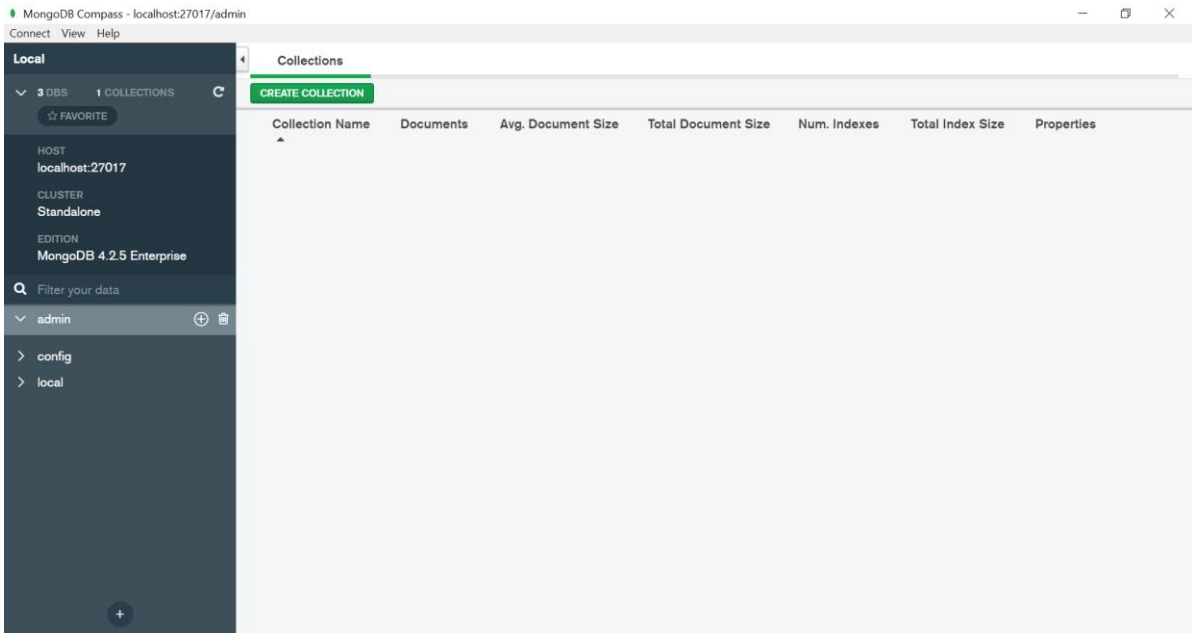
MongoDB Assignment by Priyanka Labh

Lab 1:

show dbs	Display all available database
use analisisProveedores	analisisProveedores is a database which is been in used
Aggregate	Performs aggregation tasks such as group using the aggregation framework
Count	Counts the number of documents in a collection or a view.
Distinct	Displays the distinct values found for a specified key in a collection or a view.
Group	Groups input documents by the specified _id expression and for each distinct grouping, outputs a document.
mapReduce	Performs map-reduce aggregation for large data sets.
Find	Selects documents in a collection or a view.
Insert	Inserts one or more documents.
Update	Updates one or more documents.
Delete	Deletes one or more documents.
findAndModify	Returns and modifies a single document.
Logout	Terminates the current authenticated session.
Authenticate	Starts an authenticated session using a username and password.
createUser	Creates a new user.
dropUser	Removes a single user.
grantRolesToUser	Grants a role and its privileges to a user.
usersInfo	Returns information about the specified users.
renameCollection	Rename Collection name
Copydb	Create copy of database.
dropDatabase	Removes the current database.
listCollectios	Display all available collection
drop	Removes the specified collection from the database.
create	Creates a collection or a view.
clone	Create clone
createIndexes	Create new index.
dropIndexes	Drop exixting index.
shutdoen	Shut down database.

Install MongoDB



Start MongoDB

Administrator: Command Prompt - mongo

```
Microsoft Windows [Version 10.0.17763.1098]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net start MongoDB
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.
```

Connect to MongoDB

Administrator: Command Prompt - mongo

```
C:\>cd "Program Files"

C:\Program Files>cd MongoDB

C:\Program Files\MongoDB>cd Server

C:\Program Files\MongoDB\Server>cd 4.2

C:\Program Files\MongoDB\Server\4.2>cd bin

C:\Program Files\MongoDB\Server\4.2\bin>mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3780e79c-b0b2-4302-b8f1-ee786a38be09") }
MongoDB server version: 4.2.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten]
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is
unrestricted.
2020-04-13T13:16:51.784-0500 I  CONTROL  [initandlisten]
MongoDB Enterprise >
```

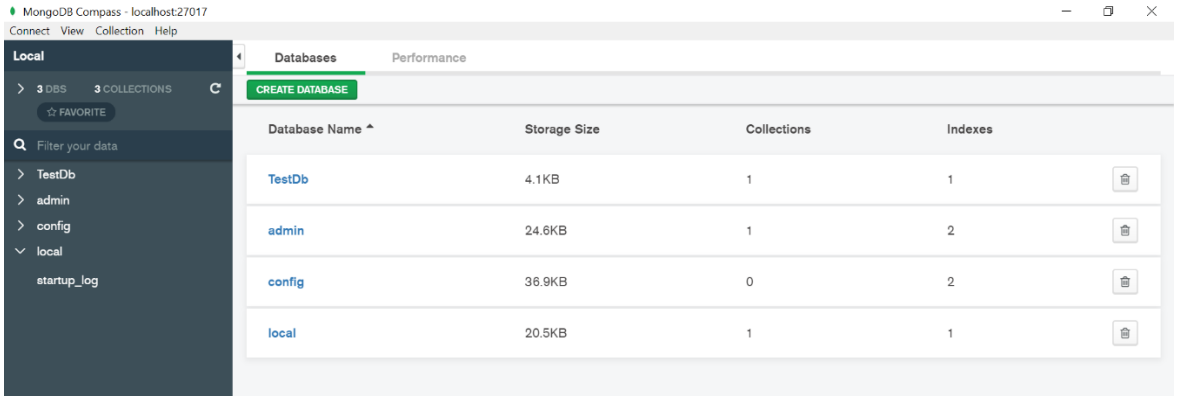
Lab 2: Importing a collection and exploring a database

Objective: To import a collection of suppliers to a database.

Administrator: Command Prompt - mongo

```
C:\Program Files\MongoDB\Server\4.2\bin>mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a377b8a2-d5a0-4df0-87a2-e0890fc53676") }
MongoDB server version: 4.2.5
Server has startup warnings:
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten]
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-13T13:16:51.783-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is
unrestricted.
2020-04-13T13:16:51.784-0500 I  CONTROL  [initandlisten]
MongoDB Enterprise >
```

2. Run the mongoimport binary



3. list the existing databases.

Administrator: Command Prompt - mongo

```
MongoDB Enterprise > show dbs
TestDb  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
MongoDB Enterprise >
```

4. Show the database you are in

Administrator: Command Prompt - mongo

```
MongoDB Enterprise > show dbs
TestDb  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
MongoDB Enterprise > use TestDb
switched to db TestDb
MongoDB Enterprise > db
TestDb
MongoDB Enterprise >
```

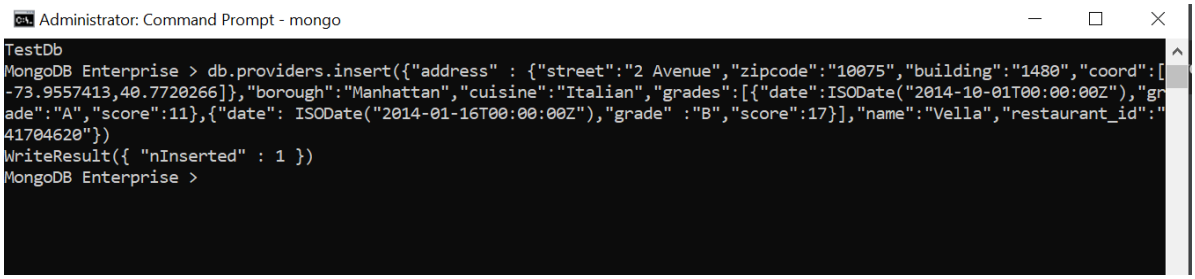
Lab 3: Querying providers

Insert, display and query documents.

Use the database you created during the past laboratory, and execute the sentences according to the requirements:

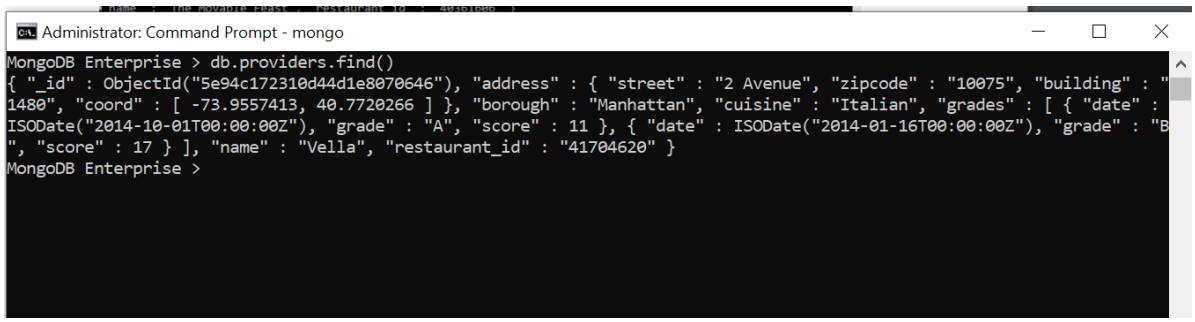
1.- Insert into the providers collection the following document:

db.providers.insert({"address" : {"street":"2 venue","zipcode":"10075","building":"1480","coord":[-73.9557413,40.7720266]},"borough":"Manhattan","cuisine":"Italian","grades":[{"date":ISODate("2014-10-01T00:00:00Z"), "grade" :"A","score":11},{"date": ISODate("2014-01-6T00:00:00Z"),"grade" :"B","score":17}],"name":"Vella","restaurant_id":"41704620"})



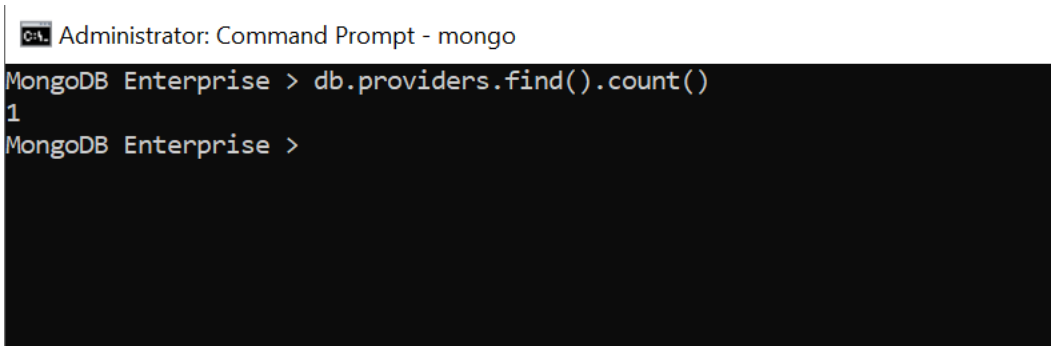
2.- Display all the documents of the providers collection

db.providers.find()



3.- Count the number of documents stored in providers collection

db.providers.find().count()



4.- For establishing conditions: Find the restaurant providers from Manhattan

db.providers.find({ "borough": "Manhattan" })

```
Select Administrator: Command Prompt - mongo
MongoDB Enterprise > db.providers.find({"borough":"Manhattan"})
{ "_id" : ObjectId("5e94c172310d44d1e8070646"), "address" : { "street" : "2 Avenue", "zipcode" : "10075", "building" : "
1480", "coord" : [ -73.9557413, 40.7720266 ] }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" :
ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B
", "score" : 17 } ], "name" : "Vella", "restaurant_id" : "41704620" }
MongoDB Enterprise >
```

5.- For counting documents: Get the number of restaurant providers from Manhattan
db.providers.find({ "borough": "Manhattan" }).count()

```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.providers.find({"borough":"Manhattan"}).count()
1
MongoDB Enterprise >
```

6.- For querying subdocument fields: Get the restaurant providers that have a zipcode 10075
db.providers.find({ "address.zipcode": "10075" }

```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.providers.find( { "address.zipcode": "10075" })
{ "_id" : ObjectId("5e94c172310d44d1e8070646"), "address" : { "street" : "2 Avenue", "zipcode" : "10075", "building" : "
1480", "coord" : [ -73.9557413, 40.7720266 ] }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" :
ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B
", "score" : 17 } ], "name" : "Vella", "restaurant_id" : "41704620" }
MongoDB Enterprise >
```

7.- For querying array fields: Get the providers with grade B

db.providers.find({ "grades.grade": "B" })

```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.providers.find( { "grades.grade": "B" } )
{ "_id" : ObjectId("5e94c172310d44d1e8070646"), "address" : { "street" : "2 Avenue", "zipcode" : "10075", "building" : "
1480", "coord" : [ -73.9557413, 40.7720266 ] }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" :
ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B
", "score" : 17 } ], "name" : "Vella", "restaurant_id" : "41704620" }
MongoDB Enterprise >
```

8.- Using operators: Get the providers that have a score greater than 10

db.providers.find({ "grades.score": { $gt: 10 } })

Get the providers that have a score less than 30

db.providers.find({ "grades.score": { $lt: 30 } })

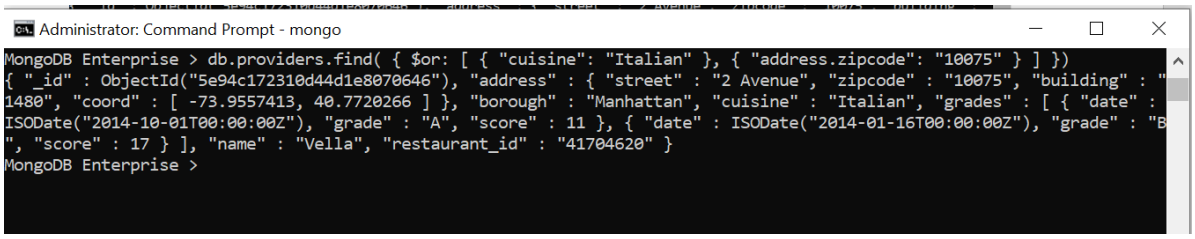
```
Administrator: Command Prompt - mongo
MongoDB Enterprise > db.providers.find({"grades.score":{$gt:10}})
{ "_id" : ObjectId("5e94c172310d44d1e8070646"), "address" : { "street" : "2 Avenue", "zipcode" : "10075", "building" : "
1480", "coord" : [ -73.9557413, 40.7720266 ] }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" :
ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B
", "score" : 17 } ], "name" : "Vella", "restaurant_id" : "41704620" }
MongoDB Enterprise > db.providers.find({"grades.score":{$lt:30}})
{ "_id" : ObjectId("5e94c172310d44d1e8070646"), "address" : { "street" : "2 Avenue", "zipcode" : "10075", "building" : "
1480", "coord" : [ -73.9557413, 40.7720266 ] }, "borough" : "Manhattan", "cuisine" : "Italian", "grades" : [ { "date" :
ISODate("2014-10-01T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-01-16T00:00:00Z"), "grade" : "B
", "score" : 17 } ], "name" : "Vella", "restaurant_id" : "41704620" }
MongoDB Enterprise >
```

9.-Find providers of Italian cuisine and zipcode 10075

db.providers.find({ "cuisine": "Italian", "address.zipcode": "10075" })

10.- Find providers of Italian cuisine or those with zipcode 10075

db.providers.find({ $or: [{ "cuisine": "Italian" }, { "address.zipcode": "10075" }] })



11.- For ordering results: Get providers ordered by borough and zipcode
db.proveedores.find().sort({ "borough": 1, "address.zipcode": 1 })