

Exercise Questions

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

```
Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find();
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a98"),
    address: {
      building: '469',
      coord: [ -73.961704, 40.662942 ],
```

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

```
atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({}, {restaurant_id:1, name:1, borough:1, cuisine:1});

{
  _id: ObjectId("61f01914c0aa4384fc005a97"),
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a98"),
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: "Wendy'S",
  restaurant_id: '30112340'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a99"),
  borough: 'Manhattan',
  cuisine: 'Irish',
  name: 'Dj Reynolds Pub And Restaurant',
  restaurant_id: '30191841'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9a"),
  borough: 'Brooklyn',
  cuisine: 'American ',
  name: 'Riviera Caterer',
  restaurant_id: '40356018'
}
atlas atlas-w4167a-shard-0 [primary] addresses>
{
  _id: ObjectId("61f01914c0aa4384fc005a9b"),
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  name: 'Tov Kosher Kitchen',
  restaurant_id: '40356068'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9c"),
  borough: 'Queens',
  cuisine: 'American ',
  name: 'Brunos On The Boulevard',
  restaurant_id: '40356151'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9d"),
  borough: 'Staten Island',
  cuisine: 'Jewish/Kosher',
```

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`.

```
Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({}, {restaurant_id:1,name:1,borough:1,cuisine:1,_id:0});
[
  {
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'Wendy'S',
    restaurant_id: '30112340'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356068'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  ],_id: ObjectId("61f01914c0aa4384fc005a9b"),
```

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`.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({}, {restaurant_id:1,name:1,borough:1,cuisine:1,address:{zipcode:1},_id:0});
[
  {
    address: { zipcode: '10462' },
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    address: { zipcode: '11225' },
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'Wendy'S',
    restaurant_id: '30112340'
  },
  {
    address: { zipcode: '10019' },
    borough: 'Manhattan',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    address: { zipcode: '11224' },
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    address: { zipcode: '11374' },
    borough: 'Queens',

```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.aggregate([{$match:{borough:"Bronx"}},{ $limit:5}]);
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa1"),
    address: {
      building: '2300',
      coord: [ -73.8786113, 40.8502883 ],
      street: 'Southern Boulevard',
      zipcode: '10460'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]

```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.aggregate([{$match:{borough:"Bronx"}}]);
[
  {
    grade: 'A',
    {
      score: 12
    },
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    address: {
      building: '1007',2012-08-29T00:00:00.000Z",
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    {
      borough: 'Bronx',("2012-02-13T00:00:00.000Z"),
      cuisine: 'Bakery',
      grades: [: 16
        {
          date: ISODate("2014-03-03T00:00:00.000Z"),
          grade: 'A',e',
          score: 2d: '40366748'
        },
        {
          date: ISODate("2013-09-11T00:00:00.000Z"),
          grade: 'A',
          score: 6'4340',
        },ord: [ -73.8194559, 40.8899176 ],
        {treet: 'Boston Road',
          date: ISODate("2013-01-24T00:00:00.000Z"),
          grade: 'A',
          score: 10nx',
        },ine: 'Pancakes/Waffles',
        {des: [
          date: ISODate("2011-11-23T00:00:00.000Z"),
          grade: 'A',te("2014-09-22T00:00:00.000Z"),
          score: 9A',
        },score: 11
        {,
          date: ISODate("2011-03-10T00:00:00.000Z"),
          grade: 'B',te("2014-05-16T00:00:00.000Z"),
          score: 14',
        } score: 9
      ],},
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'0-30T00:00:00.000Z"),
  },
  {
    grade: 'A',
    {
      score: 3
    },
    _id: ObjectId("61f01914c0aa4384fc005aa1"),
    address: {
      building: '2300',2013-06-05T00:00:00.000Z"),
      coord: [ -73.8786113, 40.8502883 ],

```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.aggregate([{$match:{borough:"Bronx"}},{ $skip:5},{ $limit:5}]);
[
  {
    _id: ObjectId("61f01914c0aa4384fc005ad4"),
    address: {
      building: '658',
      coord: [ -73.81363999999999, 40.82941100000001 ],
      street: 'Clarence Ave',
      zipcode: '10465'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-21T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2012-07-11T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ],
    name: 'Manhem Club',
    restaurant_id: '40364363'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aec"),
    address: {
      building: '2222',
      coord: [ -73.84971759999999, 40.8304811 ],
      street: 'Haviland Avenue',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-12-18T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2014-05-01T00:00:00.000Z"),
        grade: 'B',
        score: 17
      }
    ]
  }
]

```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.aggregate([{$match:{"grades.score":{$gt:90}}}] );
[
  {
    _id: ObjectId("61f01914c0aa4384fc005bf5"),
    address: {
      building: '65',
      coord: [ -73.9782725, 40.7624022 ],
      street: 'West 54 Street',
      zipcode: '10019'
    },
    borough: 'Manhattan',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-08-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-03-28T00:00:00.000Z"),
        grade: 'C',
        score: 131
      },
      {
        date: ISODate("2013-09-25T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2013-04-08T00:00:00.000Z"),
        grade: 'B',
        score: 25
      },
      {
        date: ISODate("2012-10-15T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-10-19T00:00:00.000Z"),
        grade: 'A',
        score: 13
      }
    ],
    name: "Murals On 54/Randolphs'S",
    restaurant_id: '40372466'
  },
]

```

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


```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({grades:{$elemMatch:{"score":{$gt:80,$lt:100}}}});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005c96"),
    address: {
      building: '345',
      coord: [ -73.9864626, 40.7266739 ],
      street: 'East 6 Street',
      zipcode: '10003'
    },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades: [
      {
        date: ISODate("2014-09-15T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-04-24T00:00:00.000Z"),
        grade: 'P',
        score: 2
      },
      {
        date: ISODate("2012-10-01T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-04-06T00:00:00.000Z"),
        grade: 'C',
        score: 92
      },
      {
        date: ISODate("2011-11-03T00:00:00.000Z"),
        grade: 'C',
        score: 41
      }
    ]
  }
]

```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"address.coord":{"$lt":-95.754168}});
[
  {
    _id: ObjectId("61f01915c0aa4384fc0060df"),
    address: {
      building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372'
    },
    borough: 'Queens',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-04T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2013-11-07T00:00:00.000Z"),
        grade: 'B',
        score: 19
      },
      {
        date: ISODate("2013-05-17T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-08-29T00:00:00.000Z"),
        date: ISODate("2012-08-29T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2012-04-03T00:00:00.000Z"),
        date: ISODate("2012-04-03T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2011-11-16T00:00:00.000Z"),
        date: ISODate("2011-11-16T00:00:00.000Z"),
        grade: 'A',
        score: 7
      }
    ],
    me: 'Burger King',
    name: 'Burger King',4067
    restaurant_id: '40534067'
  },
]

```

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.

```
Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({$and:[{"cuisine":{"$ne:"American"}},{ "grades.score":{"$gt:70"}},{ "address.coord":{"$lt:-65.754168}}]});
```

```
[
  {
    _id: ObjectId("61f01914c0aa4384fc005c96"),
    address: {
      building: '345',
      coord: [ -73.9864626, 40.7266739 ],
      street: 'East 6 Street',
      zipcode: '10003'
    },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades: [
      {
        date: ISODate("2014-09-15T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-05-30T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2013-04-24T00:00:00.000Z"),
        grade: 'P',
        score: 2
      },
      {
        date: ISODate("2012-10-01T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-04-06T00:00:00.000Z"),
        grade: 'C',
        score: 92
      },
      {
        date: ISODate("2011-11-03T00:00:00.000Z"),
        grade: 'C',
        score: 41
      }
    ]
  }
]
```

Activa
Go to Se

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.

```
Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({$and:[{"cuisine":{"$ne:"American "}},{"grades.score":{"$gt:70}},{"address.coord":{"$lt:-65.754168}}]});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005c96"),
    address: {
      building: '345',
      coord: [ -73.9864626, 40.7266739 ],
      street: 'East 6 Street',
      zipcode: '10003'
    },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades: [
      {
        date: ISODate("2014-09-15T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-05-30T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2013-04-24T00:00:00.000Z"),
        grade: 'P',
        score: 2
      },
      {
        date: ISODate("2012-10-01T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-04-06T00:00:00.000Z"),
        grade: 'C',
        score: 92
      },
      {
        date: ISODate("2011-11-03T00:00:00.000Z"),
        grade: 'C',
        score: 41
      }
    ]
  }
]
```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({$and:[{"cuisine":{"$ne:"American "}},{"grades.grade":"A"},{"borough":{"$ne:"Brooklyn"}}]}).sort({"cuisine":-1});
[
  {
    _id: ObjectId("61f01915c0aa4384fc0061a3"),
    address: {
      building: '89',
      coord: [ -73.9995899, 40.7168015 ],
      street: 'Baxter Street',
      zipcode: '10013'
    },
    borough: 'Manhattan',
    cuisine: 'Vietnamese/Cambodian/Malaysia',
    grades: [
      {
        date: ISODate("2014-08-21T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-08-31T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-04-11T00:00:00.000Z"),
        grade: 'C',
        score: 3
      },
      {
        date: ISODate("2012-10-17T00:00:00.000Z"),
        grade: 'A',
        score: 4
      },
      {
        date: ISODate("2012-05-15T00:00:00.000Z"),
        grade: 'A',
        score: 10
      }
    ],
    name: 'Thai Son',
    restaurant_id: '40559606'
  }
]

```

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.

```

tlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({name:/^Wil/},{restaurant_id:1,name:1,borough:1,cuisine:1});

{
  _id: ObjectId("61f01914c0aa4384fc005a9e"),
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  name: "Wilken'S Fine Food",
  restaurant_id: '40356483'
},
{
  _id: ObjectId("61f01914c0aa4384fc005aa1"),
  borough: 'Bronx',
  cuisine: 'American ',
  name: 'Wild Asia',
  restaurant_id: '40357217'
},
{
  _id: ObjectId("61f01916c0aa4384fc0068a6"),
  borough: 'Bronx',
  cuisine: 'Pizza',
  name: 'Wilbel Pizza',
  restaurant_id: '40871979'
}

tlas atlas-w4167a-shard-0 [primary] addresses>

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({name:/ces$/},{restaurant_id:1,name:1,borough:1,cuisine:1});
[
  {
    _id: ObjectId("61f01915c0aa4384fc005f2a"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Pieces',
    restaurant_id: '40399910'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc005fe9"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'S.M.R Restaurant Services',
    restaurant_id: '40403857'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc005fef"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Good Shepherd Services',
    restaurant_id: '40403989'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc0064a2"),
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'The Ice Box-Ralph'S Famous Italian Ices',
    restaurant_id: '40690899'
  },
  {
    _id: ObjectId("61f01916c0aa4384fc0066a4"),
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Alices',
    restaurant_id: '40782042'
  },
  {
    _id: ObjectId("61f01916c0aa4384fc0068c0"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Re: Sources',
    restaurant_id: '40876068'
  }
]

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({name:/Reg/},{restaurant_id:1,name:1,borough:1,cuisine:1});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a9f"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005b9c"),
    borough: 'Manhattan',
    cuisine: 'Café/Coffee/Tea',
    name: 'Caffe Reggio',
    restaurant_id: '40369418'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005cab"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc005fc8"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Regency Whist Club',
    restaurant_id: '40402377'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc0060ab"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Rego Park Cafe',
    restaurant_id: '40523342'
  },
  {
    _id: ObjectId("61f01916c0aa4384fc006719"),
    borough: 'Queens',
    cuisine: 'Pizza',
    name: 'Regina Pizza',
    restaurant_id: '40801325'
  },
  {
    _id: ObjectId("61f01916c0aa4384fc006930"),
    borough: 'Manhattan',

```

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


```

las atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({borough:"Bronx",$or:[{cuisine:"American "},{cuisine:"chinese"}]});

{
  _id: ObjectId("61f01914c0aa4384fc005aa1"),
  address: {
    building: '2300',
    coord: [ -73.8786113, 40.8502883 ],
    street: 'Southern Boulevard',
    zipcode: '10460'
  },
  borough: 'Bronx',
  cuisine: 'American ',
  grades: [
    {
      date: ISODate("2014-05-28T00:00:00.000Z"),
      grade: 'A',
      score: 11
    },
    {
      date: ISODate("2013-06-19T00:00:00.000Z"),
      grade: 'A',
      score: 4
    },
    {
      date: ISODate("2012-06-15T00:00:00.000Z"),
      grade: 'A',
      score: 3
    }
  ],
  name: 'Wild Asia',
  restaurant_id: '40357217'
},
{
  _id: ObjectId("61f01914c0aa4384fc005ad4"),
  address: {
    building: '658',
    coord: [ -73.81363999999999, 40.82941100000001 ],
    street: 'Clarence Ave',
    zipcode: '10465'
  },
  borough: 'Bronx',
  cuisine: 'American ',
  grades: [
    {
      date: ISODate("2014-06-21T00:00:00.000Z"),
      grade: 'A',
      score: 5
    },
  ],
}

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({borough:{$in:["Staten Island","Wueens","Bronx","Brooklyn"]}}, {restaurant_id:1,name:1,borough:1,cuisine:1});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a98"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9a"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9d"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island',
    restaurant_id: '40356442'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9e"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9f"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({borough:{$nin:["Staten Island","Wueens","Bronx","Brooklyn"]}}, {restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a99"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9b"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356068'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9c"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Brunos On The Boulevard',
    restaurant_id: '40356151'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa4"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa7"),
    borough: 'Queens',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Carvel Ice Cream',
    restaurant_id: '40361322'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa9"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Glorious Food',
    restaurant_id: '40361521'
  },
]

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"grades.score":{"not":{"gt:10}}},{restaurant_id:1,name:1,borough:1,cuisine:1});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005aa2"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa4"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005aa8"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies',
    restaurant_id: '40361390'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005ab1"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'White Castle',
    restaurant_id: '40362344'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005ac4"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: "Sonny'S Heros",
    restaurant_id: '40363744'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005ad4"),
    borough: 'Bronx',
    cuisine: 'American ',

```

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'.

```

tlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({$or:[{name:/"Wil/},{ "$and":[{"cuisine":{"$ne:"American "}}, {"cuisine":{"$ne:"Chinees"}}]}]}, {restaurant_id:1,name:1,borough:1,cuisine:1});

{
  _id: ObjectId("61f01914c0aa4384fc005a97"),
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a98"),
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: 'Wendy'S',
  restaurant_id: '30112340'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a99"),
  borough: 'Manhattan',
  cuisine: 'Irish',
  name: 'Dj Reynolds Pub And Restaurant',
  restaurant_id: '30191841'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9b"),
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  name: 'Tov Kosher Kitchen',
  restaurant_id: '40356068'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9d"),
  borough: 'Staten Island',
  cuisine: 'Jewish/Kosher',
  name: 'Kosher Island',
  restaurant_id: '40356442'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a9e"),
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  name: 'Wilken'S Fine Food',
  restaurant_id: '40356483'
},
{
}

```

Activate Windows
Go to Settings to activate Windows.

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..

Activate W
Go to Settings

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"

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"grades.date":ISODate("2014-08-11T00:00:00Z"),"grades.grade":"A","grades.score":11},{restaurant_id : 1,name:1,grades:1});
[
  {
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    { name: 'Taste The Tropics Ice Cream',
      _id: ObjectId("61f01914c0aa4384fc005b15"),
      grades: [
        {
          date: ISODate("2014-08-11T00:00:00Z"),
          grade: 'A',
          score: 13
        },
        { name: 'Wild Asia',
          {restaurant_id: '40357217'
            date: ISODate("2013-07-22T00:00:00Z"),
            grade: 'A',
            score: 9
          },
          { name: 'Brooklyn',
            {cuisine: 'Chinese',
              date: ISODate("2013-03-14T00:00:00Z"),
              grade: 'A',
              score: 12
            },
            { _id: ObjectId("61f01914c0aa4384fc005aa5"),
              date: ISODate("2012-07-02T00:00:00Z"),
              grade: 'A',
              score: 11
            },
            { name: 'Foods',
              {restaurant_id: '40360045'
                {
                  date: ISODate("2012-02-02T00:00:00Z"),
                  grade: 'A',
                  score: 10
                },
                { name: 'Brooklyn',
                  {cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
                    {name: 'Carvel Ice Cream',
                      date: ISODate("2011-08-24T00:00:00Z"),
                      grade: 'A',
                      score: 11
                    },
                    { _id: ObjectId("61f01914c0aa4384fc005aa7"),
                      {name: 'Queens',
                        {cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
                          {name: 'Neary's Pub',
                            {restaurant_id: '40365871'
                              {restaurant_id: '40361322'
                                {
                                  _id: ObjectId("61f01914c0aa4384fc005bf0"),
                                  grades: [
                                    {
                                      _id: ObjectId("61f01914c0aa4384fc005aa8"),
                                      {cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
                                        {name: 'Brooklyn',
                                          {
                                            {
                                              {
                                                {
                                                  {
                                                    {
                                                      {
                                                        {
                                                          {
                                                            {
                                                              {
                                                                {
                                                                  {
                                                                    {
                                                                      {
                                                                        {
                                                                          {
                                                                            {
                                                                              {
                                                                                {
                                                                                  {
                                                                                    {
                                                                                      {
                                                                                        {
                                                                                          {
                                                                                           ...
                                                                                          ...
                                                                                        ...
                                                                                      ...
                                                                                    ...
                                                                                  ...
                                                                                ...
                                                                              ...
                                                                            ...
                                                                          ...
                                                                        ...
                                                                      ...
                                                                    ...
                                                                  ...
                                                                ...
                                                              ...
                                                            ...
                                                          ...
                                                        ...
                                                      ...
                                                    ...
                                                  ...
                                                ...
                                              ...
                                            ...
                                          ...
                                        ...
                                      ...
                                    ...
                                  ...
                                ...
                              ...
                            ...
                          ...
                        ...
                      ...
                    ...
                  ...
                ...
              ...
            ...
          ...
        ...
      ...
    ...
  ]
}

```

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..

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"address.coord.1":{"$gt:42,$lte:52}},{restaurant_id : 1,name:1,address:1,coord:1});
{
  _id: ObjectId("61f01914c0aa4384fc005d39"),
  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("61f01914c0aa4384fc005d65"),
  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("61f01915c0aa4384fc005fbc"),
  address: {
    building: '3000',
    coord: [ -87.86567699999999, 42.61150920000001 ],
    street: '47 Avenue',
    zipcode: '11101'
  },
  name: "Di Luvio'S Deli",
  restaurant_id: '40402284'
},
{
  _id: ObjectId("61f01915c0aa4384fc0061f3"),
  address: {
    building: '21972199',
    coord: [ -78.589606, 42.8912372 ],
    street: 'Broadway',
    zipcode: '10024'
  },
  name: 'La Caridad 78',
  restaurant_id: '40568285'
},

```

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


```
atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find().sort({name:1})
```

```
{
  _id: ObjectId("61f01916c0aa4384fc006727"),
  address: {
    building: '129',
    coord: [ -73.962943, 40.685007 ],
    street: 'Gates Avenue',
    zipcode: '11238'
  },
  borough: 'Brooklyn',
  cuisine: 'Italian',
  grades: [
    {
      date: ISODate("2014-03-06T00:00:00.000Z"),
      grade: 'A',
      score: 5
    },
    {
      date: ISODate("2013-08-29T00:00:00.000Z"),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate("2013-03-08T00:00:00.000Z"),
      grade: 'A',
      score: 7
    },
    {
      date: ISODate("2012-06-27T00:00:00.000Z"),
      grade: 'A',
      score: 7
    },
    {
      date: ISODate("2011-11-17T00:00:00.000Z"),
      grade: 'A',
      score: 12
    }
  ],
  name: '(Lewis Drug Store) Locanda Vini E Olii',
  restaurant_id: '40804423'
},
{
  _id: ObjectId("61f01914c0aa4384fc005aa4"),
  address: {
```

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

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find().sort({"name":-1})
[
  {
    _id: ObjectId("61f01914c0aa4384fc005b56"),
    address: {
      building: '6946',
      coord: [ -73.8811834, 40.7017759 ],
      street: 'Myrtle Avenue',
      zipcode: '11385'
    },
    borough: 'Queens',
    cuisine: 'German',
    grades: [
      {
        date: ISODate("2014-09-24T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2014-04-17T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2013-03-12T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-10-02T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2012-05-09T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2011-12-28T00:00:00.000Z"),
        grade: 'B',

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find().sort({cuisine:1,borough:-1,})
[
  {
    _id: ObjectId("61f01915c0aa4384fc006182"),
    address: {
      building: '1345',
      coord: [ -73.959249, 40.768076 ],
      street: '2 Avenue',
      zipcode: '10021'
    },
    borough: 'Manhattan',
    cuisine: 'Afghan',
    grades: [
      {
        date: ISODate("2014-10-07T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2013-10-23T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2012-10-26T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2012-04-26T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-01-12T00:00:00.000Z"),
        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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find( {"address.street":{$exists:true}});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  }
]

```

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

```

class atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"address.coord":{"$type:1}});

{
  _id: ObjectId("61f01914c0aa4384fc005a97"),
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate("2014-03-03T00:00:00.000Z"),
      grade: 'A',
      score: 2
    },
    {
      date: ISODate("2013-09-11T00:00:00.000Z"),
      grade: 'A',
      score: 6
    },
    {
      date: ISODate("2013-01-24T00:00:00.000Z"),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate("2011-11-23T00:00:00.000Z"),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate("2011-03-10T00:00:00.000Z"),
      grade: 'B',
      score: 14
    }
  ],
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
},
{
  _id: ObjectId("61f01914c0aa4384fc005a98"),
  address: {

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({"grades.score":{$mod:[7,0]}},{ "restaurant_id":1,"name":1,"grades":1})
[
  {
    _id: ObjectId("61f01914c0aa4384fc005a97"),
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2011-11-23T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2011-03-10T00:00:00.000Z"),
        grade: 'B',
        score: 14
      }
    ],
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005a9a"),
    grades: [
      {
        date: ISODate("2014-06-10T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2013-06-05T00:00:00.000Z"),

```

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.

```

Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({name:{$regex:"mon.*",$options:"i"}},{name:1,borough:1,"address.coord":1,cuisine:1});
[
  {
    _id: ObjectId("61f01914c0aa4384fc005b2b"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005b34"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005b40"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005b6b"),
    address: { coord: [ -73.9812843, 40.5947365 ] },
    borough: 'Brooklyn',
    cuisine: 'Pizza/Italian',
    name: 'Lb Spumoni Gardens'
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005bbd"),
    address: { coord: [ -73.951199, 40.7166026 ] },
    borough: 'Brooklyn',
    cuisine: 'Italian',
    name: "Bamonte'S Restaurant"
  },
  {
    _id: ObjectId("61f01914c0aa4384fc005bf4"),
    address: { coord: [ -73.924072, 40.761089000000001 ] },
    borough: 'Queens',
    cuisine: 'Greek',
    name: 'Omonia Cafe'
  },
  {

```

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

```
Atlas atlas-w4167a-shard-0 [primary] addresses> db.restaurants.find({name:{$regex:/^Mad/i,}}, {name:1, borough:1, "address.coord":1, cuisine:1});
[
  {
    _id: ObjectId("61f01915c0aa4384fc005fd3"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc0060a1"),
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc00634f"),
    address: { coord: [ -74.000002, 40.72735 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madame X'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc0063ff"),
    address: { coord: [ -73.98171959999999, 40.7499406 ] },
    borough: 'Manhattan',
    cuisine: 'French',
    name: 'Madison Bistro'
  },
  {
    _id: ObjectId("61f01915c0aa4384fc006488"),
    address: { coord: [ -73.9717845, 40.6897199 ] },
    borough: 'Brooklyn',
    cuisine: 'African',
    name: 'Madiba'
  },
  {
    _id: ObjectId("61f01916c0aa4384fc00678d"),
    address: { coord: [ -73.9040753, 40.9069011 ] },
    borough: 'Bronx',
    cuisine: 'Italian',
  }
]
```