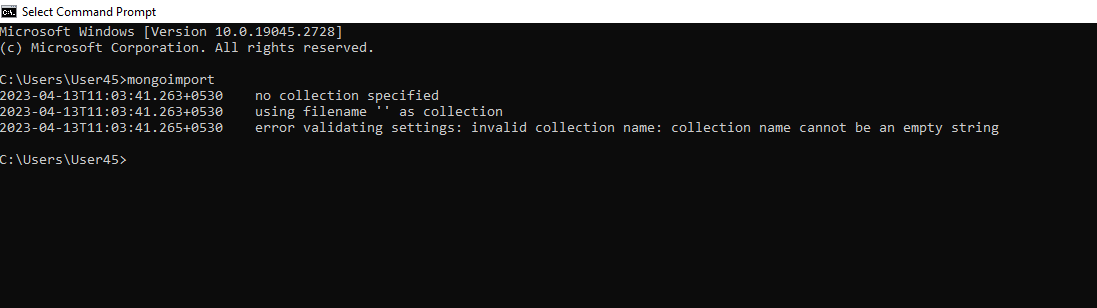
**Sakshi Chauhan MongoDBimport Day-2 13-April-2023**

Firstly, Download the **MongoDB Command Line Database Tools** and setup path in Environment Variable and execute in the command Prompt.



1. **Import vendorsJson file in database with vendors collection**

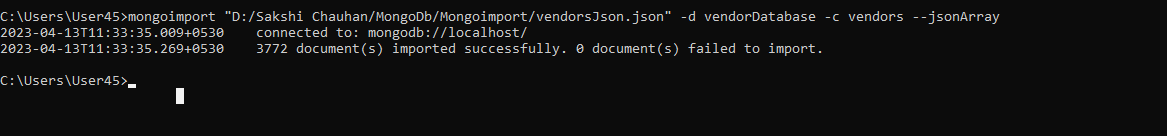
* mongoimport

"D:/Sakshi Chauhan/MongoDb/Mongoimport/vendorsJson.json"

-d vendorDatabase

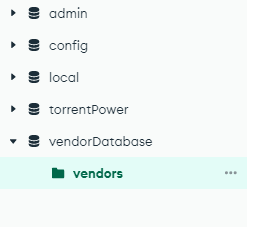
-c vendors

--jsonArray



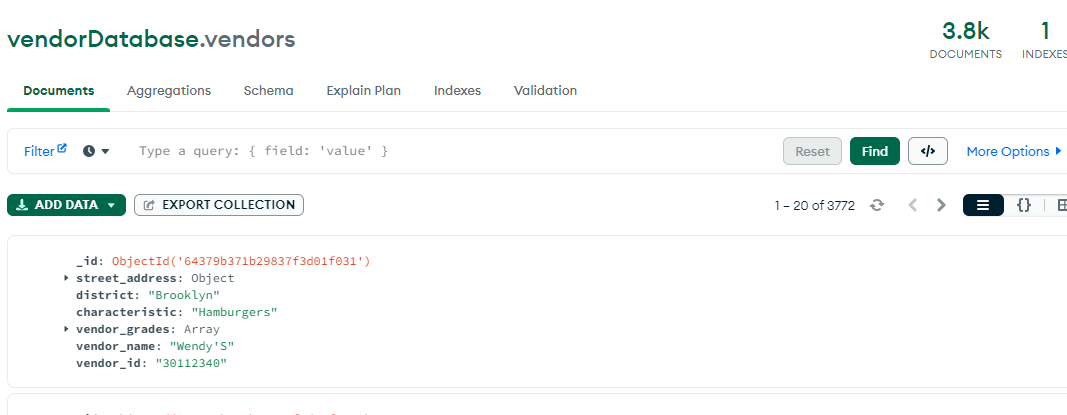
Here is my imported Database:-

**databaseName:- vendorDatabase collectionName:- vendors**

****

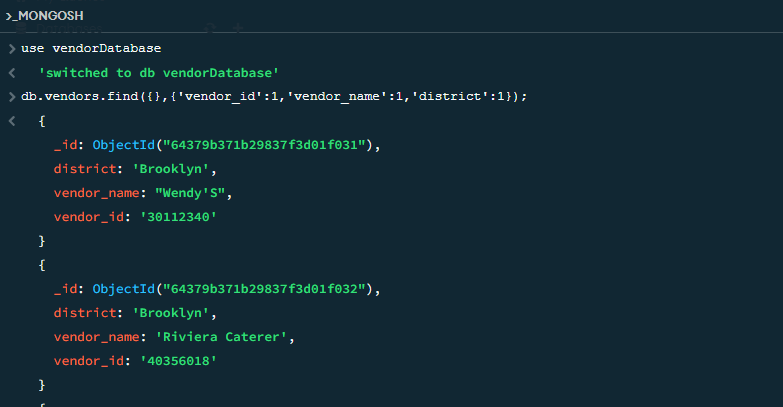
1. **Display all the documents.**

* **db.vendors.find();**

****

**3. display the fields vendor\_id, vendor\_name, district for all the documents in the collection**.

* db.vendors.find({},{'vendor\_id':1,'vendor\_name':1,'district':1});



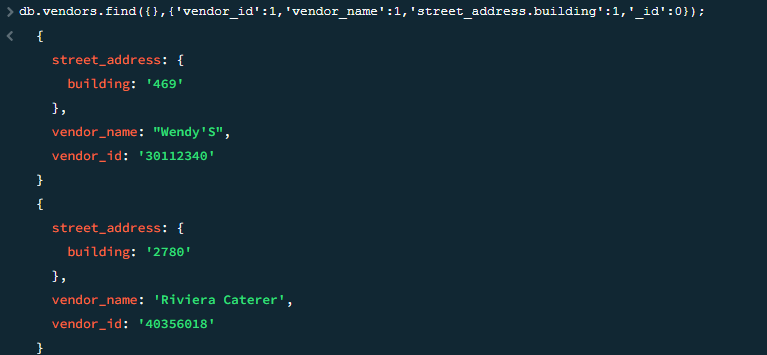
* db.vendors.find({},{'vendor\_id':0,'vendor\_name':0,'district':0});



**4. Display the fields vendor\_id, vendor\_name, street\_address. Building but removing the field \_id for all the documents in output.**

* db.vendors.find({},{'vendor\_id':1,

'vendor\_name':1, 'street\_address.building':1,'\_id':0});

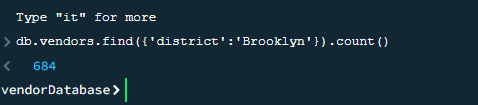


**5. display all the vendors which is in the district ‘Brooklyn’**

* db.vendors.find({'district':'Brooklyn'});



* Total count:-

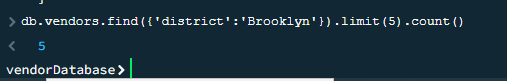


**6. display the first 5 vendors which is in the district ‘Brooklyn’**

* db.vendors.find({'district':'Brooklyn'}).limit(5)

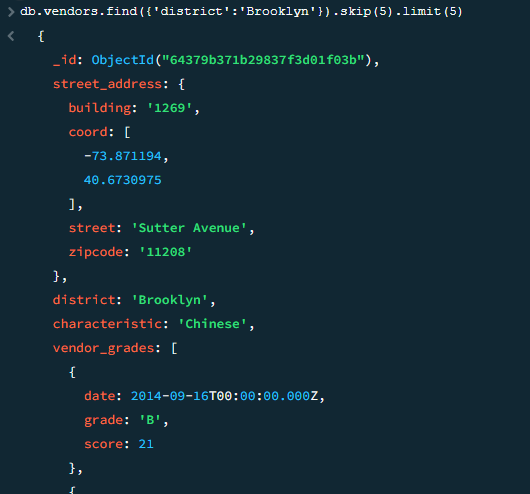


* Total count:-



**7. display the next 5 vendors after skipping first 5 which is in the district ‘Brooklyn’**

* db.vendors.find({'district':'Brooklyn'}).skip(5).limit(5)

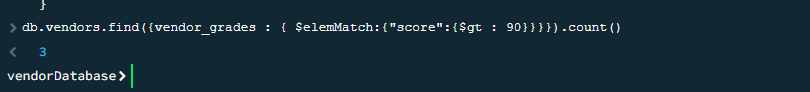
****

**8. find the vendors who achieved a score more than 90 in vendor\_grades**

* db.vendors.find({vendor\_grades : { $elemMatch:{"score":{$gt : 90}}}});
* db.vendors.find({"vendor\_grades.score" : {$gt : 90}});

****

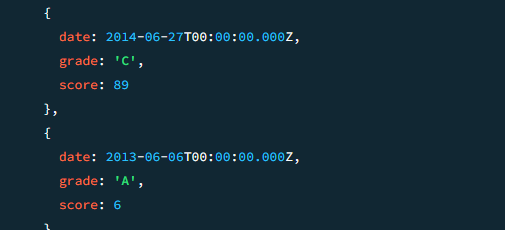
* Total count:-



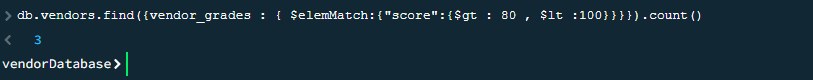
**9. find the vendors that achieved a score, more than 80 but less than 100.**

* db.vendors.find({vendor\_grades :{ $elemMatch:{"score":{$gt : 80 , $lt :100}}}});

****

****

* Total count:-

****

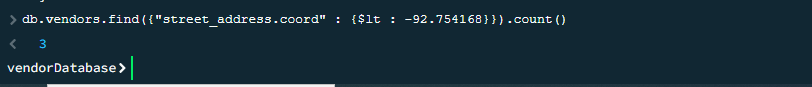
**10. find the vendors which are located in latitude(coord) value less than -92.754168.**

* db.vendors.find({"street\_address.coord" : {$lt : -92.754168}});





* Total count:-



**11. find the vendors that characteristic is not ‘Italian' and their grade score more than 70 and latitude less than -65.754168**

* db.vendors.find({

"characteristics" : {$ne : "Italian "},

"vendor\_grades.score" :{$gt: 70},

"street\_address.coord" : {$lt : -65.754168}

}); **OR**

* db.vendors.find({

"characteristics" : { $not: { $eq: "Italian" } },

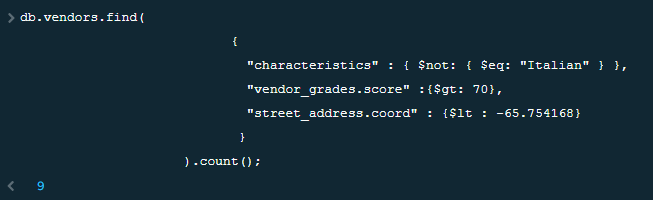
"vendor\_grades.score" :{$gt: 70},

"street\_address.coord" : {$lt : -65.754168}

});



* Total count:-



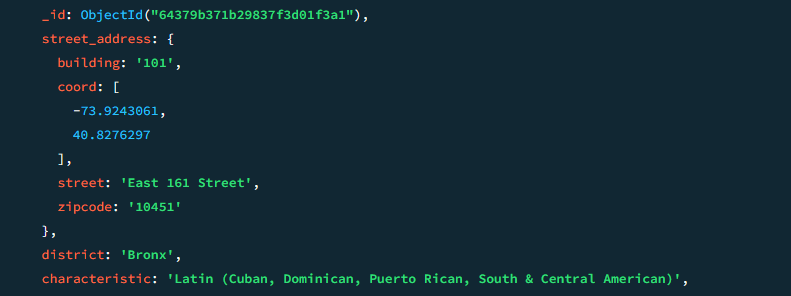
**12. find the vendors that characteristic is not ‘Italian' and their grade score more than 70 and latitude less than -65.754168.The document must be displayed according to the characteristic in descending order.**

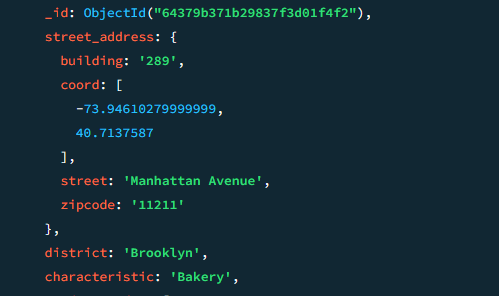
* db.vendors.find({"characteristic": { $ne: "Italian" },

"vendor\_grades.score": { $gt: 70 },

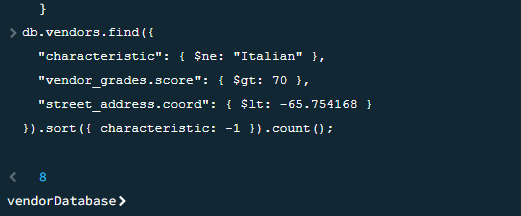
"street\_address.coord": { $lt: -65.754168 }}).sort({ characteristic: -1 })







* Total count:-



**13. find the vendor\_id, vendor\_name, for those vendors which contain 'Gio' as first three letters for its name.**

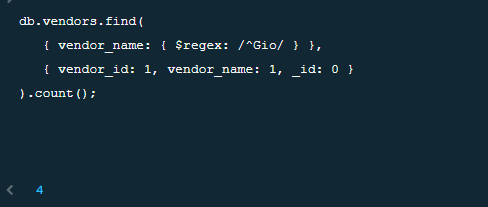
* db.vendors.find(

{vendor\_name:{$regex:/^Gio/} },

{ vendor\_id: 1, vendor\_name: 1, \_id: 0 })

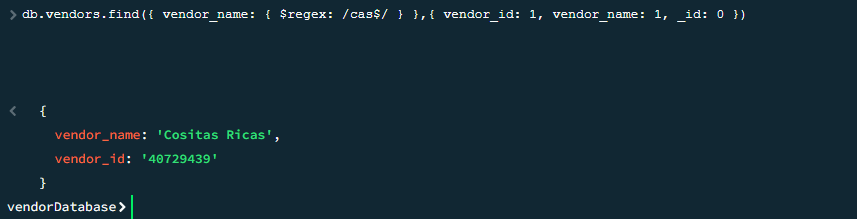


* Total count:-



**14. find the vendor\_id, vendor\_name, for those vendors which contain 'cas' as last three letters for its name.**

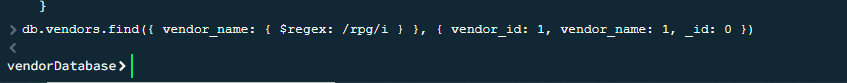
* db.vendors.find({ vendor\_name: { $regex: /cas$/ } },{ vendor\_id: 1, vendor\_name: 1, \_id: 0 })



**15. find the vendor\_id, vendor\_name, for those vendors which contain 'Rpg' as three letters somewhere in its name**

* db.vendors.find({vendor\_name:{$regex:/rpg/i}},

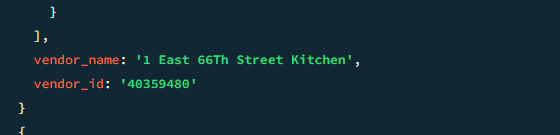
{vendor\_id: 1,vendor\_name: 1, \_id: 0 })

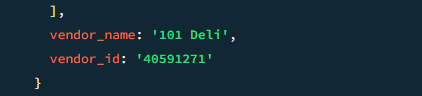
****

* 0 records in this Query.

**16. arrange the name of the vendors in ascending order along with all the columns.**

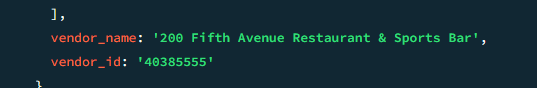
* db.vendors.find().sort({"vendor\_name":1});



****

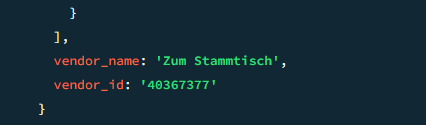
****

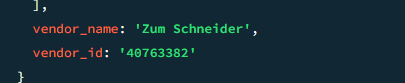
****

****

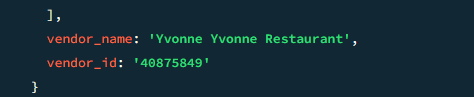
**17. arrange the name of the vendors in descending order along with all the columns.**

* db.vendors.find().sort({"vendor\_name":-1});









**18. query to know whether all the street\_address contains the street or not.**

* db.vendors.find({"street\_address.street" : { $exists : true } } );

This Query returns to if in the street\_address (street or not), $exists: true then

Check if street\_address street is there or not.

**19. query which will select all documents in the vendors collection where the ‘coord’ field value is Double.**

* db.vendors.find({"street\_address.coord" : {$type : 1}});

This query will select all documents in the vendors collection where the ‘coord’ field value is Double.

**20. find the vendors that have at least one grade with a score of less than 5 and that are located in the district of Manhattan or Brooklyn, and their characteristics are not Italian.**

* db.vendors.find({

$and: [

{ "vendor\_grades.score": { $lt: 5 } },

{ $or: [{"district": "Manhattan" }, {"district": "Brooklyn" }] },

{ "characteristic": { $ne: "Italian" } }

]

})

