

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

```
> db.hotels.find().pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66e"),
  "address" : {
    "building" : "2780",
    "coord" : [
      -73.98241999999999,
      40.579505
    ],
    "street" : "Stillwell Avenue",
    "zipcode" : "11224"
  },
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-06-10T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-06-05T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-04-13T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2011-10-12T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Riviera Caterer",
  "restaurant_id" : "40356018"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66f"),
  "address" : {
    "building" : "351",
    "coord" : [
      -73.98513559999999,
      40.7676919
    ],
    "street" : "West 57 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "Irish",
  "grades" : [
    {
      "date" : ISODate("2014-09-06T00:00:00Z"),
```

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

```
> db.hotels.find({}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
{"_id": "ObjectID('60a36a1f4dae5b9d71ff683')", "borough": "Brooklyn", "cuisine": "American", "name": "Riviera Caterer", "restaurant_id": "40356018" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6e9')", "borough": "Manhattan", "cuisine": "Irish", "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f0')", "borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop", "restaurant_id": "30075445" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f1')", "borough": "Brooklyn", "cuisine": "Hamburgers", "name": "Wendy'S", "restaurant_id": "30112340" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f2')", "borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f3')", "borough": "Staten Island", "cuisine": "Jewish/Kosher", "name": "Kosher Island", "restaurant_id": "40356442" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f4')", "borough": "Queens", "cuisine": "American", "name": "Brunos On The Boulevard", "restaurant_id": "40356151" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f5')", "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Wilken'S Fine Food", "restaurant_id": "40356483" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f6')", "borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356649" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f7')", "borough": "Bronx", "cuisine": "American", "name": "Wild Asia", "restaurant_id": "40357217" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8')", "borough": "Manhattan", "cuisine": "American", "name": "1 East 66Th Street Kitchen", "restaurant_id": "40359480" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f9')", "borough": "Brooklyn", "cuisine": "Chinese", "name": "May May Kitchen", "restaurant_id": "40358429" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6fa')", "borough": "Brooklyn", "cuisine": "Jewish/Kosher", "name": "Seuda Foods", "restaurant_id": "40360045" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6fb')", "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40360076" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6fc')", "borough": "Queens", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40361322" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6fd')", "borough": "Manhattan", "cuisine": "American", "name": "Glorious Food", "restaurant_id": "40361521" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6fe')", "borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Nordic Delicacies", "restaurant_id": "40361390" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6ff')", "borough": "Queens", "cuisine": "Delicatessen", "name": "Sal'S Deli", "restaurant_id": "40361618" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8')", "borough": "Brooklyn", "cuisine": "American", "name": "The Movable Feast", "restaurant_id": "40361606" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f9')", "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731" }
Type "it" for more
> it
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f82')", "borough": "Brooklyn", "cuisine": "American", "name": "C & C Catering Service", "restaurant_id": "40357437" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f83')", "borough": "Queens", "cuisine": "Delicatessen", "name": "Steve Chu'S Deli & Grocery", "restaurant_id": "40361998" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f84')", "borough": "Manhattan", "cuisine": "Delicatessen", "name": "Bully'S Deli", "restaurant_id": "40361708" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f85')", "borough": "Manhattan", "cuisine": "Chicken", "name": "Harriet'S Kitchen", "restaurant_id": "40362098" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f86')", "borough": "Manhattan", "cuisine": "American", "name": "Angelika Film Center", "restaurant_id": "40362274" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f87')", "borough": "Manhattan", "cuisine": "American", "name": "P & S Deli Grocery", "restaurant_id": "40362264" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f88')", "borough": "Manhattan", "cuisine": "Turkish", "name": "The Country Cafe", "restaurant_id": "40362715" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f89')", "borough": "Brooklyn", "cuisine": "Hamburgers", "name": "White Castle", "restaurant_id": "40362344" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8a')", "borough": "Brooklyn", "cuisine": "Caribbean", "name": "Shashemene Int'L Restaura", "restaurant_id": "40362869" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8b')", "borough": "Queens", "cuisine": "Chinese", "name": "Ho Mei Restaurant", "restaurant_id": "40362432" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8c')", "borough": "Bronx", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40363093" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8d')", "borough": "Brooklyn", "cuisine": "Donuts", "name": "Dunkin' Donuts", "restaurant_id": "40363098" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8e')", "borough": "Manhattan", "cuisine": "American", "name": "Downtown Deli", "restaurant_id": "40363021" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f8f')", "borough": "Brooklyn", "cuisine": "American", "name": "Mejlander & Mulgannon", "restaurant_id": "40363117" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f90')", "borough": "Manhattan", "cuisine": "Bakery", "name": "Olive'S", "restaurant_id": "40363151" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f91')", "borough": "Bronx", "cuisine": "Chinese", "name": "Happy Garden", "restaurant_id": "40363289" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f92')", "borough": "Queens", "cuisine": "Delicatessen", "name": "Tony'S Deli", "restaurant_id": "40363333" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f93')", "borough": "Manhattan", "cuisine": "American", "name": "Cafe Metro", "restaurant_id": "40363298" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f94')", "borough": "Staten Island", "cuisine": "Delicatessen", "name": "Bagels N Buns", "restaurant_id": "40363427" }
{"_id": "ObjectID('60a36a1f4dae5b9d71ff6f95')", "borough": "Manhattan", "cuisine": "Sandwiches/Salads/Mixed Buffet", "name": "Lexler Deli", "restaurant_id": "40363426" }
Type "it" for more
>
```

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

```
> db.hotels.find({}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1, _id: 0})
{"borough": "Brooklyn", "cuisine": "American", "name": "Riviera Caterer", "restaurant_id": "40356018" }
{"borough": "Manhattan", "cuisine": "Irish", "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841" }
{"borough": "Bronx", "cuisine": "Bakery", "name": "Morris Park Bake Shop", "restaurant_id": "30075445" }
{"borough": "Brooklyn", "cuisine": "Hamburgers", "name": "Wendy'S", "restaurant_id": "30112340" }
{"borough": "Queens", "cuisine": "Jewish/Kosher", "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }
{"borough": "Staten Island", "cuisine": "Jewish/Kosher", "name": "Kosher Island", "restaurant_id": "40356442" }
{"borough": "Queens", "cuisine": "American", "name": "Brunos On The Boulevard", "restaurant_id": "40356151" }
{"borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Wilken'S Fine Food", "restaurant_id": "40356483" }
{"borough": "Brooklyn", "cuisine": "American", "name": "Regina Caterers", "restaurant_id": "40356649" }
{"borough": "Bronx", "cuisine": "American", "name": "Wild Asia", "restaurant_id": "40357217" }
{"borough": "Manhattan", "cuisine": "American", "name": "1 East 66Th Street Kitchen", "restaurant_id": "40359480" }
{"borough": "Brooklyn", "cuisine": "Chinese", "name": "May May Kitchen", "restaurant_id": "40358429" }
{"borough": "Brooklyn", "cuisine": "Jewish/Kosher", "name": "Seuda Foods", "restaurant_id": "40360045" }
{"borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40360076" }
{"borough": "Queens", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40361322" }
{"borough": "Manhattan", "cuisine": "American", "name": "Glorious Food", "restaurant_id": "40361521" }
{"borough": "Brooklyn", "cuisine": "Delicatessen", "name": "Nordic Delicacies", "restaurant_id": "40361390" }
{"borough": "Queens", "cuisine": "Delicatessen", "name": "Sal'S Deli", "restaurant_id": "40361618" }
{"borough": "Brooklyn", "cuisine": "American", "name": "The Movable Feast", "restaurant_id": "40361606" }
{"borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731" }
Type "it" for more
> it
{"borough": "Brooklyn", "cuisine": "American", "name": "C & C Catering Service", "restaurant_id": "40357437" }
{"borough": "Queens", "cuisine": "Delicatessen", "name": "Steve Chu'S Deli & Grocery", "restaurant_id": "40361998" }
{"borough": "Manhattan", "cuisine": "Delicatessen", "name": "Bully'S Deli", "restaurant_id": "40361708" }
{"borough": "Manhattan", "cuisine": "Chicken", "name": "Harriet'S Kitchen", "restaurant_id": "40362098" }
{"borough": "Manhattan", "cuisine": "American", "name": "Angelika Film Center", "restaurant_id": "40362274" }
{"borough": "Manhattan", "cuisine": "American", "name": "P & S Deli Grocery", "restaurant_id": "40362264" }
{"borough": "Manhattan", "cuisine": "Turkish", "name": "The Country Cafe", "restaurant_id": "40362715" }
{"borough": "Brooklyn", "cuisine": "Hamburgers", "name": "White Castle", "restaurant_id": "40362344" }
{"borough": "Brooklyn", "cuisine": "Caribbean", "name": "Shashemene Int'L Restaura", "restaurant_id": "40362869" }
{"borough": "Queens", "cuisine": "Chinese", "name": "Ho Mei Restaurant", "restaurant_id": "40362432" }
{"borough": "Bronx", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "name": "Carvel Ice Cream", "restaurant_id": "40363093" }
{"borough": "Brooklyn", "cuisine": "Donuts", "name": "Dunkin' Donuts", "restaurant_id": "40363098" }
{"borough": "Manhattan", "cuisine": "American", "name": "Downtown Deli", "restaurant_id": "40363021" }
{"borough": "Brooklyn", "cuisine": "American", "name": "Mejlander & Mulgannon", "restaurant_id": "40363117" }
{"borough": "Manhattan", "cuisine": "Bakery", "name": "Olive'S", "restaurant_id": "40363151" }
{"borough": "Bronx", "cuisine": "Chinese", "name": "Happy Garden", "restaurant_id": "40363289" }
{"borough": "Queens", "cuisine": "Delicatessen", "name": "Tony'S Deli", "restaurant_id": "40363333" }
{"borough": "Manhattan", "cuisine": "American", "name": "Cafe Metro", "restaurant_id": "40363298" }
{"borough": "Staten Island", "cuisine": "Delicatessen", "name": "Bagels N Buns", "restaurant_id": "40363427" }
{"borough": "Manhattan", "cuisine": "Sandwiches/Salads/Mixed Buffet", "name": "Lexler Deli", "restaurant_id": "40363426" }
Type "it" for more
>
```

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

```
> db.hotels.find({borough : 'Bronx'}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff670"),
  "address" : {
    "building" : "1007",
    "coord" : [
      -73.856077,
      40.848447
    ],
    "street" : "Morris Park Ave",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "grades" : [
    {
      "date" : ISODate("2014-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-09-11T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
      "grade" : "B",
      "score" : 14
    }
  ],
  "name" : "Morris Park Bake Shop",
  "restaurant_id" : "30075445"
}
```

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

```
>
> db.hotels.find({borough : 'Bronx'}).limit(5).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff670"),
  "address" : {
    "building" : "1007",
    "coord" : [
      -73.856077,
      40.848447
    ],
    "street" : "Morris Park Ave",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "grades" : [
    {
      "date" : ISODate("2014-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-09-11T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
      "grade" : "B",
      "score" : 14
    }
  ],
  "name" : "Morris Park Bake Shop",
  "restaurant_id" : "30075445"
}
```

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

```
> db.hotels.find({borough : 'Bronx'}).skip(5).limit(5).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff6ab"),
  "address" : {
    "building" : "658",
    "coord" : [
      -73.81363999999999,
      40.82941100000001
    ],
    "street" : "Clarence Ave",
    "zipcode" : "10465"
  },
  "borough" : "Bronx",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-06-21T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2012-07-11T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ],
  "name" : "Manhem Club",
  "restaurant_id" : "40364363"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff6c6"),
  "address" : {
    "building" : "2222",
    "coord" : [
      -73.84971759999999,
      40.8304811
    ],
    "street" : "Haviland Avenue",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-12-18T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2014-05-01T00:00:00Z"),
      "grade" : "B",
      "score" : 17
    },
    {
      "date" : ISODate("2013-03-14T00:00:00Z"),
```

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

```
> db.hotels.find({ "grades.score" : {$gt : 90}}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff7cb"),
  "address" : {
    "building" : "65",
    "coord" : [
      -73.9782725,
      40.7624022
    ],
    "street" : "West 54 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-08-22T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-03-28T00:00:00Z"),
      "grade" : "C",
      "score" : 131
    },
    {
      "date" : ISODate("2013-09-25T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-04-08T00:00:00Z"),
      "grade" : "B",
      "score" : 25
    },
    {
      "date" : ISODate("2012-10-15T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2011-10-19T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    }
  ],
  "name" : "Murals On 54/Randolphs'S",
  "restaurant_id" : "40372466"
}
```


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

```
> db.hotels.find( {"grades.score" : { $gt : 80 , $lt : 100 }} ).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff7cb"),
  "address" : {
    "building" : "65",
    "coord" : [
      -73.9782725,
      40.7624022
    ],
    "street" : "West 54 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-08-22T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-03-28T00:00:00Z"),
      "grade" : "C",
      "score" : 131
    },
    {
      "date" : ISODate("2013-09-25T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-04-08T00:00:00Z"),
      "grade" : "B",
      "score" : 25
    },
    {
      "date" : ISODate("2012-10-15T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2011-10-19T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    }
  ],
  "name" : "Murals On 54/Randolphs'S",
  "restaurant_id" : "40372466"
}
```

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

```
> db.hotels.find({"address.coord" : {$lt : -95.754168}}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffcb8"),
  "address" : {
    "building" : "3707",
    "coord" : [
      -101.8945214,
      33.5197474
    ],
    "street" : "82 Street",
    "zipcode" : "11372"
  },
  "borough" : "Queens",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-06-04T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2013-11-07T00:00:00Z"),
      "grade" : "B",
      "score" : 19
    },
    {
      "date" : ISODate("2013-05-17T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-08-29T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-04-03T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2011-11-16T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    }
  ],
  "name" : "Burger King",
  "restaurant_id" : "40534067"
}
```


Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

```
> db.hotels.find({$and : [ {cuisine : {$ne : 'American'}}, {"grades.score" : {$gt : 70}}, {"address.coord": {$lt : -65.754168}} ]}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff7cb"),
  "address" : {
    "building" : "65",
    "coord" : [
      -73.9782725,
      40.7624022
    ],
    "street" : "West 54 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-08-22T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-03-28T00:00:00Z"),
      "grade" : "C",
      "score" : 131
    },
    {
      "date" : ISODate("2013-09-25T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-04-08T00:00:00Z"),
      "grade" : "B",
      "score" : 25
    },
    {
      "date" : ISODate("2012-10-15T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2011-10-19T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    }
  ],
  "name" : "Murals On 54/Randolphs'S",
  "restaurant_id" : "40372466"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff86e"),
```

Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

Note : Do this query without using \$and operator

```
> db.hotels.find({"cuisine" : {$ne : 'American'}, "grades.score" : {$gt : 70}, "address.coord" : {$lt : -65.754168}}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff7cb"),
  "address" : {
    "building" : "65",
    "coord" : [
      -73.9782725,
      40.7624022
    ],
    "street" : "West 54 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-08-22T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-03-28T00:00:00Z"),
      "grade" : "C",
      "score" : 131
    },
    {
      "date" : ISODate("2013-09-25T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-04-08T00:00:00Z"),
      "grade" : "B",
      "score" : 25
    },
    {
      "date" : ISODate("2012-10-15T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2011-10-19T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    }
  ],
  "name" : "Murals On 54/Randolphs'S",
  "restaurant_id" : "40372466"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff86e")
```

Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order

```
> db.hotels.find( { "cuisine": { $ne : 'American ' }, "grades.grade" : "A" , "borough" : "Brooklyn" }).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff671"),
  "address" : {
    "building" : "469",
    "coord" : [
      -73.961704,
      40.662942
    ],
    "street" : "Flatbush Avenue",
    "zipcode" : "11225"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Hamburgers",
  "grades" : [
    {
      "date" : ISODate("2014-12-30T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2014-07-01T00:00:00Z"),
      "grade" : "B",
      "score" : 23
    },
    {
      "date" : ISODate("2013-04-30T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2012-05-08T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Wendy's",
  "restaurant_id" : "30112340"
},
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff675"),
  "address" : {
    "building" : "7114",
    "coord" : [
      -73.9068506,
      40.6199034
    ],
    "street" : "Avenue U",
    "zipcode" : "11234"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "grades" : [
    {
      "date" : ISODate("2014-05-29T00:00:00Z"),
```

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

```
> db.hotels.find({name : /^Wil/ }, {restaurant_id : 1, name : 1, borough : 1, cuisine : 1, _id : 0} ).pretty()
{
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken'S Fine Food",
  "restaurant_id" : "40356483"
}
{
  "borough" : "Bronx",
  "cuisine" : "American ",
  "name" : "Wild Asia",
  "restaurant_id" : "40357217"
}
{
  "borough" : "Bronx",
  "cuisine" : "Pizza",
  "name" : "Wilbel Pizza",
  "restaurant_id" : "40871979"
}
> |
```

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

```
> db.hotels.find({name : /ces$/ }, {restaurant_id : 1, name : 1, borough : 1, cuisine : 1, _id : 0} ).pretty()
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Pieces",
  "restaurant_id" : "40399910"
}
{
  "borough" : "Queens",
  "cuisine" : "American ",
  "name" : "S.M.R Restaurant Services",
  "restaurant_id" : "40403857"
}
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Good Shepherd Services",
  "restaurant_id" : "40403989"
}
{
  "borough" : "Queens",
  "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
  "name" : "The Ice Box-Ralph'S Famous Italian Ices",
  "restaurant_id" : "40690899"
}
{
  "borough" : "Brooklyn",
  "cuisine" : "Jewish/Kosher",
  "name" : "Alices",
  "restaurant_id" : "40782042"
}
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Re: Sources",
  "restaurant_id" : "40876068"
}
>
```

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.hotels.find({name : /Reg/ }, {restaurant_id : 1, name : 1, borough : 1, cuisine : 1, _id : 0} ).pretty()
{
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "name" : "Regina Caterers",
  "restaurant_id" : "40356649"
}
{
  "borough" : "Manhattan",
  "cuisine" : "Café/Coffee/Tea",
  "name" : "Caffe Reggio",
  "restaurant_id" : "40369418"
}
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Regency Hotel",
  "restaurant_id" : "40382679"
}
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Regency Whist Club",
  "restaurant_id" : "40402377"
}
{
  "borough" : "Queens",
  "cuisine" : "American ",
  "name" : "Rego Park Cafe",
  "restaurant_id" : "40523342"
}
{
  "borough" : "Queens",
  "cuisine" : "Pizza",
  "name" : "Regina Pizza",
  "restaurant_id" : "40801325"
}
{
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Regal Entertainment Group",
  "restaurant_id" : "40891782"
}
>
```


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

```
> db.hotels.find( {"borough" : "Bronx", $or : [ {"cuisine" : 'American' }, {"cuisine" : 'Chinese'}] } ).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff691"),
  "address" : {
    "building" : "1236",
    "coord" : [
      -73.8893654,
      40.81376179999999
    ],
    "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("2012-06-12T00:00:00Z"),
      "grade" : "B",
      "score" : 15
    }
  ],
  "name" : "Happy Garden",
  "restaurant_id" : "40363289"
}

{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff6a2"),
  "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("2013-09-26T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    }
  ]
}
```

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.

```
> db.hotels.find({'borough': {'$in': ['Staten Island', 'Queens', 'Bronx', 'Brooklyn']}}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff6e6"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff678"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff671"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff672"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff673"), "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant_id" : "40356442" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff674"), "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff675"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Milken'S Fine Food", "restaurant_id" : "40356483" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff676"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff677"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff679"), "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant_id" : "40358429" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67a"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant_id" : "40360045" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67b"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40360076" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67c"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40361322" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67e"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67f"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant_id" : "40361618" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff680"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant_id" : "40361606" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff681"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff682"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff683"), "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant_id" : "40361998" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff689"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "White Castle", "restaurant_id" : "40362344" }
```

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

```
> db.hotels.find({'borough': {'$nin': ['Staten Island', 'Queens', 'Bronx', 'Brooklyn']}}, {restaurant_id: 1, name: 1, borough: 1, cuisine: 1})
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff66f"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff670"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff677"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant_id" : "40357217" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff678"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant_id" : "40359480" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff67d"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff684"), "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant_id" : "40361708" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff685"), "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant_id" : "40362098" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff686"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Angelika Film Center", "restaurant_id" : "40362274" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff687"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery", "restaurant_id" : "40362264" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff688"), "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe", "restaurant_id" : "40362715" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff68c"), "borough" : "Bronx", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant_id" : "40363093" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff68e"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Downtown Deli", "restaurant_id" : "40363021" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff690"), "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S", "restaurant_id" : "40363151" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff691"), "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant_id" : "40363289" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff693"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Cafe Metro", "restaurant_id" : "40363298" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff695"), "borough" : "Manhattan", "cuisine" : "Sandwiches/Salads/Mixed Buffet", "name" : "Lexler Deli", "restaurant_id" : "40363426" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff696"), "borough" : "Manhattan", "cuisine" : "Continental", "name" : "Lorenzo & Maria'S", "restaurant_id" : "40363630" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff699"), "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant_id" : "40363644" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff69c"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Berkely", "restaurant_id" : "40363685" }
{ "_id" : ObjectId("60a36a1f4dae5be9d71ff69e"), "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant_id" : "40363945" }
```

Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

```
> db.hotels.find({"grades.score" : {$not : {$gt : 10}}}, {restaurant_id : 1, name : 1, borough : 1, cuisine : 1})
{"_id" : ObjectId("60a36a1f4dae5be9d71ff678"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66th Street Kitchen", "restaurant_id" : "40359480" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff67e"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant_id" : "40361390" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff682"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff689"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "White Castle", "restaurant_id" : "40362344" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff69a"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Sonny's Heros", "restaurant_id" : "40363744" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff6ab"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Manhem Club", "restaurant_id" : "40364363" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff6b9"), "borough" : "Staten Island", "cuisine" : "American ", "name" : "Great Kills Yacht Club", "restaurant_id" : "40364610" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff6c0"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Serendipity 3", "restaurant_id" : "40364863" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff6c1"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "White Horse Tavern", "restaurant_id" : "40364958" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff6cc"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dorrian's Red Hand Restaurant", "restaurant_id" : "40365239" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff77e"), "borough" : "Manhattan", "cuisine" : "Mexican", "name" : "Mexico Lindo Restaurant", "restaurant_id" : "40367038" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff73e"), "borough" : "Brooklyn", "cuisine" : "Greek", "name" : "El Greco Diner", "restaurant_id" : "40367795" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff743"), "borough" : "Bronx", "cuisine" : "Café/Coffee/Tea", "name" : "Terrace Cafe", "restaurant_id" : "40368018" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff744"), "borough" : "Bronx", "cuisine" : "Not Listed/Not Applicable", "name" : "The Lark's Nest", "restaurant_id" : "40367946" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff746"), "borough" : "Bronx", "cuisine" : "African", "name" : "African Terrace", "restaurant_id" : "40368021" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff749"), "borough" : "Bronx", "cuisine" : "American ", "name" : "African Market (Baboon Cafe)", "restaurant_id" : "40368026" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff74a"), "borough" : "Staten Island", "cuisine" : "Italian", "name" : "Roadhouse Restaurant", "restaurant_id" : "40368034" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff76b"), "borough" : "Manhattan", "cuisine" : "French", "name" : "Pergola Des Artistes", "restaurant_id" : "40369139" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff777"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Mcdonald's", "restaurant_id" : "40369535" }
{"_id" : ObjectId("60a36a1f4dae5be9d71ff77a"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The River Cafe", "restaurant_id" : "40369608" }
type 'lt' for more
>
```

Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

```
> db.hotels.find({name : /^Wil/, $and : [ {cuisine : {$ne : 'American' }}, {cuisine : {$ne : 'Chinees' }} ], {restaurant_id : 1, name : 1, borough : 1, cuisine : 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff675"),
  "borough" : "Brooklyn",
  "cuisine" : "Delicatessen",
  "name" : "Wilken's Fine Food",
  "restaurant_id" : "40356483"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d720047e"),
  "borough" : "Bronx",
  "cuisine" : "Pizza",
  "name" : "Wilbel Pizza",
  "restaurant_id" : "40871979"
}
>
```

Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.

```
> db.hotels.find({"grades.date" : ISODate("2014-08-11T00:00:00Z"), "grades.grade" : "A", "grades.score" : 11}, {restaurant_id : 1, name: 1, grades: 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff6ec"),
  "grades" : [
    {
      "date" : ISODate("2014-08-11T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2013-07-22T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2013-03-14T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2012-07-02T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2012-02-02T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-08-24T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    }
  ],
  "name" : "Neary'S Pub",
  "restaurant_id" : "40365871"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffc6c"),
  "grades" : [
    {
      "date" : ISODate("2014-08-11T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2013-12-10T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2013-06-10T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    }
  ],
  "name" : "Club Macanudo (Cigar Bar)",
  "restaurant_id" : "40526406"
}
```

Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

```
> db.hotels.find({"grades.1.date" : ISODate("2014-08-11T00:00:00Z"), "grades.1.grade" : "A", "grades.1.score" : 9}, {restaurant_id : 1, name: 1, grades : 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffc9a"),
  "grades" : [
    {
      "date" : ISODate("2015-01-12T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2014-08-11T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2014-01-14T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2013-02-07T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2012-04-30T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    }
  ],
  "name" : "Club Macanudo (Cigar Bar)",
  "restaurant_id" : "40526406"
}
```

Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52.

```
> db.hotels.find({"address.coord.1" : {$gt : 42, $lte : 52}}, {restaurant_id : 1, name : 1, address : 1, coord : 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff90e"),
  "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("60a36a1f4dae5be9d71ff93c"),
  "address" : {
    "building" : "1",
    "coord" : [
      -0.7119979,
      51.6514664
    ],
    "street" : "Pennplaza E, Penn Sta",
    "zipcode" : "10001"
  },
  "name" : "T.G.I. Fridays",
  "restaurant_id" : "40388936"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffb95"),
  "address" : {
    "building" : "3000",
    "coord" : [
      -87.86567699999999,
      42.61150920000001
    ],
    "street" : "47 Avenue",
    "zipcode" : "11101"
  },
  "name" : "Di Luvio'S Deli",
  "restaurant_id" : "40402284"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffdca"),
  "address" : {
    "building" : "21972199",
    "coord" : [
      -78.589606,
      42.8912372
    ],
    "street" : "Broadway",

```

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

```
> db.hotels.aggregate({$sort : {name : 1}}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d72002fe"),
  "address" : {
    "building" : "129",
    "coord" : [
      -73.962943,
      40.685007
    ],
    "street" : "Gates Avenue",
    "zipcode" : "11238"
  },
  "borough" : "Brooklyn",
  "cuisine" : "Italian",
  "grades" : [
    {
      "date" : ISODate("2014-03-06T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-08-29T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-03-08T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-06-27T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2011-11-17T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "(Lewis Drug Store) Locanda Vini E Olii",
  "restaurant_id" : "40804423"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff678"),
  "address" : {
    "building" : "1",
    "coord" : [
      -73.96926909999999,
      40.7685235
    ],
    "street" : "East 66 Street",
    "zipcode" : "10065"
  },
  "borough" : "Manhattan",
  "cuisine" : "Italian",
  "grades" : [
    {
      "date" : ISODate("2014-03-06T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-08-29T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-03-08T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-06-27T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2011-11-17T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Locanda Vini E Olii",
  "restaurant_id" : "40804423"
}
```


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

```
> db.hotels.find().sort({'name' : -1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff72d"),
  "address" : {
    "building" : "6946",
    "coord" : [
      -73.8811834,
      40.7017759
    ],
    "street" : "Myrtle Avenue",
    "zipcode" : "11385"
  },
  "borough" : "Queens",
  "cuisine" : "German",
  "grades" : [
    {
      "date" : ISODate("2014-09-24T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-04-17T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2013-03-12T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2012-10-02T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2012-05-09T00:00:00Z"),
      "grade" : "A",
      "score" : 13
    },
    {
      "date" : ISODate("2011-12-28T00:00:00Z"),
      "grade" : "B",
      "score" : 24
    }
  ],
  "name" : "Zum Stammtisch",
  "restaurant_id" : "40367377"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d7200267"),
  "address" : {
    "building" : "107109",
    "coord" : [
      -73.9744668,
      40.731155
    ]
  }
}
```

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.

```
> db.hotels.find().sort({cuisine : 1, borough : -1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffd59"),
  "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" : 9
    },
    {
      "date" : ISODate("2013-10-23T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2012-10-26T00:00:00Z"),
      "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"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffe83"),
  "address" : {
    "building" : "34",
    "coord" : [
      -73.9883612,
      40.7286391
    ],
    "street" : "St Marks Place",
    "zipcode" : "10003"
  },
  "borough" : "Manhattan",
```

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

```
> db.hotels.find({"address.street" : { $exists : true }}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66e"),
  "address" : {
    "building" : "2780",
    "coord" : [
      -73.98241999999999,
      40.579505
    ],
    "street" : "Stillwell Avenue",
    "zipcode" : "11224"
  },
  "borough" : "Brooklyn",
  "cuisine" : "American ",
  "grades" : [
    {
      "date" : ISODate("2014-06-10T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-06-05T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-04-13T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2011-10-12T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Riviera Caterer",
  "restaurant_id" : "40356018"
}

{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66f"),
  "address" : {
    "building" : "351",
    "coord" : [
      -73.98513559999999,
      40.7676919
    ],
    "street" : "West 57 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "Irish",
  "grades" : [
    {
      "date" : ISODate("2014-09-06T00:00:00Z"),
      "grade" : "A",

```

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

```
> db.hotels.find({"address.coord" : {$type : 1}}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66e"),
  "address" : {
    "building" : "2780",
    "coord" : [
      -73.98241999999999,
      40.579505
    ],
    "street" : "Stillwell Avenue",
    "zipcode" : "11224"
  },
  "borough" : "Brooklyn",
  "cuisine" : "American",
  "grades" : [
    {
      "date" : ISODate("2014-06-10T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-06-05T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-04-13T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2011-10-12T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Riviera Caterer",
  "restaurant_id" : "40356018"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66f"),
  "address" : {
    "building" : "351",
    "coord" : [
      -73.98513559999999,
      40.7676919
    ],
    "street" : "West 57 Street",
    "zipcode" : "10019"
  },
  "borough" : "Manhattan",
  "cuisine" : "Irish",
  "grades" : [
    {
      "date" : ISODate("2014-09-06T00:00:00Z"),
      "grade" : "A",
```

Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.\

```
> db.hotels.find({"grades.score" : {$mod: [7, 0]}}, {restaurant_id : 1, name : 1, grades : 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff66e"),
  "grades" : [
    {
      "date" : ISODate("2014-06-10T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2013-06-05T00:00:00Z"),
      "grade" : "A",
      "score" : 7
    },
    {
      "date" : ISODate("2012-04-13T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2011-10-12T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    }
  ],
  "name" : "Riviera Caterer",
  "restaurant_id" : "40356018"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff670"),
  "grades" : [
    {
      "date" : ISODate("2014-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-09-11T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
```

Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

```
> db.hotels.find({name: {$regex: "mon.*", $options: "i"}}, {"name": 1, "borough": 1, "address.coord": 1, "cuisine": 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff6ff"),
  "address" : {
    "coord" : [
      -73.98306099999999,
      40.741419
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Desmond'S Tavern"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff70a"),
  "address" : {
    "coord" : [
      -73.8221418,
      40.7272376
    ]
  },
  "borough" : "Queens",
  "cuisine" : "Jewish/Kosher",
  "name" : "Shlmons Kosher Pizza"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff716"),
  "address" : {
    "coord" : [
      -74.10465599999999,
      40.58834
    ]
  },
  "borough" : "Staten Island",
  "cuisine" : "American ",
  "name" : "Richmond County Country Club"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ff742"),
  "address" : {
    "coord" : [
      -73.9812843,
      40.5947365
    ]
  },
  "borough" : "Brooklyn",
  "cuisine" : "Pizza/Italian",
  "name" : "Lb Spumoni Gardens"
}
```


Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

```
> db.hotels.find({name : {$regex : /^Mad/}}, {"name" : 1, "borough" : 1, "address.coord" : 1, "cuisine" : 1}).pretty()
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffba7"),
  "address" : {
    "coord" : [
      -73.9860597,
      40.7431194
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Madison Square"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71ffc76"),
  "address" : {
    "coord" : [
      -73.98302199999999,
      40.742313
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "Indian",
  "name" : "Madras Mahal"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71fff27"),
  "address" : {
    "coord" : [
      -74.000002,
      40.72735
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Madame X"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d71fffd7"),
  "address" : {
    "coord" : [
      -73.98171959999999,
      40.7499406
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "French",
  "name" : "Madison Bistro"
}
{
  "_id" : ObjectId("60a36a1f4dae5be9d7200061"),
  "address" : {
    "coord" : [
      -73.9717845,
      40.6897199
    ]
  },
  "borough" : "Manhattan",
  "cuisine" : "American ",
  "name" : "Madison Square Park"
}
```