**Practical No - 04**

**AIM: Importing The JSON Data To The Mongodb.**

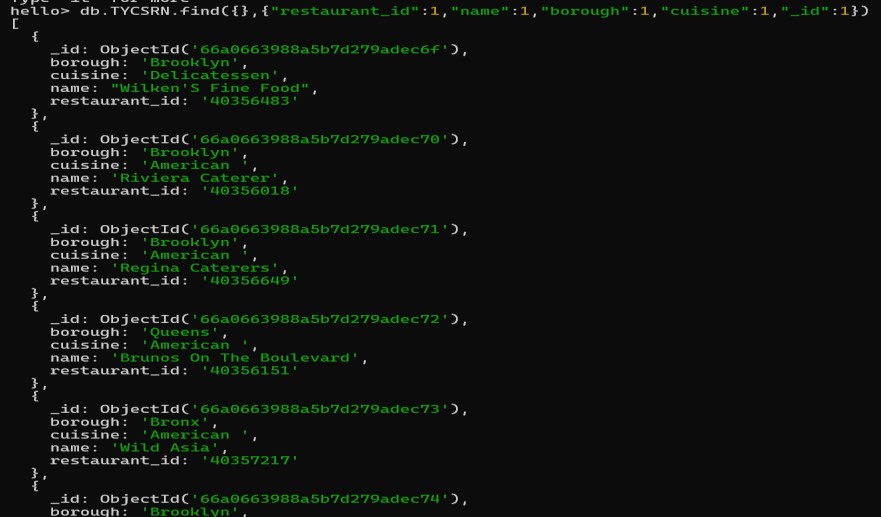
**1)** Write a MongoDB query to display all the documents in the collection TYCSRN

**Code :** db.TYCSRN.find().pretty()

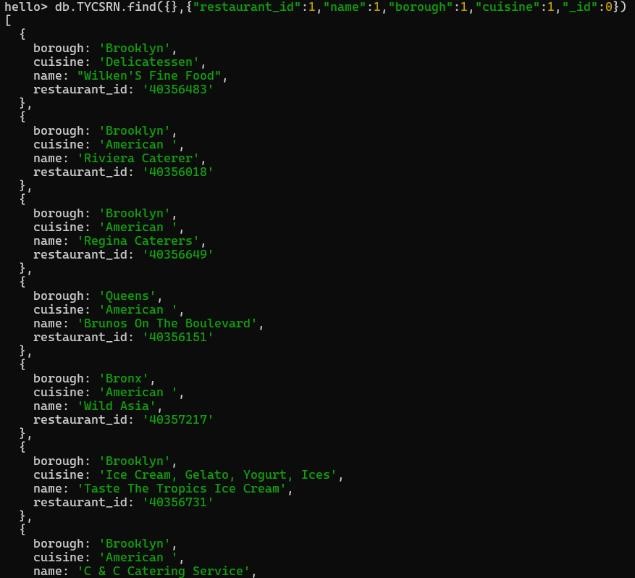


**2)** Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection TYCSRN.

# Code : db.TYCSRN.find({},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()



**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 TYCSRN.

**Code :** db.TYCSRN.find({},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1,"\_id":0}).pretty()

**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 TYCSRN.

**Code :** db.TYCSRN.find({},{"restaurant\_id":1,"name":1,"borough":1,"address.zipcode":1}).pretty()



**5)** Write a MongoDB query to display all the documents which is in the borough is Bronx.

**Code :** db.TYCSRN.find({"borough":"Bronx"}).pretty()



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

**Code :** db.TYCSRN.find({"borough":"Bronx"}).limit(5).pretty()



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

**Code :** db.TYCSRN.find({"borough":"Bronx"}).skip(5).limit(5).pretty()



**8)** Write a MongoDB query to find the TYCSRN who achieved a score more than 90.

**Code :** db.TYCSRN.find({grades:{$elemMatch:{"score":{$gt:90}}}}).pretty()



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

**Code :**  db.TYCSRN.find({grades:{$elemMatch:{"score":{$gt:80,$lt:100}}}}).pretty()



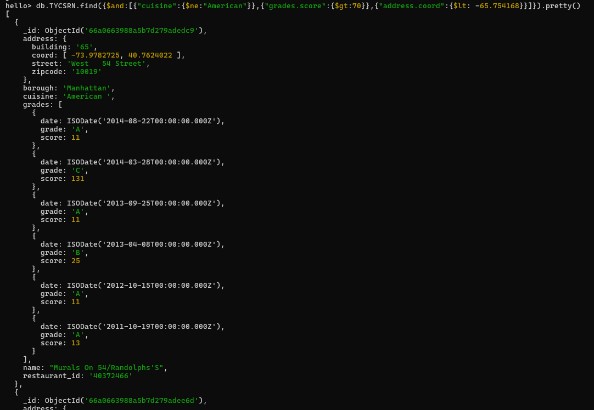
**10)** Write a MongoDB query to find the TYCSRN which locate in latitude value less than - 95.754168.

# Code : db.TYCSRN.find({"address.coord":{$lt: -95.754168}}).pretty()



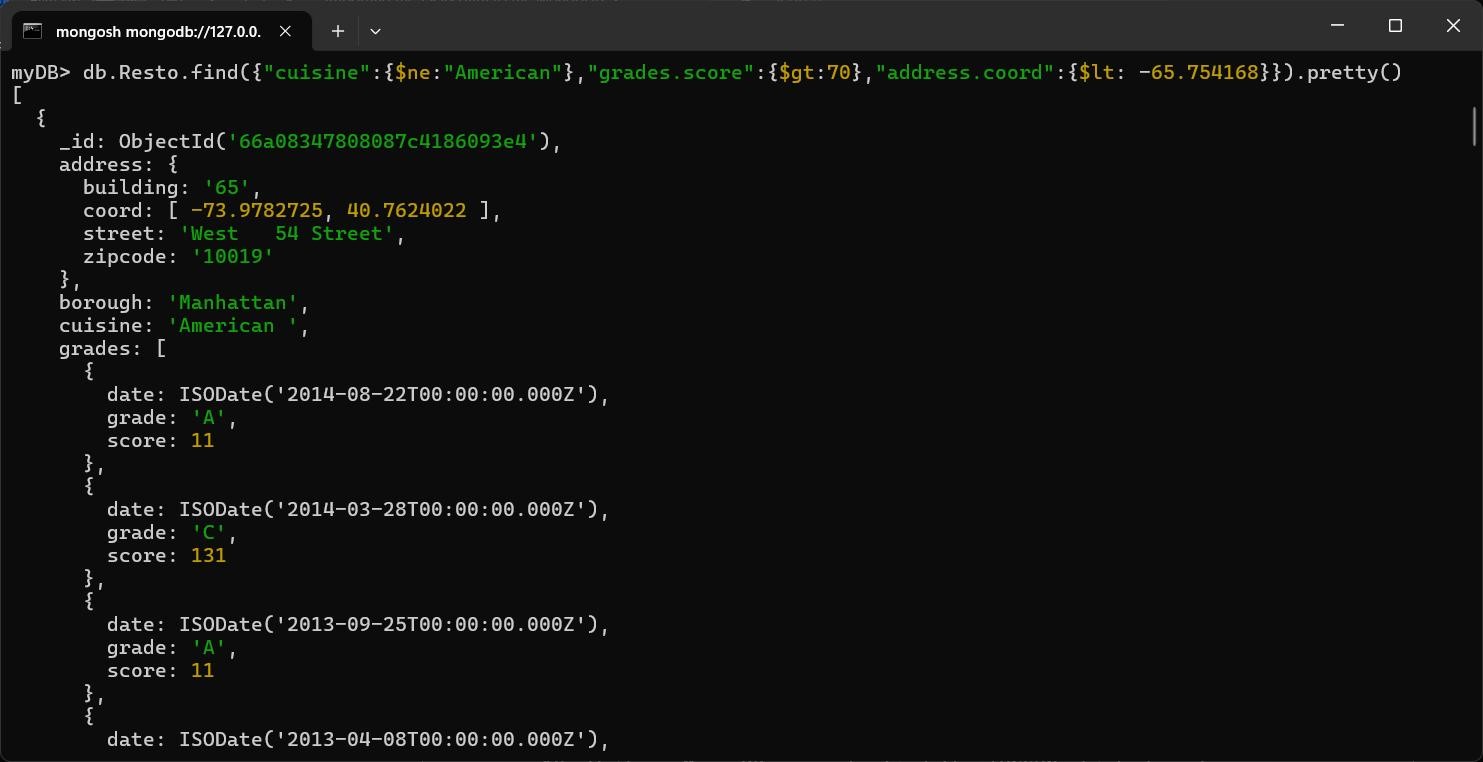
**11)** Write a MongoDB query to find the TYCSRN that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than 65.754168.

# Code : db.TYCSRN.find({$and:[{"cuisine":{$ne:"American"}},{"grades.score":{$gt:70}},{"address.coord ":{$lt: -65.754168}}]}).pretty()



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

**Code:**db.Resto.find({"cuisine":{$ne:"American"},"grades.score":{$gt:70},"address.coord":{$lt: 65.754168}}).pretty()



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

# Code : db.TYCSRN.find({"cuisine":{$ne:"American"},"grades.grade":"A","borough":"Brooklyn"}).sor t({"cuisine": -1}).pretty()

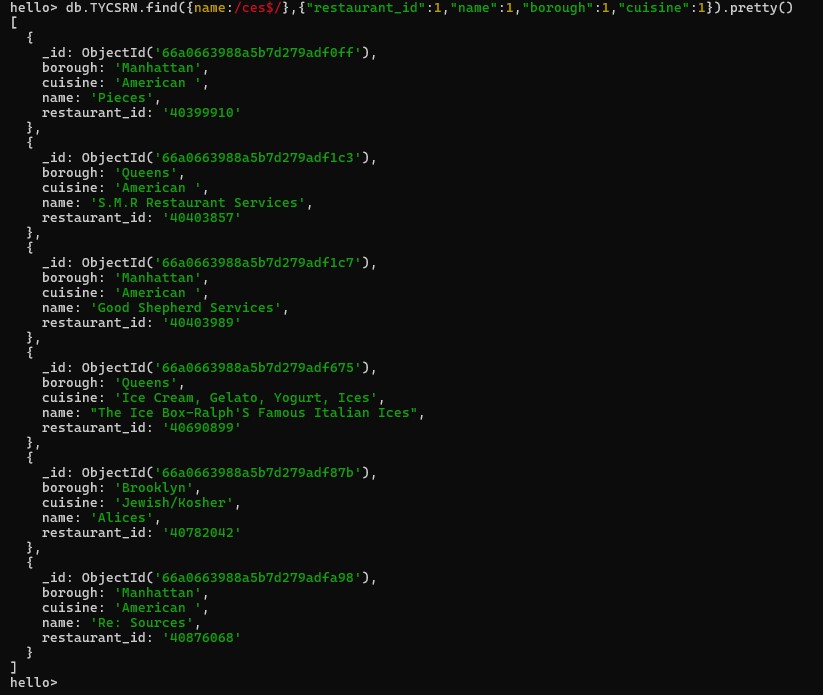


**14)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which contain 'Wil' as first three letters for its name.

**Code :** db.TYCSRN.find({name:/^Wil/},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()

**15)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which contain 'ces' as last three letters for its name.

# Code : db.TYCSRN.find({name:/ces$/},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()



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

# Code : db.TYCSRN.find({"name":/.\*Reg.\*/},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pre tty()

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

# Code : db.TYCSRN.find({"borough":"Bronx",$or:[{"cuisine":"American"},{"cuisine":"Chinese"}]}).pr etty()



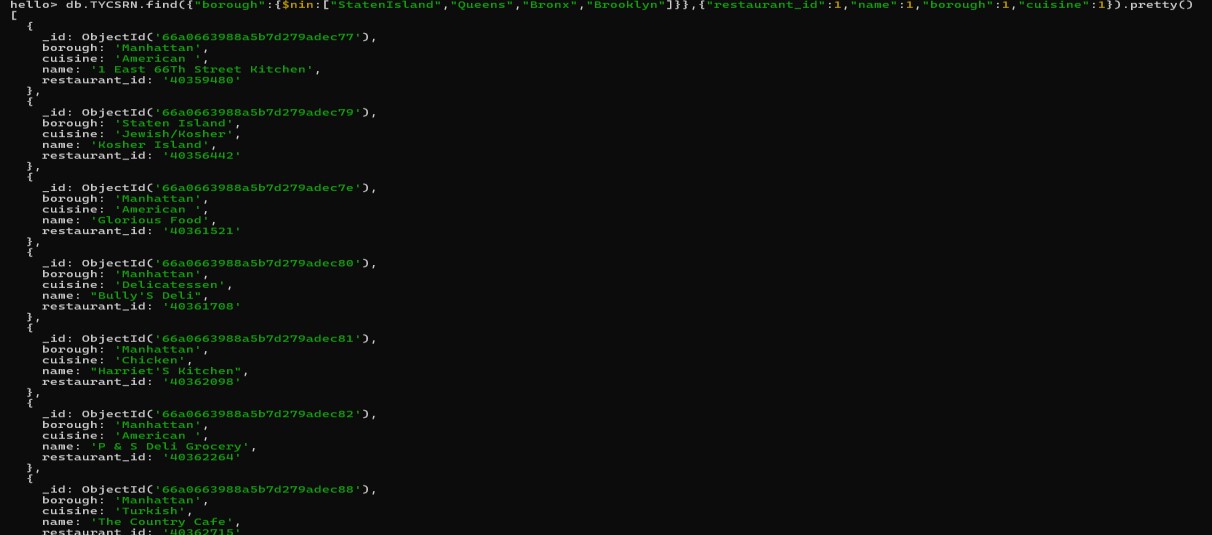
**18)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

# Code : db.TYCSRN.find({"borough":{$in:["StatenIsland","Queens","Bronx","Brooklyn"]}},{"restauran t\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()



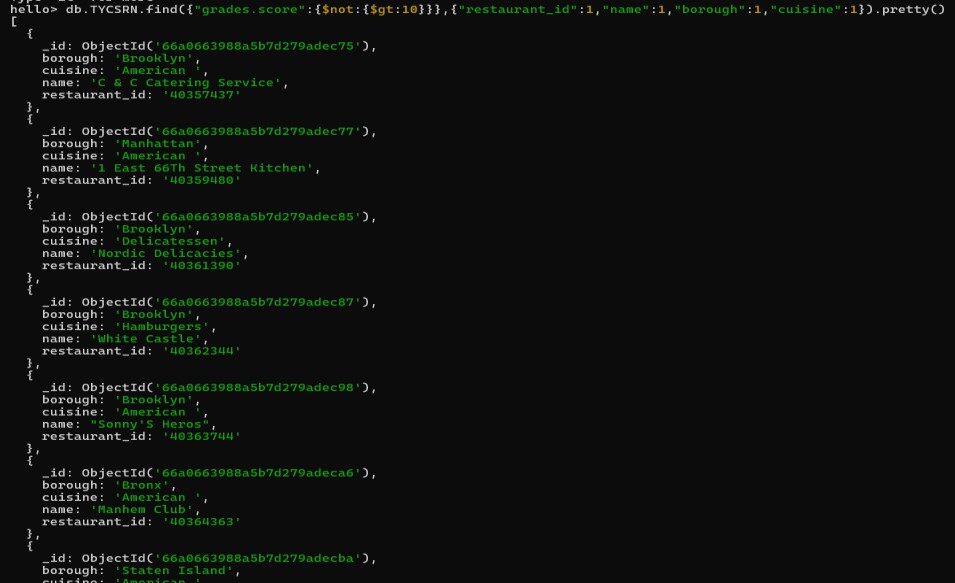
**19)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

# Code : db.TYCSRN.find({"borough":{$nin:["StatenIsland","Queens","Bronx","Brooklyn"]}},{"restaura nt\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()



**20)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which achieved a score which is not more than 10.

# Code : db.TYCSRN.find({"grades.score":{$not:{$gt:10}}},{"restaurant\_id":1,"name":1,"borough":1,"cu isine":1}).pretty()



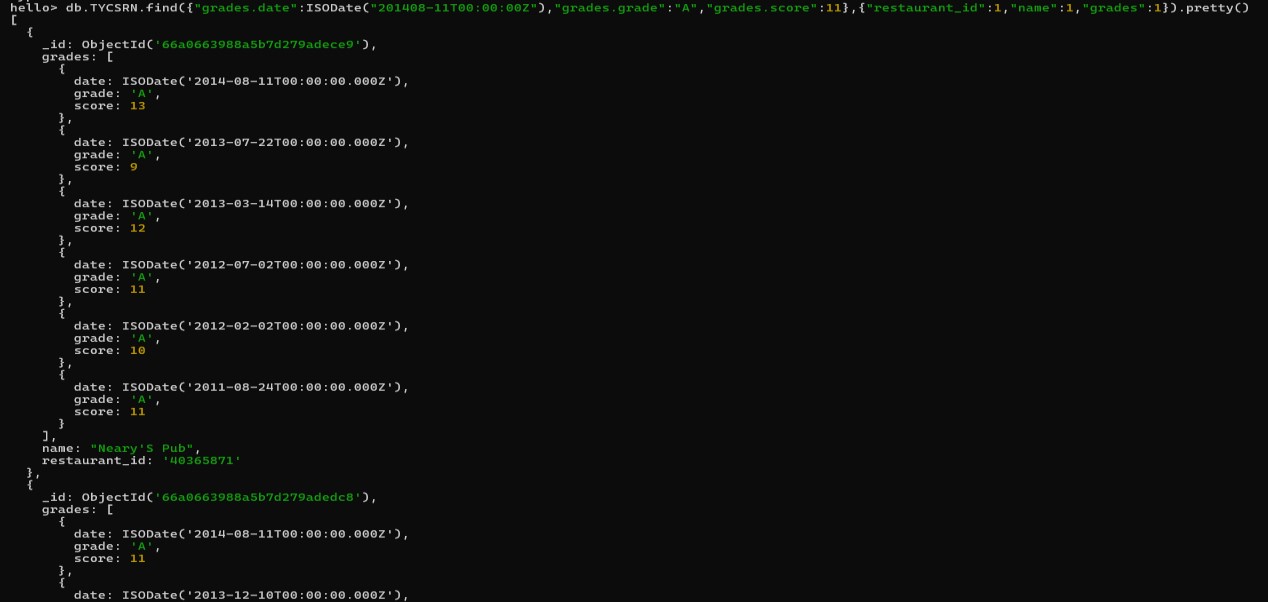
**21)** Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those TYCSRN which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

# Code : db.TYCSRN.find({$or:[{name:/^Wil/},{"$and":[{"cuisine":{$ne:"American"}},{"cuisine":{$ne:" Chinees"}}]}]},{"restaurant\_id":1,"name":1,"borough":1,"cuisine":1}).pretty()



**22)** Write a MongoDB query to find the restaurant Id, name, and grades for those TYCSRN which achieved a grade of "A" and scored 11 on an ISODate "201408-11T00:00:00Z" among many of survey dates.

# Code:db.TYCSRN.find({"grades.date":ISODate("20140811T00:00:00Z"),"grades.grade":"A","grades.score":11},{"restaurant\_id":1,"name":1,"grades":1}). pretty()



**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".

# Code:db.TYCSRN.find({"grades.1.date":ISODate("20140811T00:00:00Z"),"grades.1.grade":"A","grades.1.score":9},{"restaurant\_id":1,"name":1,"grades":1} ).pretty()

**24)** Write a MongoDB query to find the restaurant Id, name, address and geographical location for those TYCSRN where 2nd element of coord array contains a value which is more than 42 and upto 52.

# Code : db.TYCSRN.find({"address.coord.1":{$gt:42,$lte:52}},{"restaurant\_id":1,"name":1,"address":1, "coord":1}).pretty()



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

# Code : db.TYCSRN.find().sort({"name":1}).pretty()



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

# Code ; db.TYCSRN.find().sort({"name": -1}).pretty()



**27)** Write a MongoDB query to arranged the name of the cuisine in ascending order and borough should be in descending order.

# Code : db.TYCSRN.find().sort({"cuisine": 1,"borough": -1}).pretty()



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

# Code : db.TYCSRN.find({"address.street":{$exists:true}}).pretty()



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

# Code : db.TYCSRN.find({"address.coord":{$type:1}}).pretty()



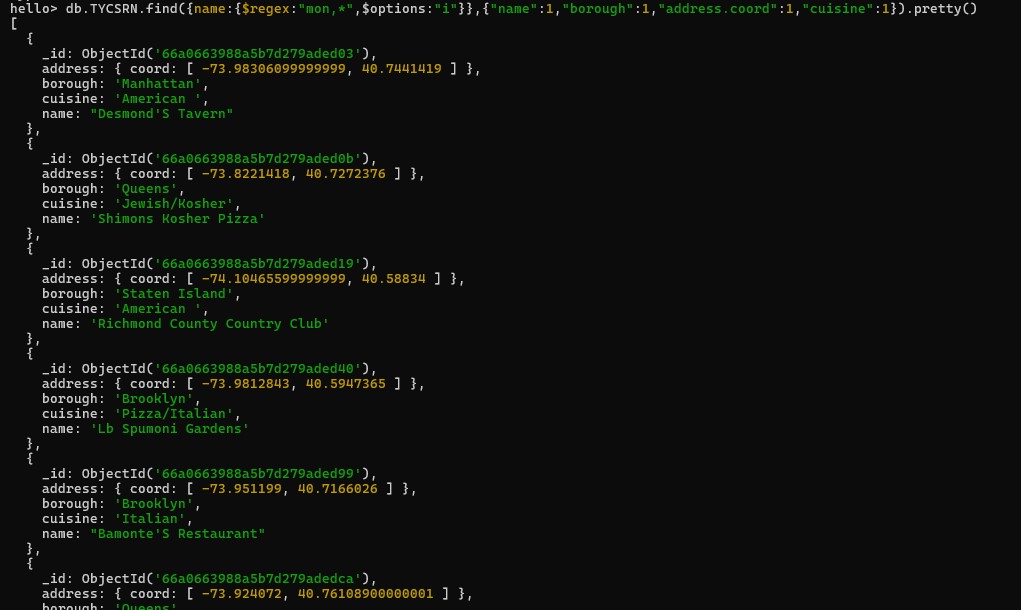
**30)** Write a MongoDB query which will select the restaurant Id, name and grades for those TYCSRN which returns 0 as a remainder after dividing the score by 7.

# Code : db.TYCSRN.find({"grades.score":{$mod:[7,0]}},{"restaurant\_id":1,"name":1,"grades":1}).pretty ()



**31)** Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those TYCSRN which contains 'mon' as three letters somewhere in its name.

# Code : db.TYCSRN.find({name:{$regex:"mon,\*",$options:"i"}},{"name":1,"borough":1,"address.coord":1, "cuisine":1}).pretty()



**32)** Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those TYCSRN which contain 'Mad' as first three letters of its name.

**Code :**db.TYCSRN.find({name:{$regex:/^Mad/i}},{"name":1,"borough":1,"address.coord":1,"cuisine":1}).

# pretty()

