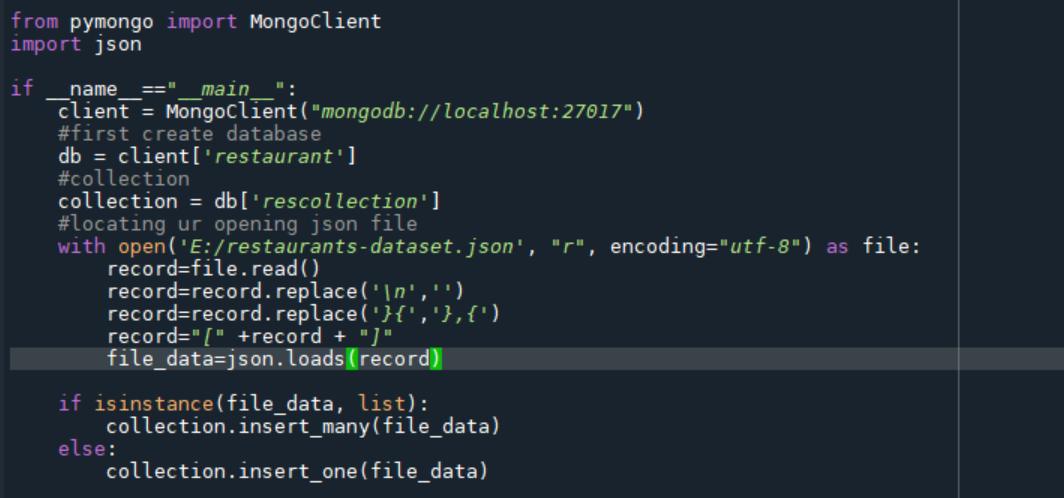
**Assignment 5**

**MongoDB Restaurant Analysis**

1. Create database – restaurant, create collection – rescollection. Insert the documents into collections.

Query-



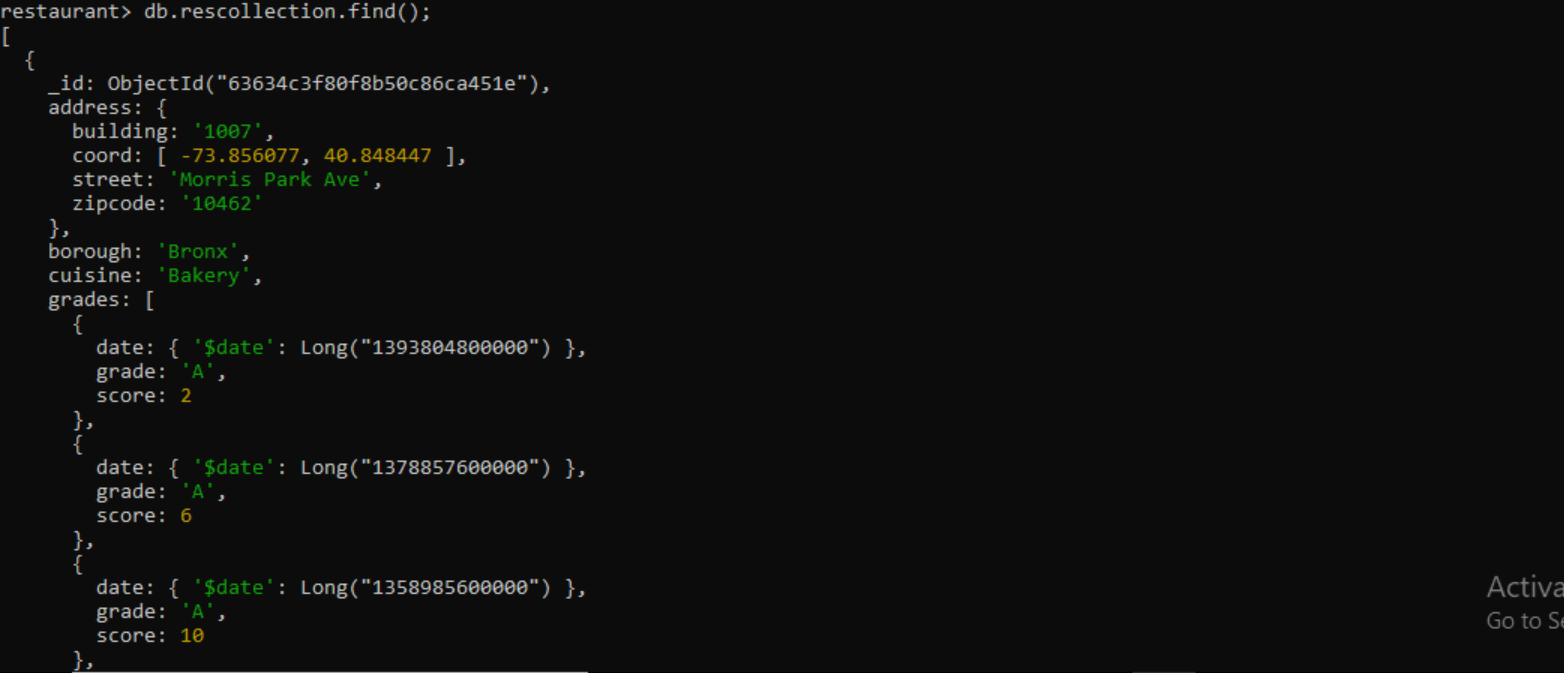
Output-



1. Display all the documents in the collection restaurants.

Query- db.rescollection.find()

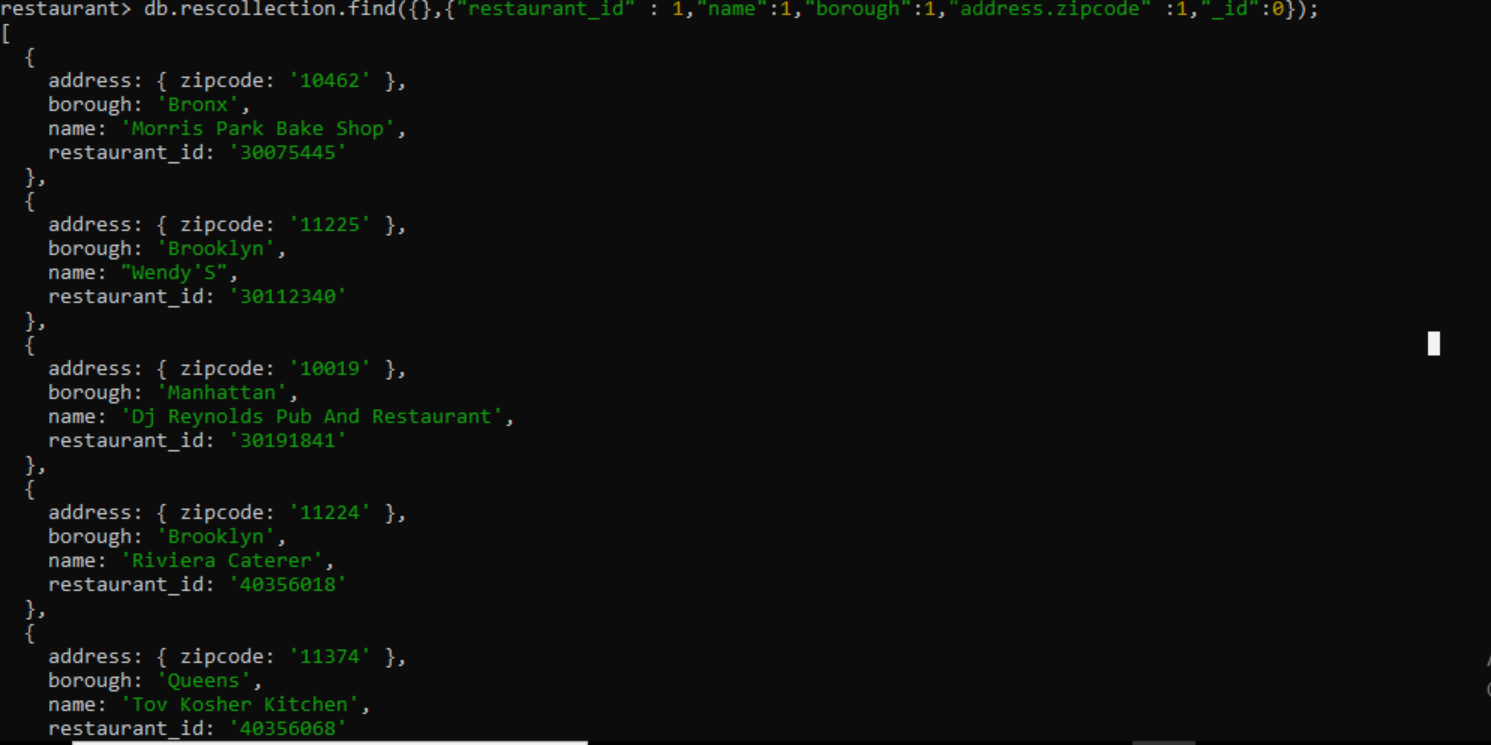
Output-



1. Display the fields restaurant\_id, name, borough, and zip code, but exclude the field \_id for all the documents in the collection restaurant.

Query- db.rescollection.find({},{"restaurant\_id" : 1,"name":1,"borough":1,"address.zipcode" :1,"\_id":0});

Output-



1. Find the restaurants who achieved a score more than 90.

Query- db.rescollection.find({grades : { $elemMatch:{"score":{$gt : 90}}}});

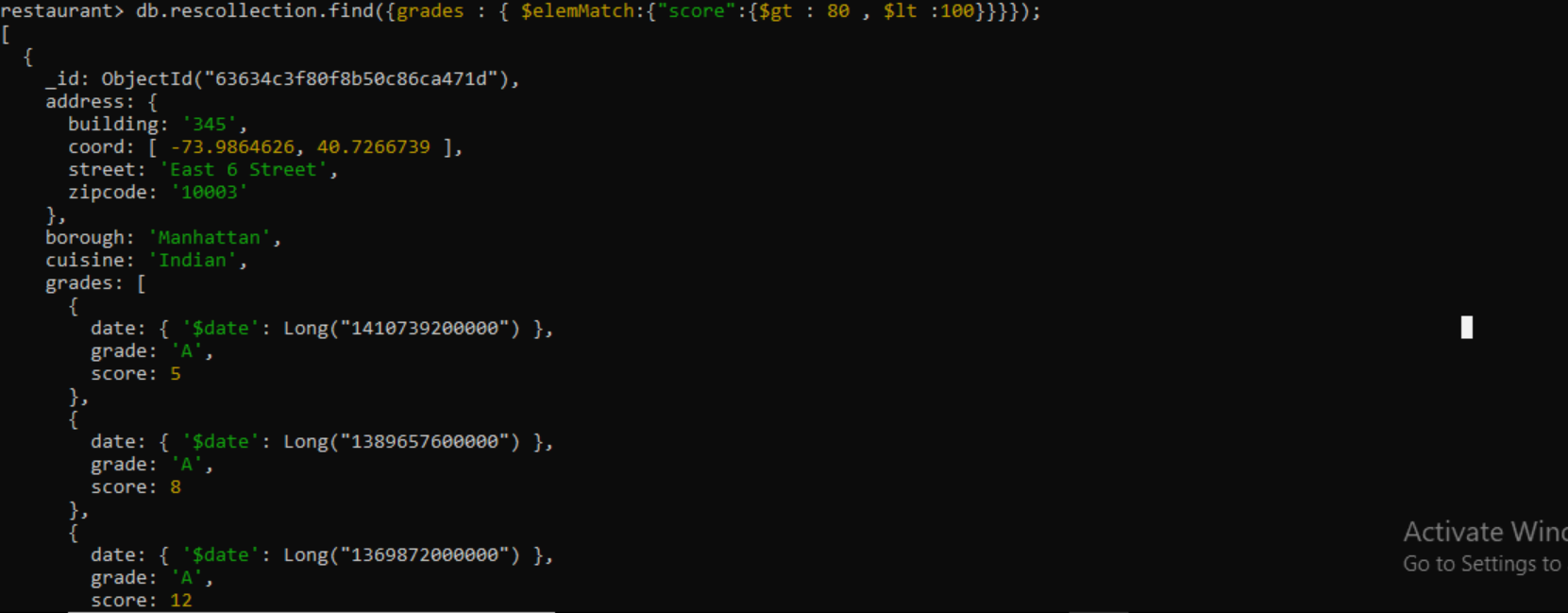
Output-



1. Show the restaurants that achieved a score, more than 80 but less than 100.

Query- db.rescollection.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt :100}}}});

Output-



1. Write Query to show the restaurants that do not prepare any cuisine of american & their grade score > 70.

Query- db.rescollection.find(

{$and:

[

{"cuisine" : {$ne :"American "}},

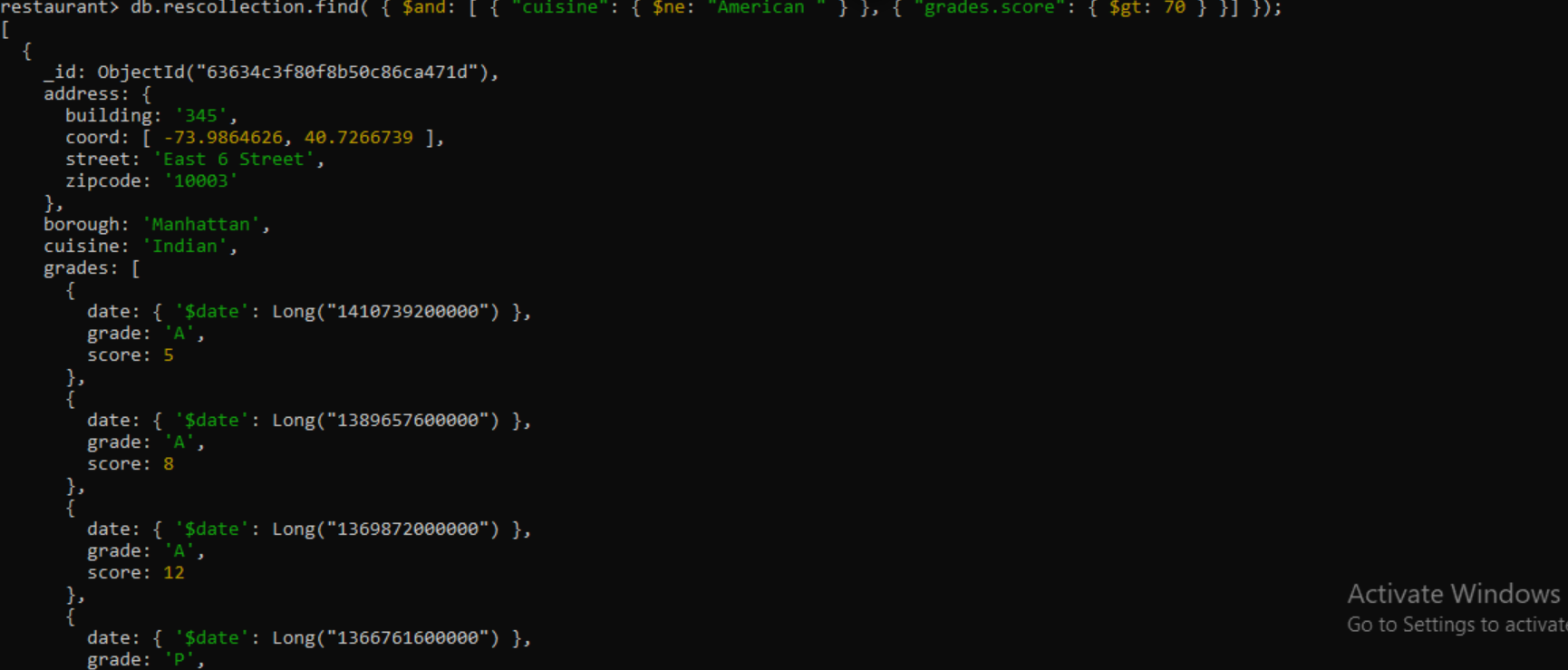
{"grades.score" : {$gt : 70}},

]

}

);

Output-



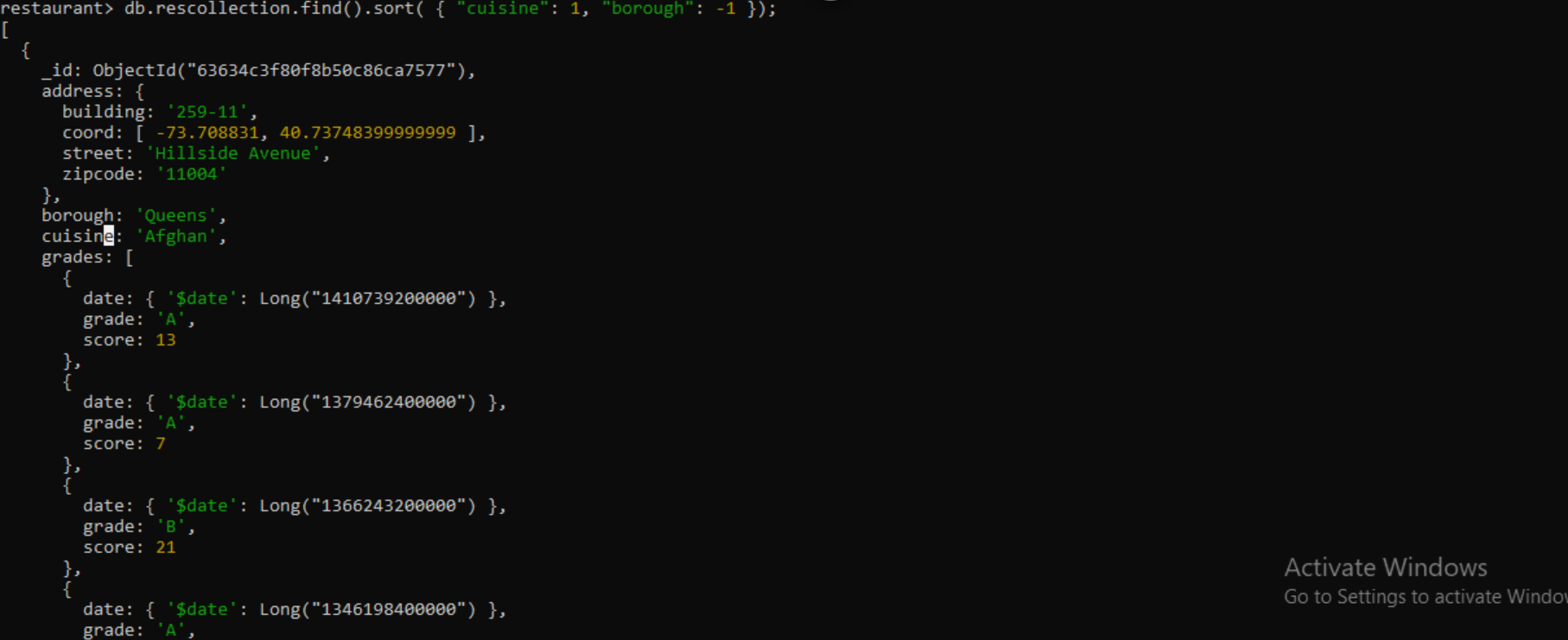
1. Write a MongoDB query to arrange the name of the cuisine in an ascending order and for that same borough arranged in descending order.

Query- db.rescollection.find().sort(

{"cuisine":1,"borough" : -1,}

);

Output-



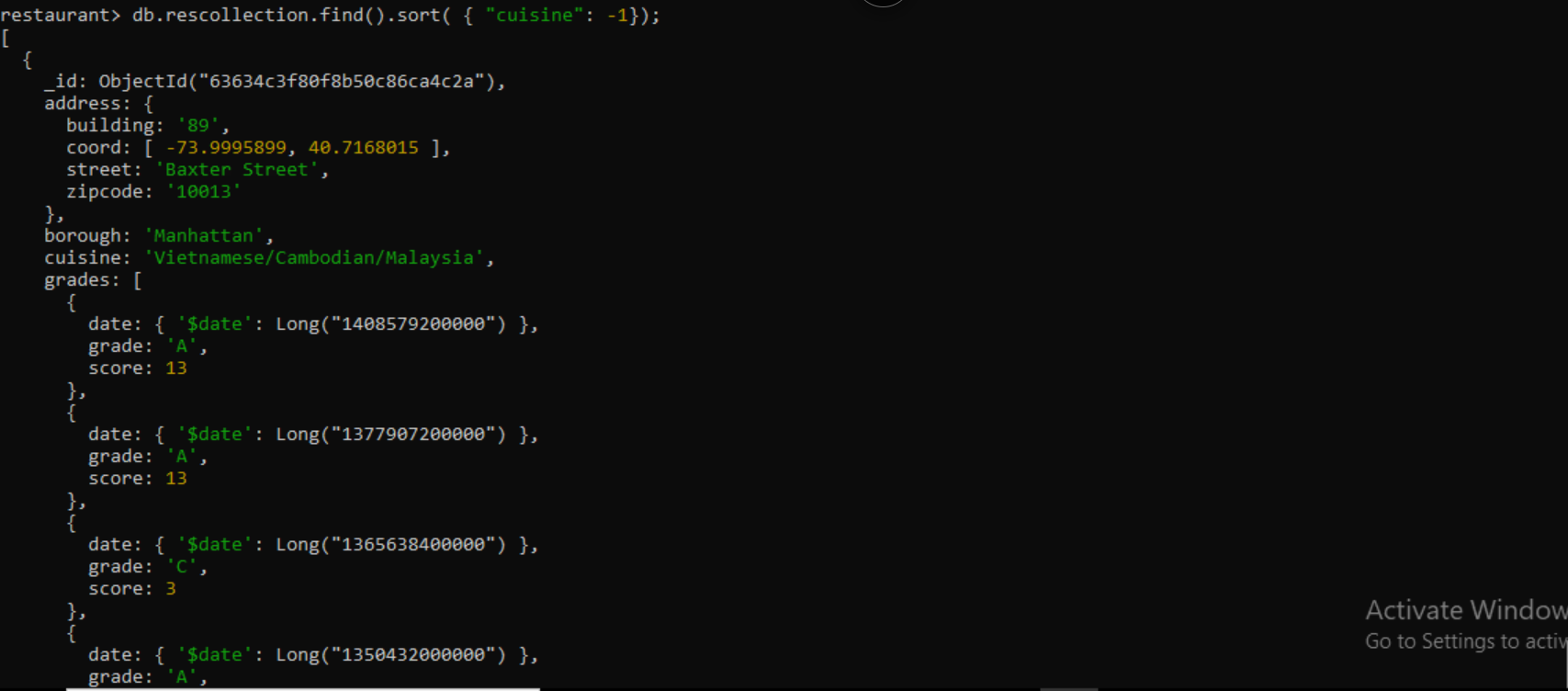
1. Write a MongoDB query to arrange the name of the cuisine in descending order.

Query- db.rescollection.find().sort(

{"cuisine":-1}

);

Output-



1. Show the restaurant Id, name, borough and cuisines for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Bil'.

Query- db.rescollection.find(

{$or: [

{name: /^Bil/},

{"$and": [

{"cuisine" : {$ne :"American "}},

{"cuisine" : {$ne :"Chinees"}}

]}

]}

,{"restaurant\_id" : 1,"name":1,"borough":1,"cuisine" :1}

);

Output-



1. Show the restaurant Id, name, borough and cuisines and score for restaurant's name begins with letter 'Bil'.

Query- db.rescollection.find(

{name: /^Bil/},

{

"restaurant\_id" : 1,

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

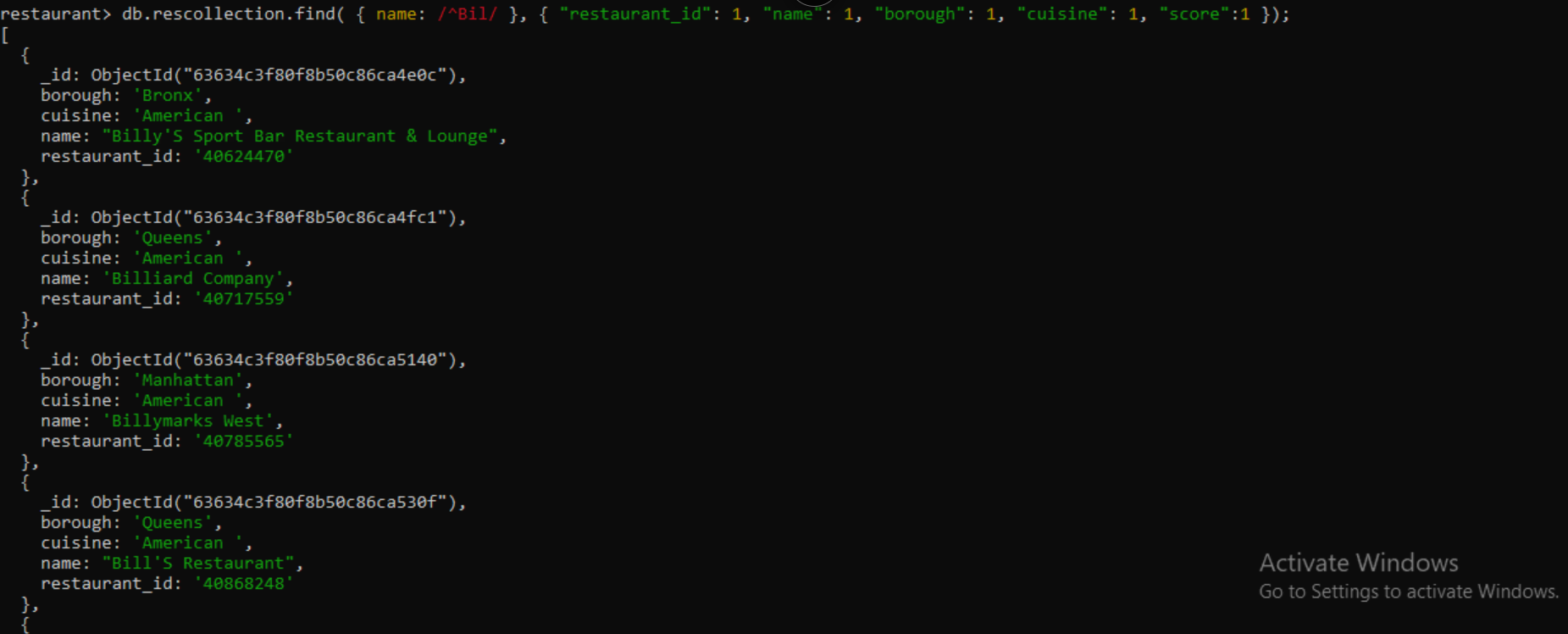
"cuisine" :1

“score”:1

}

);

Output-



1. Show the restaurant Id, name, borough and cuisines and score for restaurant serving “Indian” as cuisines.

Query-

Output-

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

Query- db.rescollection.find(

{name: /bi$/},

{

"restaurant\_id" : 1,

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

"cuisine" :1

“score”:1

}

);

Output-



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

Query- Query- db.rescollection.find(

{name: /il$/},

{

"restaurant\_id" : 1,

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

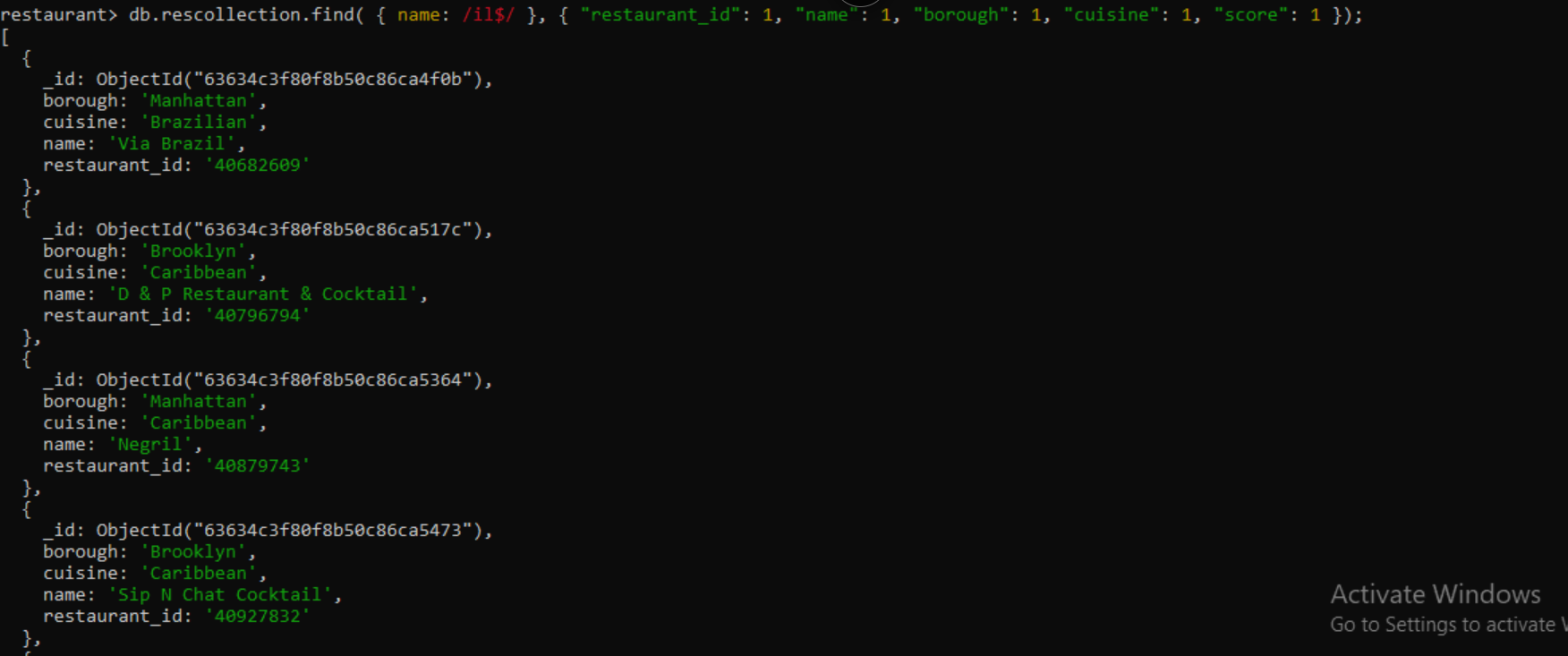
"cuisine" :1

“score”:1

}

);

Output-



1. Write a query to show all the restaurant Id, name, borough, cuisines, and score for those restaurants which contain 'il' anywhere in its name.

Query- db.rescollection.find(

{ name :

{ $regex : "il.\*"}

},

{

“restaurant\_id”:1

"name":1,

"borough":1,

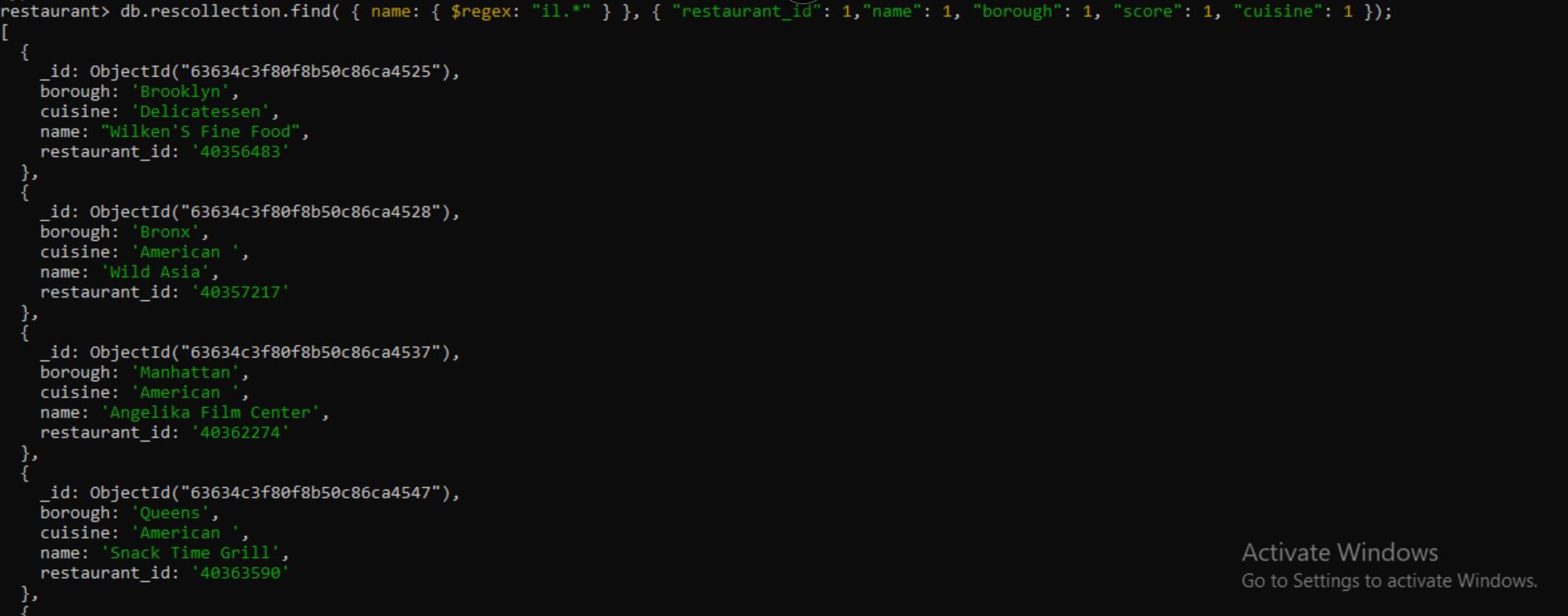
"score":1,

"cuisine" :1

}

);

Output-



1. 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.

Query- db.rescollection.find(

{"grades.score" :

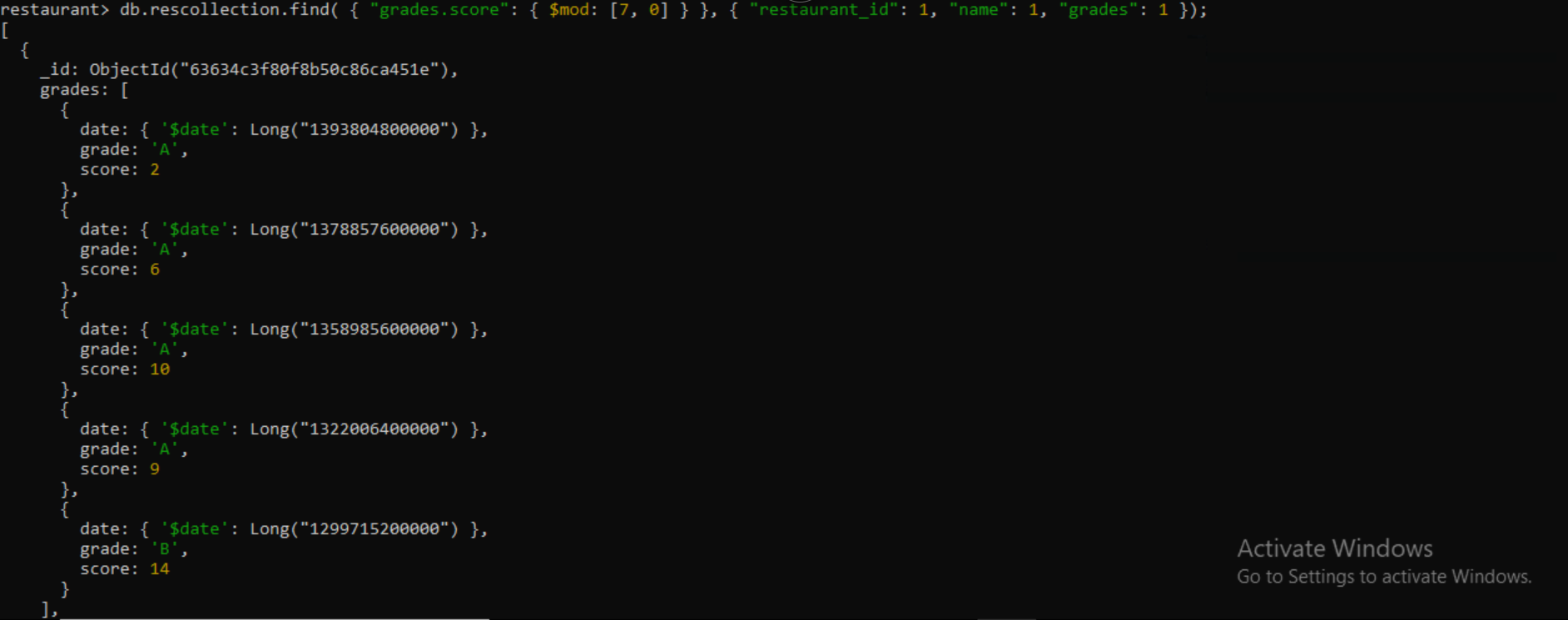
{$mod : [7,0]}

},

{"restaurant\_id" : 1,"name":1,"grades":1}

);

Output-



1. Show document/record counts that has street and not street in addresses.

Query- db.rescollection.find(

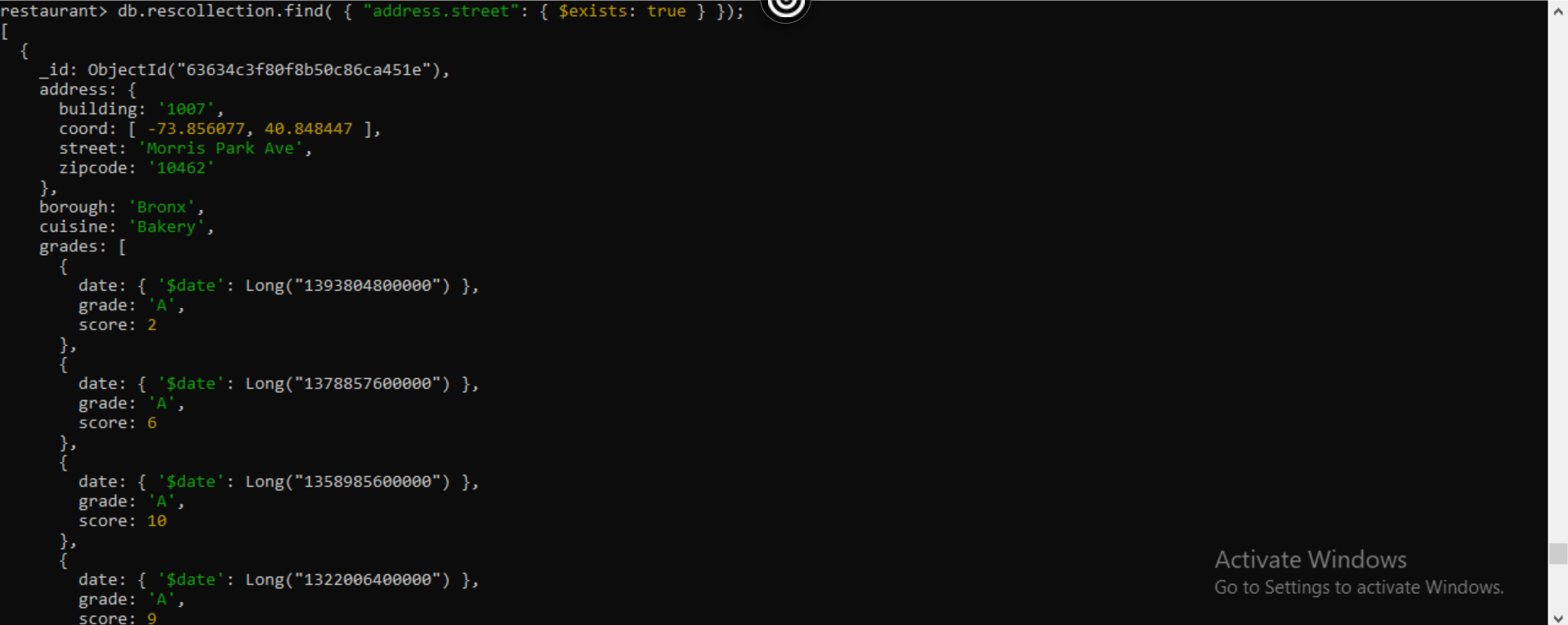
{"address.street" :

{ $exists : true }

}

);

Output-



1. 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

Query- db.rescollection.find(

{

"cuisine" : {$ne : "American "},

"grades.score" :{$gt: 70},

"address.coord" : {$lt : -65.754168}

}

);

Output-

