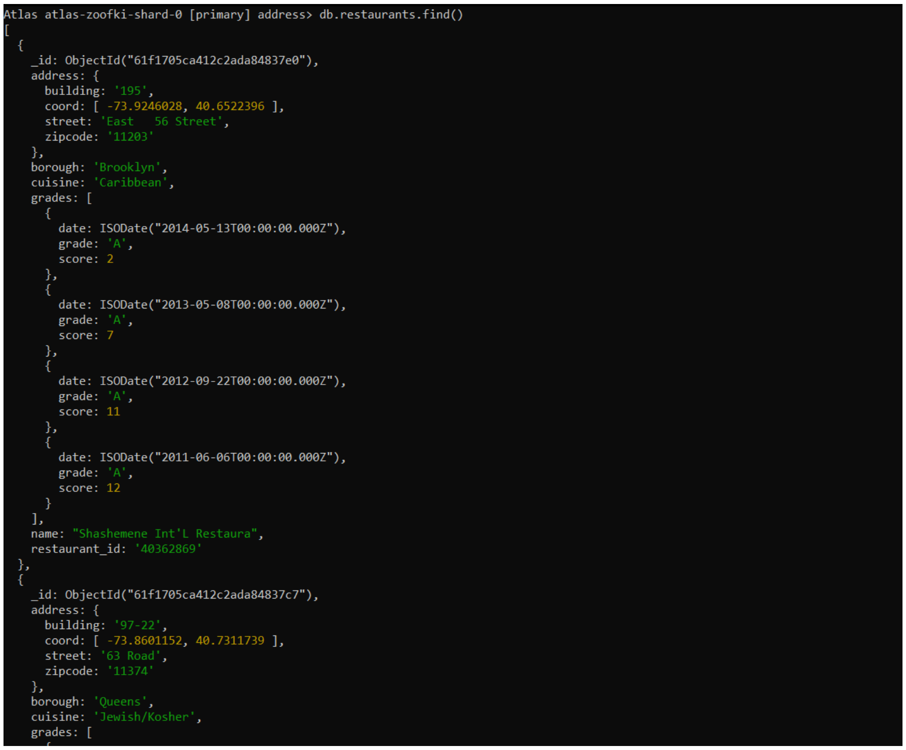
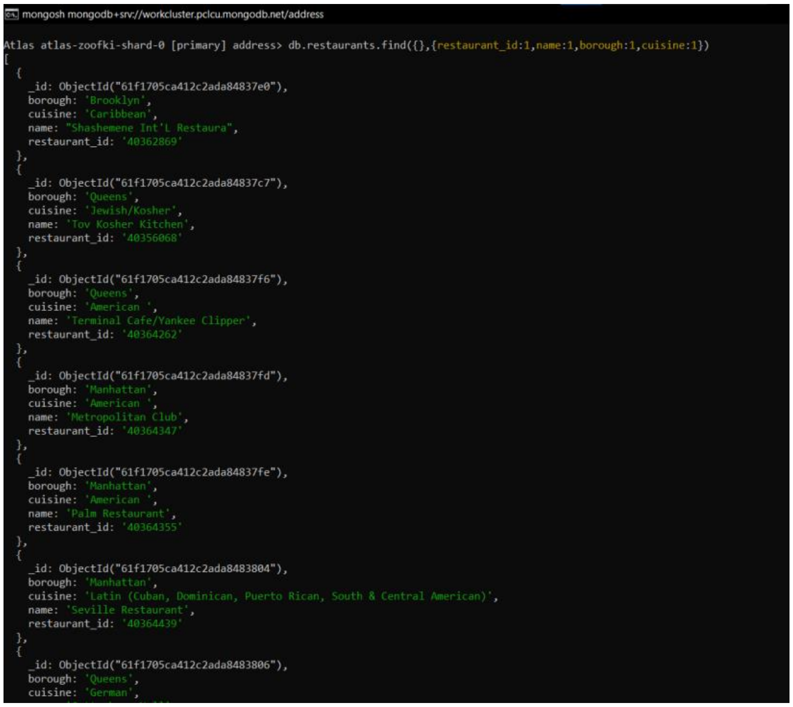
Exercise Questions

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



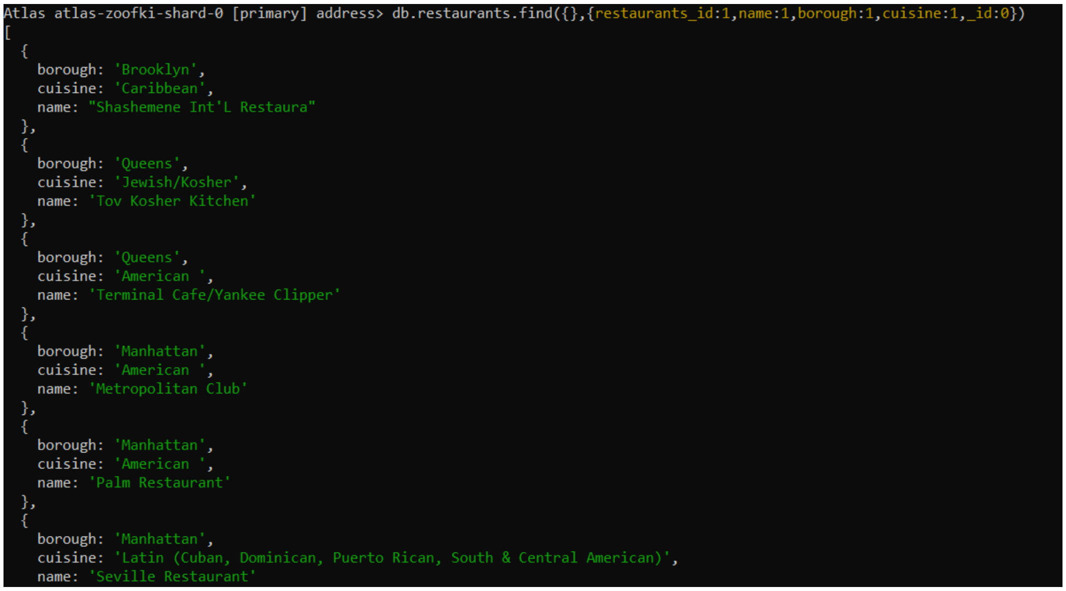
```
Atlas atlas-zoofki-shard-0 [primary] address> db.restaurants.find()
[
  {
    _id: ObjectId("61f1705ca412c2ada84837e0"),
    address: {
      building: '195',
      coord: [ -73.9246028, 40.6522396 ],
      street: 'East   56 Street',
      zipcode: '11203'
    },
    borough: 'Brooklyn',
    cuisine: 'Caribbean',
    grades: [
      {
        date: ISODate("2014-05-13T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-05-08T00:00:00.000Z"),
        grade: 'A',
        score: 7
      },
      {
        date: ISODate("2012-09-22T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
        date: ISODate("2011-06-06T00:00:00.000Z"),
        grade: 'A',
        score: 12
      }
    ],
    name: "Shashemene Int'L Restaura",
    restaurant_id: '40362869'
  },
  {
    _id: ObjectId("61f1705ca412c2ada84837c7"),
    address: {
      building: '97-22',
      coord: [ -73.8601152, 40.7311739 ],
      street: '63 Road',
      zipcode: '11374'
    },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    grades: [
```

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



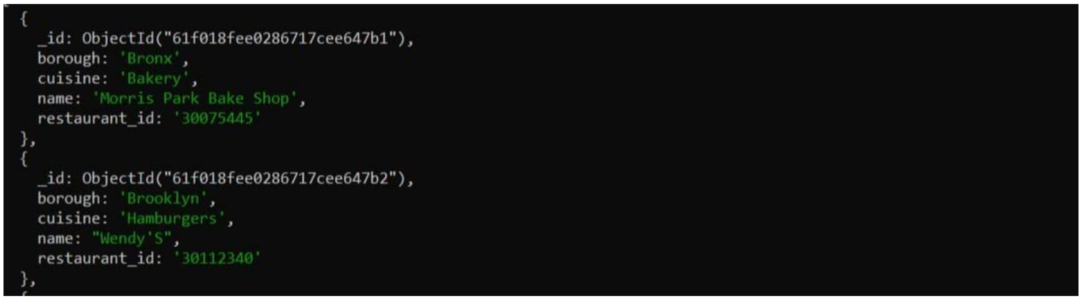
```
Atlas atlas-zoofki-shard-0 [primary] address> db.restaurants.find({},{restaurant_id:1,name:1,borough:1,cuisine:1})
[
  {
    _id: ObjectId("61f1705ca412c2ada84837e0"),
    borough: 'Brooklyn',
    cuisine: 'Caribbean',
    name: "Shashemene Int'L Restaura",
    restaurant_id: '40362869'
  },
  {
    _id: ObjectId("61f1705ca412c2ada84837c7"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen',
    restaurant_id: '40356068'
  },
  {
    _id: ObjectId("61f1705ca412c2ada84837f6"),
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Terminal Cafe/Yankee Clipper',
    restaurant_id: '40364262'
  },
  {
    _id: ObjectId("61f1705ca412c2ada84837fd"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Metropolitan Club',
    restaurant_id: '40364347'
  },
  {
    _id: ObjectId("61f1705ca412c2ada84837fe"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Palm Restaurant',
    restaurant_id: '40364355'
  },
  {
    _id: ObjectId("61f1705ca412c2ada8483804"),
    borough: 'Manhattan',
    cuisine: 'Latin (Cuban, Dominican, Puerto Rican, South & Central American)',
    name: 'Seville Restaurant',
    restaurant_id: '40364439'
  },
  {
    _id: ObjectId("61f1705ca412c2ada8483806"),
    borough: 'Queens',
    cuisine: 'German',
```

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



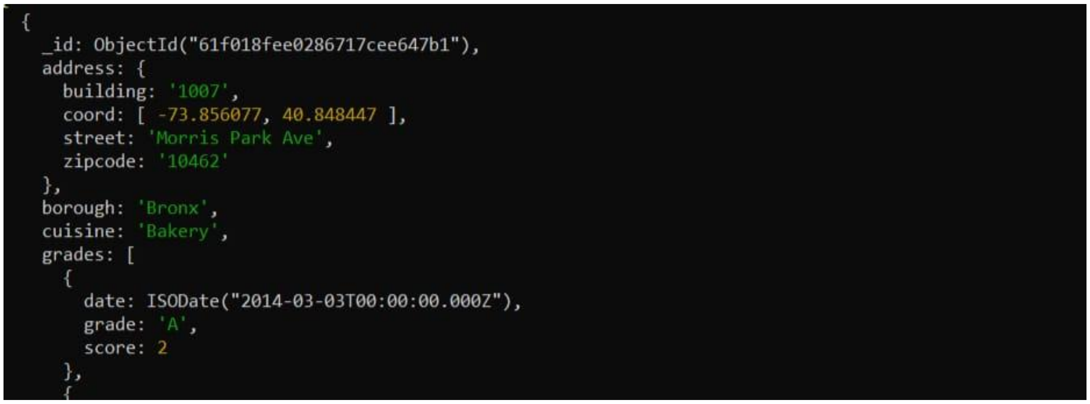
```
Atlas atlas-zoofki-shard-0 [primary] address> db.restaurants.find({},{restaurants_id:1,name:1,borough:1,cuisine:1,_id:0})
[
  {
    borough: 'Brooklyn',
    cuisine: 'Caribbean',
    name: "Shashemene Int'L Restaura"
  },
  {
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    borough: 'Queens',
    cuisine: 'American ',
    name: 'Terminal Cafe/Yankee Clipper'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Metropolitan Club'
  },
  {
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Palm Restaurant'
  },
  {
    borough: 'Manhattan',
    cuisine: 'Latin (Cuban, Dominican, Puerto Rican, South & Central American)',
    name: 'Seville Restaurant'
```

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



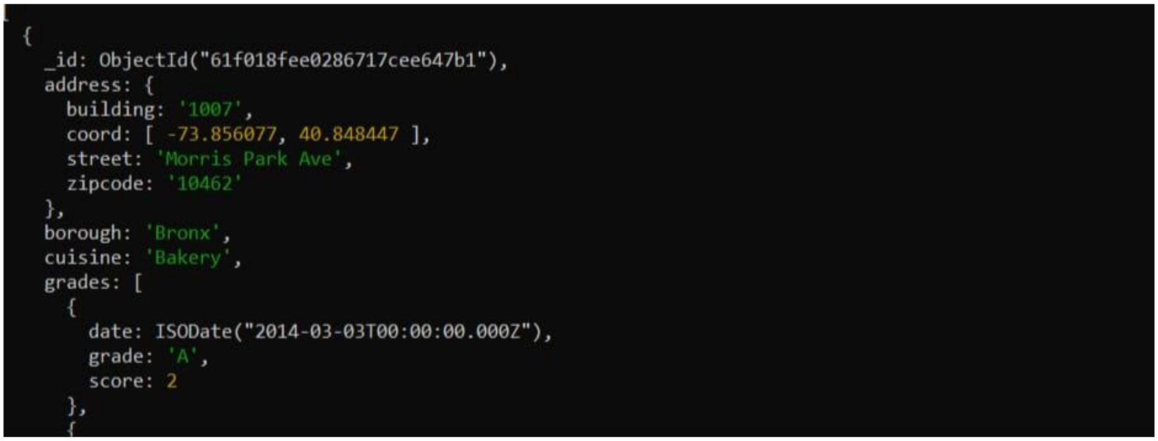
```
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f018fee0286717cee647b2"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
```

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



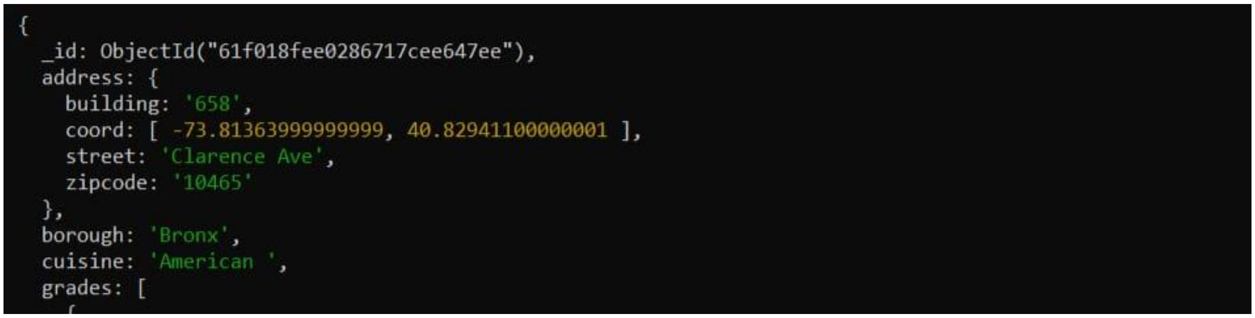
```
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
```

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



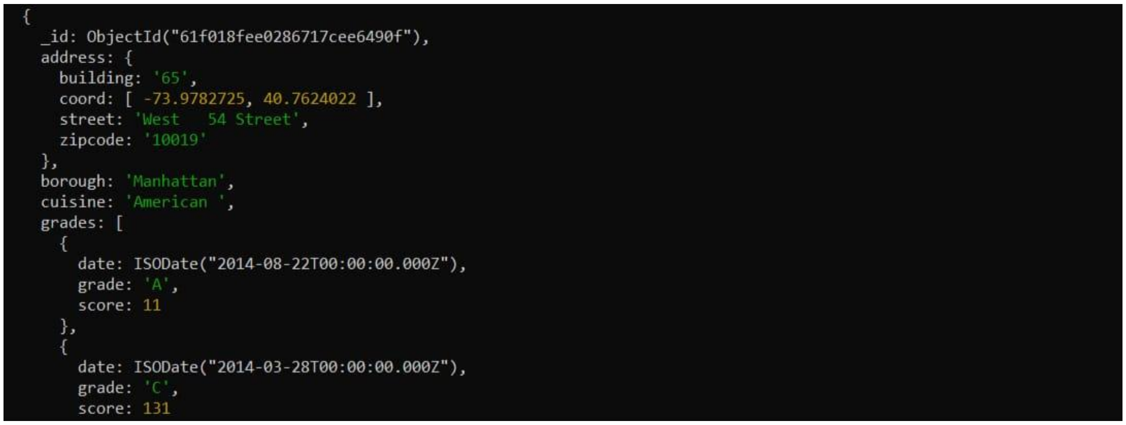
```
{
  _id: ObjectId("61f018fee0286717cee647b1"),
  address: {
    building: '1007',
    coord: [ -73.856077, 40.848447 ],
    street: 'Morris Park Ave',
    zipcode: '10462'
  },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades: [
    {
      date: ISODate("2014-03-03T00:00:00.000Z"),
      grade: 'A',
      score: 2
    },
    {
```

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



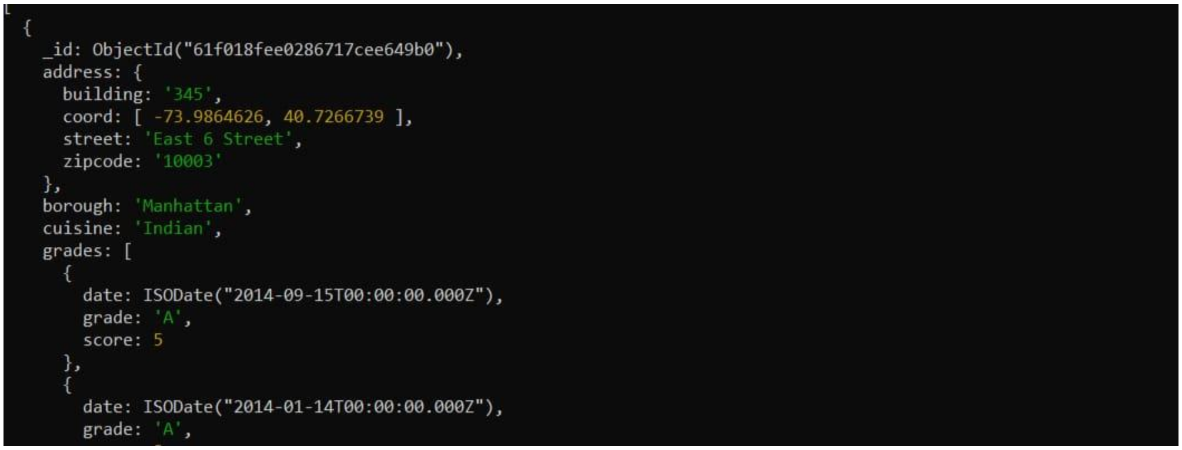
```
{
  _id: ObjectId("61f018fee0286717cee647ee"),
  address: {
    building: '658',
    coord: [ -73.81363999999999, 40.82941100000001 ],
    street: 'Clarence Ave',
    zipcode: '10465'
  },
  borough: 'Bronx',
  cuisine: 'American ',
  grades: [
    {
```

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



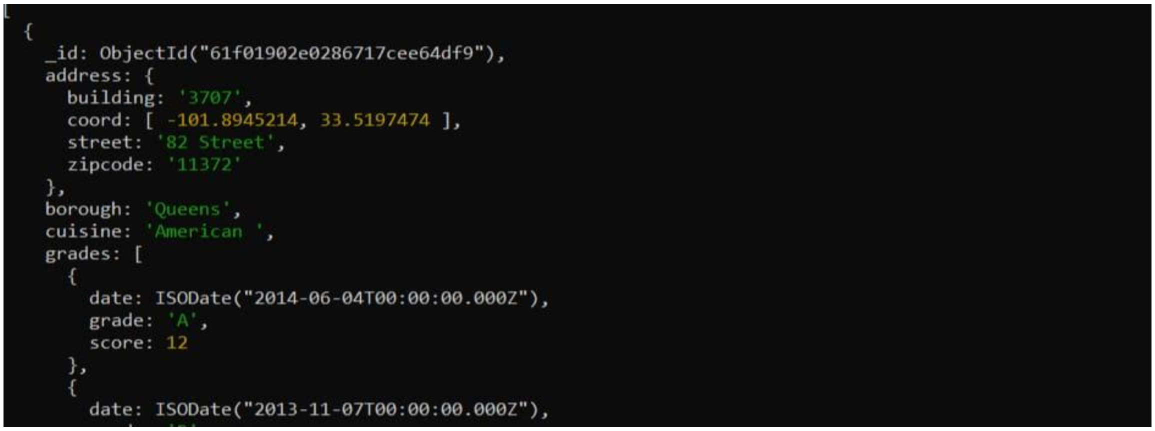
```
{
  _id: ObjectId("61f018fee0286717cee6490f"),
  address: {
    building: '65',
    coord: [ -73.9782725, 40.7624022 ],
    street: 'West   54 Street',
    zipcode: '10019'
  },
  borough: 'Manhattan',
  cuisine: 'American ',
  grades: [
    {
      date: ISODate("2014-08-22T00:00:00.000Z"),
      grade: 'A',
      score: 11
    },
    {
      date: ISODate("2014-03-28T00:00:00.000Z"),
      grade: 'C',
      score: 131
```

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



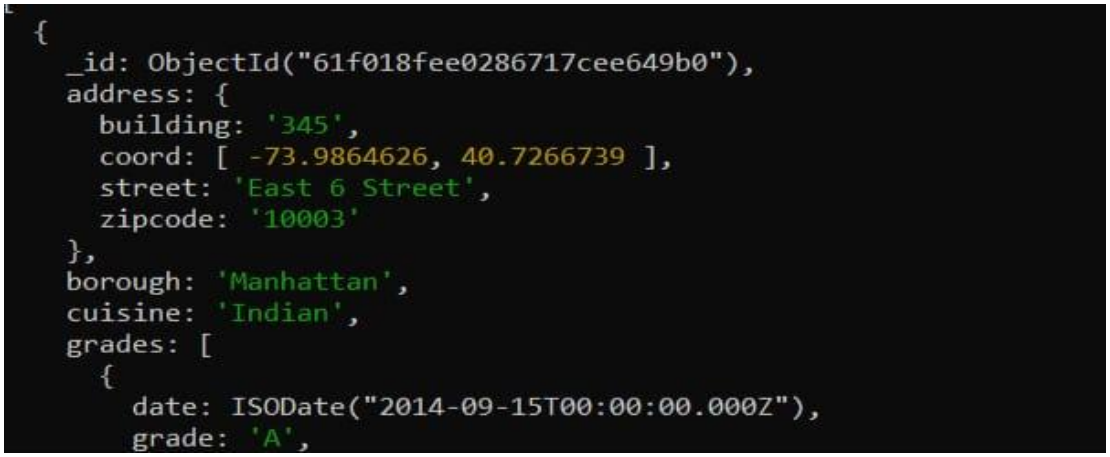
```
[
  {
    _id: ObjectId("61f018fee0286717cee649b0"),
    address: {
      building: '345',
      coord: [ -73.9864626, 40.7266739 ],
      street: 'East 6 Street',
      zipcode: '10003'
    },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades: [
      {
        date: ISODate("2014-09-15T00:00:00.000Z"),
        grade: 'A',
        score: 5
      },
      {
        date: ISODate("2014-01-14T00:00:00.000Z"),
        grade: 'A',
```

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



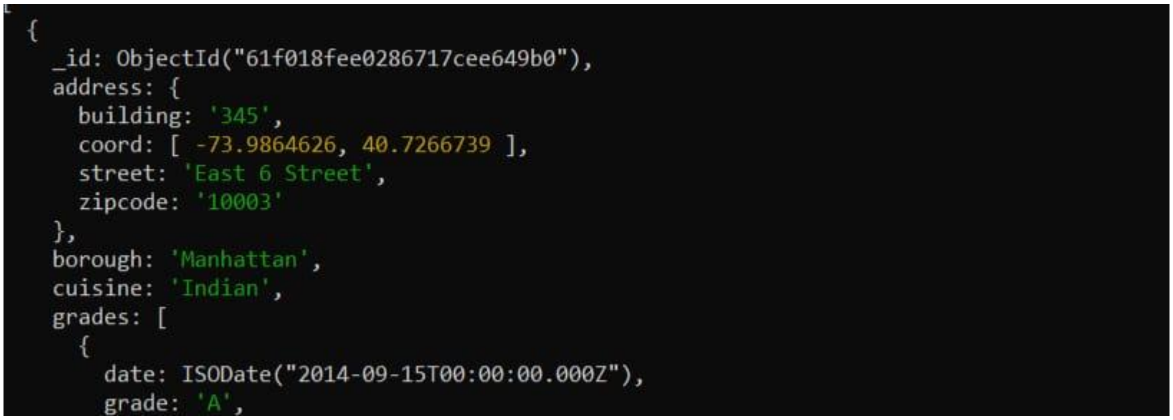
```
[
  {
    _id: ObjectId("61f01902e0286717cee64df9"),
    address: {
      building: '3707',
      coord: [ -101.8945214, 33.5197474 ],
      street: '82 Street',
      zipcode: '11372'
    },
    borough: 'Queens',
    cuisine: 'American ',
    grades: [
      {
        date: ISODate("2014-06-04T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2013-11-07T00:00:00.000Z"),
```

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.



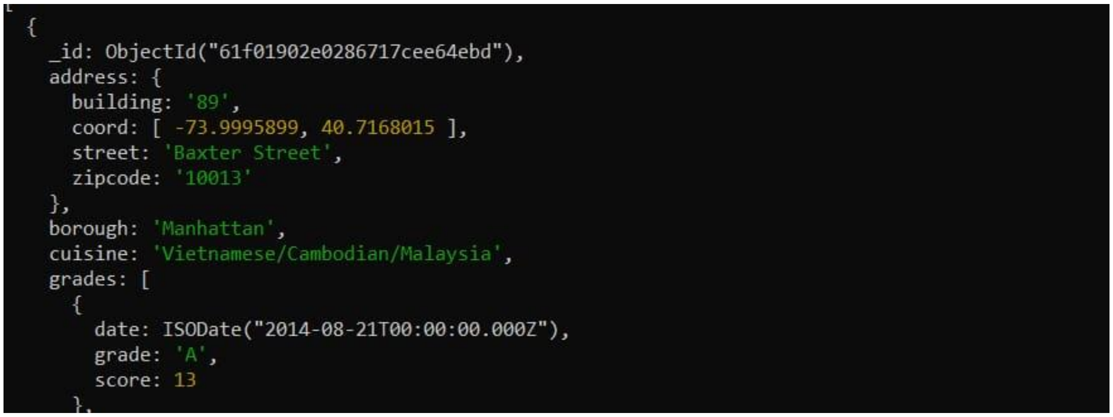
```
[
  {
    _id: ObjectId("61f018fee0286717cee649b0"),
    address: {
      building: '345',
      coord: [ -73.9864626, 40.7266739 ],
      street: 'East 6 Street',
      zipcode: '10003'
    },
    borough: 'Manhattan',
    cuisine: 'Indian',
    grades: [
      {
        date: ISODate("2014-09-15T00:00:00.000Z"),
        grade: 'A',
```

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.



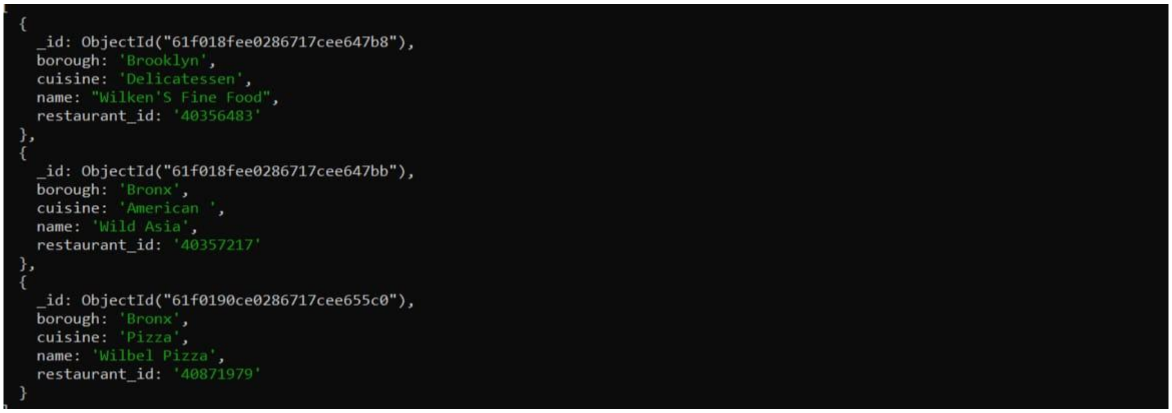
```
{
  _id: ObjectId("61f018fee0286717cee649b0"),
  address: {
    building: '345',
    coord: [ -73.9864626, 40.7266739 ],
    street: 'East 6 Street',
    zipcode: '10003'
  },
  borough: 'Manhattan',
  cuisine: 'Indian',
  grades: [
    {
      date: ISODate("2014-09-15T00:00:00.000Z"),
      grade: 'A',
```

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



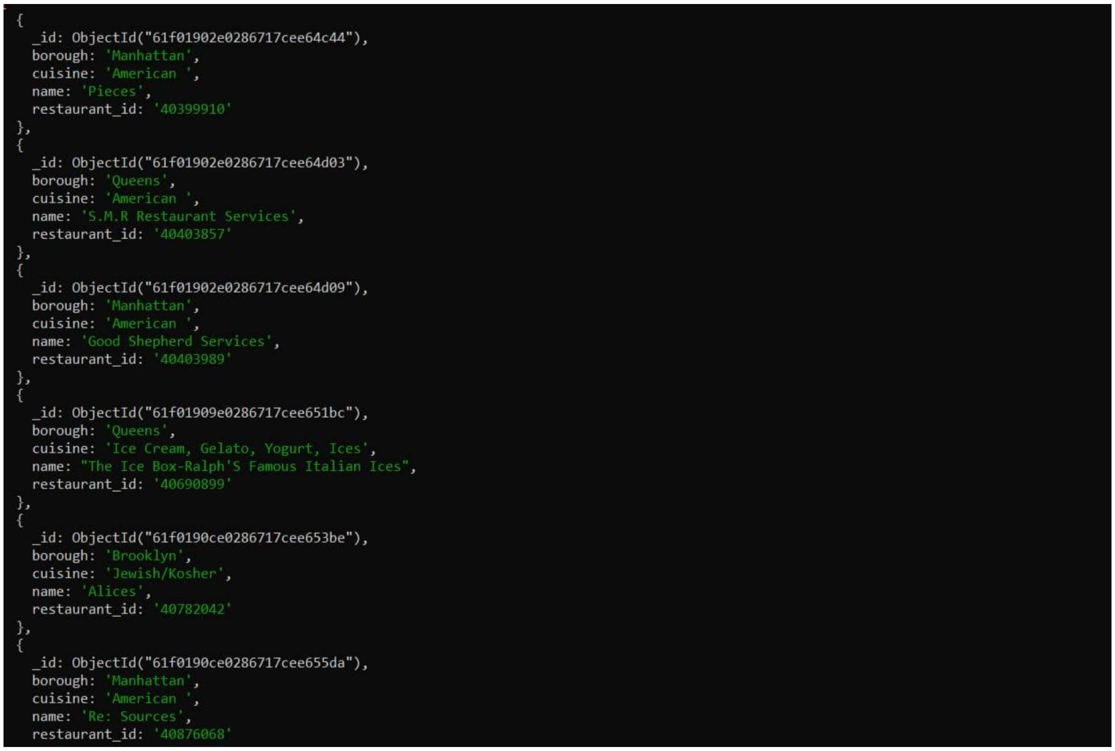
```
{
  _id: ObjectId("61f01902e0286717cee64ebd"),
  address: {
    building: '89',
    coord: [ -73.9995899, 40.7168015 ],
    street: 'Baxter Street',
    zipcode: '10013'
  },
  borough: 'Manhattan',
  cuisine: 'Vietnamese/Cambodian/Malaysia',
  grades: [
    {
      date: ISODate("2014-08-21T00:00:00.000Z"),
      grade: 'A',
      score: 13
    },
```

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



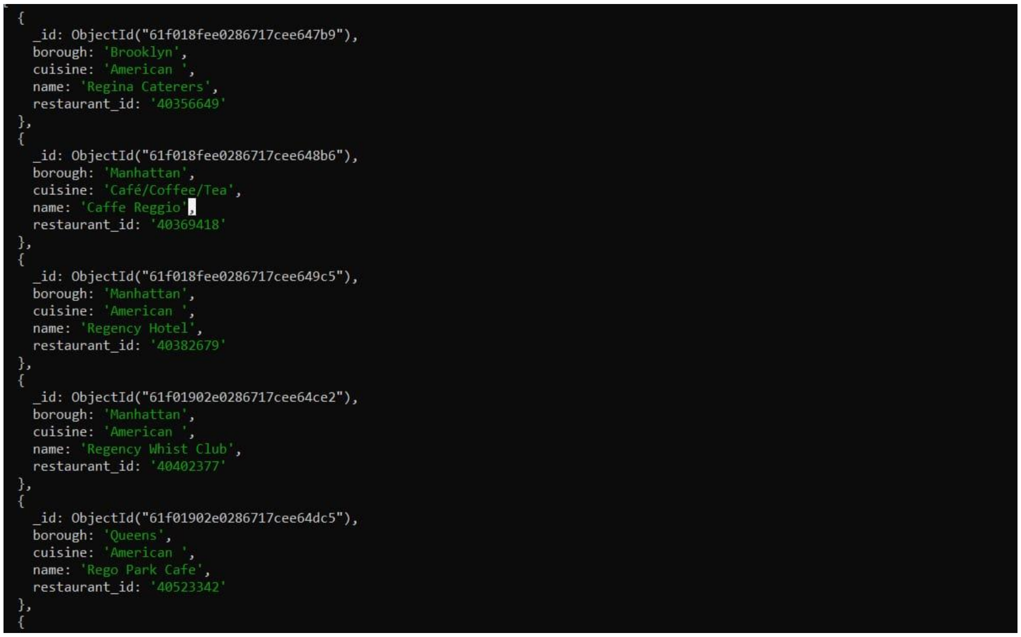
```
{
  _id: ObjectId("61f018fee0286717cee647b8"),
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  name: "Wilken'S Fine Food",
  restaurant_id: '40356483'
},
{
  _id: ObjectId("61f018fee0286717cee647bb"),
  borough: 'Bronx',
  cuisine: 'American ',
  name: 'Wild Asia',
  restaurant_id: '40357217'
},
{
  _id: ObjectId("61f0190ce0286717cee655c0"),
  borough: 'Bronx',
  cuisine: 'Pizza',
  name: 'Wilbel Pizza',
  restaurant_id: '40871979'
}
```

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.



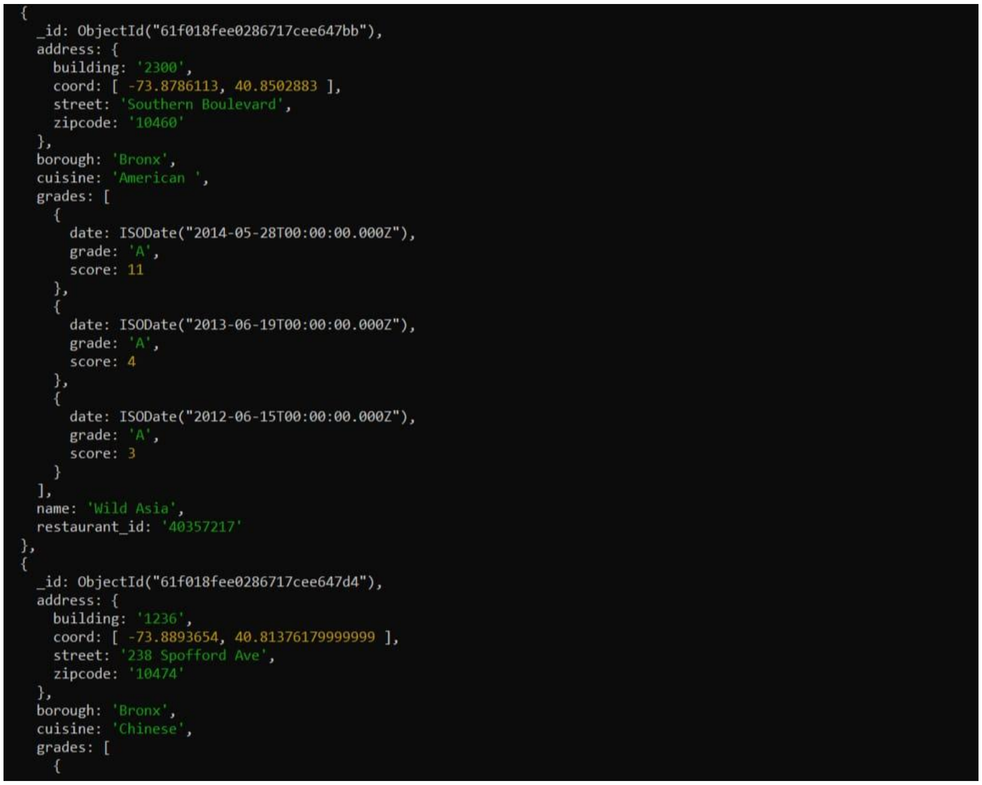
```
{
  _id: ObjectId("61f01902e0286717cee64c44"),
  borough: 'Manhattan',
  cuisine: 'American ',
  name: 'Pieces',
  restaurant_id: '40399910'
},
{
  _id: ObjectId("61f01902e0286717cee64d03"),
  borough: 'Queens',
  cuisine: 'American ',
  name: 'S.M.R Restaurant Services',
  restaurant_id: '40403857'
},
{
  _id: ObjectId("61f01902e0286717cee64d09"),
  borough: 'Manhattan',
  cuisine: 'American ',
  name: 'Good Shepherd Services',
  restaurant_id: '40403989'
},
{
  _id: ObjectId("61f01909e0286717cee651bc"),
  borough: 'Queens',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  name: "The Ice Box-Ralph'S Famous Italian Ices",
  restaurant_id: '40690899'
},
{
  _id: ObjectId("61f0190ce0286717cee653be"),
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Alices',
  restaurant_id: '40782042'
},
{
  _id: ObjectId("61f0190ce0286717cee655da"),
  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



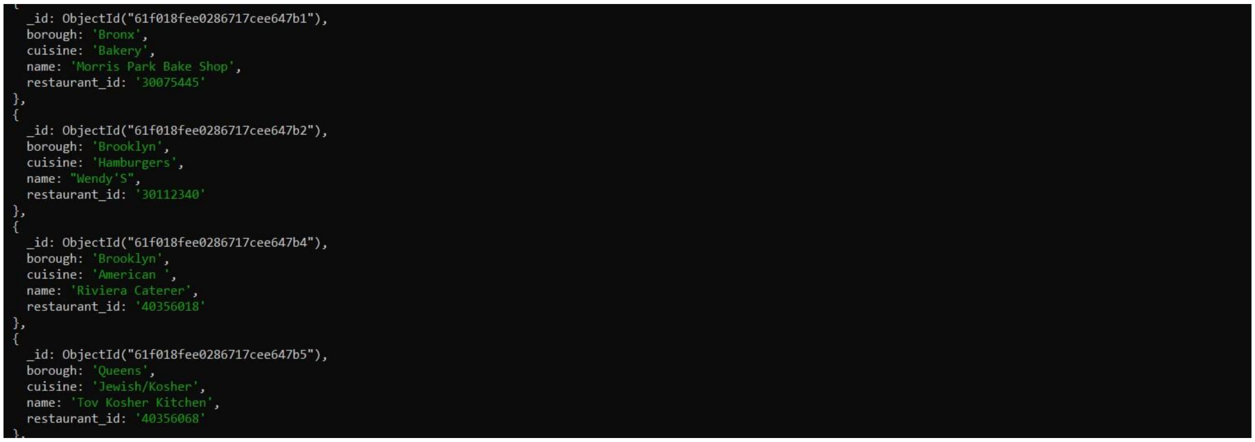
```
{
  _id: ObjectId("61f018fee0286717cee647b9"),
  borough: 'Brooklyn',
  cuisine: 'American ',
  name: 'Regina Caterers',
  restaurant_id: '40356649'
},
{
  _id: ObjectId("61f018fee0286717cee648b6"),
  borough: 'Manhattan',
  cuisine: 'Café/Coffee/Tea',
  name: 'Caffe Reggio',
  restaurant_id: '40369418'
},
{
  _id: ObjectId("61f018fee0286717cee649c5"),
  borough: 'Manhattan',
  cuisine: 'American ',
  name: 'Regency Hotel',
  restaurant_id: '40382679'
},
{
  _id: ObjectId("61f01902e0286717cee64ce2"),
  borough: 'Manhattan',
  cuisine: 'American ',
  name: 'Regency Whist Club',
  restaurant_id: '40402377'
},
{
  _id: ObjectId("61f01902e0286717cee64dc5"),
  borough: 'Queens',
  cuisine: 'American ',
  name: 'Rego Park Cafe',
  restaurant_id: '40523342'
},
{
```

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



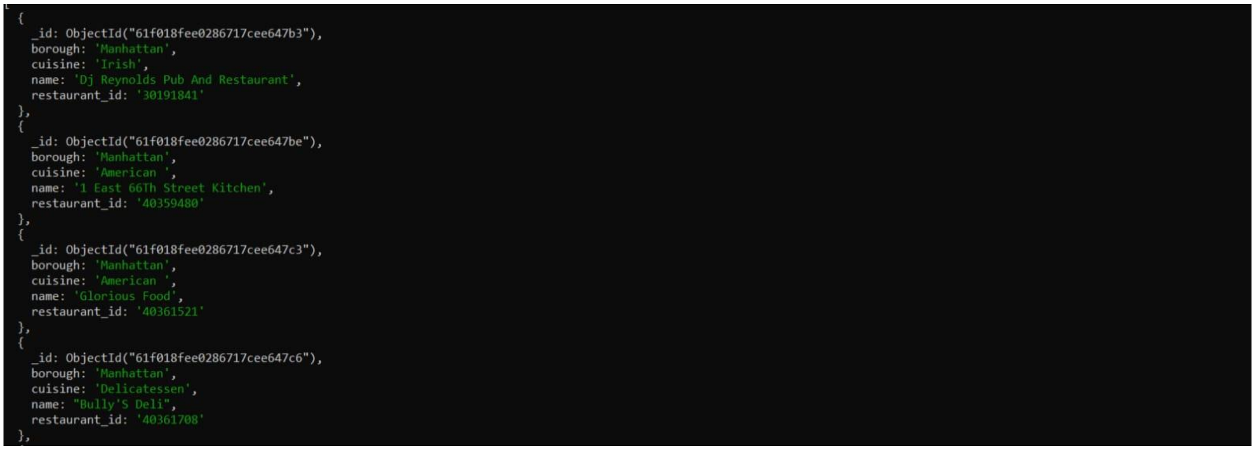
```
{
  _id: ObjectId("61f018fee0286717cee647bb"),
  address: {
    building: '2300',
    coord: [ -73.8786113, 40.8502883 ],
    street: 'Southern Boulevard',
    zipcode: '10460'
  },
  borough: 'Bronx',
  cuisine: 'American ',
  grades: [
    {
      date: ISODate("2014-05-28T00:00:00.000Z"),
      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("61f018fee0286717cee647d4"),
  address: {
    building: '1236',
    coord: [ -73.8893654, 40.81376179999999 ],
    street: '238 Spofford Ave',
    zipcode: '10474'
  },
  borough: 'Bronx',
  cuisine: 'Chinese',
  grades: [
    {
```

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.



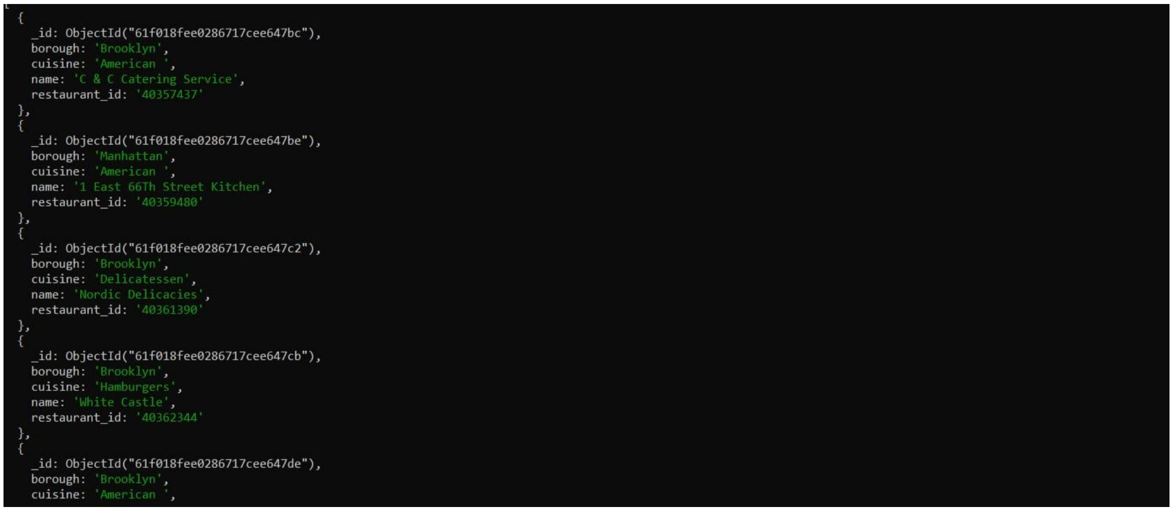
```
{
  _id: ObjectId("61f018fee0286717cee647b1"),
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
},
{
  _id: ObjectId("61f018fee0286717cee647b2"),
  borough: 'Brooklyn',
  cuisine: 'Hamburgers',
  name: "Wendy'S",
  restaurant_id: '30112340'
},
{
  _id: ObjectId("61f018fee0286717cee647b4"),
  borough: 'Brooklyn',
  cuisine: 'American ',
  name: 'Riviera Caterer',
  restaurant_id: '40356018'
},
{
  _id: ObjectId("61f018fee0286717cee647b5"),
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  name: 'Tov Kosher Kitchen',
  restaurant_id: '40356068'
},
```

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



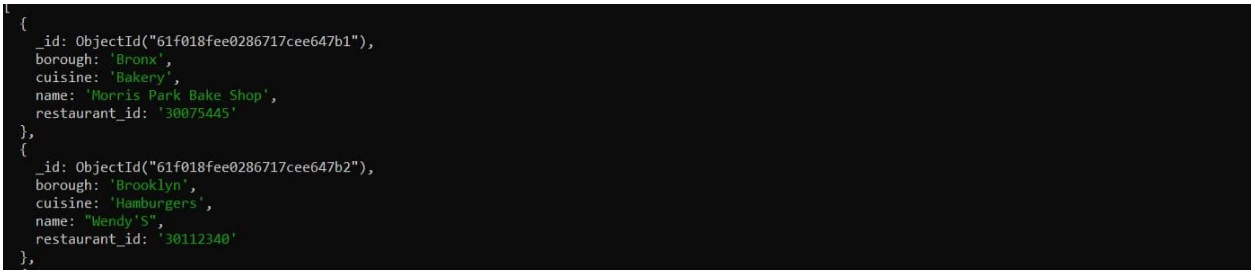
```
  {
    _id: ObjectId("61f018fee0286717cee647b3"),
    borough: 'Manhattan',
    cuisine: 'Irish',
    name: 'Dj Reynolds Pub And Restaurant',
    restaurant_id: '30191841'
  },
  {
    _id: ObjectId("61f018fee0286717cee647be"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f018fee0286717cee647c3"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Glorious Food',
    restaurant_id: '40361521'
  },
  {
    _id: ObjectId("61f018fee0286717cee647c6"),
    borough: 'Manhattan',
    cuisine: 'Delicatessen',
    name: "Bully'S Deli",
    restaurant_id: '40361708'
  },
```

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



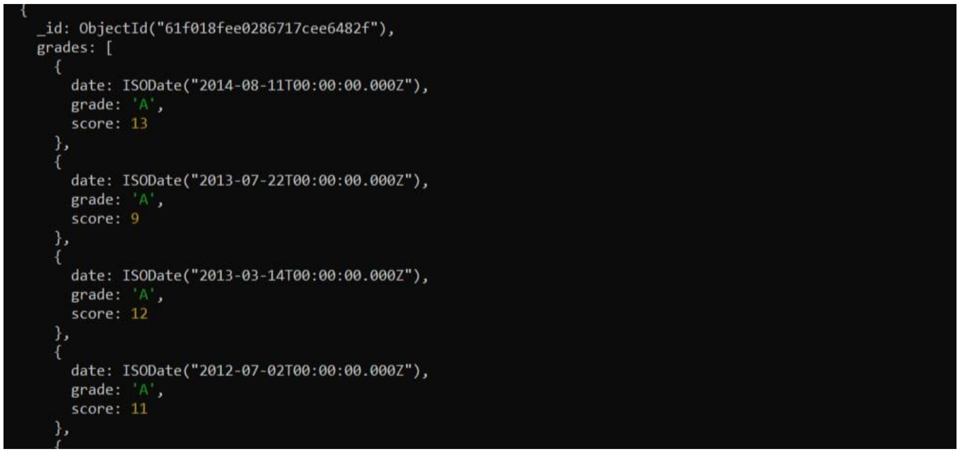
```
  {
    _id: ObjectId("61f018fee0286717cee647bc"),
    borough: 'Brooklyn',
    cuisine: 'American ',
    name: 'C & C Catering Service',
    restaurant_id: '40357437'
  },
  {
    _id: ObjectId("61f018fee0286717cee647be"),
    borough: 'Manhattan',
    cuisine: 'American ',
    name: '1 East 66Th Street Kitchen',
    restaurant_id: '40359480'
  },
  {
    _id: ObjectId("61f018fee0286717cee647c2"),
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: 'Nordic Delicacies',
    restaurant_id: '40361390'
  },
  {
    _id: ObjectId("61f018fee0286717cee647cb"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: 'White Castle',
    restaurant_id: '40362344'
  },
  {
    _id: ObjectId("61f018fee0286717cee647de"),
    borough: 'Brooklyn',
    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'.



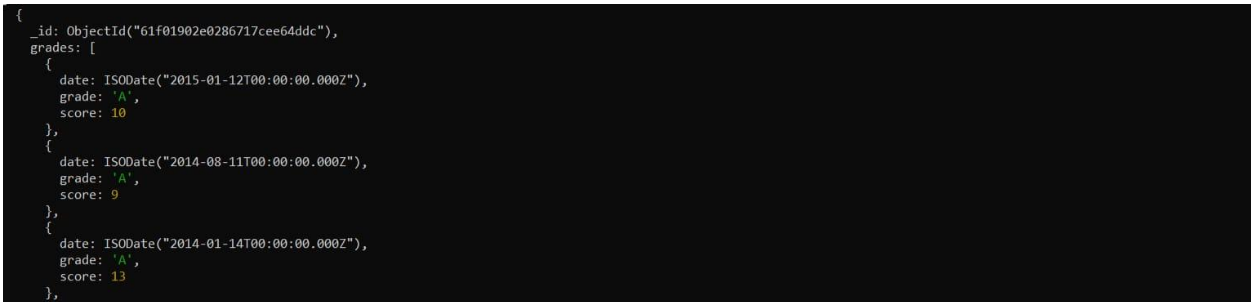
```
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    borough: 'Bronx',
    cuisine: 'Bakery',
    name: 'Morris Park Bake Shop',
    restaurant_id: '30075445'
  },
  {
    _id: ObjectId("61f018fee0286717cee647b2"),
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
  },
```

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



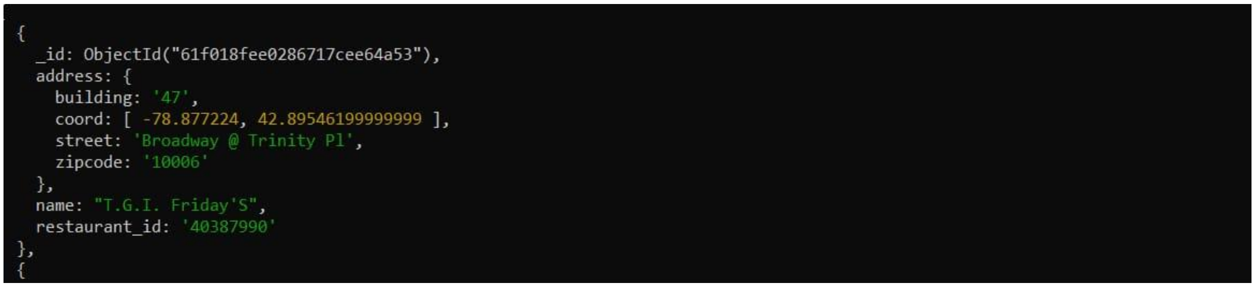
```
  {
    _id: ObjectId("61f018fee0286717cee6482f"),
    grades: [
      {
        date: ISODate("2014-08-11T00:00:00.000Z"),
        grade: 'A',
        score: 13
      },
      {
        date: ISODate("2013-07-22T00:00:00.000Z"),
        grade: 'A',
        score: 9
      },
      {
        date: ISODate("2013-03-14T00:00:00.000Z"),
        grade: 'A',
        score: 12
      },
      {
        date: ISODate("2012-07-02T00:00:00.000Z"),
        grade: 'A',
        score: 11
      },
      {
```

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"



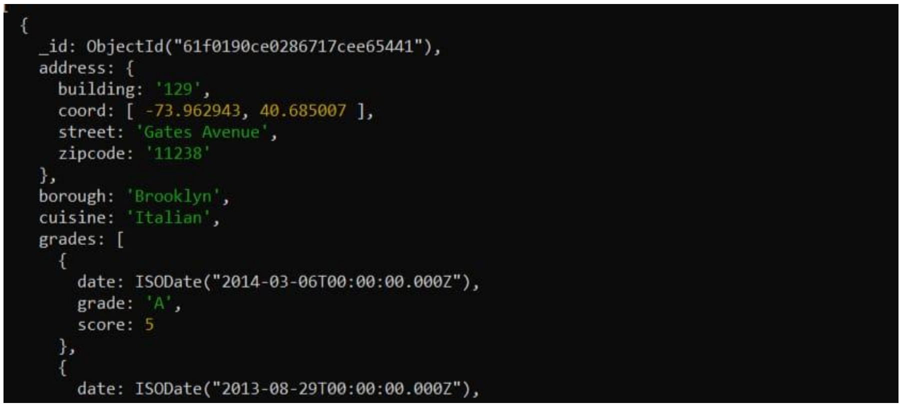
```
{
  _id: ObjectId("61f01902e0286717cee64ddc"),
  grades: [
    {
      date: ISODate("2015-01-12T00:00:00.000Z"),
      grade: 'A',
      score: 10
    },
    {
      date: ISODate("2014-08-11T00:00:00.000Z"),
      grade: 'A',
      score: 9
    },
    {
      date: ISODate("2014-01-14T00:00:00.000Z"),
      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 upto52..



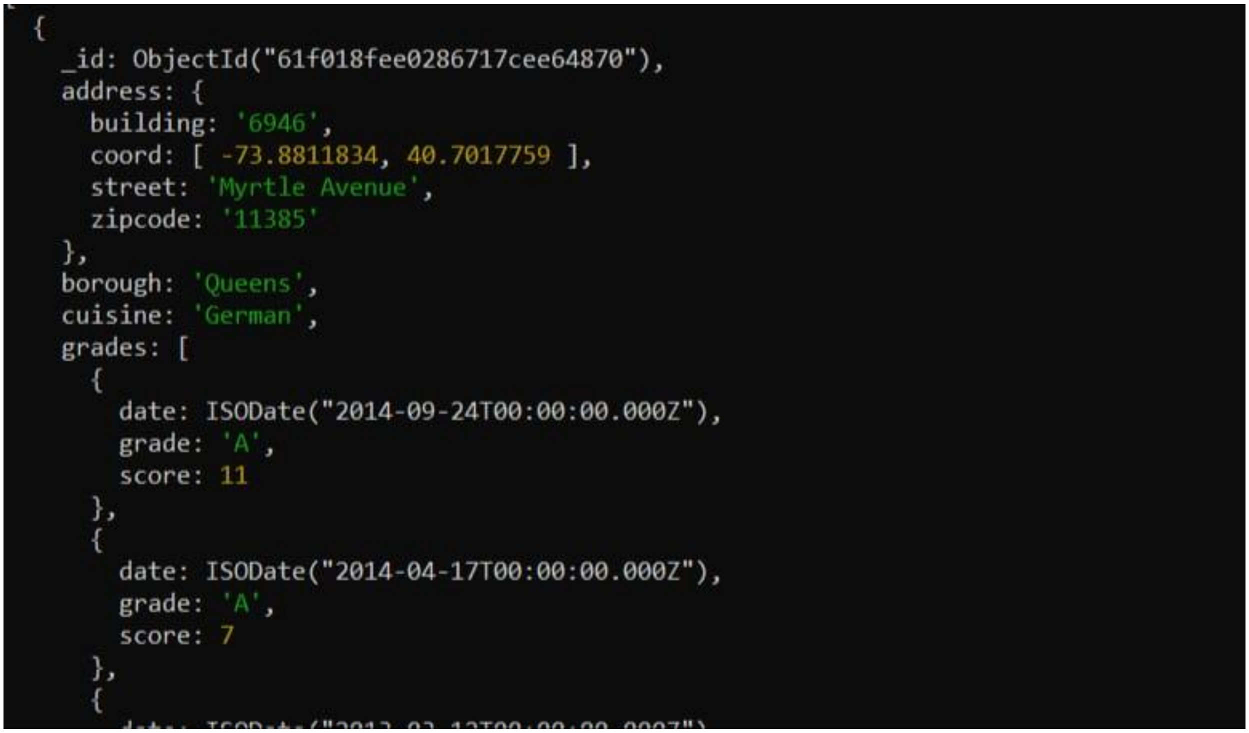
```
{
  _id: ObjectId("61f018fee0286717cee64a53"),
  address: {
    building: '47',
    coord: [ -78.877224, 42.89546199999999 ],
    street: 'Broadway @ Trinity Pl',
    zipcode: '10006'
  },
  name: "T.G.I. Friday'S",
  restaurant_id: '40387990'
},
{
```

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



```
{
  _id: ObjectId("61f0190ce0286717cee65441"),
  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"),
```

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

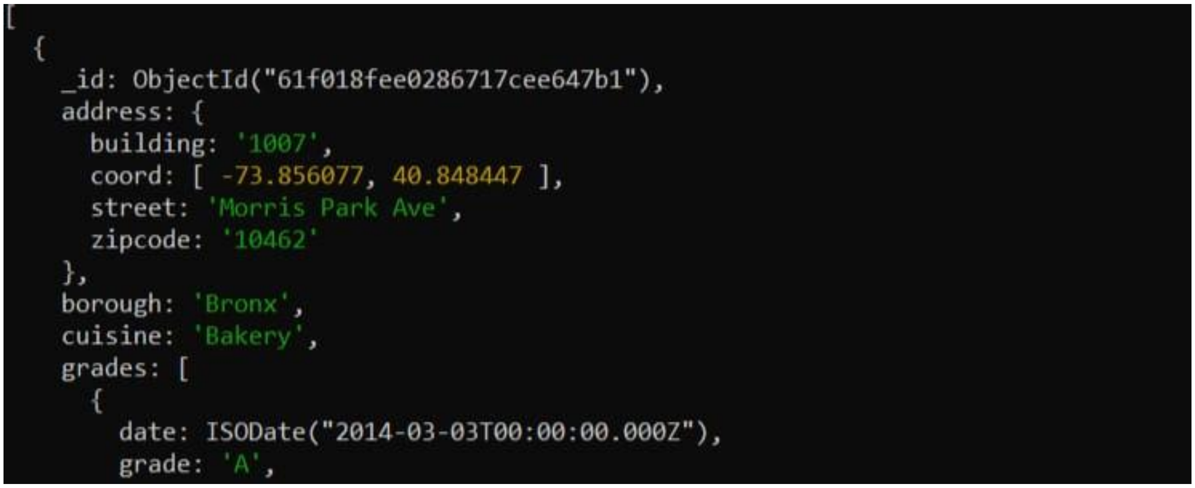


```
{
  _id: ObjectId("61f018fee0286717cee64870"),
  address: {
    building: '6946',
    coord: [ -73.8811834, 40.7017759 ],
    street: 'Myrtle Avenue',
    zipcode: '11385'
  },
  borough: 'Queens',
  cuisine: 'German',
  grades: [
    {
      date: ISODate("2014-09-24T00:00:00.000Z"),
      grade: 'A',
      score: 11
    },
    {
      date: ISODate("2014-04-17T00:00:00.000Z"),
      grade: 'A',
      score: 7
    },
    {
```

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.

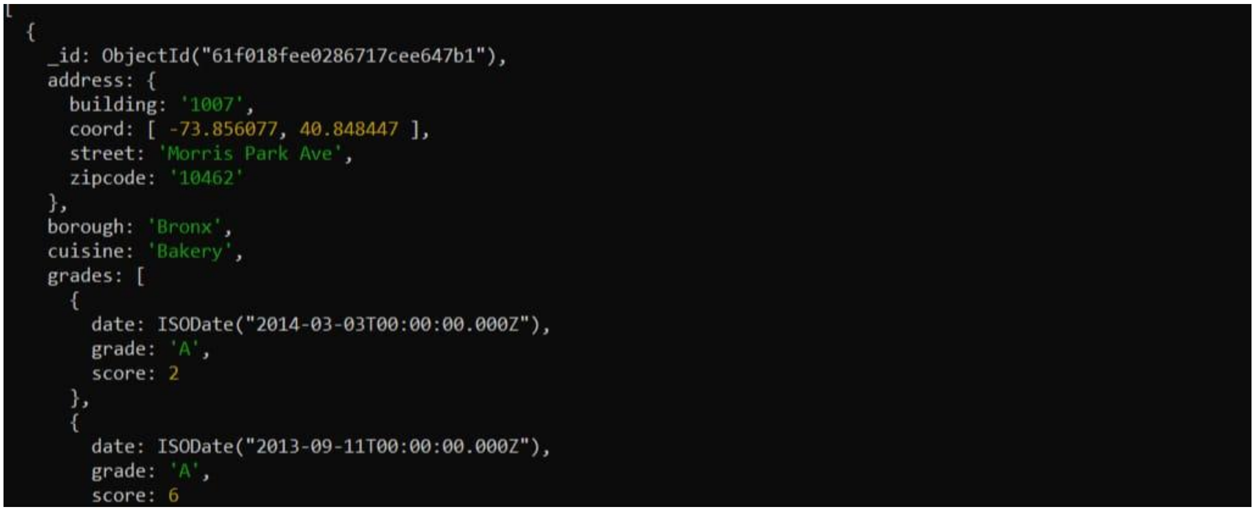
```
db.restaurants.find().sort({"cuisine":1,"borough" : -1,})
```

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



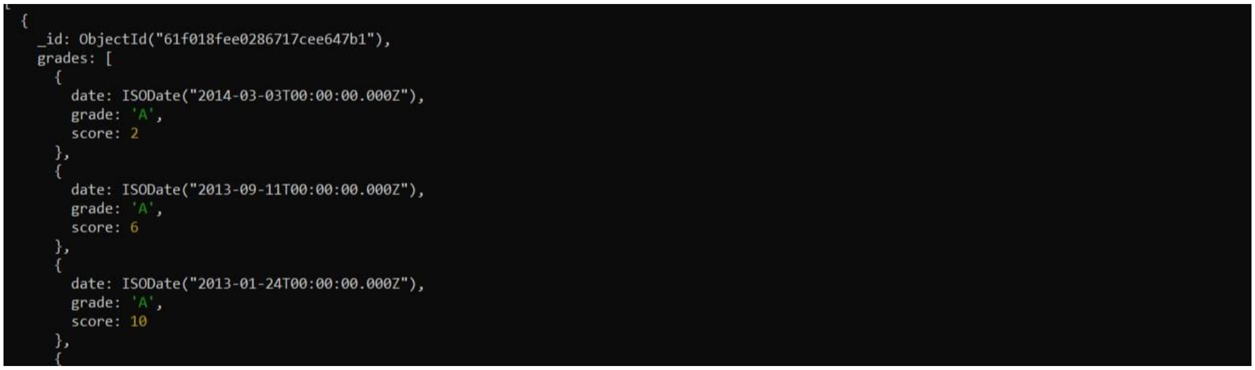
```
[
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
```

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



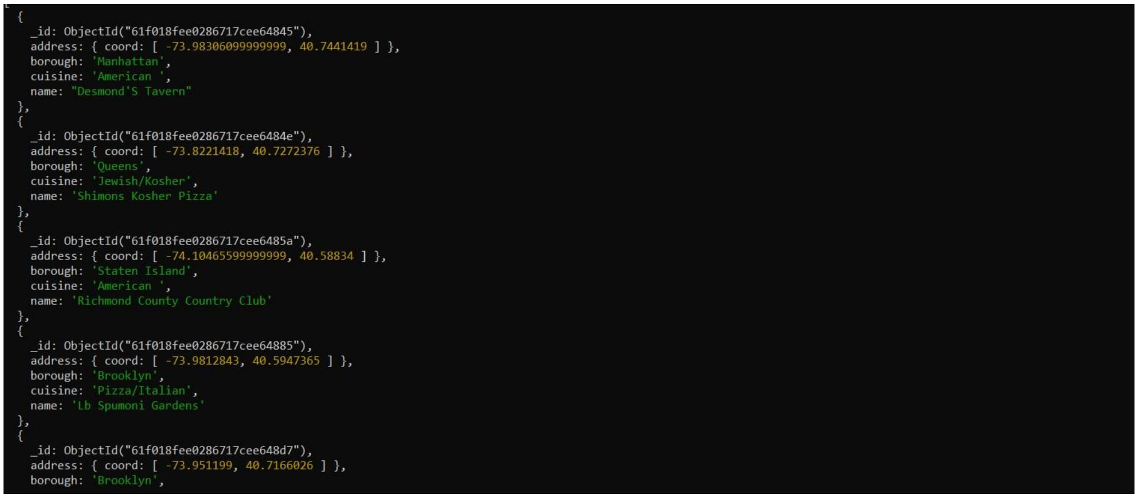
```
[
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    address: {
      building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
```

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.



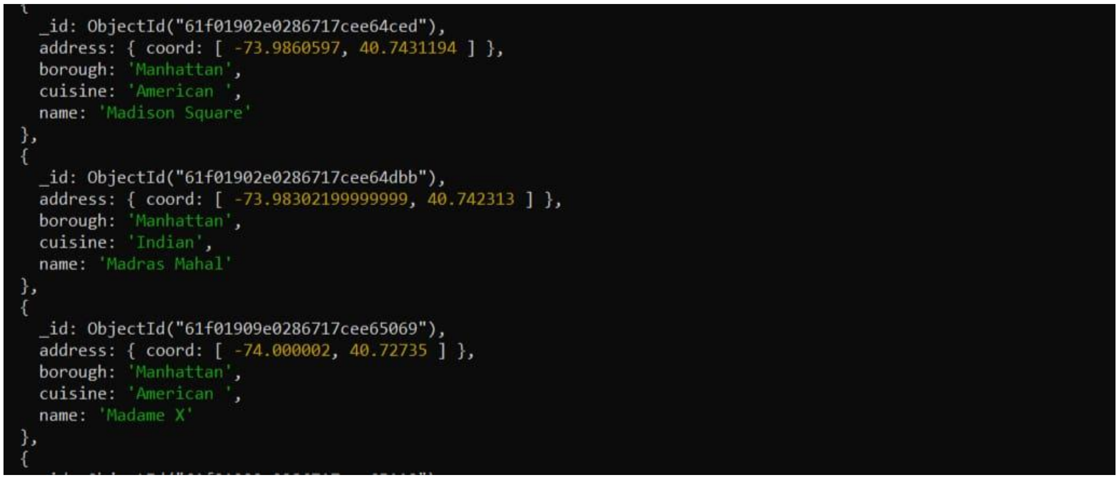
```
[
  {
    _id: ObjectId("61f018fee0286717cee647b1"),
    grades: [
      {
        date: ISODate("2014-03-03T00:00:00.000Z"),
        grade: 'A',
        score: 2
      },
      {
        date: ISODate("2013-09-11T00:00:00.000Z"),
        grade: 'A',
        score: 6
      },
      {
        date: ISODate("2013-01-24T00:00:00.000Z"),
        grade: 'A',
        score: 10
      },
      {
```

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.



```
  {
    _id: ObjectId("61f018fee0286717cee64845"),
    address: { coord: [ -73.98306099999999, 40.7441419 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: "Desmond'S Tavern"
  },
  {
    _id: ObjectId("61f018fee0286717cee6484e"),
    address: { coord: [ -73.8221418, 40.7272376 ] },
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Shimons Kosher Pizza'
  },
  {
    _id: ObjectId("61f018fee0286717cee6485a"),
    address: { coord: [ -74.10465599999999, 40.58834 ] },
    borough: 'Staten Island',
    cuisine: 'American ',
    name: 'Richmond County Country Club'
  },
  {
    _id: ObjectId("61f018fee0286717cee64885"),
    address: { coord: [ -73.9812843, 40.5947365 ] },
    borough: 'Brooklyn',
    cuisine: 'Pizza/Italian',
    name: 'Lb Spumoni Gardens'
  },
  {
    _id: ObjectId("61f018fee0286717cee648d7"),
    address: { coord: [ -73.951199, 40.7166026 ] },
    borough: 'Brooklyn',
```

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



```
    _id: ObjectId("61f01902e0286717cee64ced"),
    address: { coord: [ -73.9860597, 40.7431194 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madison Square'
  },
  {
    _id: ObjectId("61f01902e0286717cee64dbb"),
    address: { coord: [ -73.98302199999999, 40.742313 ] },
    borough: 'Manhattan',
    cuisine: 'Indian',
    name: 'Madras Mahal'
  },
  {
    _id: ObjectId("61f01909e0286717cee65069"),
    address: { coord: [ -74.000002, 40.72735 ] },
    borough: 'Manhattan',
    cuisine: 'American ',
    name: 'Madame X'
  },
  {
```