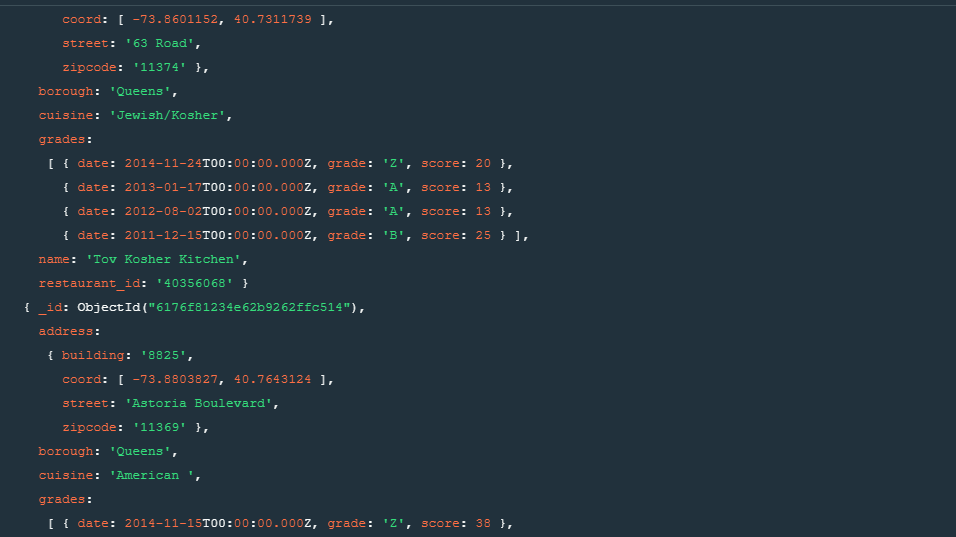
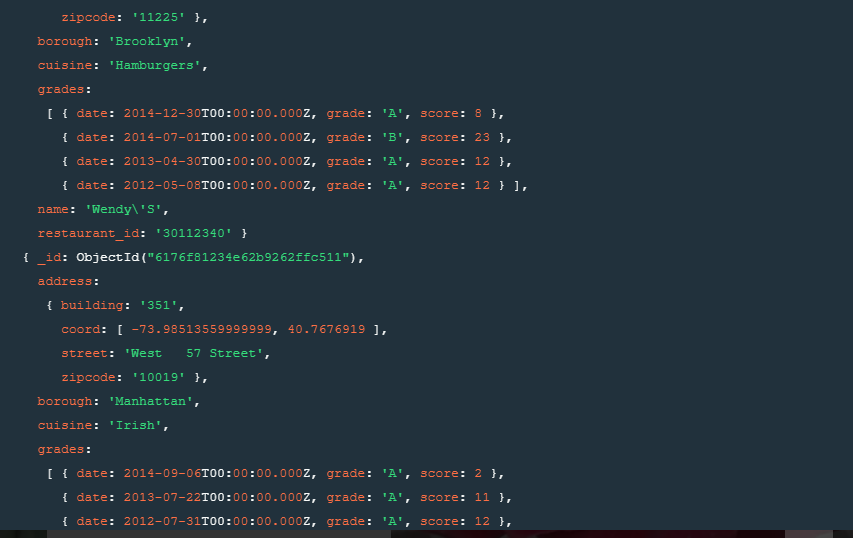
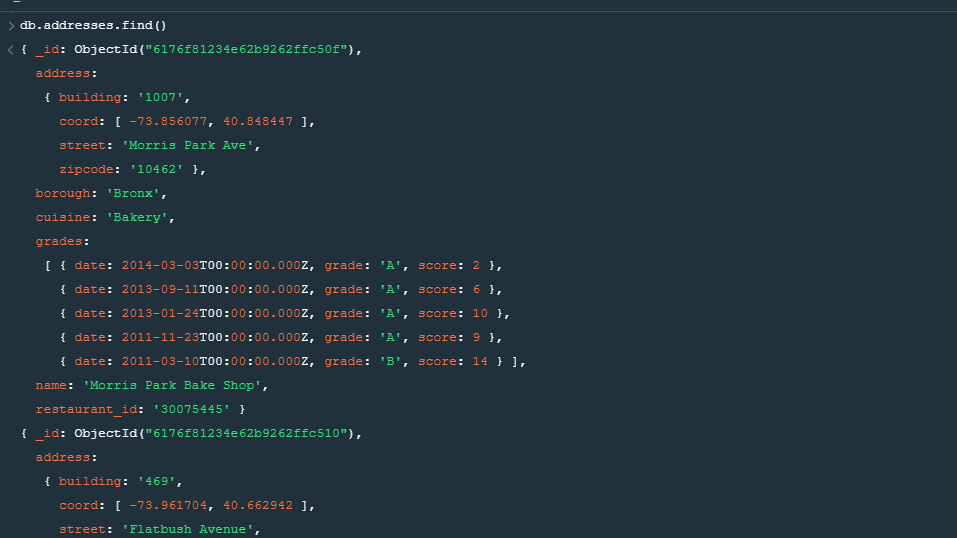
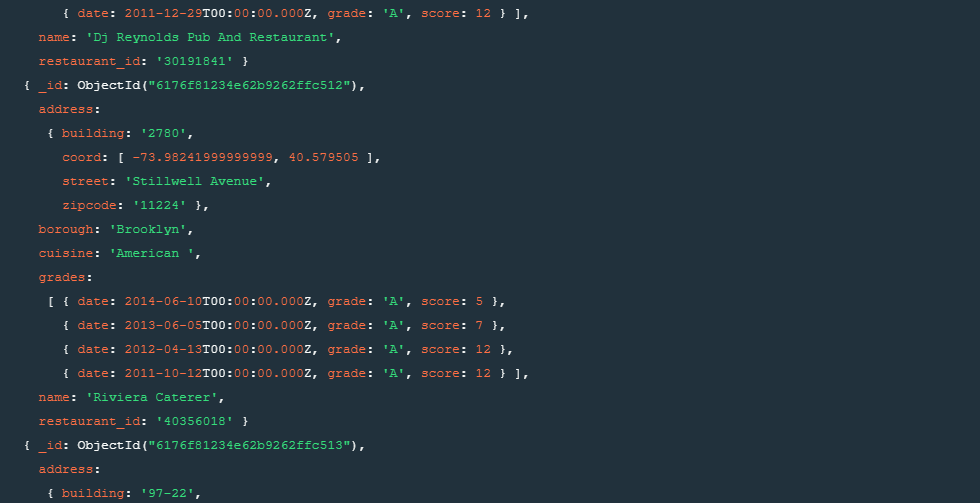
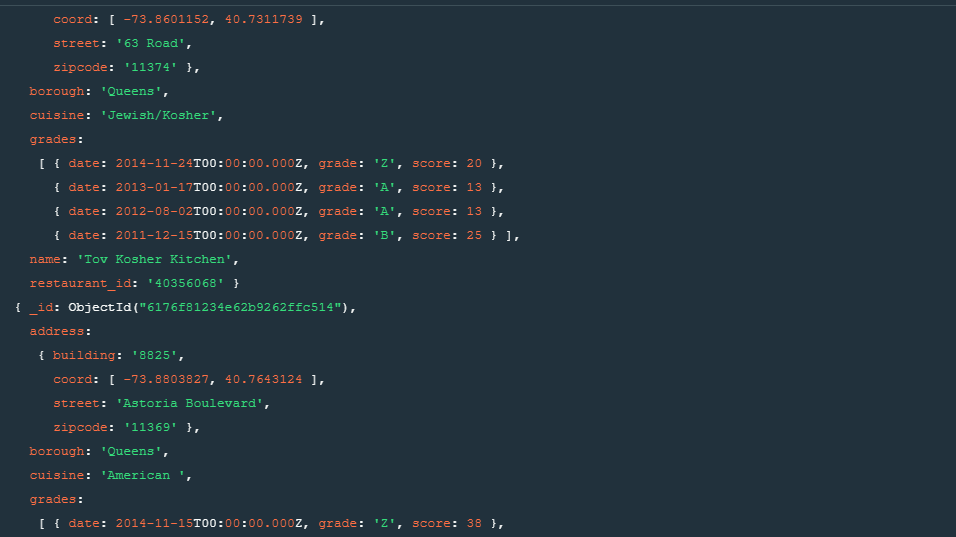
**MongoDB – Complex Queries**

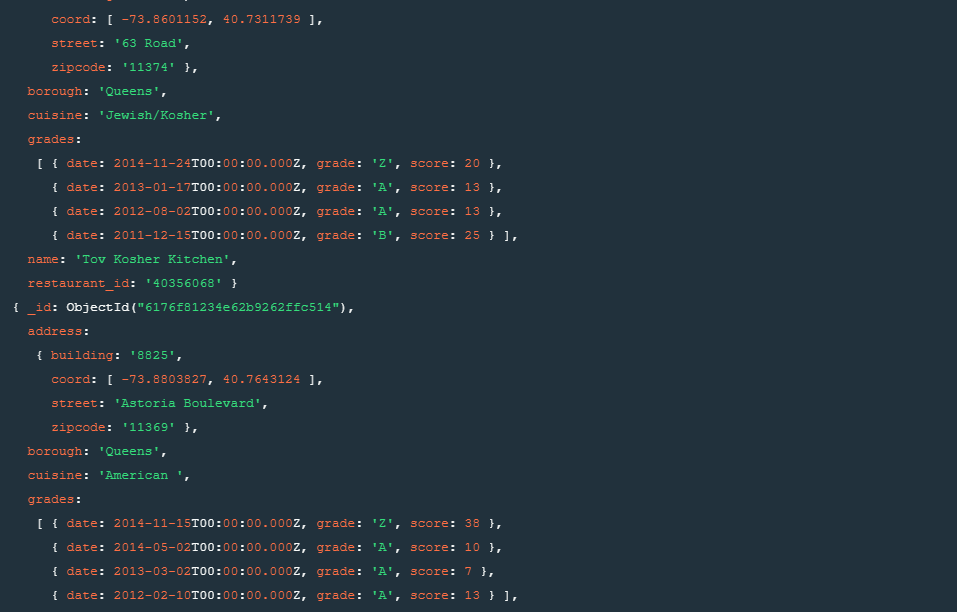
**Mongo DB Exercises - With the Restaurants Data Set**

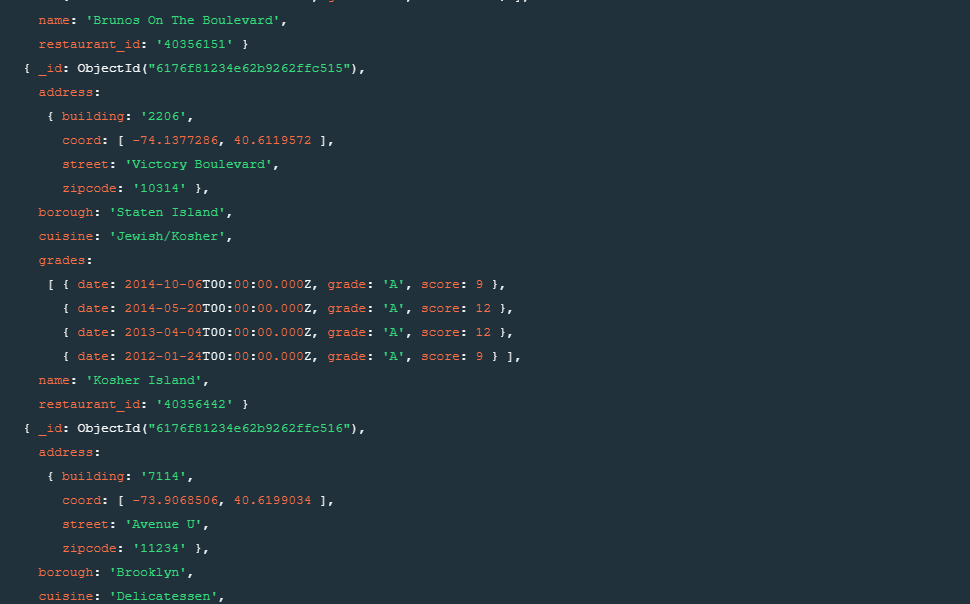
**Exercise Questions:-**

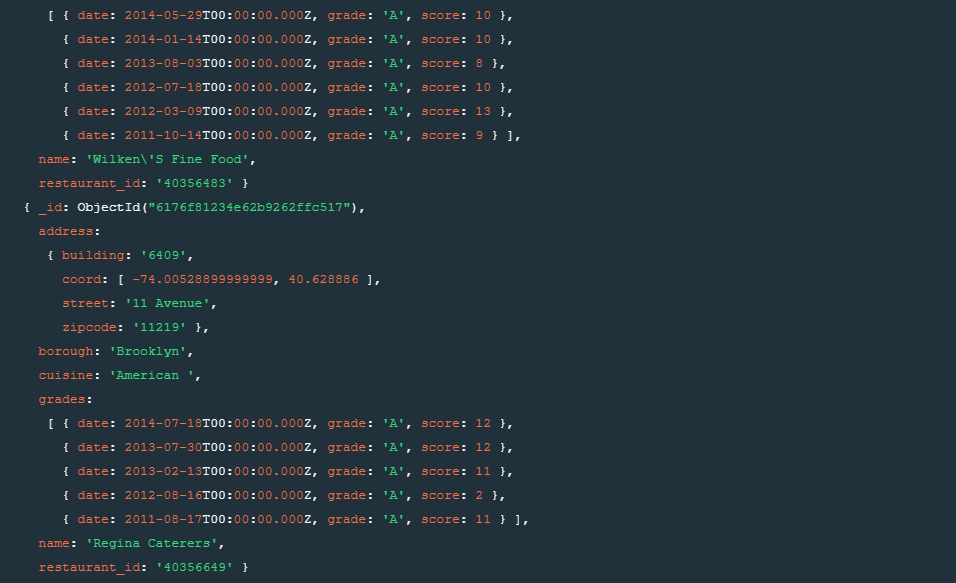
1.Write a MongoDB query to display all the documents in the collection restaurants.

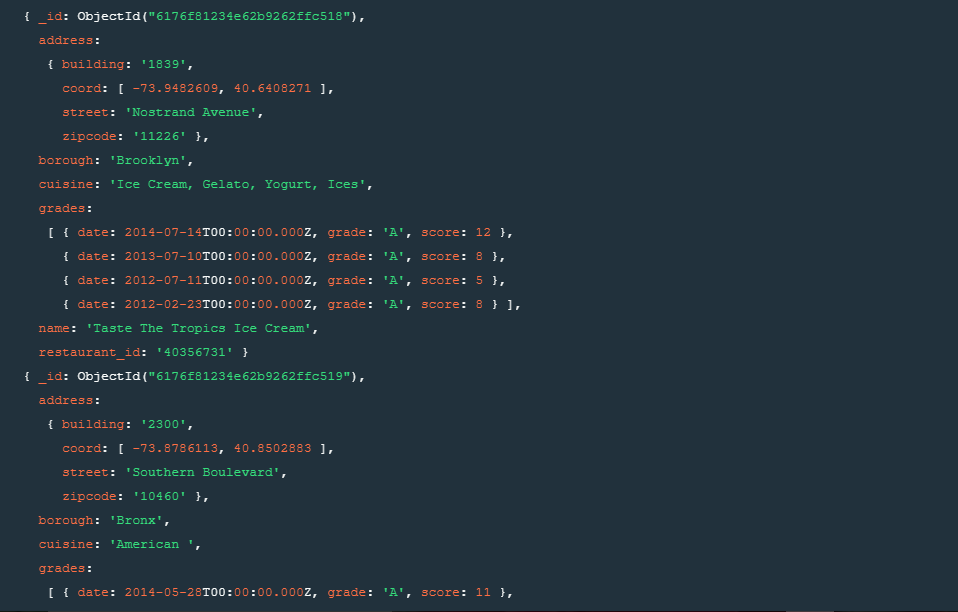


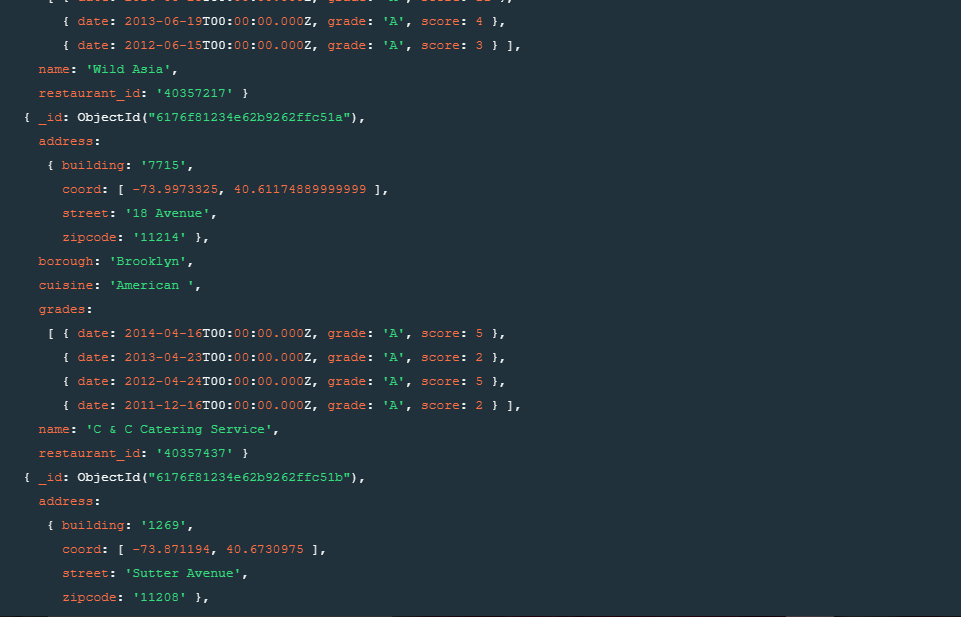


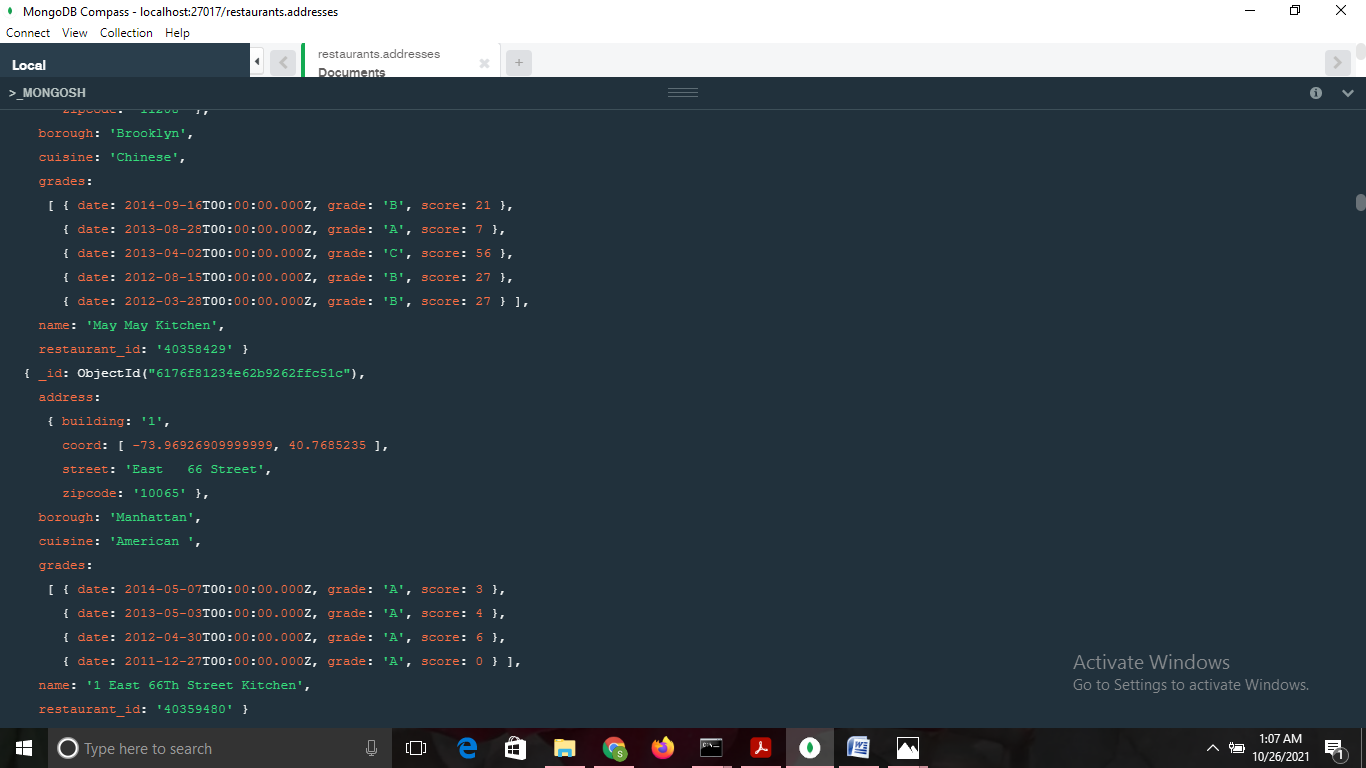


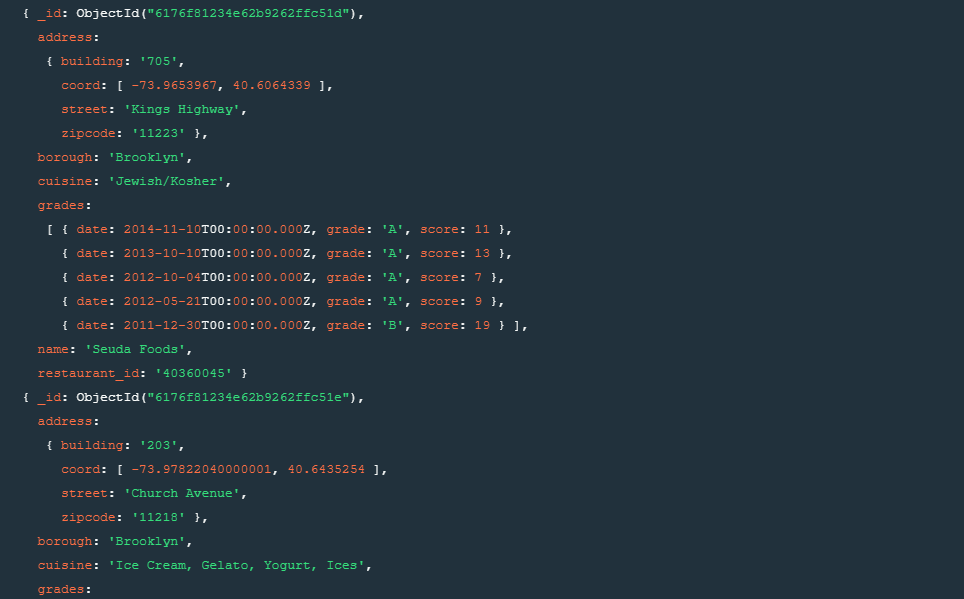


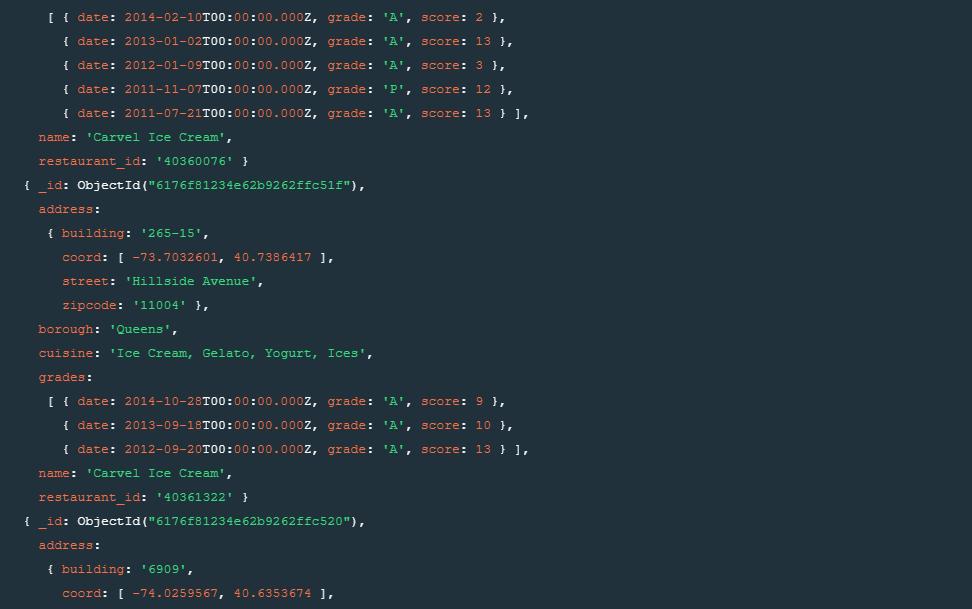


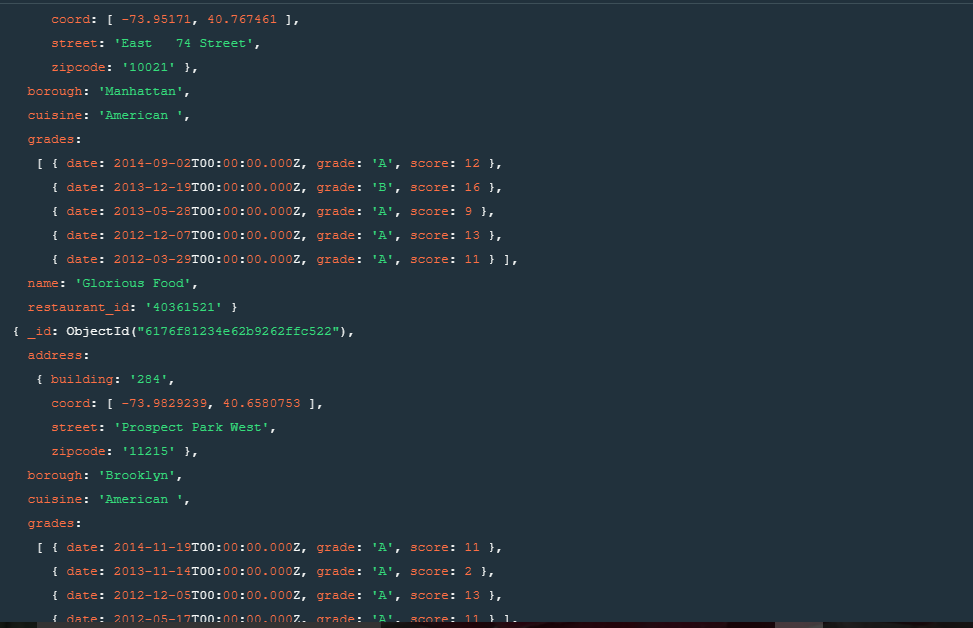






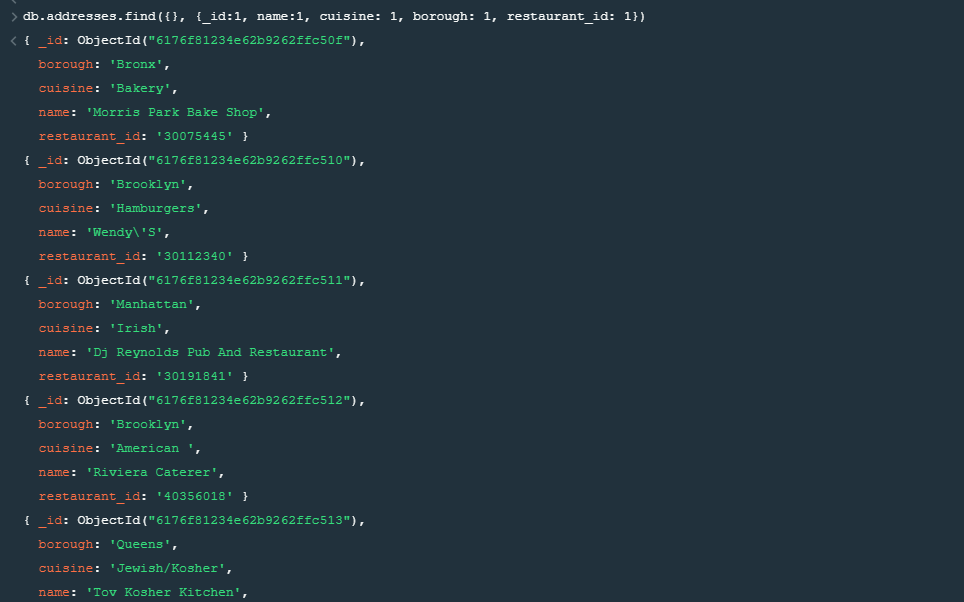


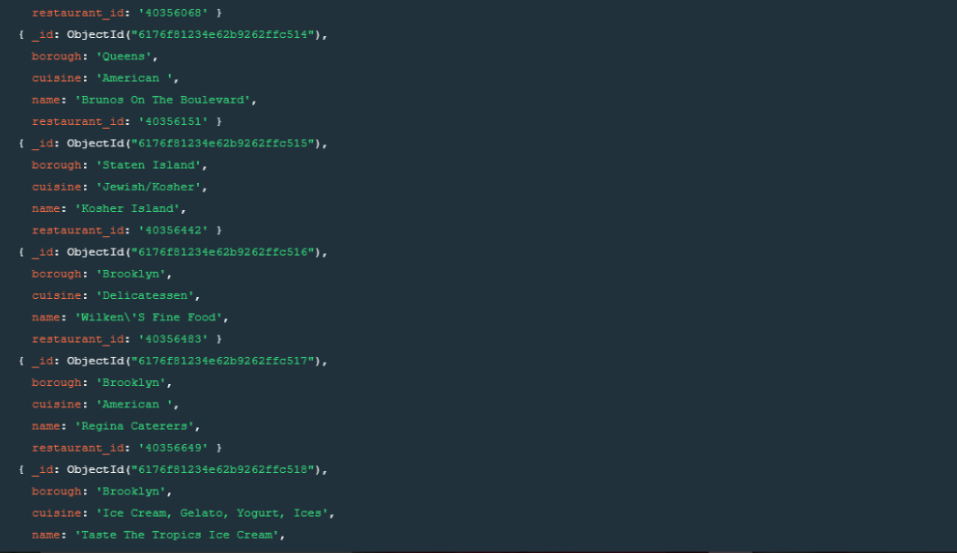


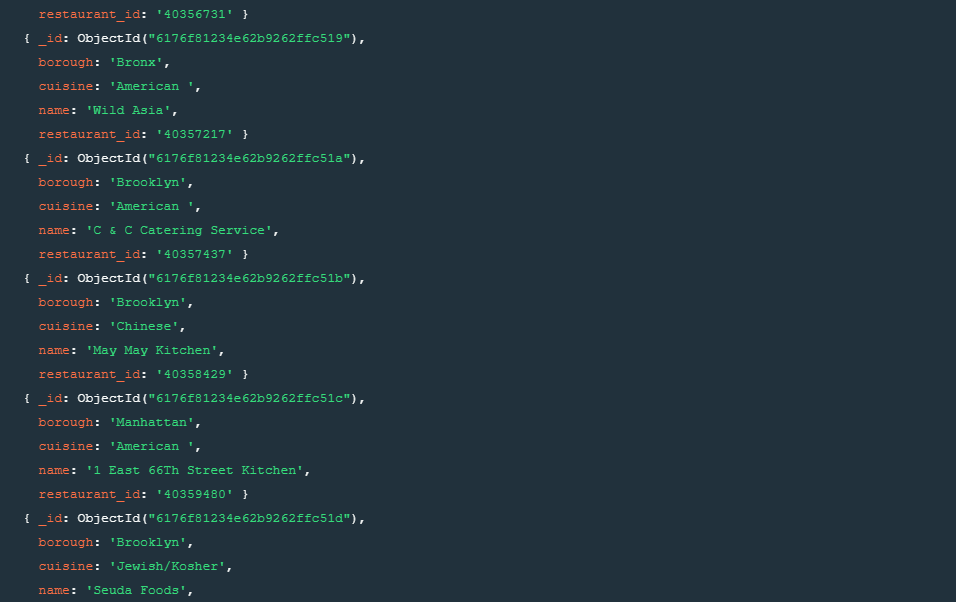


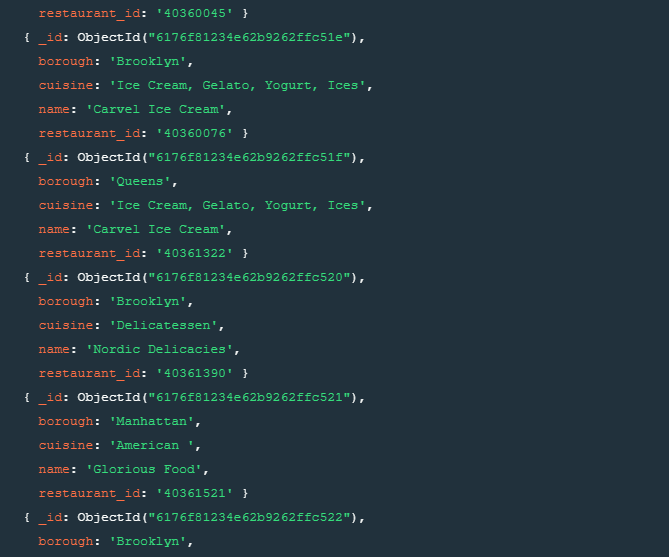


1. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.



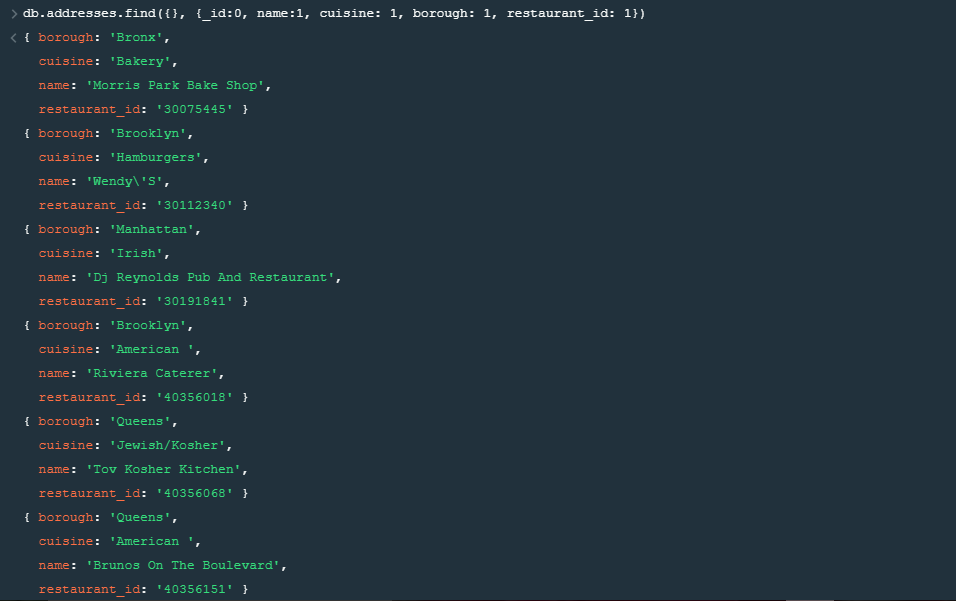


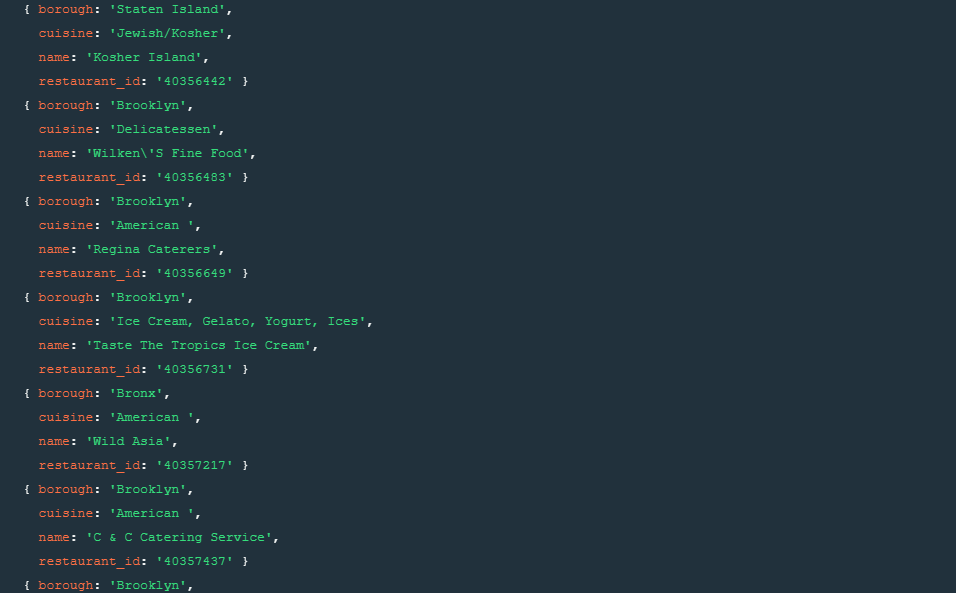


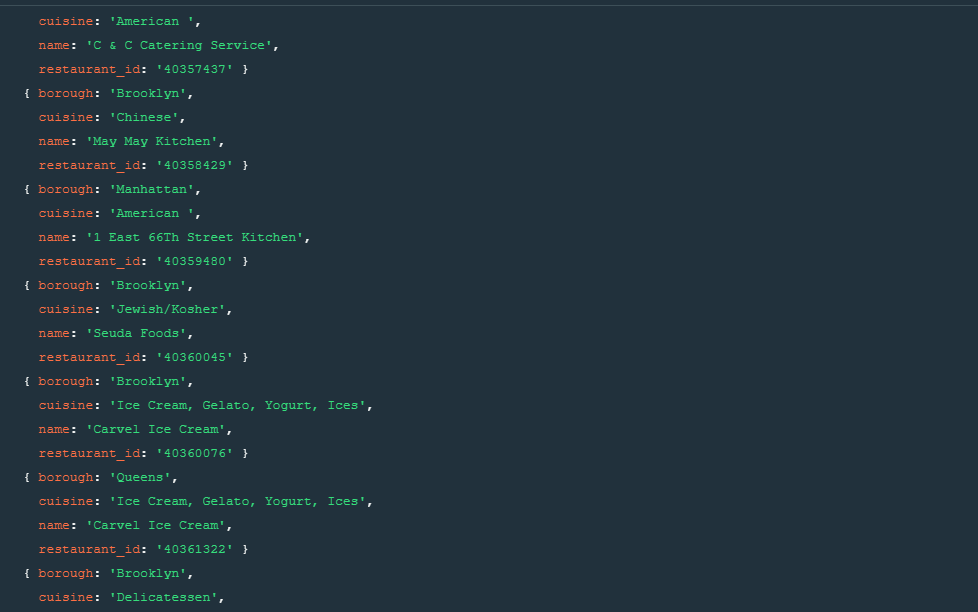




1. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.

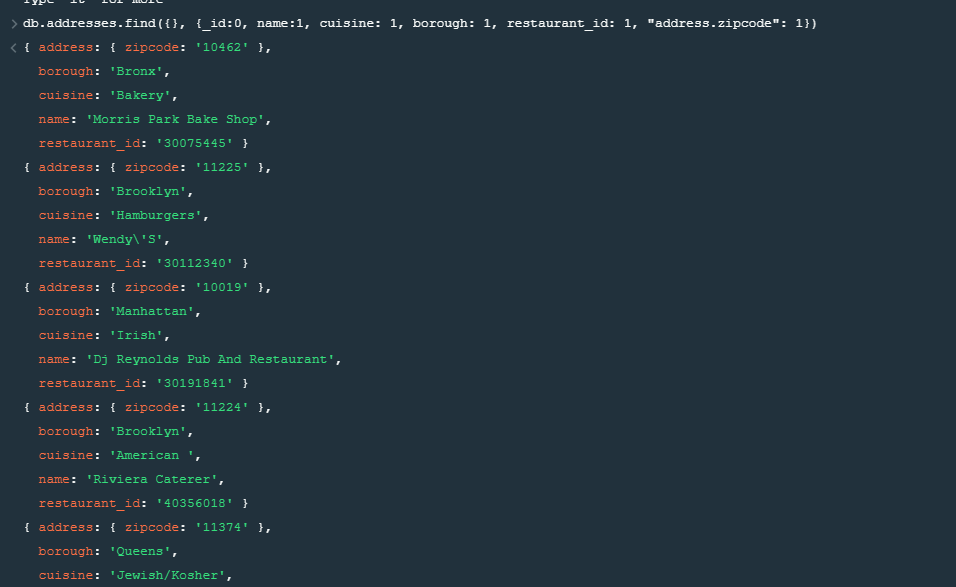


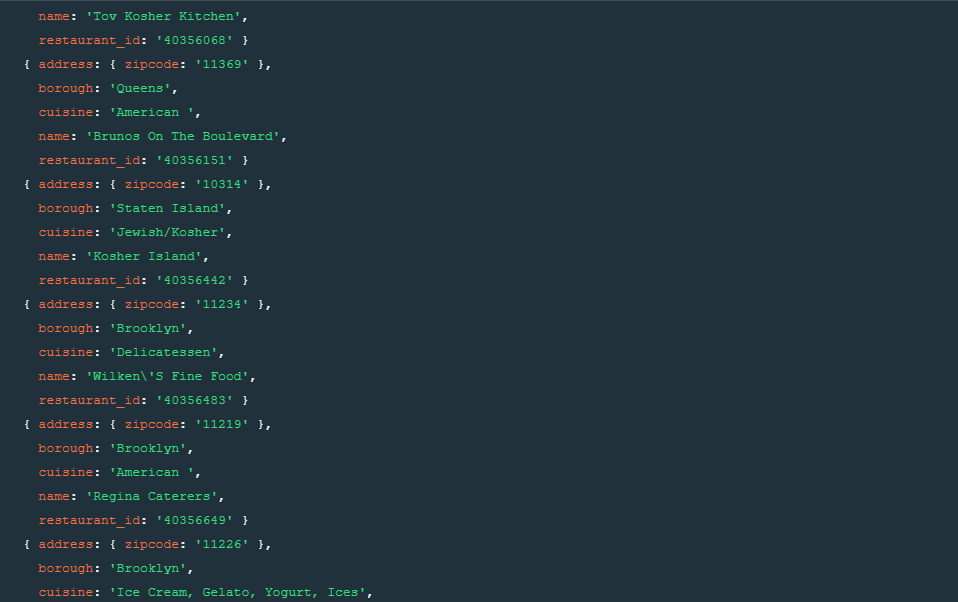


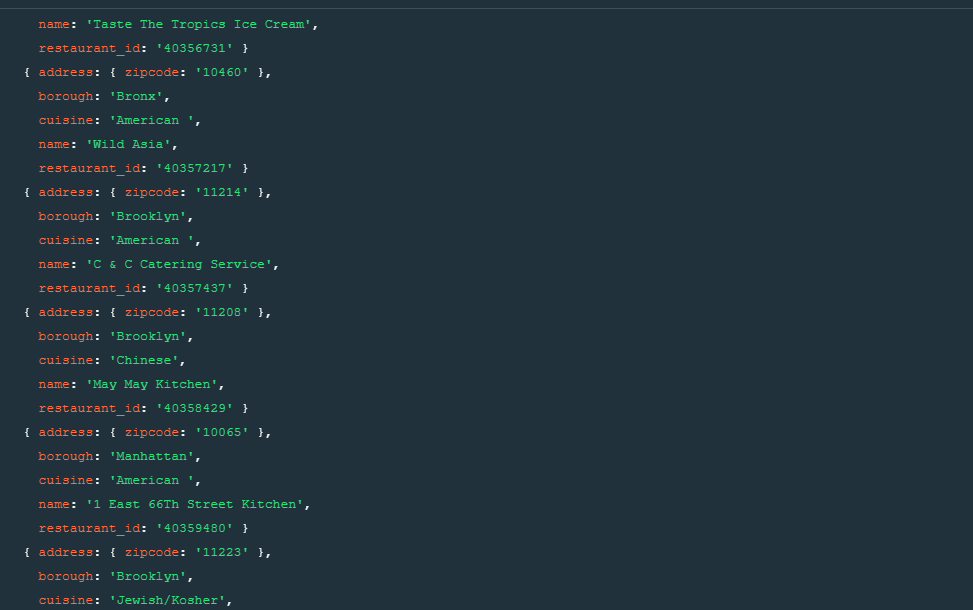


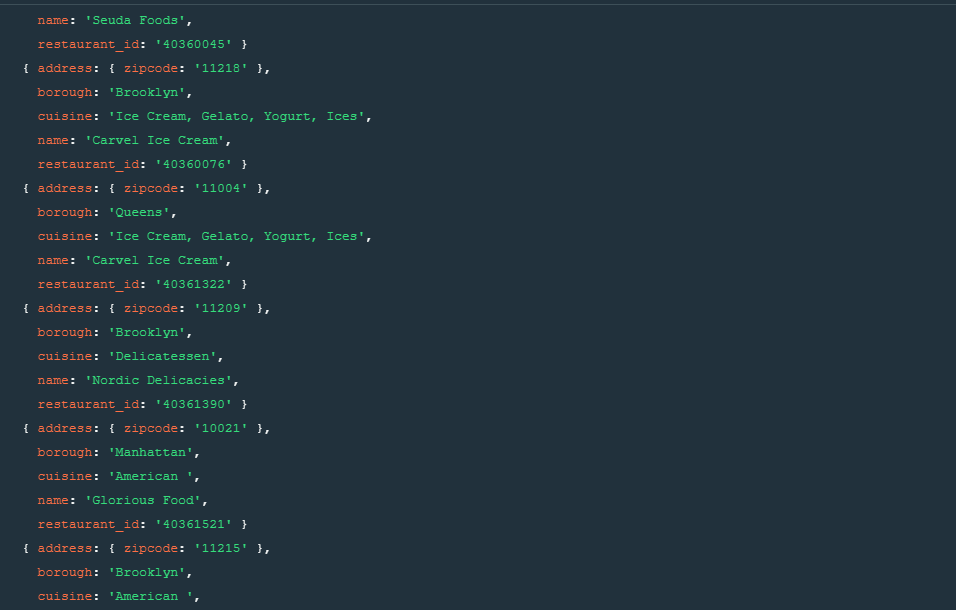


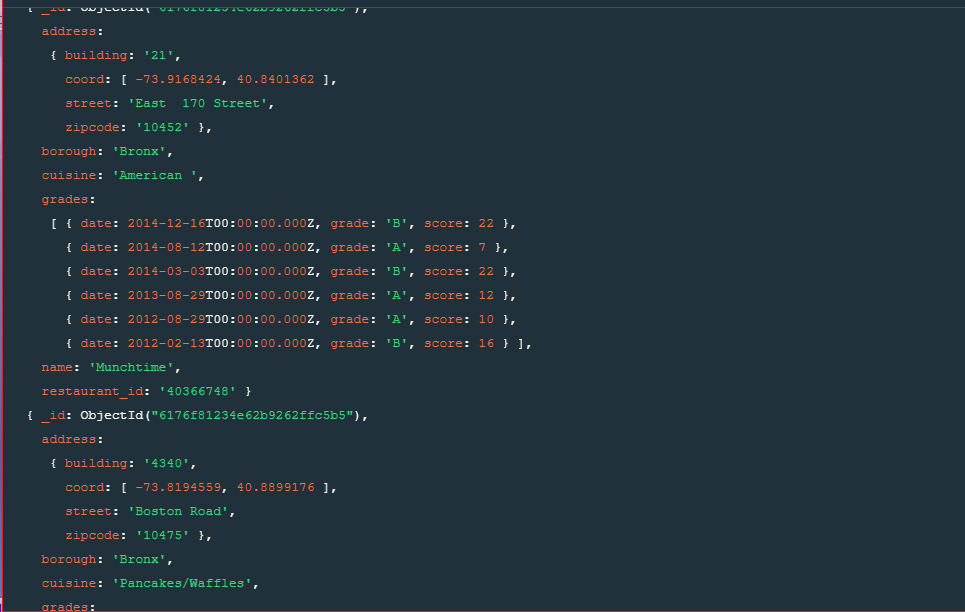
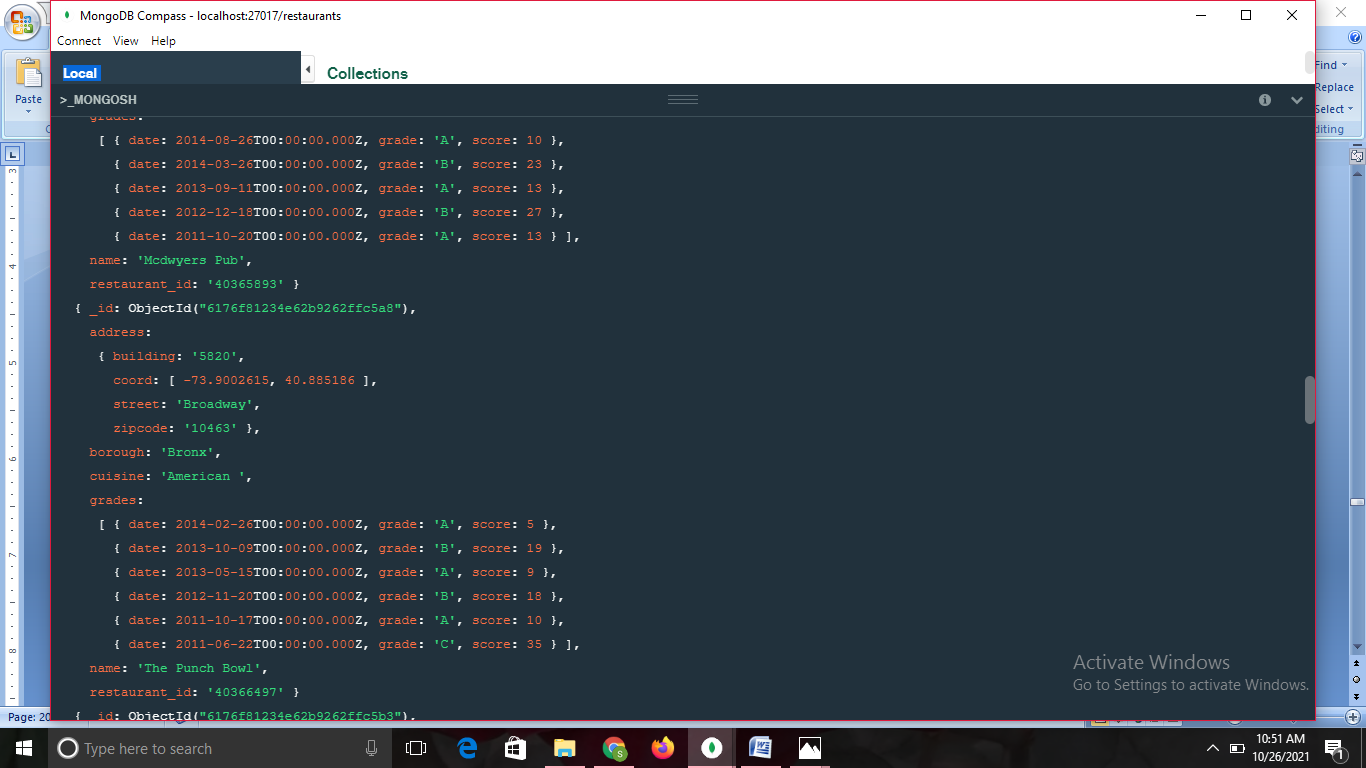
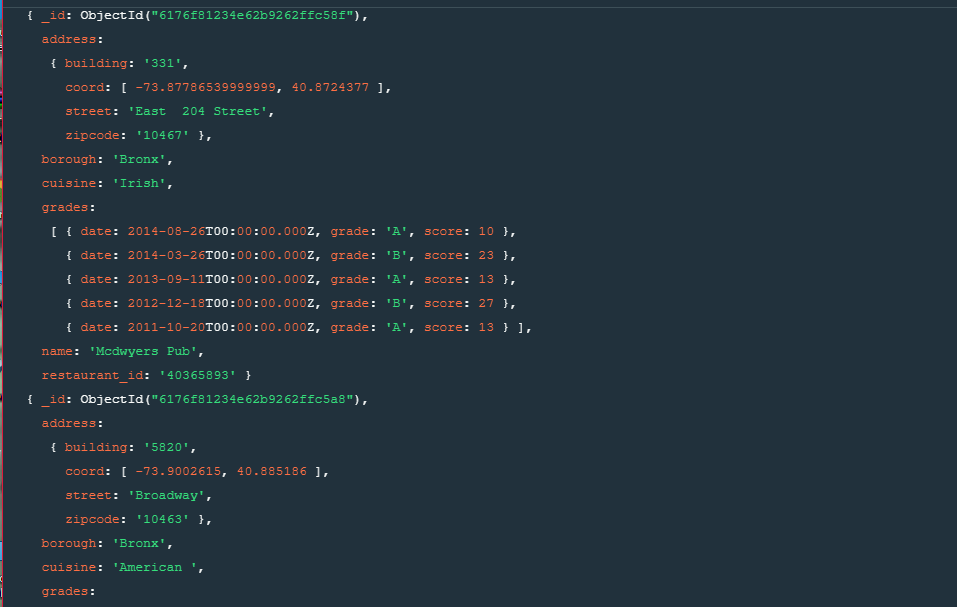
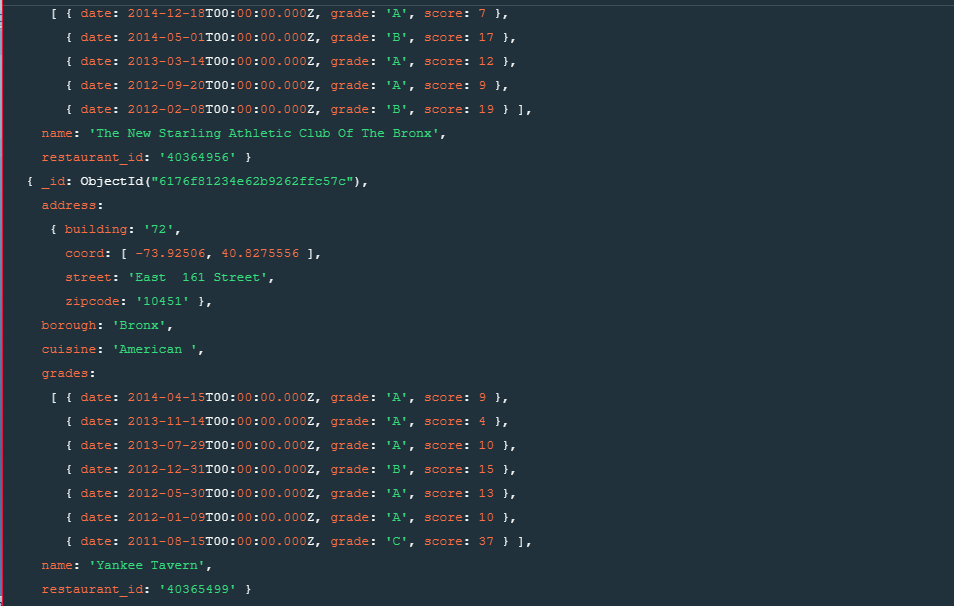
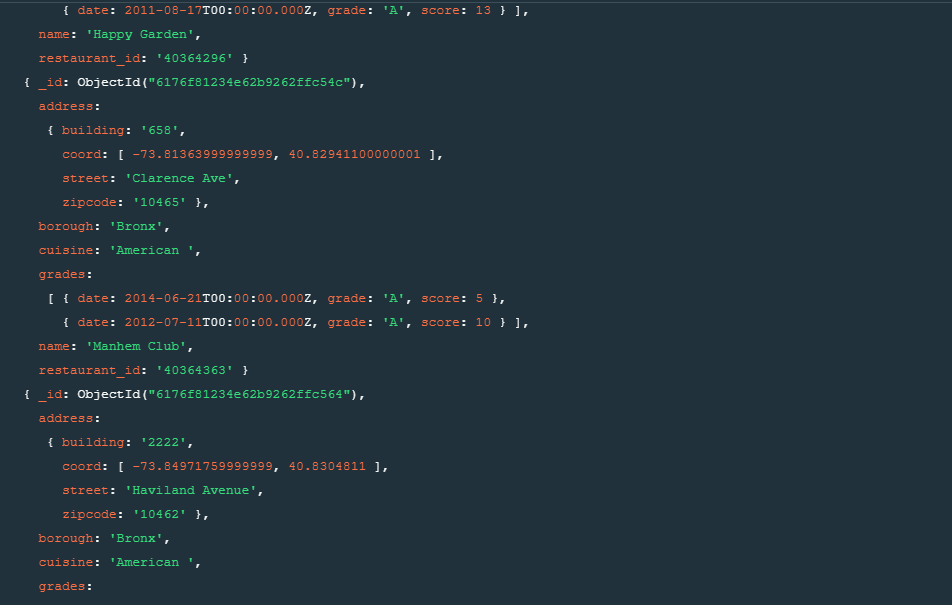
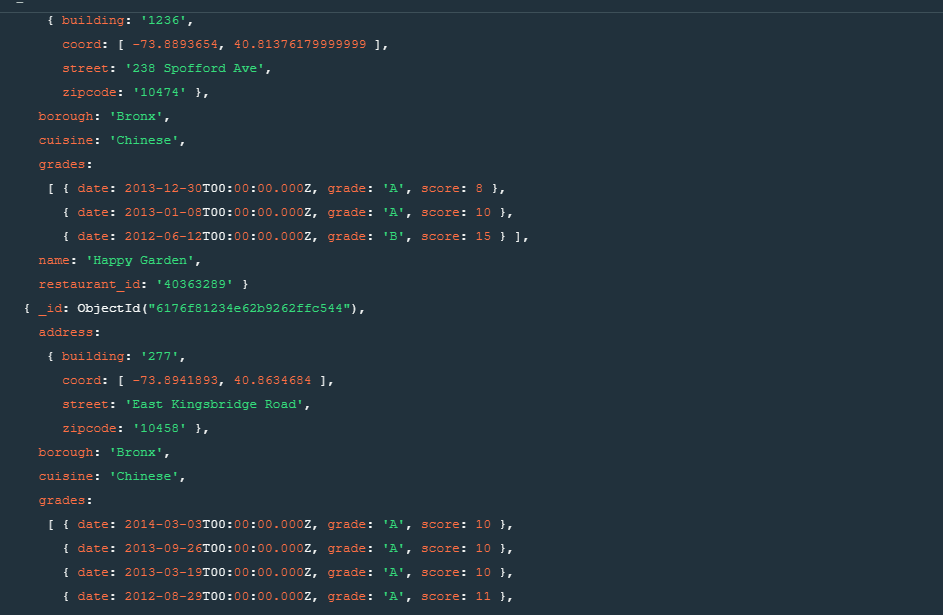
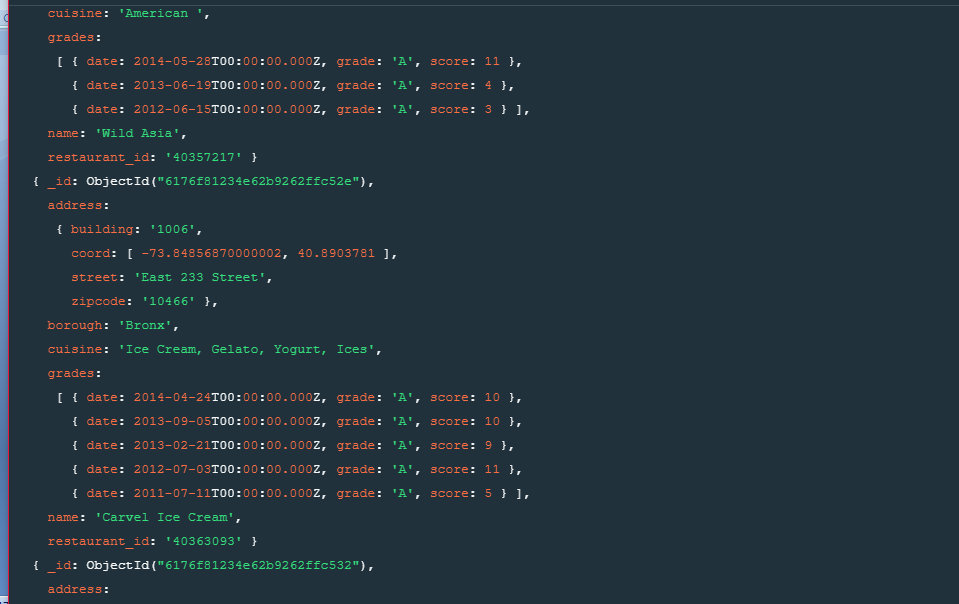
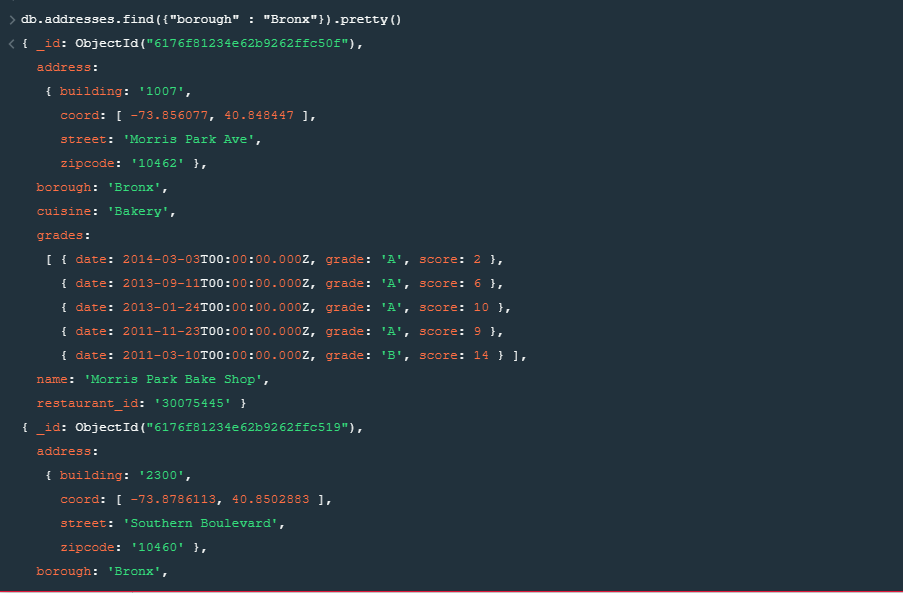
4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.



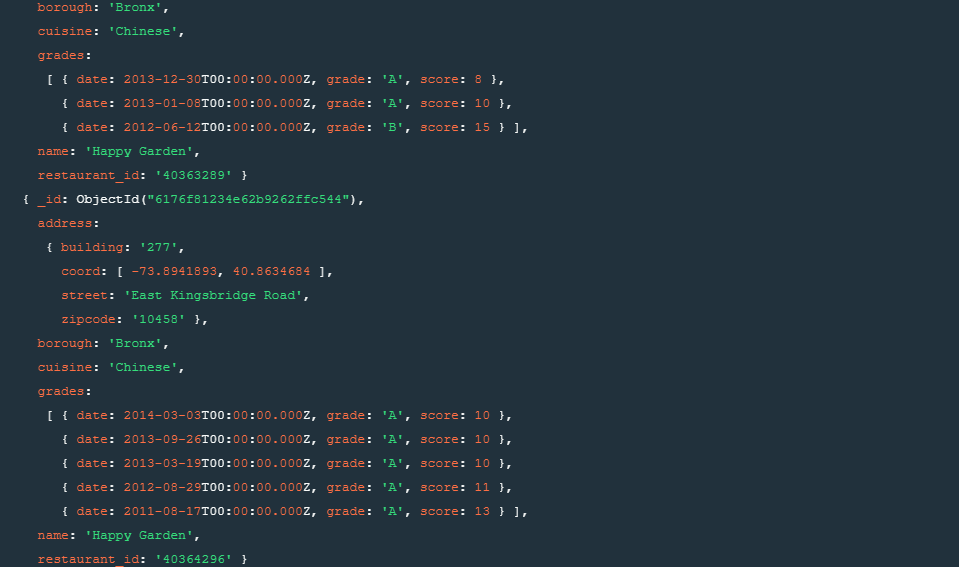
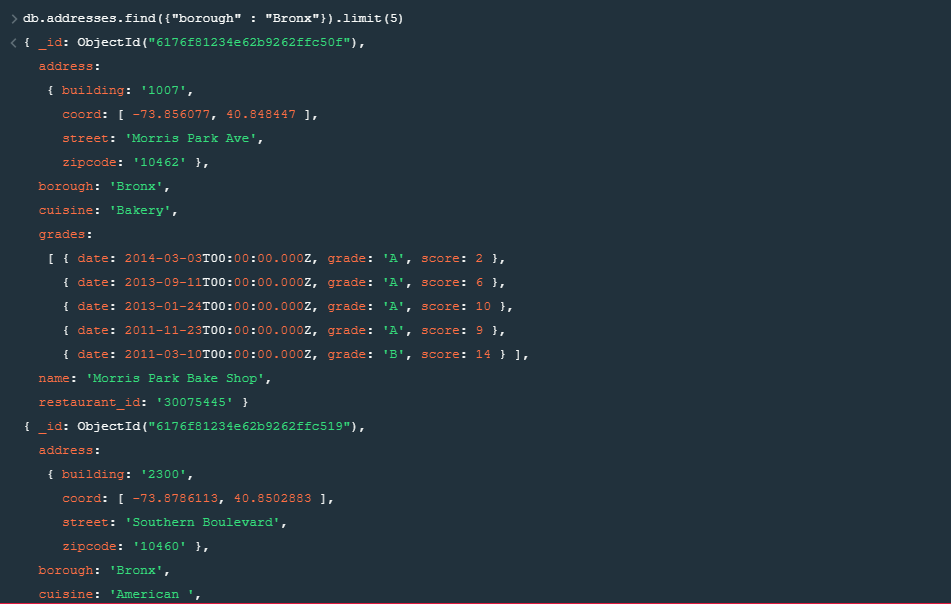




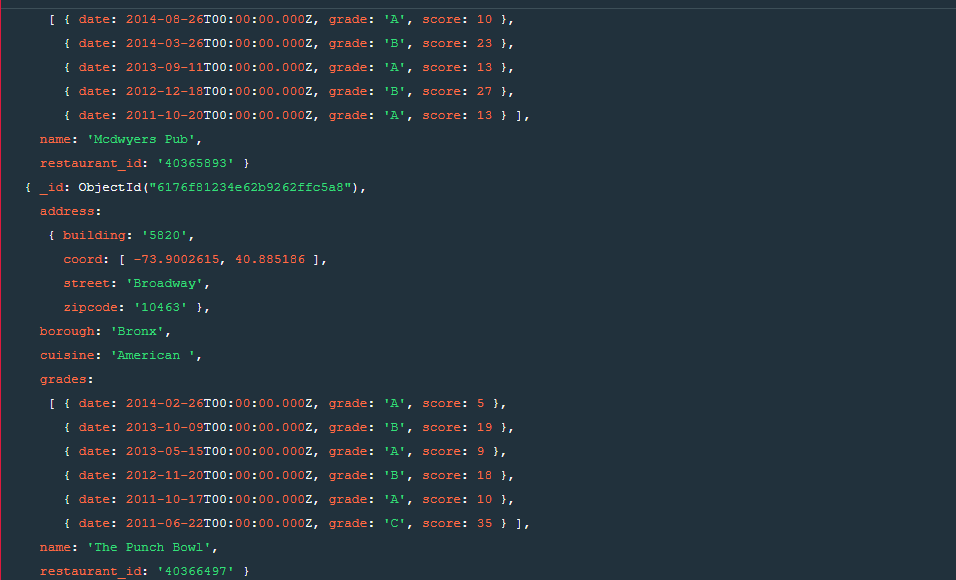
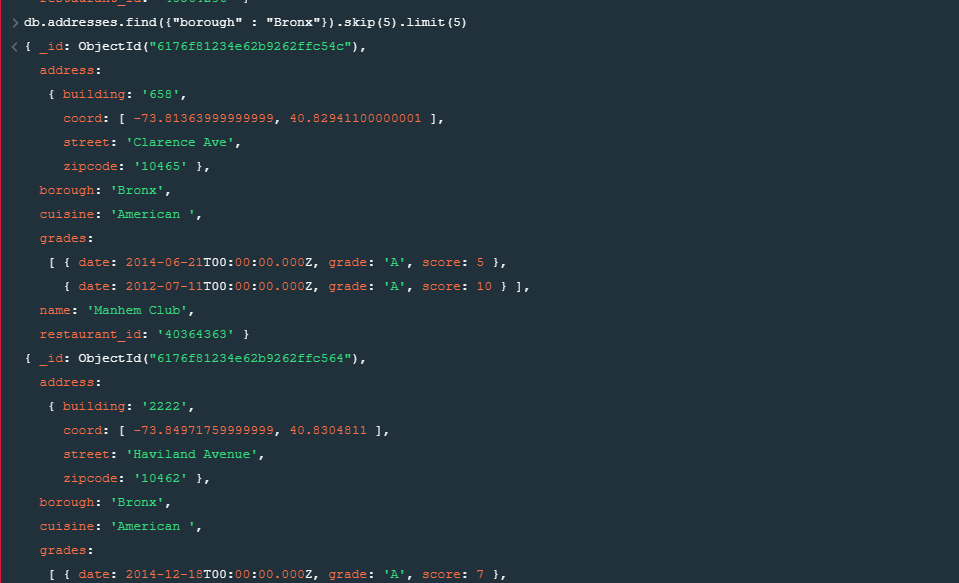


5.display all the restaurant which is in the boroughBronx. 

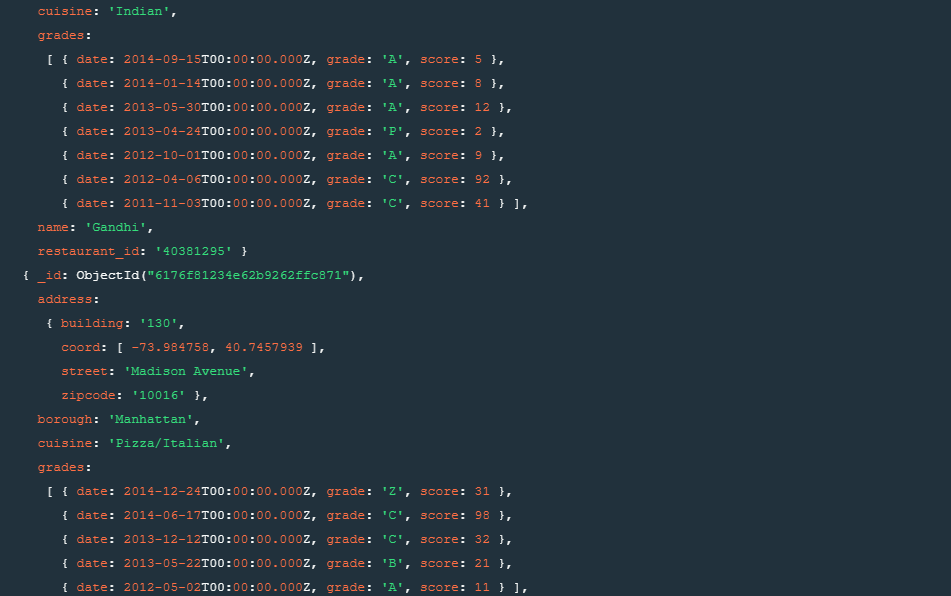
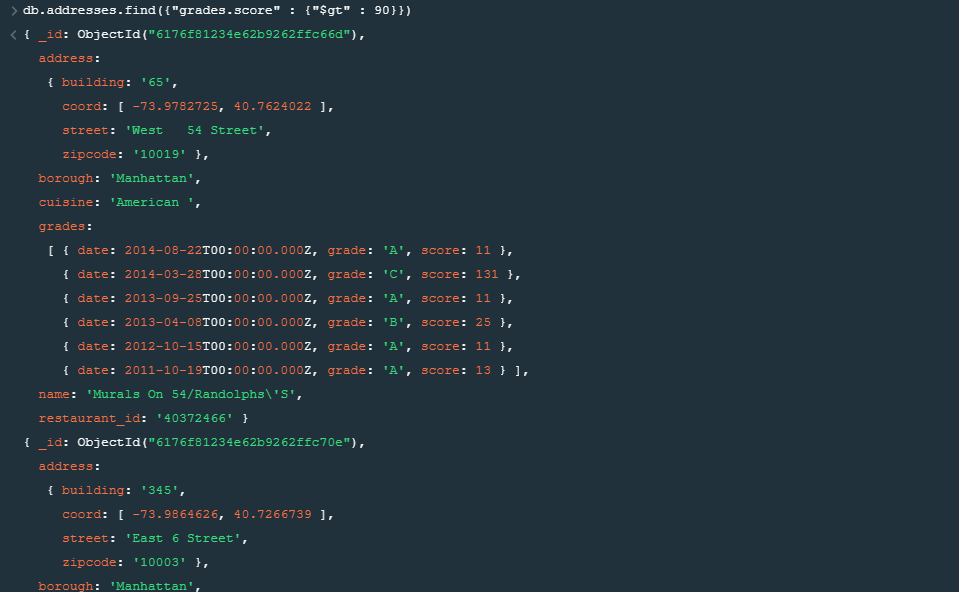
6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.



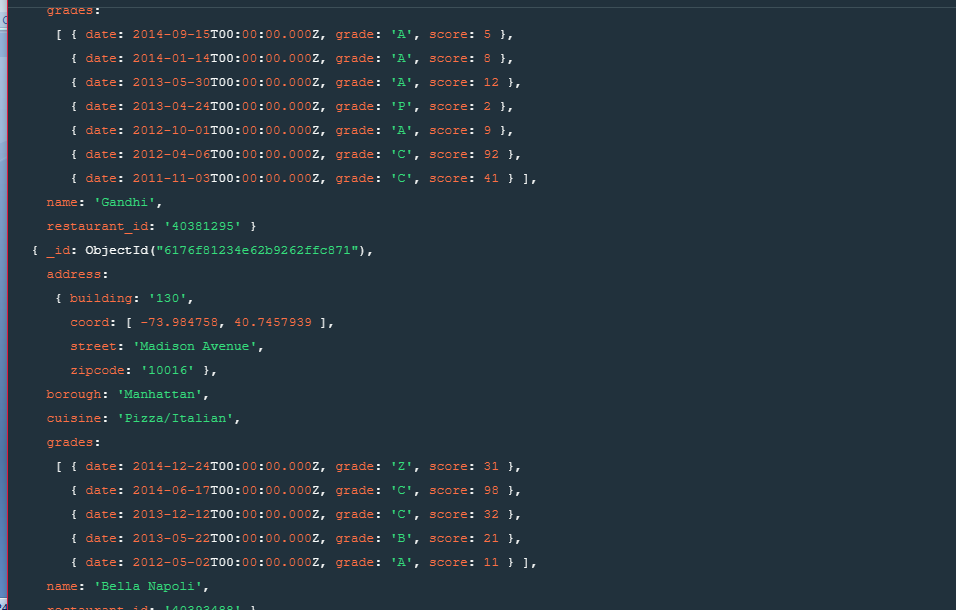
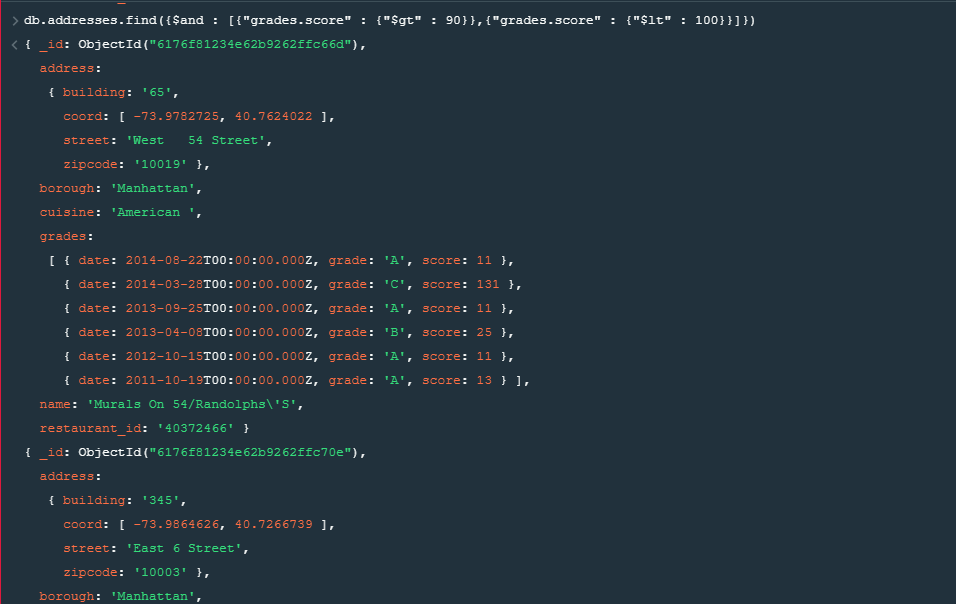
7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.



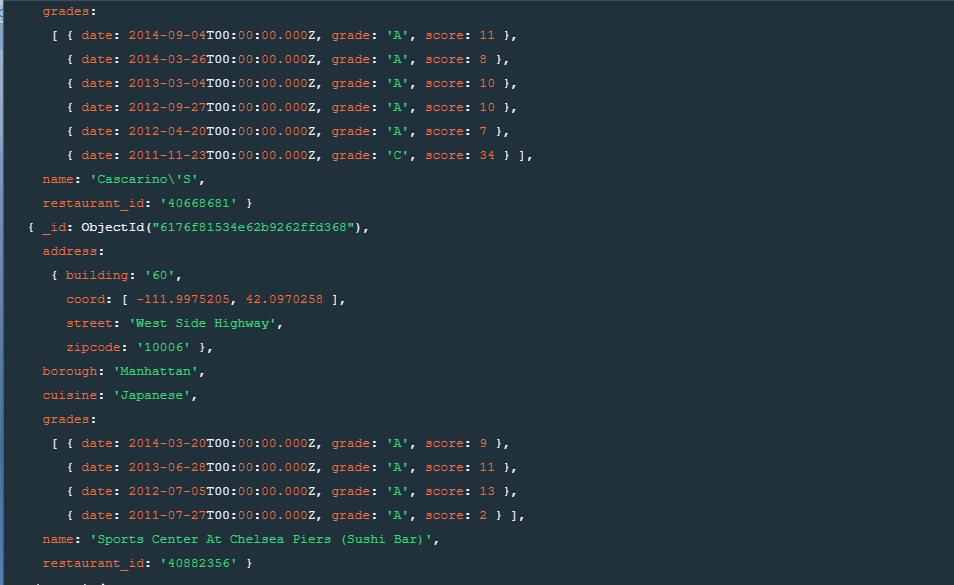
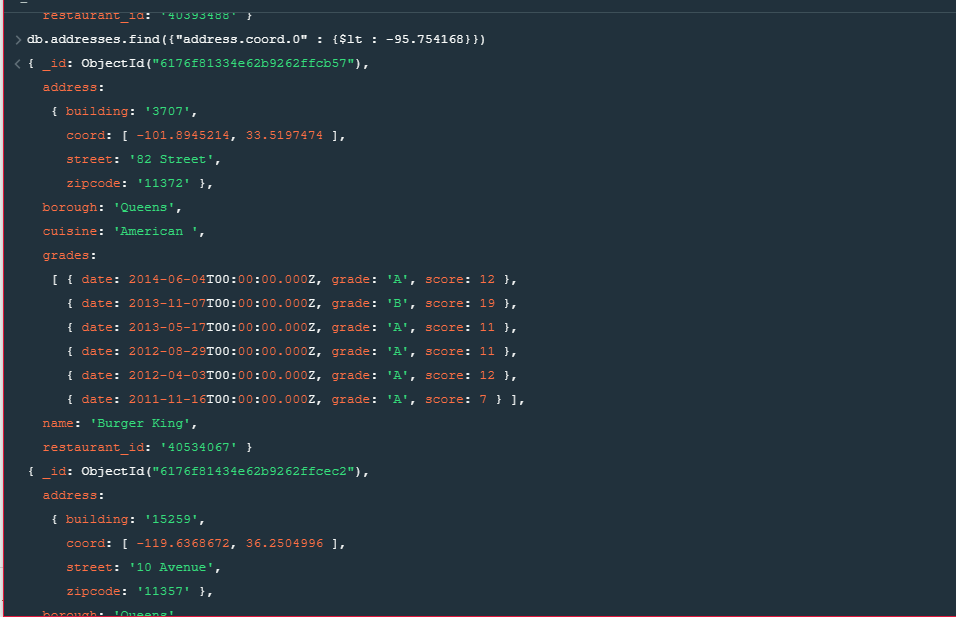
8. Write a MongoDB query to find the restaurants who achieved a score more than 90.



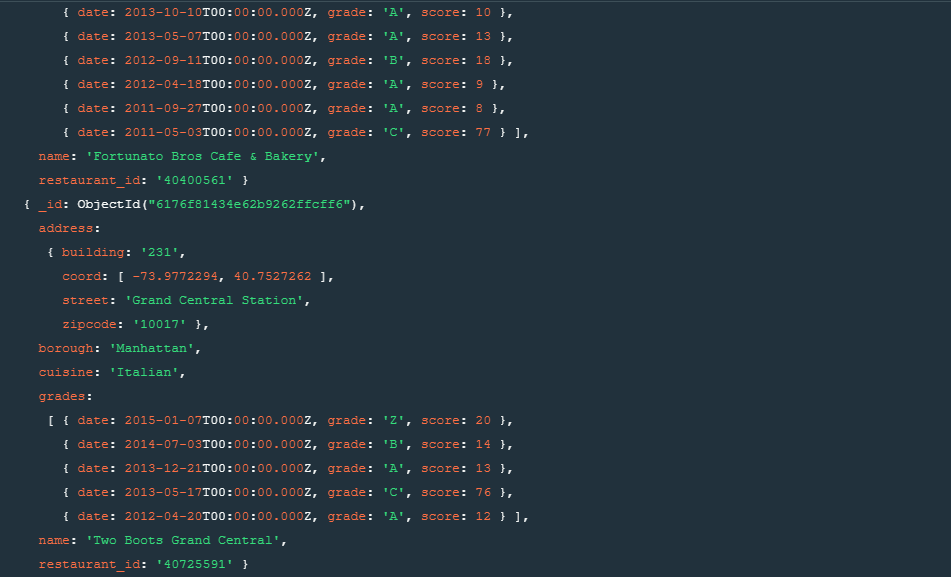
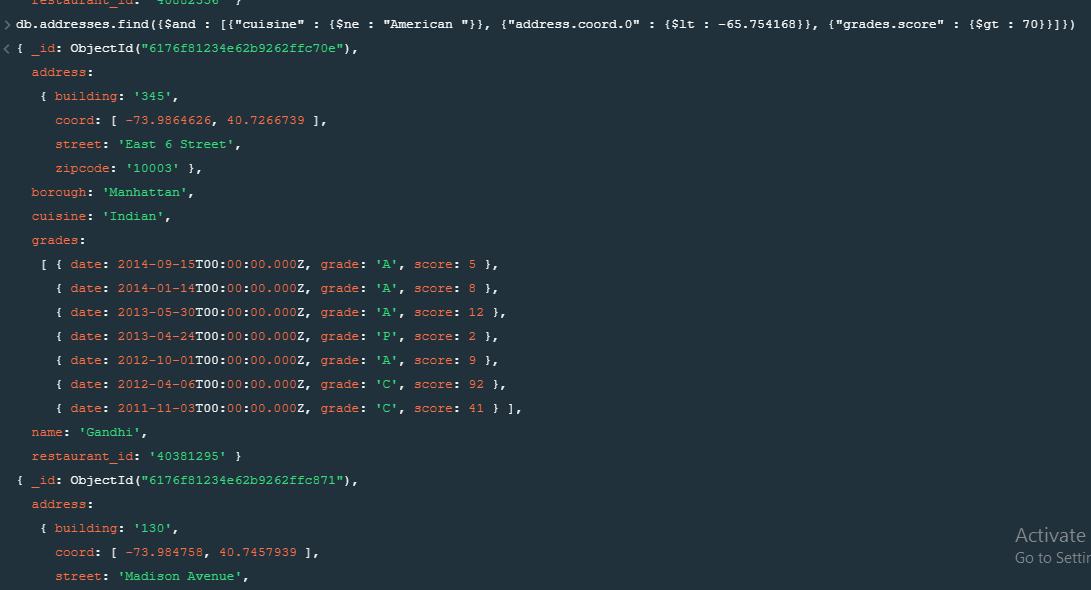
9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.



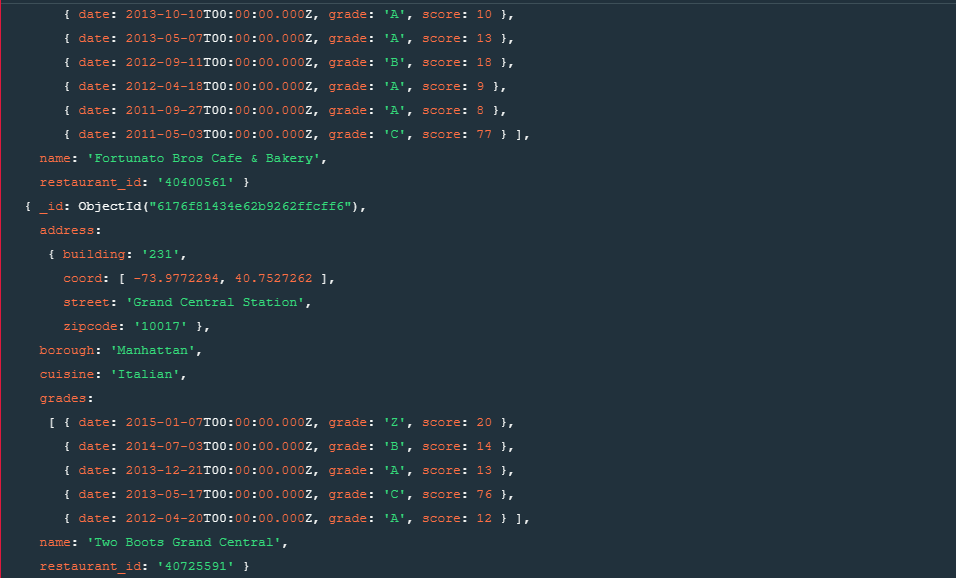
10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.



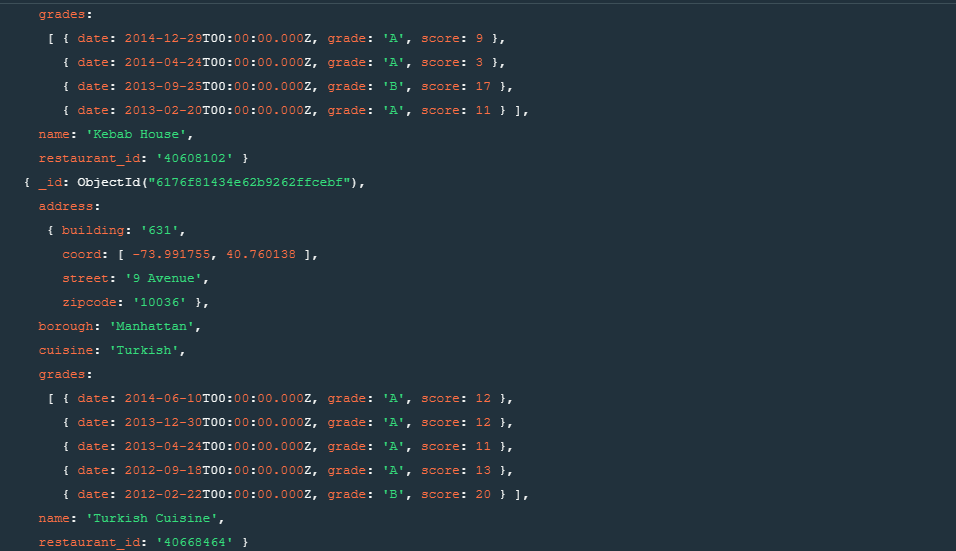
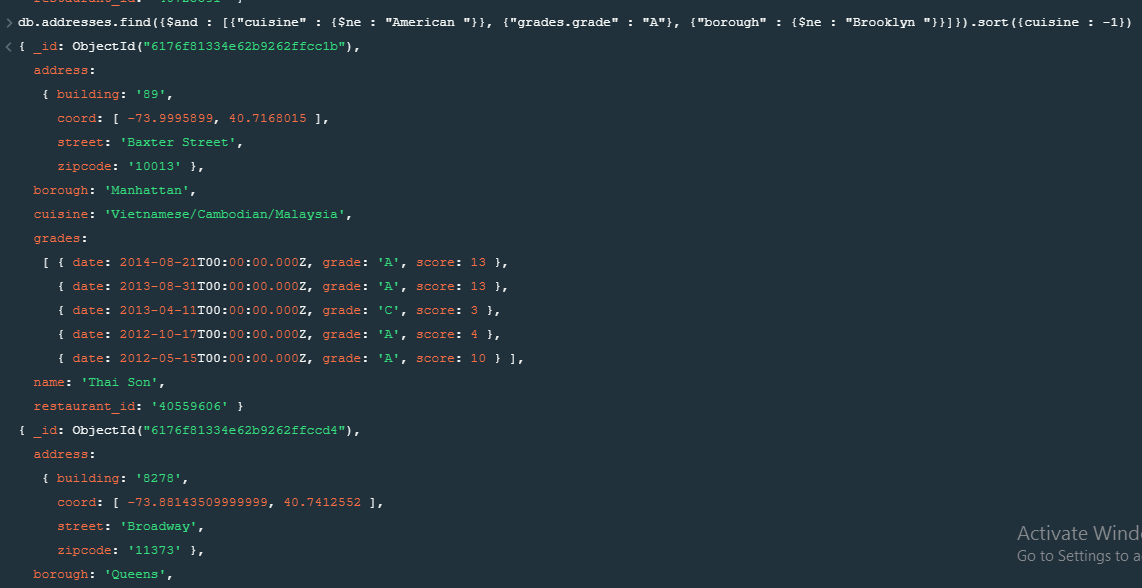
11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.



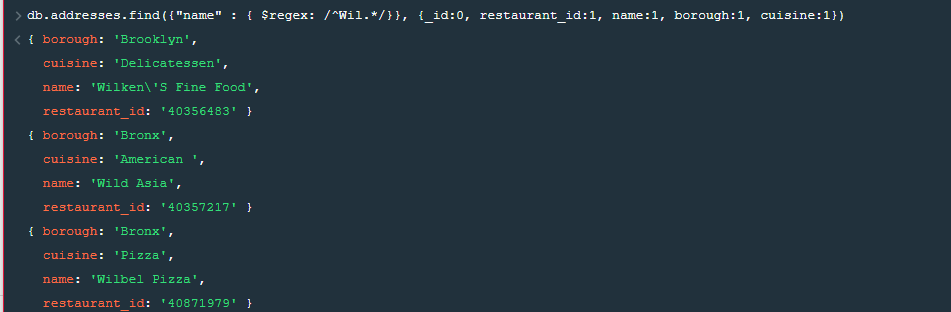
12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.



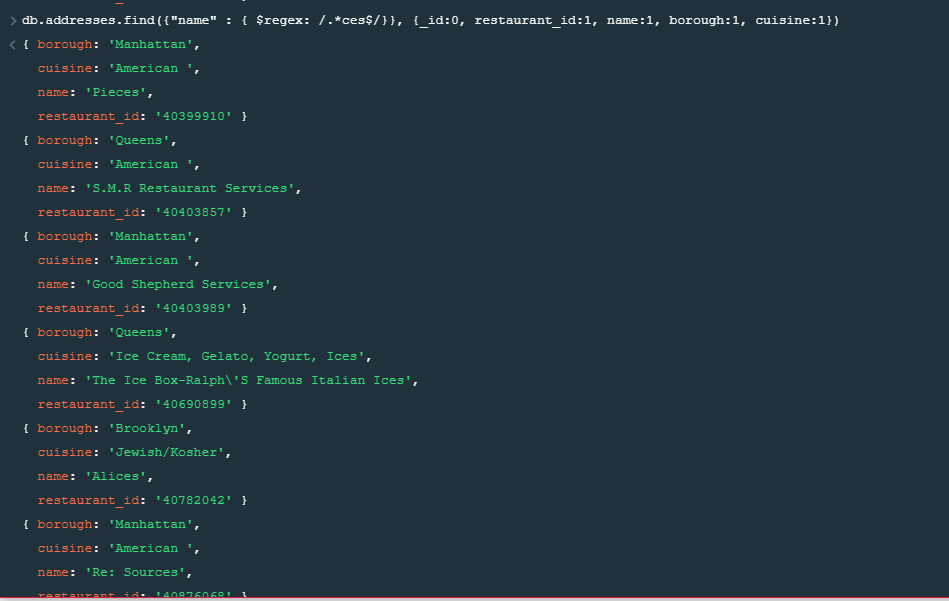
13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.



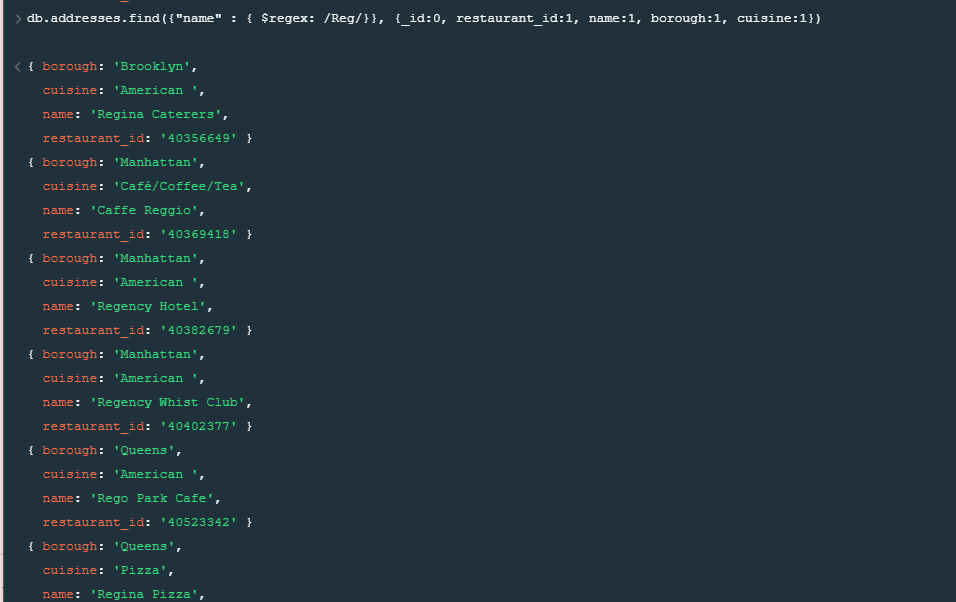
14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.



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.

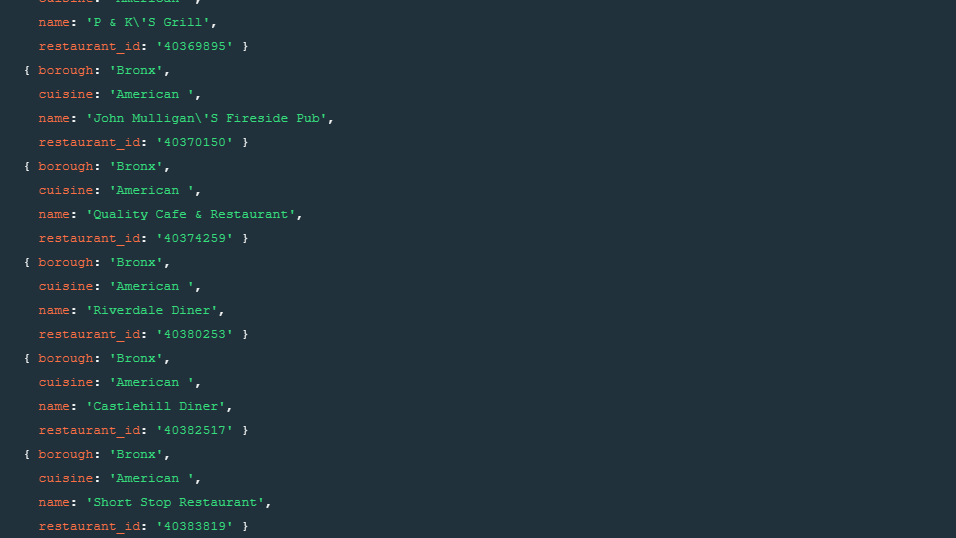
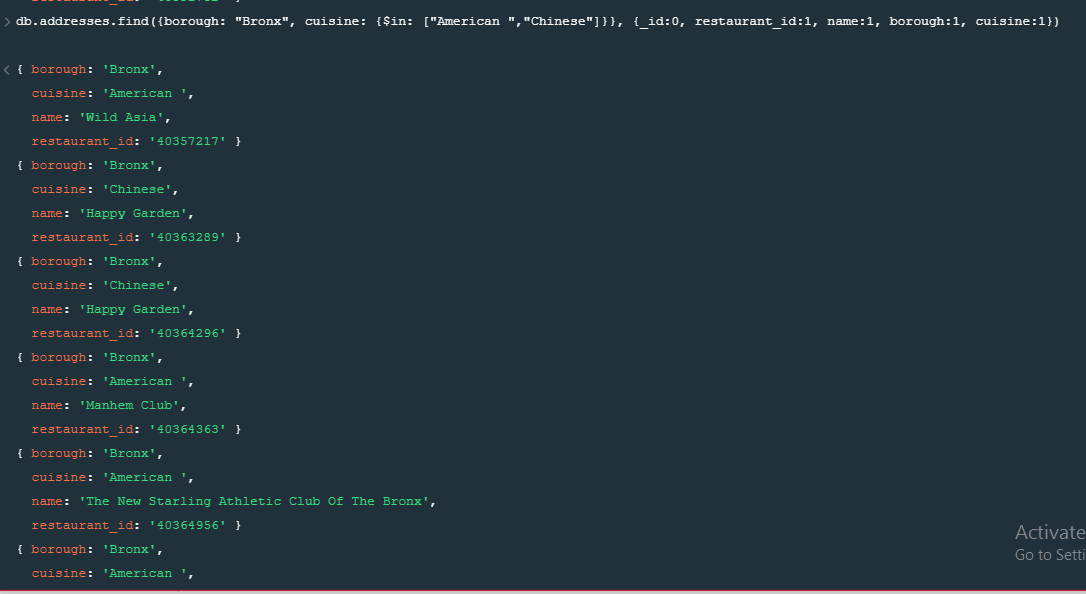


16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.



17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and

prepared either American or Chinese dish.



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.

db.addresses.find(

{"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},

{

"restaurant\_id" : 1,

"name":1,"borough":1,

"cuisine" :1

}

);

{ \_id: ObjectId("6176f81234e62b9262ffc50f"),

borough: 'Bronx',

cuisine: 'Bakery',

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445' }

{ \_id: ObjectId("6176f81234e62b9262ffc510"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: 'Wendy\'S',

restaurant\_id: '30112340' }

{ \_id: ObjectId("6176f81234e62b9262ffc512"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Riviera Caterer',

restaurant\_id: '40356018' }

{ \_id: ObjectId("6176f81234e62b9262ffc513"),

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068' }

{ \_id: ObjectId("6176f81234e62b9262ffc514"),

borough: 'Queens',

cuisine: 'American ',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151' }

{ \_id: ObjectId("6176f81234e62b9262ffc515"),

borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442' }

{ \_id: ObjectId("6176f81234e62b9262ffc516"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: 'Wilken\'S Fine Food',

restaurant\_id: '40356483' }

{ \_id: ObjectId("6176f81234e62b9262ffc517"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649' }

{ \_id: ObjectId("6176f81234e62b9262ffc518"),

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731' }

{ \_id: ObjectId("6176f81234e62b9262ffc519"),

borough: 'Bronx',

cuisine: 'American ',

name: 'Wild Asia',

restaurant\_id: '40357217' }

{ \_id: ObjectId("6176f81234e62b9262ffc51a"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'C & C Catering Service',

restaurant\_id: '40357437' }

{ \_id: ObjectId("6176f81234e62b9262ffc51b"),

borough: 'Brooklyn',

cuisine: 'Chinese',

name: 'May May Kitchen',

restaurant\_id: '40358429' }

{ \_id: ObjectId("6176f81234e62b9262ffc51d"),

borough: 'Brooklyn',

cuisine: 'Jewish/Kosher',

name: 'Seuda Foods',

restaurant\_id: '40360045' }

{ \_id: ObjectId("6176f81234e62b9262ffc51e"),

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40360076' }

{ \_id: ObjectId("6176f81234e62b9262ffc51f"),

borough: 'Queens',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40361322' }

{ \_id: ObjectId("6176f81234e62b9262ffc520"),

borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: 'Nordic Delicacies',

restaurant\_id: '40361390' }

{ \_id: ObjectId("6176f81234e62b9262ffc522"),

borough: 'Brooklyn',

cuisine: 'American ',

name: 'The Movable Feast',

restaurant\_id: '40361606' }

{ \_id: ObjectId("6176f81234e62b9262ffc523"),

borough: 'Queens',

cuisine: 'Delicatessen',

name: 'Sal\'S Deli',

restaurant\_id: '40361618' }

{ \_id: ObjectId("6176f81234e62b9262ffc525"),

borough: 'Queens',

cuisine: 'Delicatessen',

name: 'Steve Chu\'S Deli & Grocery',

restaurant\_id: '40361998' }

{ \_id: ObjectId("6176f81234e62b9262ffc529"),

borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: 'White Castle',

restaurant\_id: '40362344' }

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.

db.addresses.find( {borough: {$nin: ["Staten Island","Queens","Bronx","Brooklyn"]}} , {\_id:0, restaurant\_id:1, name:1, borough:1, cuisine:1})

{ borough: 'Manhattan',

cuisine: 'Irish',

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Glorious Food',

restaurant\_id: '40361521' }

{ borough: 'Manhattan',

cuisine: 'Delicatessen',

name: 'Bully\'S Deli',

restaurant\_id: '40361708' }

{ borough: 'Manhattan',

cuisine: 'Chicken',

name: 'Harriet\'S Kitchen',

restaurant\_id: '40362098' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'P & S Deli Grocery',

restaurant\_id: '40362264' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Angelika Film Center',

restaurant\_id: '40362274' }

{ borough: 'Manhattan',

cuisine: 'Turkish',

name: 'The Country Cafe',

restaurant\_id: '40362715' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Downtown Deli',

restaurant\_id: '40363021' }

{ borough: 'Manhattan',

cuisine: 'Bakery',

name: 'Olive\'S',

restaurant\_id: '40363151' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Cafe Metro',

restaurant\_id: '40363298' }

{ borough: 'Manhattan',

cuisine: 'Sandwiches/Salads/Mixed Buffet',

name: 'Lexler Deli',

restaurant\_id: '40363426' }

{ borough: 'Manhattan',

cuisine: 'Continental',

name: 'Lorenzo & Maria\'S',

restaurant\_id: '40363630' }

{ borough: 'Manhattan',

cuisine: 'Pizza',

name: 'Domino\'S Pizza',

restaurant\_id: '40363644' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Berkely',

restaurant\_id: '40363685' }

{ borough: 'Manhattan',

cuisine: 'Pizza',

name: 'Domino\'S Pizza',

restaurant\_id: '40363945' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Spoon Bread Catering',

restaurant\_id: '40364179' }

{ borough: 'Manhattan',

cuisine: 'Chicken',

name: 'Texas Rotisserie',

restaurant\_id: '40364304' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Metropolitan Club',

restaurant\_id: '40364347' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Palm Restaurant',

restaurant\_id: '40364355' }

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.

db.addresses.find({"grades.score": {$lte: 10}}, {\_id:0, restaurant\_id:1, name:1, borough:1, cuisine:1})

{ 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: 'American ',

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151' }

{ borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442' }

{ borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: 'Wilken\'S Fine Food',

restaurant\_id: '40356483' }

{ borough: 'Brooklyn',

cuisine: 'American ',

name: 'Regina Caterers',

restaurant\_id: '40356649' }

{ borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731' }

{ borough: 'Bronx',

cuisine: 'American ',

name: 'Wild Asia',

restaurant\_id: '40357217' }

{ borough: 'Brooklyn',

cuisine: 'American ',

name: 'C & C Catering Service',

restaurant\_id: '40357437' }

{ borough: 'Brooklyn',

cuisine: 'Chinese',

name: 'May May Kitchen',

restaurant\_id: '40358429' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480' }

{ borough: 'Brooklyn',

cuisine: 'Jewish/Kosher',

name: 'Seuda Foods',

restaurant\_id: '40360045' }

{ borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40360076' }

{ borough: 'Queens',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40361322' }

{ borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: 'Nordic Delicacies',

restaurant\_id: '40361390' }

{ borough: 'Manhattan',

cuisine: 'American ',

name: 'Glorious Food',

restaurant\_id: '40361521' }

{ borough: 'Brooklyn',

cuisine: 'American ',

name: 'The Movable Feast',

restaurant\_id: '40361606' }

{ borough: 'Queens',

cuisine: 'Delicatessen',

name: 'Sal\'S Deli',

restaurant\_id: '40361618' }

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'.

db.addresses.find({$nor: [{cuisine: {$in: ["American ","Chinese"]}},{name: /^Wil.\*/}]},{\_id:0, restaurant\_id:1, name:1, borough:1, cuisine:1})

{ 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: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068' }

{ borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

name: 'Kosher Island',

restaurant\_id: '40356442' }

{ borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731' }

{ borough: 'Brooklyn',

cuisine: 'Jewish/Kosher',

name: 'Seuda Foods',

restaurant\_id: '40360045' }

{ borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40360076' }

{ borough: 'Queens',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40361322' }

{ borough: 'Brooklyn',

cuisine: 'Delicatessen',

name: 'Nordic Delicacies',

restaurant\_id: '40361390' }

{ borough: 'Queens',

cuisine: 'Delicatessen',

name: 'Sal\'S Deli',

restaurant\_id: '40361618' }

{ borough: 'Manhattan',

cuisine: 'Delicatessen',

name: 'Bully\'S Deli',

restaurant\_id: '40361708' }

{ borough: 'Queens',

cuisine: 'Delicatessen',

name: 'Steve Chu\'S Deli & Grocery',

restaurant\_id: '40361998' }

{ borough: 'Manhattan',

cuisine: 'Chicken',

name: 'Harriet\'S Kitchen',

restaurant\_id: '40362098' }

{ borough: 'Brooklyn',

cuisine: 'Hamburgers',

name: 'White Castle',

restaurant\_id: '40362344' }

{ borough: 'Manhattan',

cuisine: 'Turkish',

name: 'The Country Cafe',

restaurant\_id: '40362715' }

{ borough: 'Brooklyn',

cuisine: 'Caribbean',

name: 'Shashemene Int\'L Restaura',

restaurant\_id: '40362869' }

{ borough: 'Bronx',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

name: 'Carvel Ice Cream',

restaurant\_id: '40363093' }

{ borough: 'Brooklyn',

cuisine: 'Donuts',

name: 'Dunkin\' Donuts',

restaurant\_id: '40363098' }

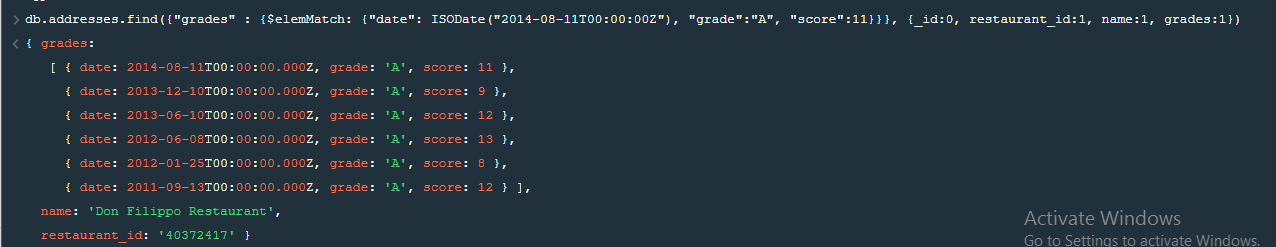
{ borough: 'Manhattan',

cuisine: 'Bakery',

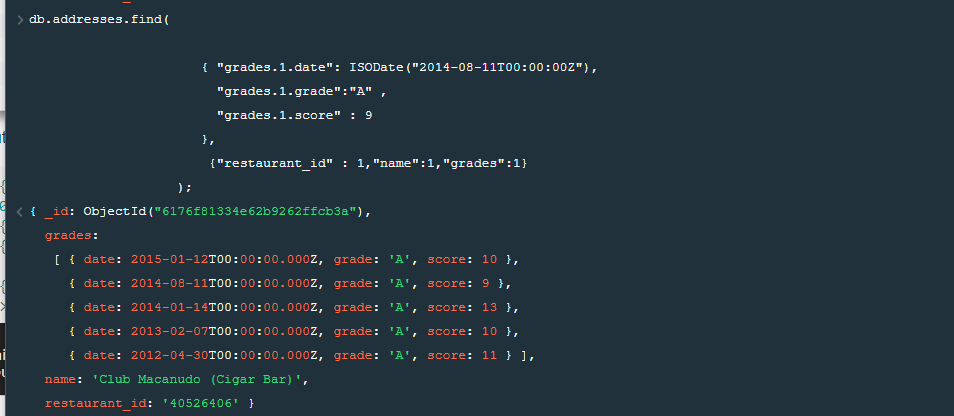
name: 'Olive\'S',

restaurant\_id: '40363151' }

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"



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..

db.addresses.find({$and : [{"address.coord.1": {$gt : 42}},{"address.coord.1": {$lte : 52}}]}, {\_id:0, restaurant\_id:1, name:1, address:1})

{ address:

{ building: '47',

coord: [ -78.877224, 42.89546199999999 ],

street: 'Broadway @ Trinity Pl',

zipcode: '10006' },

name: 'T.G.I. Friday\'S',

restaurant\_id: '40387990' }

{ address:

{ building: '1',

coord: [ -0.7119979, 51.6514664 ],

street: 'Pennplaza E, Penn Sta',

zipcode: '10001' },

name: 'T.G.I. Fridays',

restaurant\_id: '40388936' }

{ address:

{ building: '3000',

coord: [ -87.86567699999999, 42.61150920000001 ],

street: '47 Avenue',

zipcode: '11101' },

name: 'Di Luvio\'S Deli',

restaurant\_id: '40402284' }

{ address:

{ building: '21972199',

coord: [ -78.589606, 42.8912372 ],

street: 'Broadway',

zipcode: '10024' },

name: 'La Caridad 78',

restaurant\_id: '40568285' }

{ address:

{ building: '7981',

coord: [ -84.9751215, 45.4713351 ],

street: 'Hoyt Street',

zipcode: '11201' },

name: 'Bijan\'S',

restaurant\_id: '40876618' }

{ address:

{ building: '0',

coord: [ -88.0778799, 42.4154769 ],

street: '& Grand Central',

zipcode: '10017' },

name: 'Hyatt, Ny Central/Room Service',

restaurant\_id: '40879243' }

{ address:

{ building: '60',

coord: [ -111.9975205, 42.0970258 ],

street: 'West Side Highway',

zipcode: '10006' },

name: 'Sports Center At Chelsea Piers (Sushi Bar)',

restaurant\_id: '40882356' }

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

db.addresses.find({},{\_id:0, name:1}).sort( {name: 1})

{ name: '(Lewis Drug Store) Locanda Vini E Olii' }

{ name: '1 East 66Th Street Kitchen' }

{ name: '101 Deli' }

{ name: '101 Restaurant And Bar' }

{ name: '1020 Bar' }

{ name: '104-01 Foster Avenue Coffee Shop(Ups)' }

{ name: '10Th Avenue Pizza & Cafe' }

{ name: '111 Restaurant' }

{ name: '15 East Restaurant' }

{ name: '200 Fifth Avenue Restaurant & Sports Bar' }

{ name: '21 Club' }

{ name: '2A' }

{ name: '3 Deli & Grill' }

{ name: '3 Guys' }

{ name: '3 Guys Resturant' }

{ name: '42Nd Street Pizza Diner' }

{ name: '44 & X Hell\'S Kitchen' }

{ name: '44 Sw Ristorante & Bar' }

{ name: '5 Burro Cafe' }

{ name: '525 Lex Restaurant & Bar' }

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

db.addresses.find({},{\_id:0, name:1}).sort( {name: -1})

{ name: 'Zum Stammtisch' }

{ name: 'Zum Schneider' }

{ name: 'Zorba\'S' }

{ name: 'Zebu Grill' }

{ name: 'Zaro\'S Bread Basket' }

{ name: 'Zaro\'S Bread Basket' }

{ name: 'Zaro\'S Bread Basket' }

{ name: 'Zaro\'S Bread Basket' }

{ name: 'Zaro\'S Bread Basket' }

{ name: 'Zaro\'S Bakery' }

{ name: 'Zaro\'S Bakery' }

{ name: 'Zafi\'S Luncheonette' }

{ name: 'Yvonne Yvonne Restaurant' }

{ name: 'Yura & Company On Madison' }

{ name: 'Yummy Kitchen' }

{ name: 'Your Bakery' }

{ name: 'Yonah Shimmels Knishes' }

{ name: 'Yolanda Pizzeria Restaurant' }

{ name: 'Yip\'S' }

{ name: 'Yen Yen Restaurant' }

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.addresses.find({}, {\_id:0, cuisine:1, borough:1}).sort({cuisine: 1, borough: -1})

{ borough: 'Manhattan', cuisine: 'Afghan' }

{ borough: 'Manhattan', cuisine: 'Afghan' }

{ borough: 'Manhattan', cuisine: 'Afghan' }

{ borough: 'Manhattan', cuisine: 'Afghan' }

{ borough: 'Queens', cuisine: 'African' }

{ borough: 'Brooklyn', cuisine: 'African' }

{ borough: 'Bronx', cuisine: 'African' }

{ borough: 'Bronx', cuisine: 'African' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

{ borough: 'Staten Island', cuisine: 'American ' }

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

db.addresses.find(

{"address.street" :

{ $exists : true }

}

);

{ \_id: ObjectId("6176f81234e62b9262ffc50f"),

address:

{ building: '1007',

coord: [ -73.856077, 40.848447 ],

street: 'Morris Park Ave',

zipcode: '10462' },

borough: 'Bronx',

cuisine: 'Bakery',

grades:

[ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },

{ date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445' }

{ \_id: ObjectId("6176f81234e62b9262ffc510"),

address:

{ building: '469',

coord: [ -73.961704, 40.662942 ],

street: 'Flatbush Avenue',

zipcode: '11225' },

borough: 'Brooklyn',

cuisine: 'Hamburgers',

grades:

[ { date: 2014-12-30T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2014-07-01T00:00:00.000Z, grade: 'B', score: 23 },

{ date: 2013-04-30T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2012-05-08T00:00:00.000Z, grade: 'A', score: 12 } ],

name: 'Wendy\'S',

restaurant\_id: '30112340' }

{ \_id: ObjectId("6176f81234e62b9262ffc511"),

address:

{ building: '351',

coord: [ -73.98513559999999, 40.7676919 ],

street: 'West 57 Street',

zipcode: '10019' },

borough: 'Manhattan',

cuisine: 'Irish',

grades:

[ { date: 2014-09-06T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2013-07-22T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2012-07-31T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2011-12-29T00:00:00.000Z, grade: 'A', score: 12 } ],

name: 'Dj Reynolds Pub And Restaurant',

restaurant\_id: '30191841' }

{ \_id: ObjectId("6176f81234e62b9262ffc512"),

address:

{ building: '2780',

coord: [ -73.98241999999999, 40.579505 ],

street: 'Stillwell Avenue',

zipcode: '11224' },

borough: 'Brooklyn',

cuisine: 'American ',

grades:

[ { date: 2014-06-10T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2013-06-05T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-04-13T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2011-10-12T00:00:00.000Z, grade: 'A', score: 12 } ],

name: 'Riviera Caterer',

restaurant\_id: '40356018' }

{ \_id: ObjectId("6176f81234e62b9262ffc513"),

address:

{ building: '97-22',

coord: [ -73.8601152, 40.7311739 ],

street: '63 Road',

zipcode: '11374' },

borough: 'Queens',

cuisine: 'Jewish/Kosher',

grades:

[ { date: 2014-11-24T00:00:00.000Z, grade: 'Z', score: 20 },

{ date: 2013-01-17T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-08-02T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2011-12-15T00:00:00.000Z, grade: 'B', score: 25 } ],

name: 'Tov Kosher Kitchen',

restaurant\_id: '40356068' }

{ \_id: ObjectId("6176f81234e62b9262ffc514"),

address:

{ building: '8825',

coord: [ -73.8803827, 40.7643124 ],

street: 'Astoria Boulevard',

zipcode: '11369' },

borough: 'Queens',

cuisine: 'American ',

grades:

[ { date: 2014-11-15T00:00:00.000Z, grade: 'Z', score: 38 },

{ date: 2014-05-02T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2013-03-02T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-02-10T00:00:00.000Z, grade: 'A', score: 13 } ],

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151' }

{ \_id: ObjectId("6176f81234e62b9262ffc515"),

address:

{ building: '2206',

coord: [ -74.1377286, 40.6119572 ],

street: 'Victory Boulevard',

zipcode: '10314' },

borough: 'Staten Island',

cuisine: 'Jewish/Kosher',

grades:

[ { date: 2014-10-06T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2014-05-20T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-04-04T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2012-01-24T00:00:00.000Z, grade: 'A', score: 9 } ],

name: 'Kosher Island',

restaurant\_id: '40356442' }

{ \_id: ObjectId("6176f81234e62b9262ffc516"),

address:

{ building: '7114',

coord: [ -73.9068506, 40.6199034 ],

street: 'Avenue U',

zipcode: '11234' },

borough: 'Brooklyn',

cuisine: 'Delicatessen',

grades:

[ { date: 2014-05-29T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2014-01-14T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2013-08-03T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2012-07-18T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2012-03-09T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2011-10-14T00:00:00.000Z, grade: 'A', score: 9 } ],

name: 'Wilken\'S Fine Food',

restaurant\_id: '40356483' }

{ \_id: ObjectId("6176f81234e62b9262ffc517"),

address:

{ building: '6409',

coord: [ -74.00528899999999, 40.628886 ],

street: '11 Avenue',

zipcode: '11219' },

borough: 'Brooklyn',

cuisine: 'American ',

grades:

[ { date: 2014-07-18T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-07-30T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-02-13T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2012-08-16T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2011-08-17T00:00:00.000Z, grade: 'A', score: 11 } ],

name: 'Regina Caterers',

restaurant\_id: '40356649' }

{ \_id: ObjectId("6176f81234e62b9262ffc518"),

address:

{ building: '1839',

coord: [ -73.9482609, 40.6408271 ],

street: 'Nostrand Avenue',

zipcode: '11226' },

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

grades:

[ { date: 2014-07-14T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-07-10T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2012-07-11T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2012-02-23T00:00:00.000Z, grade: 'A', score: 8 } ],

name: 'Taste The Tropics Ice Cream',

restaurant\_id: '40356731' }

{ \_id: ObjectId("6176f81234e62b9262ffc519"),

address:

{ building: '2300',

coord: [ -73.8786113, 40.8502883 ],

street: 'Southern Boulevard',

zipcode: '10460' },

borough: 'Bronx',

cuisine: 'American ',

grades:

[ { date: 2014-05-28T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-06-19T00:00:00.000Z, grade: 'A', score: 4 },

{ date: 2012-06-15T00:00:00.000Z, grade: 'A', score: 3 } ],

name: 'Wild Asia',

restaurant\_id: '40357217' }

{ \_id: ObjectId("6176f81234e62b9262ffc51a"),

address:

{ building: '7715',

coord: [ -73.9973325, 40.61174889999999 ],

street: '18 Avenue',

zipcode: '11214' },

borough: 'Brooklyn',

cuisine: 'American ',

grades:

[ { date: 2014-04-16T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2013-04-23T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2012-04-24T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2011-12-16T00:00:00.000Z, grade: 'A', score: 2 } ],

name: 'C & C Catering Service',

restaurant\_id: '40357437' }

{ \_id: ObjectId("6176f81234e62b9262ffc51b"),

address:

{ building: '1269',

coord: [ -73.871194, 40.6730975 ],

street: 'Sutter Avenue',

zipcode: '11208' },

borough: 'Brooklyn',

cuisine: 'Chinese',

grades:

[ { date: 2014-09-16T00:00:00.000Z, grade: 'B', score: 21 },

{ date: 2013-08-28T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2013-04-02T00:00:00.000Z, grade: 'C', score: 56 },

{ date: 2012-08-15T00:00:00.000Z, grade: 'B', score: 27 },

{ date: 2012-03-28T00:00:00.000Z, grade: 'B', score: 27 } ],

name: 'May May Kitchen',

restaurant\_id: '40358429' }

{ \_id: ObjectId("6176f81234e62b9262ffc51c"),

address:

{ building: '1',

coord: [ -73.96926909999999, 40.7685235 ],

street: 'East 66 Street',

zipcode: '10065' },

borough: 'Manhattan',

cuisine: 'American ',

grades:

[ { date: 2014-05-07T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2013-05-03T00:00:00.000Z, grade: 'A', score: 4 },

{ date: 2012-04-30T00:00:00.000Z, grade: 'A', score: 6 },

{ date: 2011-12-27T00:00:00.000Z, grade: 'A', score: 0 } ],

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480' }

{ \_id: ObjectId("6176f81234e62b9262ffc51d"),

address:

{ building: '705',

coord: [ -73.9653967, 40.6064339 ],

street: 'Kings Highway',

zipcode: '11223' },

borough: 'Brooklyn',

cuisine: 'Jewish/Kosher',

grades:

[ { date: 2014-11-10T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-10-10T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-10-04T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-05-21T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2011-12-30T00:00:00.000Z, grade: 'B', score: 19 } ],

name: 'Seuda Foods',

restaurant\_id: '40360045' }

{ \_id: ObjectId("6176f81234e62b9262ffc51e"),

address:

{ building: '203',

coord: [ -73.97822040000001, 40.6435254 ],

street: 'Church Avenue',

zipcode: '11218' },

borough: 'Brooklyn',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

grades:

[ { date: 2014-02-10T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2013-01-02T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-01-09T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2011-11-07T00:00:00.000Z, grade: 'P', score: 12 },

{ date: 2011-07-21T00:00:00.000Z, grade: 'A', score: 13 } ],

name: 'Carvel Ice Cream',

restaurant\_id: '40360076' }

{ \_id: ObjectId("6176f81234e62b9262ffc51f"),

address:

{ building: '265-15',

coord: [ -73.7032601, 40.7386417 ],

street: 'Hillside Avenue',

zipcode: '11004' },

borough: 'Queens',

cuisine: 'Ice Cream, Gelato, Yogurt, Ices',

grades:

[ { date: 2014-10-28T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2013-09-18T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2012-09-20T00:00:00.000Z, grade: 'A', score: 13 } ],

name: 'Carvel Ice Cream',

restaurant\_id: '40361322' }

{ \_id: ObjectId("6176f81234e62b9262ffc520"),

address:

{ building: '6909',

coord: [ -74.0259567, 40.6353674 ],

street: '3 Avenue',

zipcode: '11209' },

borough: 'Brooklyn',

cuisine: 'Delicatessen',

grades:

[ { date: 2014-08-21T00:00:00.000Z, grade: 'A', score: 4 },

{ date: 2014-03-05T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2013-01-10T00:00:00.000Z, grade: 'A', score: 10 } ],

name: 'Nordic Delicacies',

restaurant\_id: '40361390' }

{ \_id: ObjectId("6176f81234e62b9262ffc521"),

address:

{ building: '522',

coord: [ -73.95171, 40.767461 ],

street: 'East 74 Street',

zipcode: '10021' },

borough: 'Manhattan',

cuisine: 'American ',

grades:

[ { date: 2014-09-02T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-12-19T00:00:00.000Z, grade: 'B', score: 16 },

{ date: 2013-05-28T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2012-12-07T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-03-29T00:00:00.000Z, grade: 'A', score: 11 } ],

name: 'Glorious Food',

restaurant\_id: '40361521' }

{ \_id: ObjectId("6176f81234e62b9262ffc522"),

address:

{ building: '284',

coord: [ -73.9829239, 40.6580753 ],

street: 'Prospect Park West',

zipcode: '11215' },

borough: 'Brooklyn',

cuisine: 'American ',

grades:

[ { date: 2014-11-19T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-11-14T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2012-12-05T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-05-17T00:00:00.000Z, grade: 'A', score: 11 } ],

name: 'The Movable Feast',

restaurant\_id: '40361606'

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

db.addresses.find({"address.coord": {$type: "double"}}, {\_id:0, address:1})

{ address:

{ building: '1007',

coord: [ -73.856077, 40.848447 ],

street: 'Morris Park Ave',

zipcode: '10462' } }

{ address:

{ building: '469',

coord: [ -73.961704, 40.662942 ],

street: 'Flatbush Avenue',

zipcode: '11225' } }

{ address:

{ building: '351',

coord: [ -73.98513559999999, 40.7676919 ],

street: 'West 57 Street',

zipcode: '10019' } }

{ address:

{ building: '2780',

coord: [ -73.98241999999999, 40.579505 ],

street: 'Stillwell Avenue',

zipcode: '11224' } }

{ address:

{ building: '97-22',

coord: [ -73.8601152, 40.7311739 ],

street: '63 Road',

zipcode: '11374' } }

{ address:

{ building: '8825',

coord: [ -73.8803827, 40.7643124 ],

street: 'Astoria Boulevard',

zipcode: '11369' } }

{ address:

{ building: '2206',

coord: [ -74.1377286, 40.6119572 ],

street: 'Victory Boulevard',

zipcode: '10314' } }

{ address:

{ building: '7114',

coord: [ -73.9068506, 40.6199034 ],

street: 'Avenue U',

zipcode: '11234' } }

{ address:

{ building: '6409',

coord: [ -74.00528899999999, 40.628886 ],

street: '11 Avenue',

zipcode: '11219' } }

{ address:

{ building: '1839',

coord: [ -73.9482609, 40.6408271 ],

street: 'Nostrand Avenue',

zipcode: '11226' } }

{ address:

{ building: '2300',

coord: [ -73.8786113, 40.8502883 ],

street: 'Southern Boulevard',

zipcode: '10460' } }

{ address:

{ building: '7715',

coord: [ -73.9973325, 40.61174889999999 ],

street: '18 Avenue',

zipcode: '11214' } }

{ address:

{ building: '1269',

coord: [ -73.871194, 40.6730975 ],

street: 'Sutter Avenue',

zipcode: '11208' } }

{ address:

{ building: '1',

coord: [ -73.96926909999999, 40.7685235 ],

street: 'East 66 Street',

zipcode: '10065' } }

{ address:

{ building: '705',

coord: [ -73.9653967, 40.6064339 ],

street: 'Kings Highway',

zipcode: '11223' } }

{ address:

{ building: '203',

coord: [ -73.97822040000001, 40.6435254 ],

street: 'Church Avenue',

zipcode: '11218' } }

{ address:

{ building: '265-15',

coord: [ -73.7032601, 40.7386417 ],

street: 'Hillside Avenue',

zipcode: '11004' } }

{ address:

{ building: '6909',

coord: [ -74.0259567, 40.6353674 ],

street: '3 Avenue',

zipcode: '11209' } }

{ address:

{ building: '522',

coord: [ -73.95171, 40.767461 ],

street: 'East 74 Street',

zipcode: '10021' } }

{ address:

{ building: '284',

coord: [ -73.9829239, 40.6580753 ],

street: 'Prospect Park West',

zipcode: '11215' } }

restaurants

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.

db.addresses.find({"grades": {$elemMatch: {"score": {$mod: [7,0]}}}},{\_id:0, restaurant\_id:1, name:1, grades:1})

{ grades:

[ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },

{ date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],

name: 'Morris Park Bake Shop',

restaurant\_id: '30075445' }

{ grades:

[ { date: 2014-06-10T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2013-06-05T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-04-13T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2011-10-12T00:00:00.000Z, grade: 'A', score: 12 } ],

name: 'Riviera Caterer',

restaurant\_id: '40356018' }

{ grades:

[ { date: 2014-11-15T00:00:00.000Z, grade: 'Z', score: 38 },

{ date: 2014-05-02T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2013-03-02T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-02-10T00:00:00.000Z, grade: 'A', score: 13 } ],

name: 'Brunos On The Boulevard',

restaurant\_id: '40356151' }

{ grades:

[ { date: 2014-09-16T00:00:00.000Z, grade: 'B', score: 21 },

{ date: 2013-08-28T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2013-04-02T00:00:00.000Z, grade: 'C', score: 56 },

{ date: 2012-08-15T00:00:00.000Z, grade: 'B', score: 27 },

{ date: 2012-03-28T00:00:00.000Z, grade: 'B', score: 27 } ],

name: 'May May Kitchen',

restaurant\_id: '40358429' }

{ grades:

[ { date: 2014-05-07T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2013-05-03T00:00:00.000Z, grade: 'A', score: 4 },

{ date: 2012-04-30T00:00:00.000Z, grade: 'A', score: 6 },

{ date: 2011-12-27T00:00:00.000Z, grade: 'A', score: 0 } ],

name: '1 East 66Th Street Kitchen',

restaurant\_id: '40359480' }

{ grades:

[ { date: 2014-11-10T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-10-10T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-10-04T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-05-21T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2011-12-30T00:00:00.000Z, grade: 'B', score: 19 } ],

name: 'Seuda Foods',

restaurant\_id: '40360045' }

{ grades:

[ { date: 2014-08-16T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-08-27T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2012-09-20T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2011-09-29T00:00:00.000Z, grade: 'A', score: 10 } ],

name: 'Sal\'S Deli',

restaurant\_id: '40361618' }

{ grades:

[ { date: 2014-03-19T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2013-03-13T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2012-03-27T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2011-04-05T00:00:00.000Z, grade: 'A', score: 7 } ],

name: 'Steve Chu\'S Deli & Grocery',

restaurant\_id: '40361998' }

{ grades:

[ { date: 2014-09-15T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2014-03-04T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2013-07-18T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2013-01-09T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2012-04-10T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2011-11-15T00:00:00.000Z, grade: 'A', score: 7 } ],

name: 'Harriet\'S Kitchen',

restaurant\_id: '40362098' }

{ grades:

[ { date: 2014-03-18T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2013-03-18T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2012-10-10T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2011-09-22T00:00:00.000Z, grade: 'A', score: 2 } ],

name: 'White Castle',

restaurant\_id: '40362344' }

{ grades:

[ { date: 2014-04-21T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2013-11-12T00:00:00.000Z, grade: 'A', score: 5 },

{ date: 2013-06-04T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2012-11-14T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2012-10-11T00:00:00.000Z, grade: 'P', score: 0 },

{ date: 2012-05-24T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2011-12-08T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2011-07-20T00:00:00.000Z, grade: 'A', score: 11 } ],

name: 'Ho Mei Restaurant',

restaurant\_id: '40362432' }

{ grades:

[ { date: 2014-05-13T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2013-05-08T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-09-22T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2011-06-06T00:00:00.000Z, grade: 'A', score: 12 } ],

name: 'Shashemene Int\'L Restaura',

restaurant\_id: '40362869' }

{ grades:

[ { date: 2014-10-17T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-09-18T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2013-04-30T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-04-20T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2011-12-19T00:00:00.000Z, grade: 'A', score: 3 } ],

name: 'Olive\'S',

restaurant\_id: '40363151' }

{ grades:

[ { date: 2014-02-21T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2013-09-13T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2012-08-28T00:00:00.000Z, grade: 'A', score: 0 },

{ date: 2011-09-13T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2011-05-03T00:00:00.000Z, grade: 'A', score: 5 } ],

name: 'Lexler Deli',

restaurant\_id: '40363426' }

{ grades:

[ { date: 2014-02-25T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2013-08-14T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2012-08-07T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-03-26T00:00:00.000Z, grade: 'A', score: 10 },

{ date: 2011-11-04T00:00:00.000Z, grade: 'A', score: 0 },

{ date: 2011-06-29T00:00:00.000Z, grade: 'A', score: 4 } ],

name: 'Snack Time Grill',

restaurant\_id: '40363590' }

{ grades:

[ { date: 2014-06-02T00:00:00.000Z, grade: 'A', score: 9 },

{ date: 2013-12-27T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2013-03-18T00:00:00.000Z, grade: 'B', score: 26 },

{ date: 2012-02-01T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2011-07-06T00:00:00.000Z, grade: 'B', score: 25 } ],

name: 'Lorenzo & Maria\'S',

restaurant\_id: '40363630' }

{ grades:

[ { date: 2014-08-05T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2014-03-06T00:00:00.000Z, grade: 'A', score: 11 },

{ date: 2013-07-09T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-01-30T00:00:00.000Z, grade: 'A', score: 4 },

{ date: 2012-01-05T00:00:00.000Z, grade: 'A', score: 2 },

{ date: 2011-09-26T00:00:00.000Z, grade: 'A', score: 0 } ],

name: 'Domino\'S Pizza',

restaurant\_id: '40363644' }

{ grades:

[ { date: 2014-02-05T00:00:00.000Z, grade: 'A', score: 0 },

{ date: 2013-01-29T00:00:00.000Z, grade: 'A', score: 3 },

{ date: 2011-12-08T00:00:00.000Z, grade: 'A', score: 10 } ],

name: 'Sonny\'S Heros',

restaurant\_id: '40363744' }

{ grades:

[ { date: 2014-11-18T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-11-07T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2013-04-24T00:00:00.000Z, grade: 'A', score: 7 },

{ date: 2012-03-20T00:00:00.000Z, grade: 'A', score: 5 } ],

name: 'Plaza Bagels & Deli',

restaurant\_id: '40364286' }

{ grades:

[ { date: 2015-01-06T00:00:00.000Z, grade: 'A', score: 12 },

{ date: 2014-07-15T00:00:00.000Z, grade: 'C', score: 48 },

{ date: 2013-05-02T00:00:00.000Z, grade: 'A', score: 13 },

{ date: 2012-09-24T00:00:00.000Z, grade: 'A', score: 8 },

{ date: 2012-04-19T00:00:00.000Z, grade: 'A', score: 7 } ],

name: 'Texas Rotisserie',

restaurant\_id: '40364304' }

restaurants

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.

db.addresses.find({name: {$regex: /mon/}},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})

{ address: { coord: [ -73.98306099999999, 40.7441419 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Desmond\'S Tavern' }

{ address: { coord: [ -73.8221418, 40.7272376 ] },

borough: 'Queens',

cuisine: 'Jewish/Kosher',

name: 'Shimons Kosher Pizza' }

{ address: { coord: [ -74.10465599999999, 40.58834 ] },

borough: 'Staten Island',

cuisine: 'American ',

name: 'Richmond County Country Club' }

{ address: { coord: [ -73.9812843, 40.5947365 ] },

borough: 'Brooklyn',

cuisine: 'Pizza/Italian',

name: 'Lb Spumoni Gardens' }

{ address: { coord: [ -73.951199, 40.7166026 ] },

borough: 'Brooklyn',

cuisine: 'Italian',

name: 'Bamonte\'S Restaurant' }

{ address: { coord: [ -73.924072, 40.76108900000001 ] },

borough: 'Queens',

cuisine: 'Greek',

name: 'Omonia Cafe' }

{ address: { coord: [ -73.9901605, 40.7526176 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Delmonico\'S Kitchen' }

{ address: { coord: [ -73.9707505, 40.7635651 ] },

borough: 'Manhattan',

cuisine: 'Delicatessen',

name: 'Delmonico Gourmet' }

{ address: { coord: [ -73.9760637, 40.7508686 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Delmonico Gourmet' }

{ address: { coord: [ -73.97198209999999, 40.764464 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Harmonie Club' }

{ address: { coord: [ -73.79571990000001, 40.7095637 ] },

borough: 'Queens',

cuisine: 'Bakery',

name: 'Ramona\'S Bakery' }

{ address: { coord: [ -73.98463480000001, 40.7630755 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Neil Simon Theatre' }

{ address: { coord: [ -74.028486, 40.630438 ] },

borough: 'Brooklyn',

cuisine: 'Mediterranean',

name: 'Omonia Cafe' }

{ address: { coord: [ -73.9849976, 40.7276766 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Simone Bar Cafe' }

{ address: { coord: [ -74.01032339999999, 40.7050156 ] },

borough: 'Manhattan',

cuisine: 'Italian',

name: 'Delmonicos' }

{ address: { coord: [ -73.8706606, 40.7342757 ] },

borough: 'Queens',

cuisine: 'American ',

name: 'Cinnabon World Famous Cinnamon Rolls' }

{ address: { coord: [ -73.9640299, 40.807001 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Ferris Booth Commons - Alfred Lerner Hall' }

{ address: { coord: [ -74.10465599999999, 40.58834 ] },

borough: 'Staten Island',

cuisine: 'American ',

name: 'Richmond County Country Club - Pool Snack Bar' }

{ address: { coord: [ -74.1110561, 40.5884772 ] },

borough: 'Staten Island',

cuisine: 'American ',

name: 'Richmond County Country Club(10Th Hole)' }

{ address: { coord: [ -74.112758, 40.5833299 ] },

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.

db.addresses.find({name: {$regex: /^Mad.\*/}},{\_id:0, name:1, borough:1, "address.coord":1, cuisine:1})

{ address: { coord: [ -73.9860597, 40.7431194 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Madison Square' }

{ address: { coord: [ -73.98302199999999, 40.742313 ] },

borough: 'Manhattan',

cuisine: 'Indian',

name: 'Madras Mahal' }

{ address: { coord: [ -74.000002, 40.72735 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Madame X' }

{ address: { coord: [ -73.98171959999999, 40.7499406 ] },

borough: 'Manhattan',

cuisine: 'French',

name: 'Madison Bistro' }

{ address: { coord: [ -73.9717845, 40.6897199 ] },

borough: 'Brooklyn',

cuisine: 'African',

name: 'Madiba' }

{ address: { coord: [ -73.9040753, 40.9069011 ] },

borough: 'Bronx',

cuisine: 'Italian',

name: 'Madison\'S' }

{ address: { coord: [ -73.9886598, 40.7565811 ] },

borough: 'Manhattan',

cuisine: 'Hotdogs',

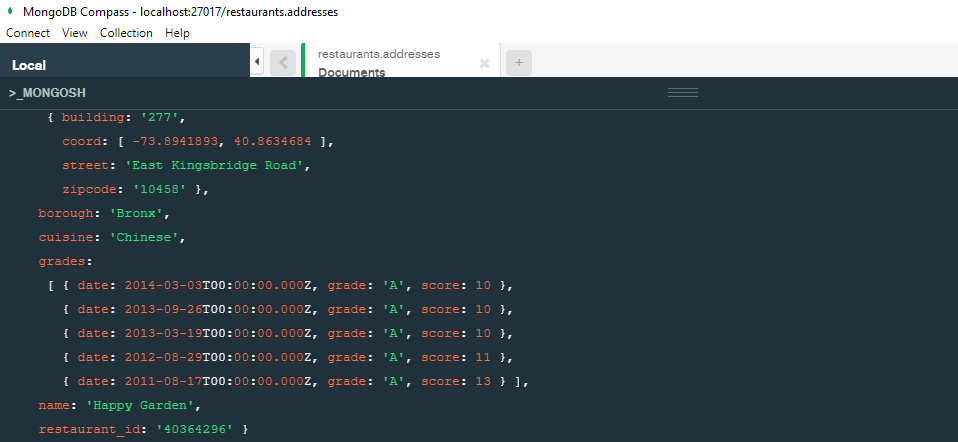
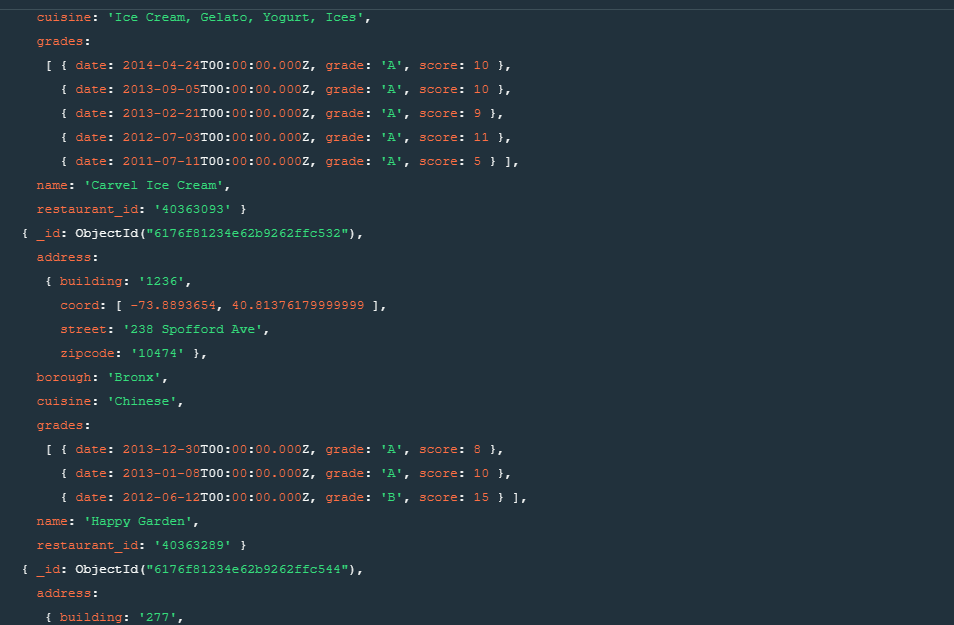
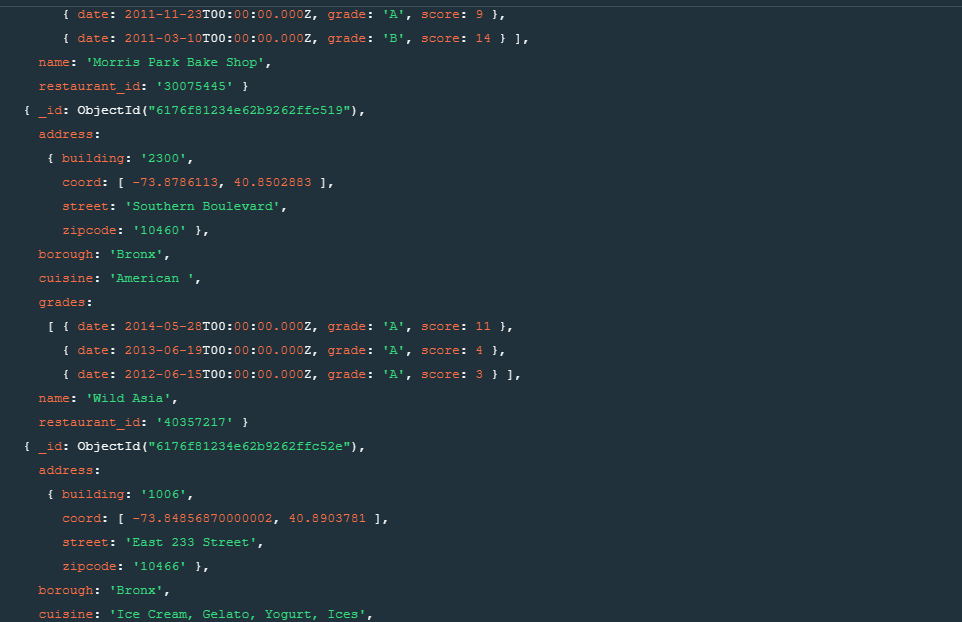
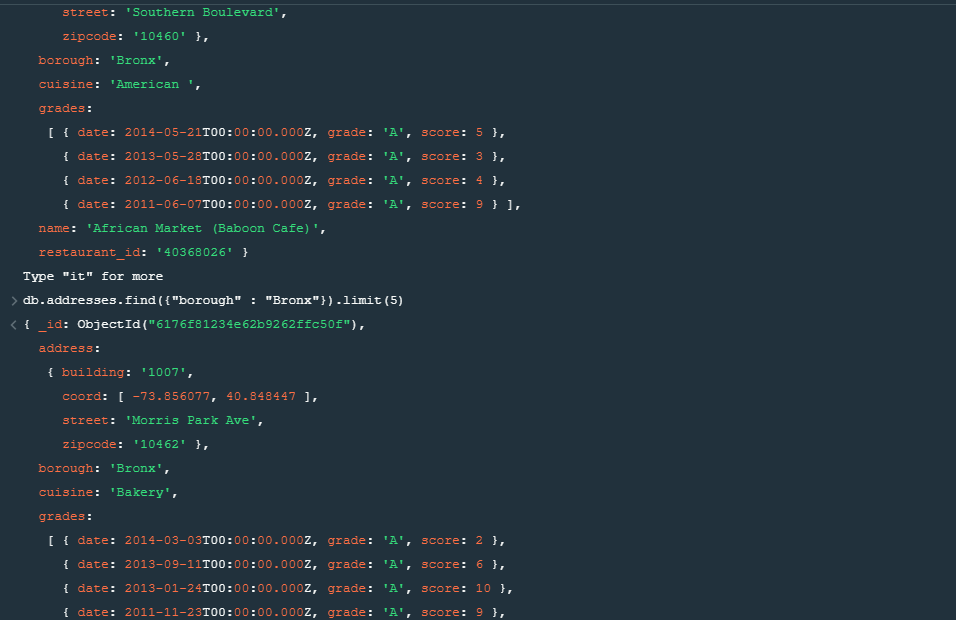
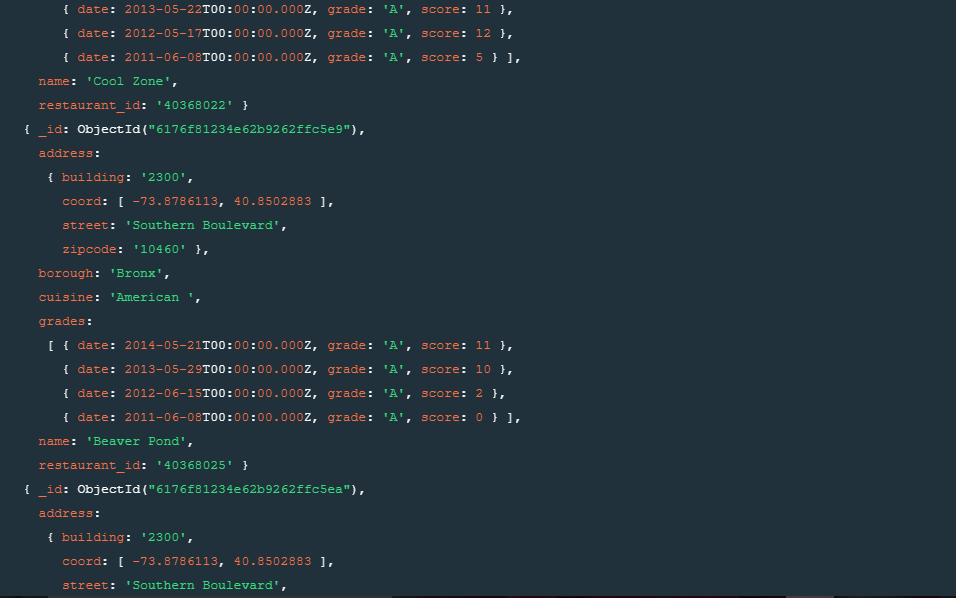
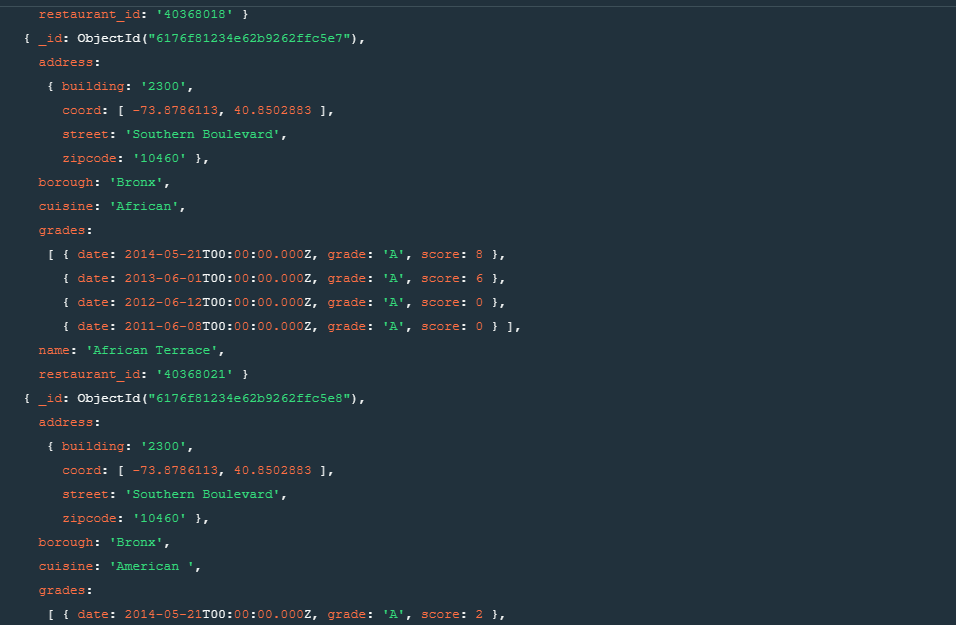
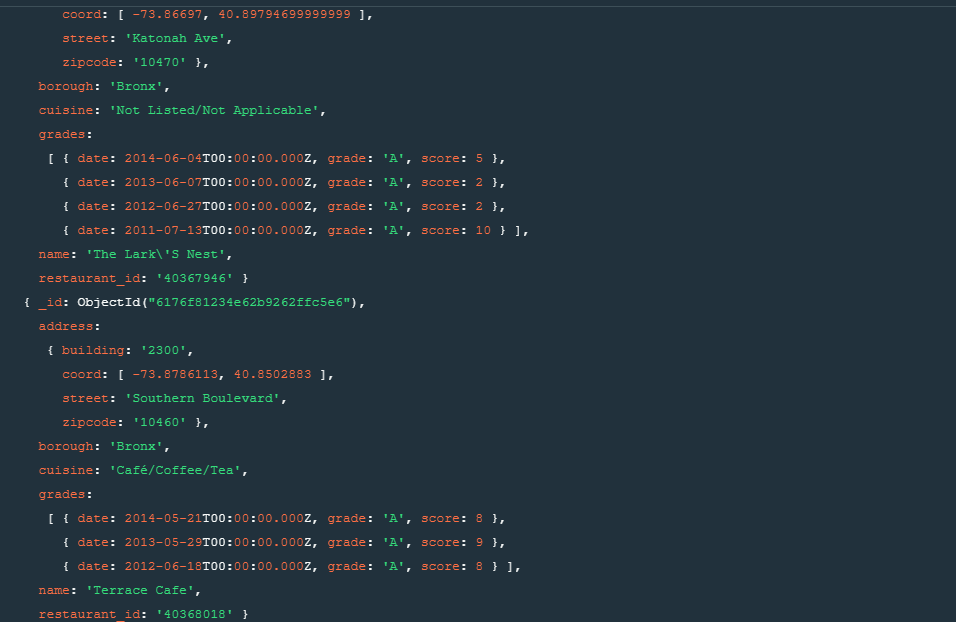
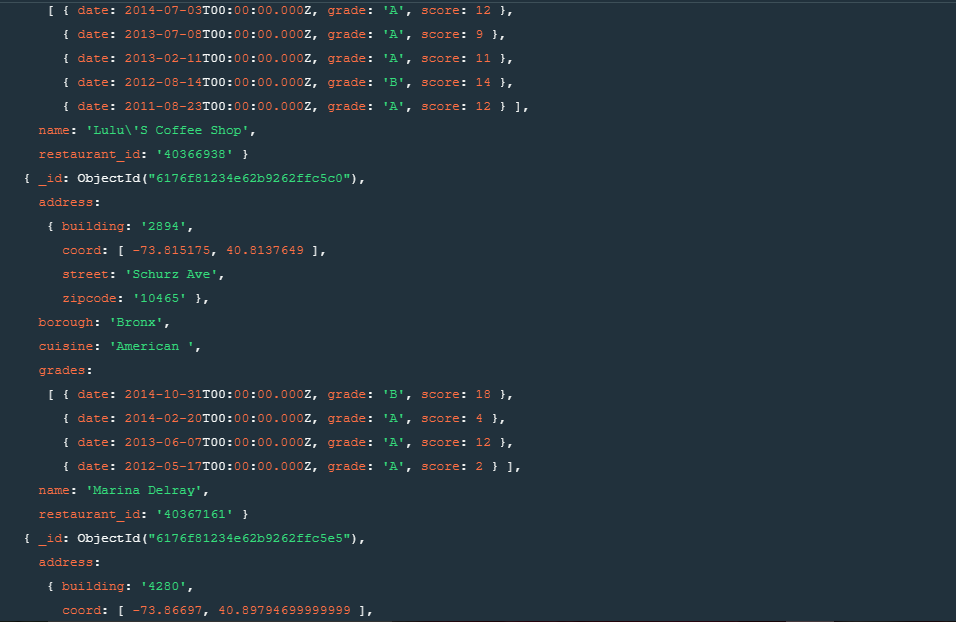
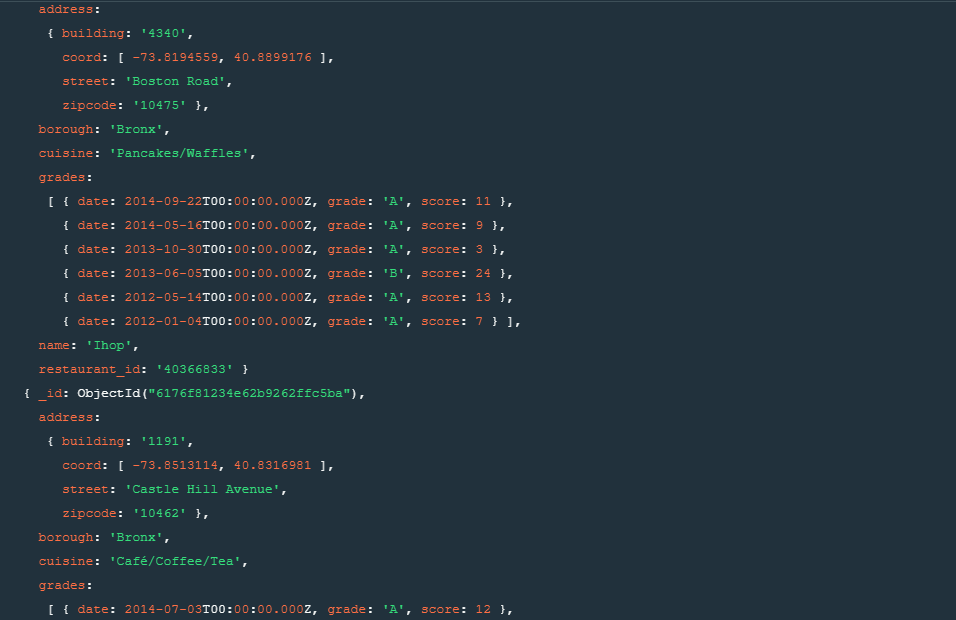
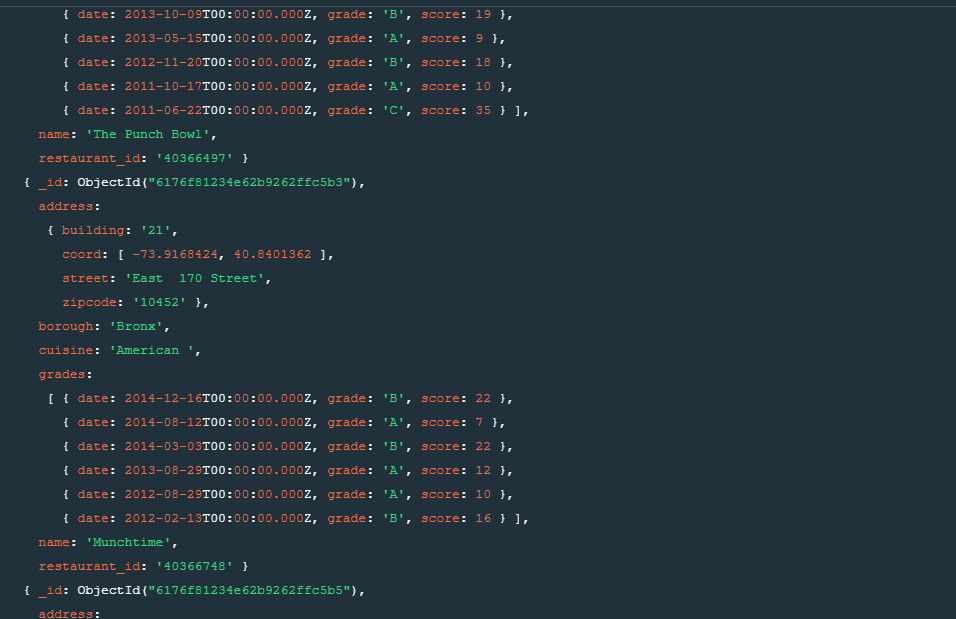
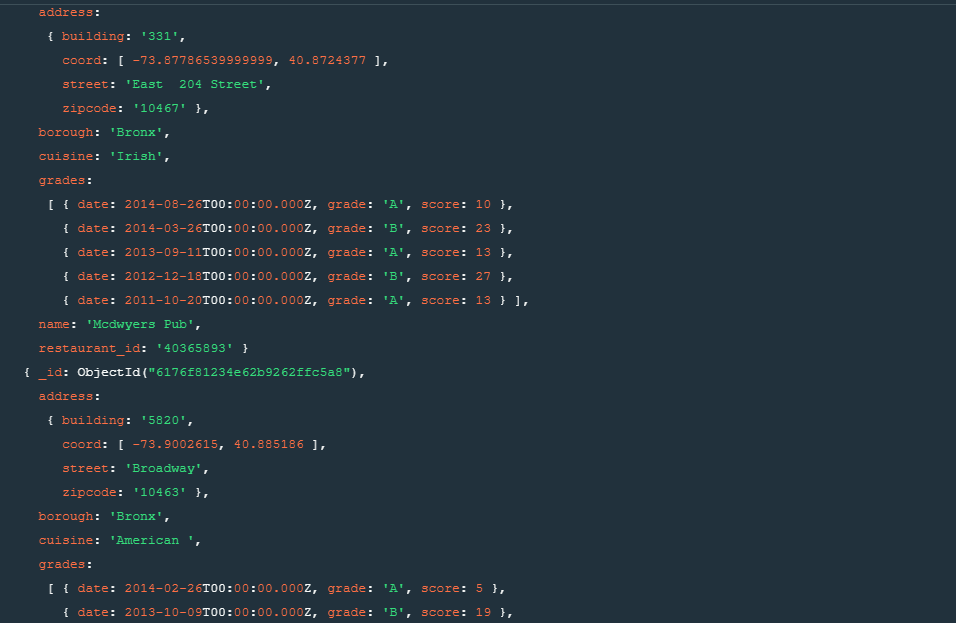
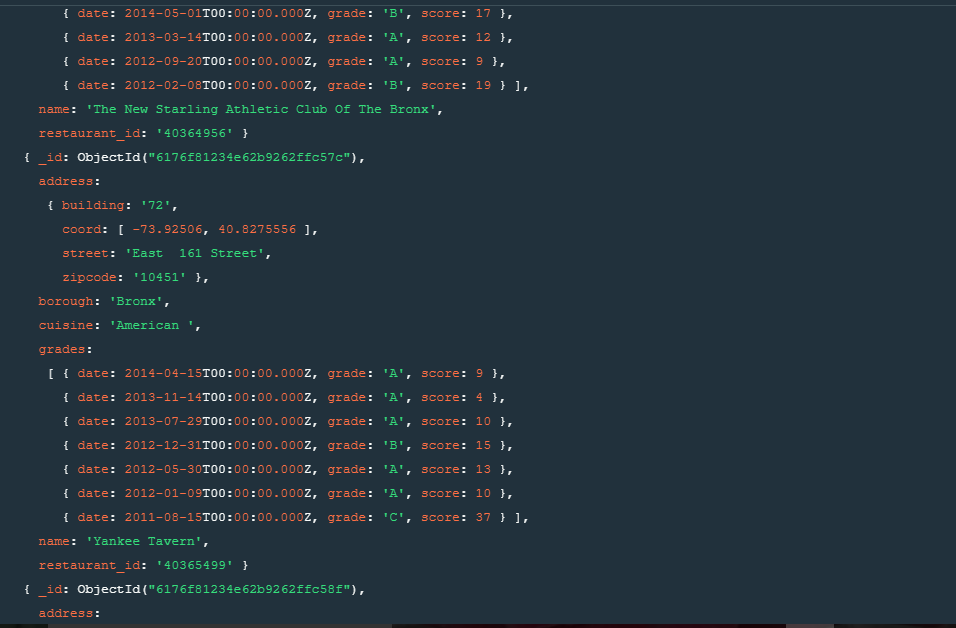
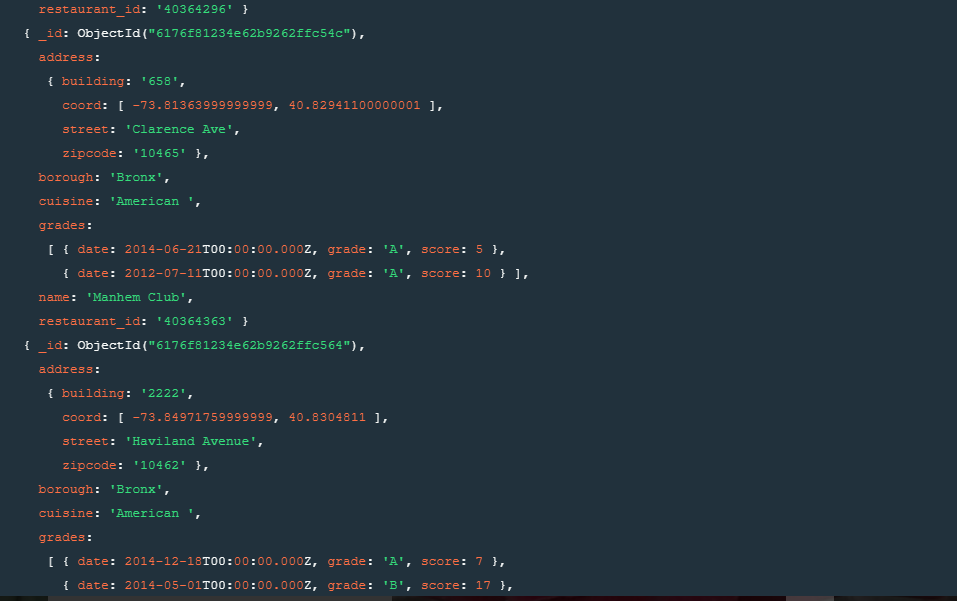
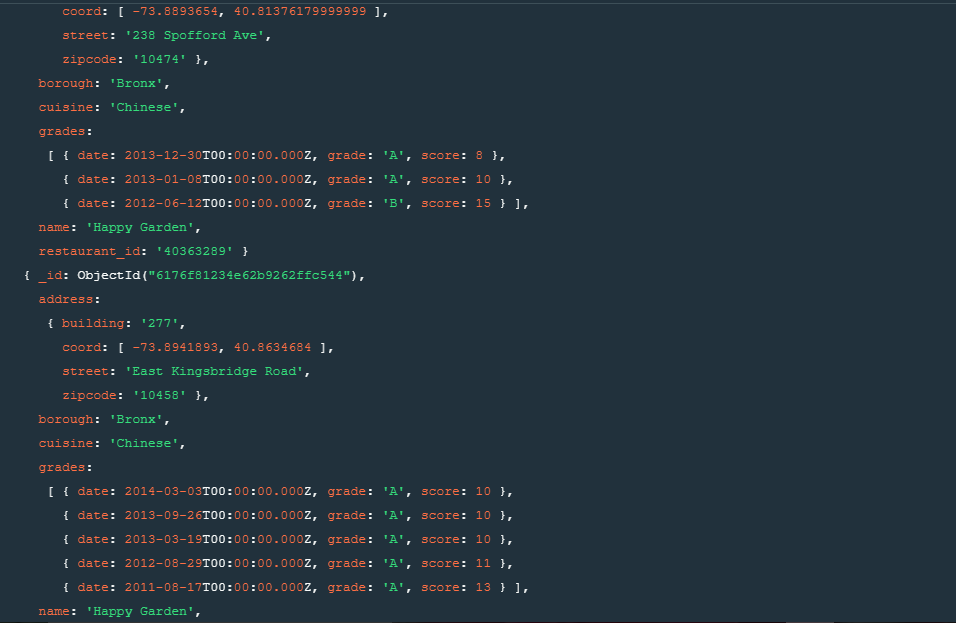
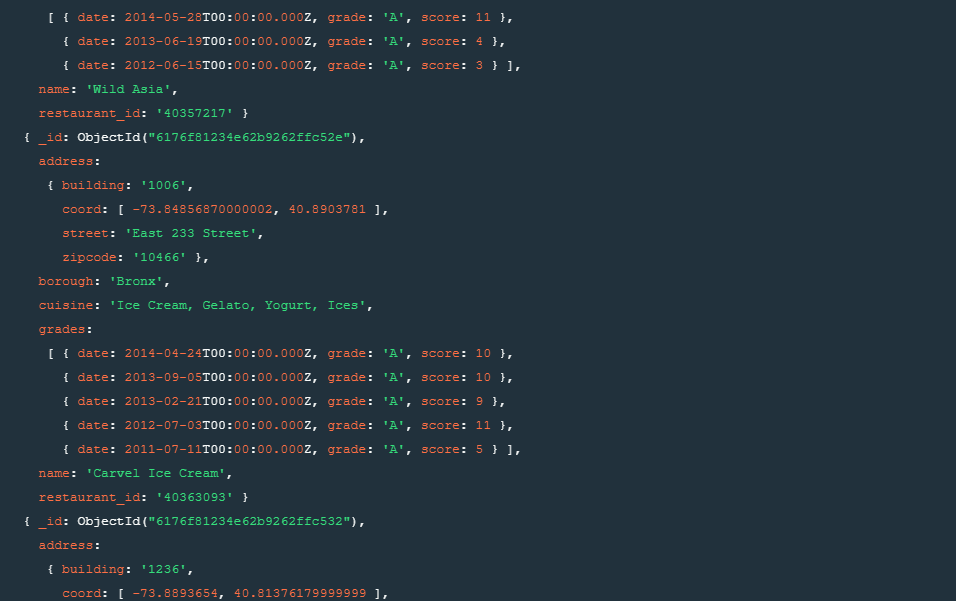
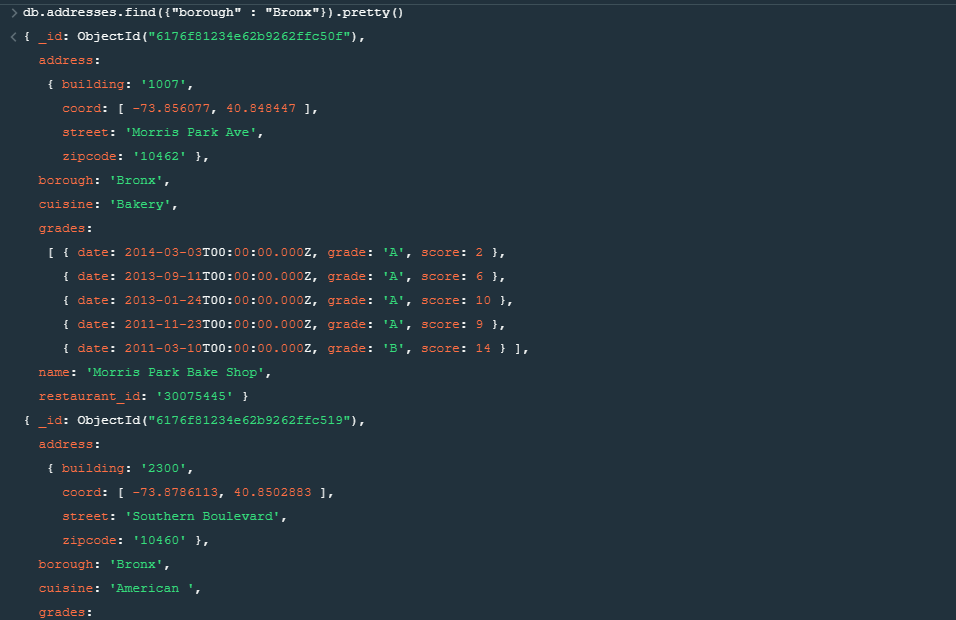
name: 'Madame Tussaud\'S' }

{ address: { coord: [ -73.95623719999999, 40.7761697 ] },

borough: 'Manhattan',

cuisine: 'American ',

name: 'Mad River Bar & Grille' }



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