### AIM: Performing queries based on AND, OR, Limit, Sort and Projection and apply some queries to get specified output.

1. **AND , OR, SORT, LIMIT, SKIP, count, projections SYNTAX**
   * $and

*Syntax*: { $and: [ { <expression1> }, { <expression2> } , ... , { <expressionN> } ]

}

### db.inventory.find( { $and: [ { price: { $ne: 1.99 } }, { price: 10} ] } )

* + $or

{ $or: [ { <expression1> }, { <expression2> }, ... , { <expressionN> } ] }

### db.inventory.find( { $or: [ { quantity: { $lt: 20 } }, { price: 10 } ] } )

* + skip

If we want to fetch two documents after the first two documents from the collection 'userdetails', the following mongodb command can be used :

>db.userdetails.find().skip(2).pretty();

* + limit()

To limit the records in MongoDB, you need to use **limit()** method, which is the number of documents that you want to be displayed.

>db.COLLECTION\_NAME.find().limit(NUMBER)

* + count

The db.collection.count() method is used to return the count of documents that would match a find() query. Also counts number of records.

count();

restaurants. restaurants.

db.

db.

find({"cuisine" : "American "}).count()

* + projection

Consider the collection mycol has the following data −

{ "\_id" : ObjectId(5983548781331adf45ec5), "title":"MongoDB Overview"}

{ "\_id" : ObjectId(5983548781331adf45ec6), "title":"NoSQL Overview"}

{ "\_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}

Following example will display the title of the document while querying the document.

>db.mycol.find({},{"title":1,\_ id:0})

{"title":"MongoDB Overview"}

{"title":"NoSQL Overview"}

{"title":"Tutorials Point Overview"}

>

Please note \_id field is always displayed while executing find() method, if you don't want this field, then you need to set it as 0.

# PRACTICE QUESTIONS:

**Create collection name as “restaurants”. Also insert 10 more documents in following format which satisfy all requirement of query.**

## {

"address": { "building": "1007",

"coord": [ -73.856077, 40.848447 ],

"street": "Morris Park Ave", "zipcode": "10462"

},

"borough": "Bronx",

"cuisine": "Bakery", "grades": [

{"grade": "A", "score": 2 },

{"grade": "A", "score": 6 },

{"grade": "A", "score": 10 },

{ "grade": "A", "score": 9 },

{ "grade": "B", "score": 14 }

],

"name": "Morris Park Bake Shop", "restaurant\_id": "30075445"

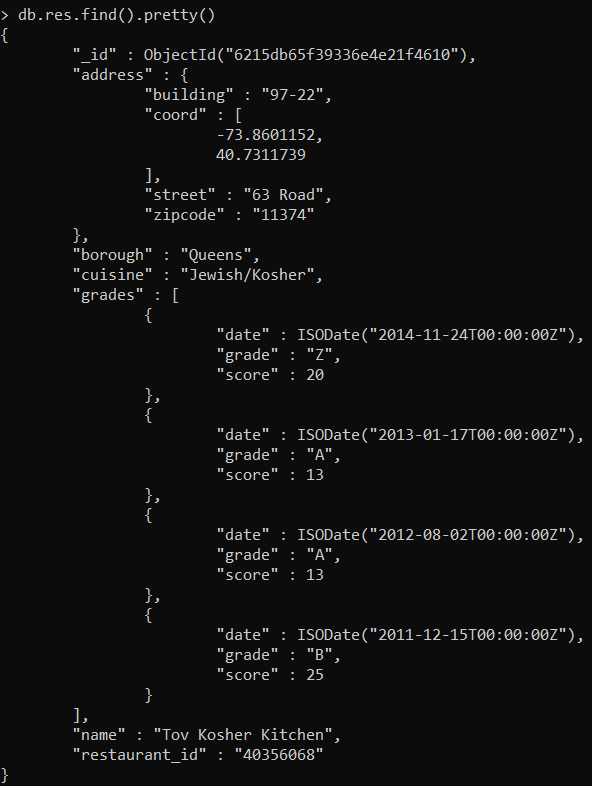
}

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

**Input code:-**

db.res.find().pretty()

**Output:-**

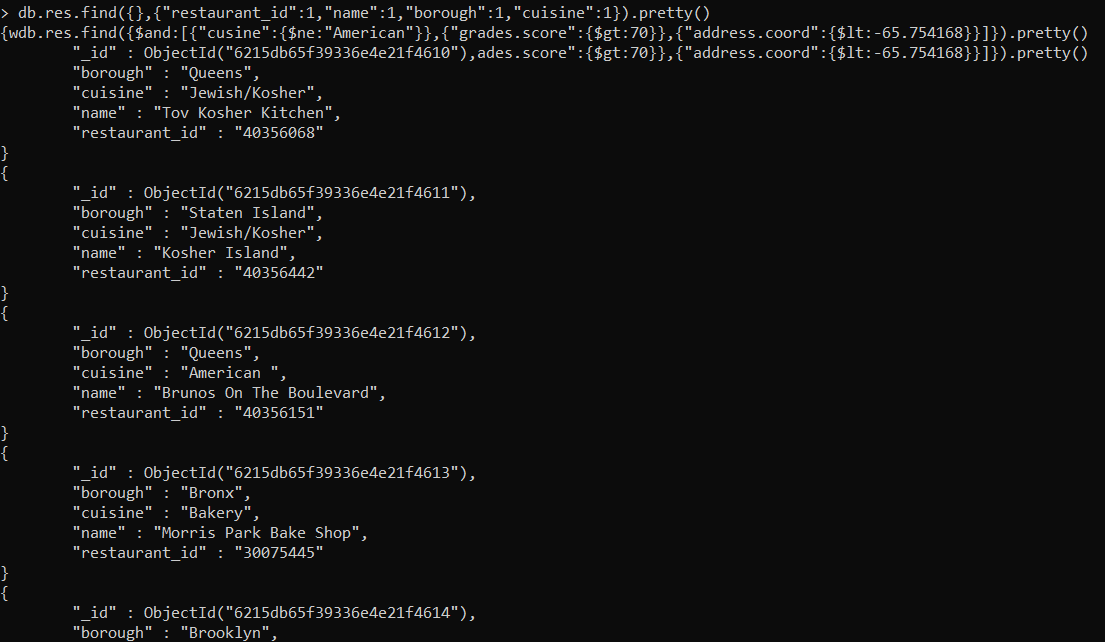
****

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

**Input code:-**

db.res.find({},{“restaurant\_id”:1,”name”:1,”borough”:1,”cusine”:1}).pretty()

**Output:-**

****

1. 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. (USING PROJECTION)

**Input code:-**

db.res.find({},{“restaurant\_id”:1,”name”:1,”borough”:1,”address.zipcode”:1,”\_id”:0}).pretty()

**Output:-**

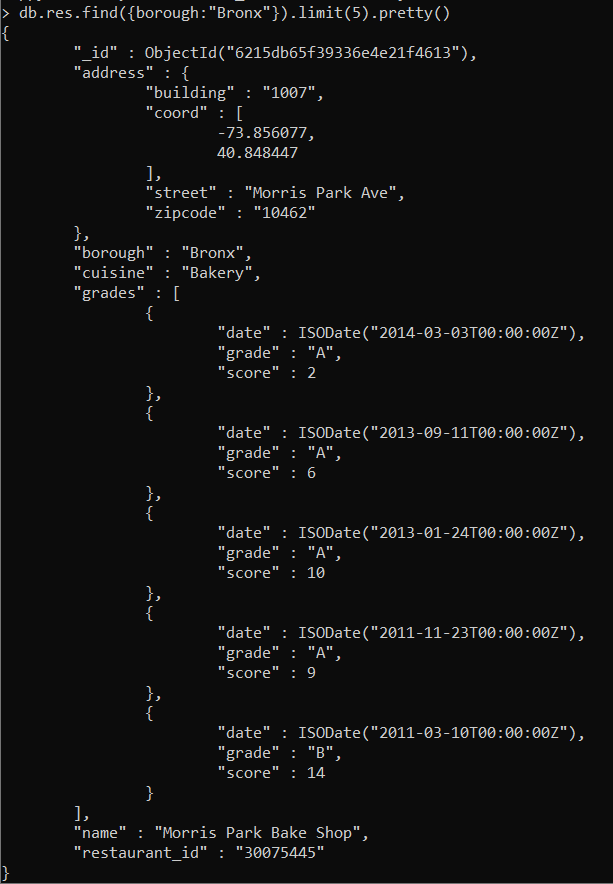
****

1. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx. (USING LIMIT)

**Input code:-**

db.res.find({borough:”Bronx”}).limit(5).pretty()

**Output:-**

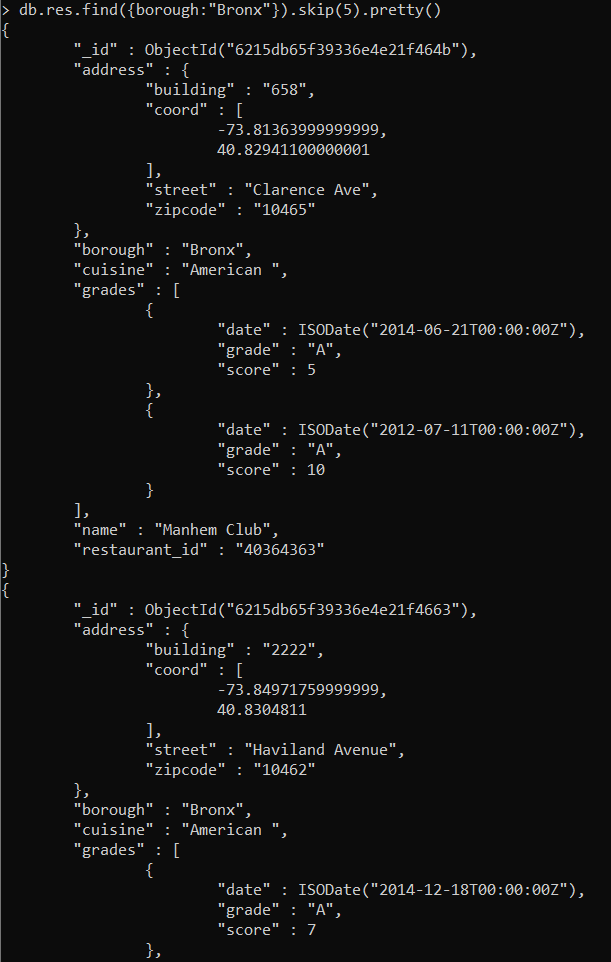
****

1. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx. (USING SKIP)

**Input code:-**

db.res.find({borough:”Bronx”}).skip(5).pretty()

**Output:-**

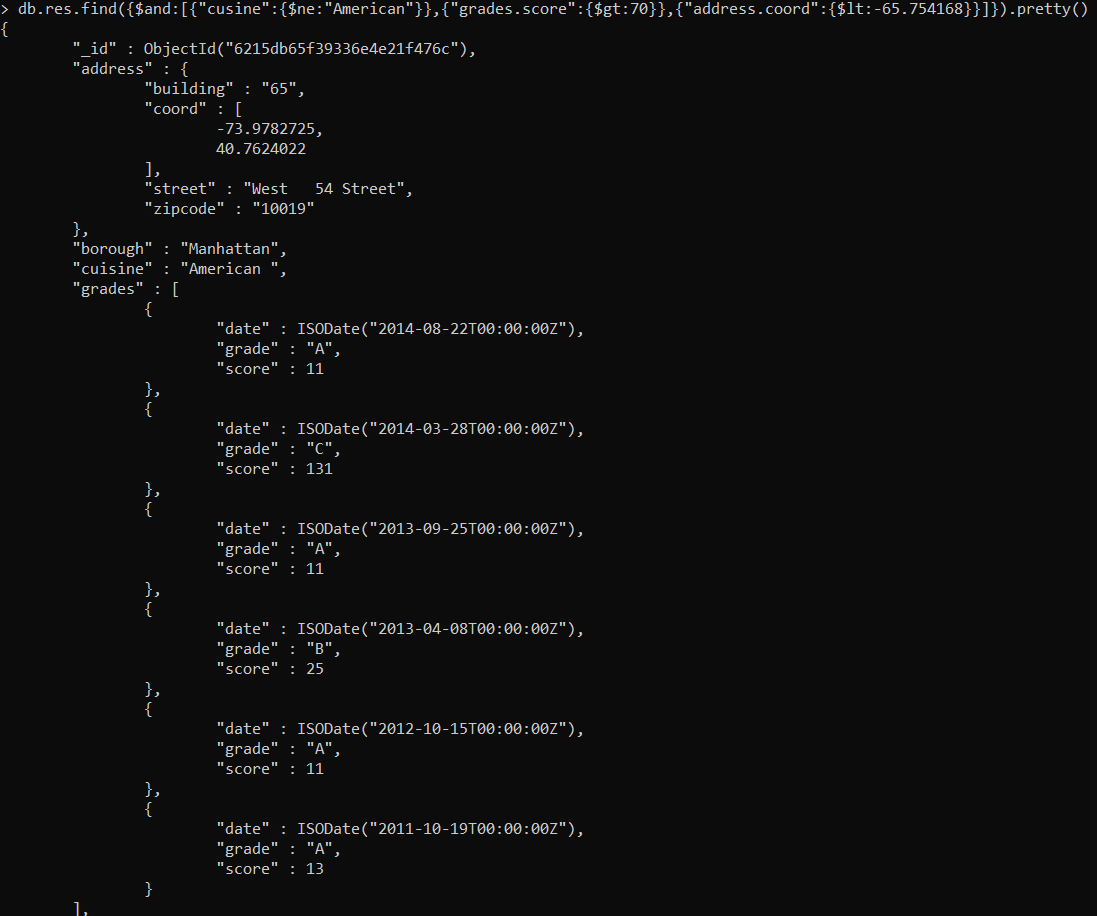


1. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168. (USING AND)

**Input code:-**

db.res.find({$and:[{“cusine”:{$ne:”American”}},{“grades.score”:{$gt:70}},{“address.coord”:{$lt:-65.754168}}]}).pretty()

**Output:-**

****

1. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish. (USING OR)

**Input code:-**

Db.res.find({borough:”Bronox”,$or:[{“cuisine”:”Americaan”},{“cusine”:”Chinese”}]}).pretty()

**Output:-**

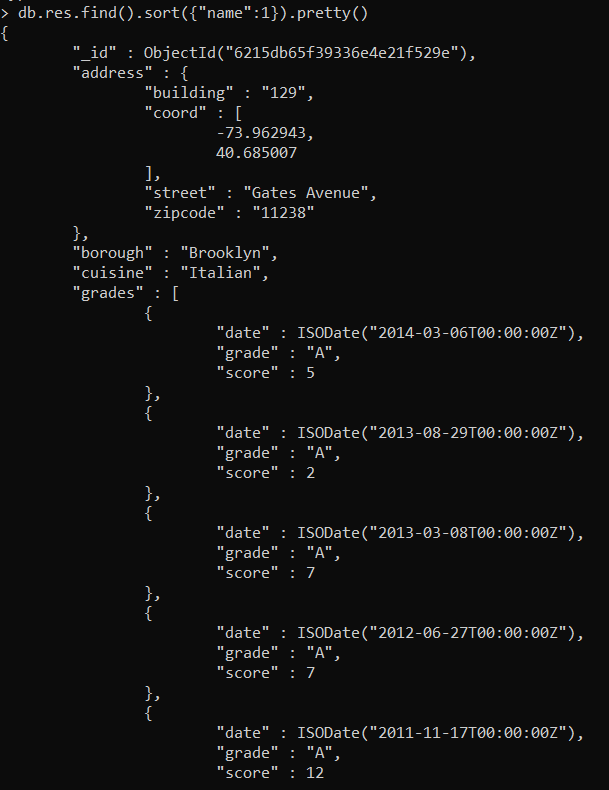


1. Write a MongoDB query to arrange the name of the restaurants in ascending / descending order along with all the columns. (USING SORT)

**Input code:-**

db.res.find().sort({“name”:1}).pretty()

**Output:-**

****