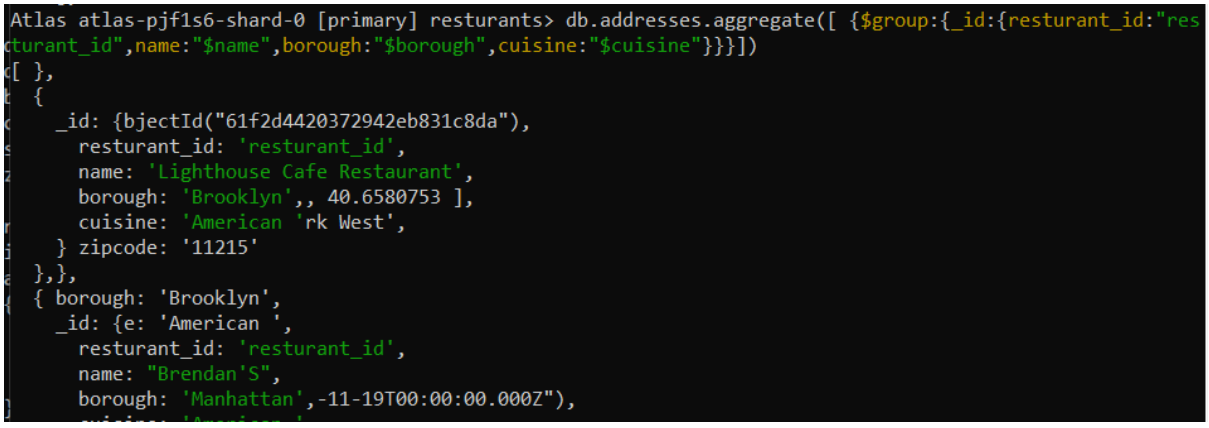
1. Write a MongoDB query to display all the documents in the collection restaurants.

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find().pretty()
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    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',te("2013-05-17T00:00:00.000Z"),
        score: 2A',
      },score: 11
      {,
```

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



```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([ {$group:{_id:{resturant_id:"res
turant_id",name:"$name",borough:"$borough",cuisine:"$cuisine"}}}])
[ },
  {
    _id: {bjectId("61f2d4420372942eb831c8da"),
      resturant_id: 'resturant_id',
      name: 'Lighthouse Cafe Restaurant',
      borough: 'Brooklyn',, 40.6580753 ],
      cuisine: 'American 'rk West',
    } zipcode: '11215'
  },},
  { borough: 'Brooklyn',
    _id: {e: 'American ',
      resturant_id: 'resturant_id',
      name: "Brendan'S",
      borough: 'Manhattan',-11-19T00:00:00.000Z"),
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({},{resturant_id:1,name:1,borough:1,co
usine:1,_id:0})
[     {
  { borough: 'Bronx', name: 'Morris Park Bake Shop' },
  { borough: 'Brooklyn', name: "Wendy'S" },
  { borough: 'Manhattan', name: 'Dj Reynolds Pub And Restaurant' },
  { borough: 'Brooklyn', name: 'Riviera Caterer' },
  { borough: 'Queens', name: 'Tov Kosher Kitchen' },
  { borough: 'Queens', name: 'Brunos On The Boulevard' },
  { borough: 'Staten Island', name: 'Kosher Island' },
  { borough: 'Brooklyn', name: "Wilken'S Fine Food" },
  { borough: 'Brooklyn', name: 'Regina Caterers' },
  { borough: 'Brooklyn', name: 'Taste The Tropics Ice Cream' },
  { borough: 'Bronx', name: 'Wild Asia' },.000Z"),
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([ { $project: { resturant_id: 1,
name: 1, borough: 1, "address.zipcode": 1, _id: 0 } }])
[   ],
  { name: 'Gandhi',
    address: { zipcode: '10462' },
    borough: 'Bronx',
    name: 'Morris Park Bake Shop'
  },_id: ObjectId("61f2d4420372942eb831cb54"),
  { address: {
    address: { zipcode: '11225' },
    borough: 'Brooklyn',09, 40.740735 ],
    name: "Wendy'S"   23 Street',
  },  zipcode: '10010'
  { },
    address: { zipcode: '10019' },
    borough: 'Manhattan',
    name: 'Dj Reynolds Pub And Restaurant'
  },  {
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([{$match:{borough:"Bronx"}},{$lim
it:5}])
[     {
  {     date: ISODate("2013-04-11T00:00:00.000Z"),
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    address: { 25
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',5T00:00:00.000Z"),
      zipcode: '10462'
    },  score: 12
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [ ISODate("2012-10-10T00:00:00.000Z"),
      { grade: 'P',
        date: ISODate("2014-03-03T00:00:00.000Z"),
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([{$match:{borough:"Bronx"}}])
[
  { grades: [
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    address: {ISODate("2014-06-27T00:00:00.000Z"),
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },  date: ISODate("2013-06-06T00:00:00.000Z"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 23
      },
      {
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([{$match:{borough:"Bronx"}},{$ski
p:5}])
[       score: 10
  {   },
    _id: ObjectId("61f2d4420372942eb831c904"),
    address: {ISODate("2014-01-14T00:00:00.000Z"),
      building: '658',
      coord: [ -73.81363999999999, 40.82941100000001 ],
      street: 'Clarence Ave',
      zipcode: '10465'
    },  date: ISODate("2013-07-25T00:00:00.000Z"),
    borough: 'Bronx',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-21T00:00:00.000Z"),
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([{$match:{"grades.score":{$gt:90}}}])
[
  {
    _id: ObjectId("61f2d4420372942eb831ca25"),
    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
      },
      {
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.aggregate([{$match:{$and:[{"grades.score":{$gt:80}},{"grades.score":{$lt:100}}]}}])
[
  {
    _id: ObjectId("61f2d4420372942eb831ca25"),
    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
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"address.coord":{$lt:-95.754168}})
[
  {
    _id: ObjectId("61f2d4430372942eb831cf0f"),
    address: {
      building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372'
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({$and:[{"cuisine":{$ne:"American"}},
grades.score":{$gt:7}},{"address.coord":{$lt:-65.754168}}]})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    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
```

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.

```
tlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({$and:[{cuisine:{$ne:"American "}},{"g
ades.grade":"A"},{borough:{$ne:"Brooklyn"}}]}).sort({cuisine:-1})

{
  _id: ObjectId("61f2d4430372942eb831cfd3"),
  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
    },
```

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.

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:/^Wil/},{restaurant_id:1,name:1,
borough:1,cousicousine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8ce"),
    borough: 'Brooklyn',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8d1"),
    borough: 'Bronx',
    name: 'Wild Asia',
    restaurant_id: '40357217'
  },
  {
    _id: ObjectId("61f2d4430372942eb831d6d6"),
    borough: 'Bronx',
    name: 'Wilbel Pizza',
    restaurant_id: '40871979'
  }
]
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:/ces$/},{restaurant_id:1,name:1,
borough:1,cousine:1})
[
  {
    _id: ObjectId("61f2d4430372942eb831cd5a"),
    borough: 'Manhattan',
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:/ces$/},{restaurant_id:1,name:1,
borough:1,cousine:1})
[
  {
    _id: ObjectId("61f2d4430372942eb831cd5a"),
    borough: 'Manhattan',
    name: 'Pieces',
    restaurant_id: '40399910'
  },
  {
    _id: ObjectId("61f2d4430372942eb831ce19"),
    borough: 'Queens',
    name: 'S.M.R Restaurant Services',
    restaurant_id: '40403857'
  },
  {
    _id: ObjectId("61f2d4430372942eb831ce1f"),
    borough: 'Manhattan',
    name: 'Good Shepherd Services',
    restaurant_id: '40403989'
  },
  {
    _id: ObjectId("61f2d4430372942eb831d2d2"),
    borough: 'Queens',
    name: "The Ice Box-Ralph'S Famous Italian Ices",
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:/Reg/},{restaurant_id:1,name:1,b
orough:1,cousine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8cf"),
    borough: 'Brooklyn',
    name: 'Regina Caterers',
    restaurant_id: '40356649'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c9cc"),
    borough: 'Manhattan',
    name: 'Caffe Reggio',
    restaurant_id: '40369418'
  },
  {
    _id: ObjectId("61f2d4420372942eb831cadb"),
    borough: 'Manhattan',
    name: 'Regency Hotel',
    restaurant_id: '40382679'
  },
  {
    _id: ObjectId("61f2d4430372942eb831cdf8"),
    borough: 'Manhattan',
    name: 'Regency Whist Club',
    restaurant_id: '40402377'
  },
  {
    _id: ObjectId("61f2d4430372942eb831cedb"),
    borough: 'Queens',
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({borough:"Bronx",$or:[{cuisine:"Americ
an"},{cuisine:"Chinese"}]})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8ea"),
    address: {
      building: '1236',
      coord: [ -73.8893654, 40.81376179999999 ],
      street: '238 Spofford Ave',
      zipcode: '10474'
    },
    borough: 'Bronx',
    cuisine: 'Chinese',
    grades: [
      {
        date: ISODate("2013-12-30T00:00:00.000Z"),
        grade: 'A',
        score: 8
      },
      {
        date: ISODate("2013-01-08T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2012-06-12T00:00:00.000Z"),
```

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 Bronxor Brooklyn.

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({borough:{$in:["Staten Island","Queens
","Bronx","Brooklyn"]}},{restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8c8"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8ca"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8cb"),
    borough: 'Queens',
```

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 Bronxor Brooklyn.

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({borough:{$nin:["Staten Island","Queen
s","Bronx","Brooklyn"]}},{restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c9"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8d4"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8d9"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Glorious Food',
    restaurant_id: '40361521'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8dc"),
    borough: 'Manhattan',
    cuisine: 'Delicatessen'
```

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.

```
Type "it" for more
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"grades.score":{$not:{$gt:10}}},{rest
aurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8d2"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8d4"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8d8"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies',
    restaurant_id: '40361390'
```

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'.

```
Type "it" for more
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({$or:[{name:/^Wil/},{"$and":[{"cuisine
":{$ne:"American"}},{cuisine: {$ne:"Chinese"}}]}]},{restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8c8"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8c9"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c8ca"),
```

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..

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"grades.1.date":ISODate("2014-08-11T0
0:00:00Z"),"grades.1.score":"A","grades.1.score":9},{"restaurant_id":1,"name":1,"grades":1})
[
  {
    _id: ObjectId("61f2d4430372942eb831cef2"),
    grades: [
      {
        date: ISODate("2015-01-12T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"address.coord.1":{$gt:42,$lte:52}},{
restaurant_id:1,name:1,address:1,coord:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831cb69"),
    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("61f2d4420372942eb831cb95"),
    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("61f2d4430372942eb831cdee"),
    address: {
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find().sort({"name":1})
[
  {
    _id: ObjectId("61f2d4430372942eb831d557"),
    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
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find().sort({"name":-1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c986"),
    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"),
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find().sort({"cuisine":1,"borough":-1})
[
  {
    _id: ObjectId("61f2d4430372942eb831cfb2"),
    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
      },
      {
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"address.street":{$exists:true}})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    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
```

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

```
Atlas atlas-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"address.coord":{$type:1}})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    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',
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({"grades.score":{$mod:[7,0]}},{restaur
ant_id:1,grades:1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c8c7"),
    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
      },
      {
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:{$regex:"mon.*",$options:"i"}},{
"name":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("61f2d4420372942eb831c95b"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    _id: ObjectId("61f2d4420372942eb831c964"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("61f2d4420372942eb831c970"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
  {
```

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-pjf1s6-shard-0 [primary] resturants> db.addresses.find({name:{$regex:"mad",$options:"i"}},{"n
ame":1,"borough":1,"address.coord":1,"cuisine":1})
[
  {
    _id: ObjectId("61f2d4420372942eb831caf1"),
    address: { coord: [ -74.158427, 40.626607 ] },
    borough: 'Staten Island',
Atlas atlas-pjf1s6-shard-0 [primary] resturants>
    name: 'Real Madrid Restaurant'
  },
  {
    _id: ObjectId("61f2d4430372942eb831ce03"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("61f2d4430372942eb831ce70"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'Café/Coffee/Tea',
    name: 'Cafe Madison'
  },
  {
    _id: ObjectId("61f2d4430372942eb831ce85"),
    address: { coord: [ -73.989578, 40.7129199 ] },
```