord": [ -74.000002, 40.72735 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Madame X"
  "_id": ObjectId("6205310e02631db2ef50a8c8"), "address": { "coord": [ -73.98171599999999, 40.7499406 ] }, "borough": "Manhattan", "cuisine": "French", "name": "Madison Bistro"
  "_id": ObjectId("6205310e02631db2ef50a8d23"), "address": { "coord": [ -73.9717845, 40.6897199 ] }, "borough": "Brooklyn", "cuisine": "African", "name": "Nadiba"
  "_id": ObjectId("6205310e02631db2ef50b015"), "address": { "coord": [ -73.9040745, 40.9069011 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Madison"
  "_id": ObjectId("6205310e02631db2ef50b001"), "address": { "coord": [ -73.9886598, 40.7565811 ] }, "borough": "Manhattan", "cuisine": "Hot Dogs", "name": "Madame Tussaud'S"
  "_id": ObjectId("6205310e02631db2ef50b0bcc"), "address": { "coord": [ -73.95623719999999, 40.7761697 ] }, "borough": "Manhattan", "cuisine": "American", "name": "Mad River Bar & Grille"
}
>
```

Happy Coding!!!