ADVANCED DB: MONGO DB LAB

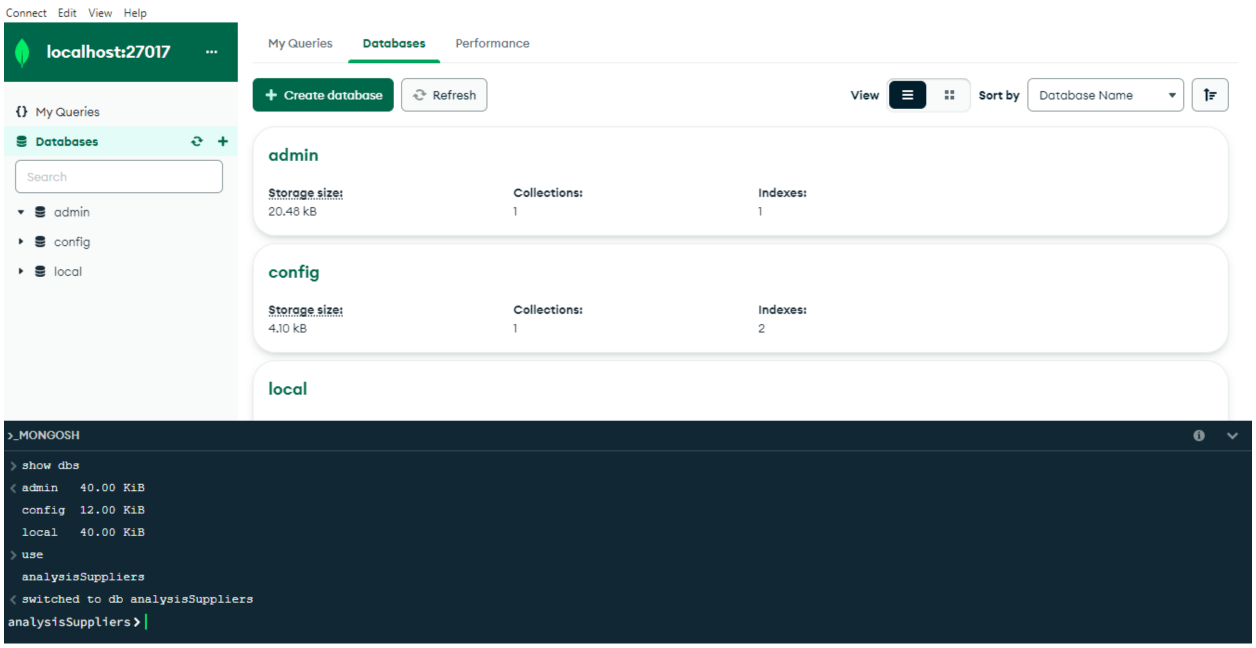
ICS 3C

Singh Sehmi – 146254 - BICS

LAB 1:

5.- Execute the following commands and in a table explain what each one is for:

Command	Explanation
Show dbs	The following command shows what current databases are being used
Use analysisSuppliers	Allows you to switch to the database named “Analysis Supplier”



	Storage size:	Collections:	Indexes:
admin	20.48 kB	1	1
config	4.10 kB	1	2
local			

```
>_MONGOSH
> show dbs
< admin   40.00 KiB
  config  12.00 KiB
  local   40.00 KiB
> use
  analysisSuppliers
< switched to db analysisSuppliers
analysisSuppliers>
```

Fig 1a: Shows the output after running the above commands.

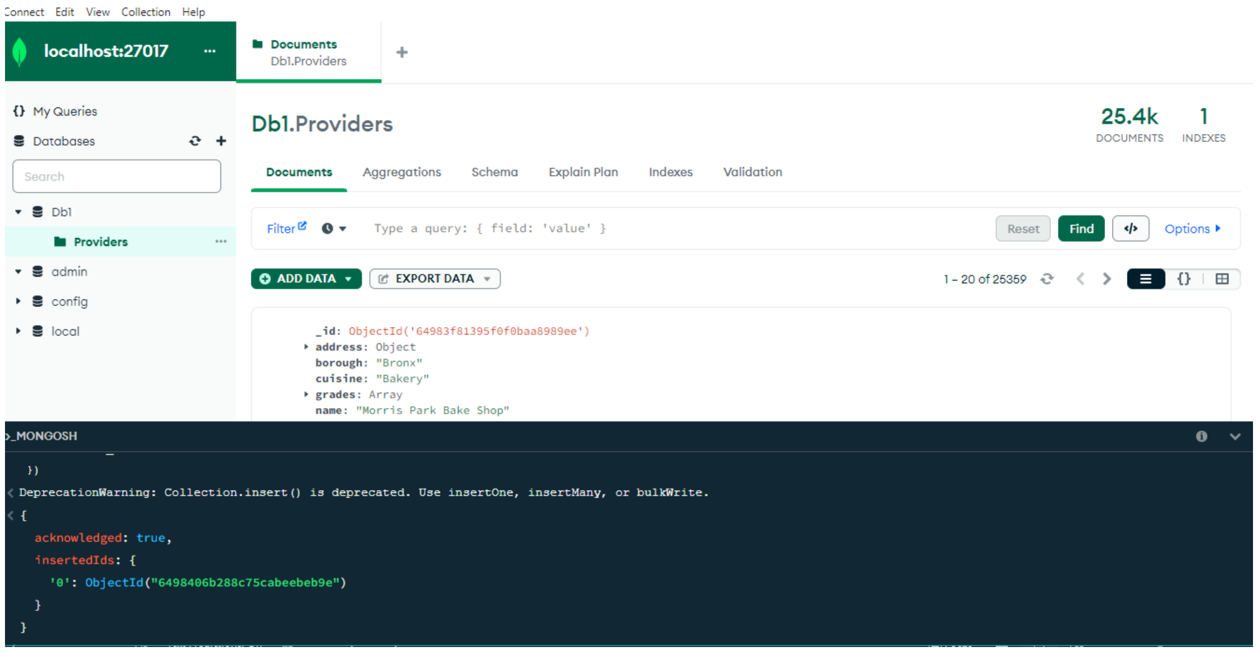
6.

Command	Function
Aggregate	Executes aggregation operations, such as $group, by utilizing an aggregation pipeline.
Count	Calculates the total count of documents within a collection or a view.
Distinct	Retrieves the unique values observed for a given key within a collection or a view and presents them as output.
Group	Groups documents in a collection based on a specified key and applies aggregations to the grouped data.

mapReduce	Executes map-reduce aggregation on extensive datasets, facilitating data analysis and summarization.
Find	Filters and retrieves specific documents from a collection or a view based on defined criteria.
Insert	Inserts one or more documents.
Update	Updates one or more documents.
Delete	Deletes in one or more documents
findAndModify	Returns and modifies a single document.
parallelColllectionScan	perform a parallel scan operation on a collection
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	Re-names the collection
Copydb	Copies the database
dropDatabase	Deletes the database
listCollection	retrieve a list of collections within a specific database
Drop	Removes the specified collection from the database.
Create	Creates a collection or a view.
Clone	Copies a collection or a view.
CreateIndexes	Creates a new index
shutdoen	Shuts down the collection/view

LAB 3:

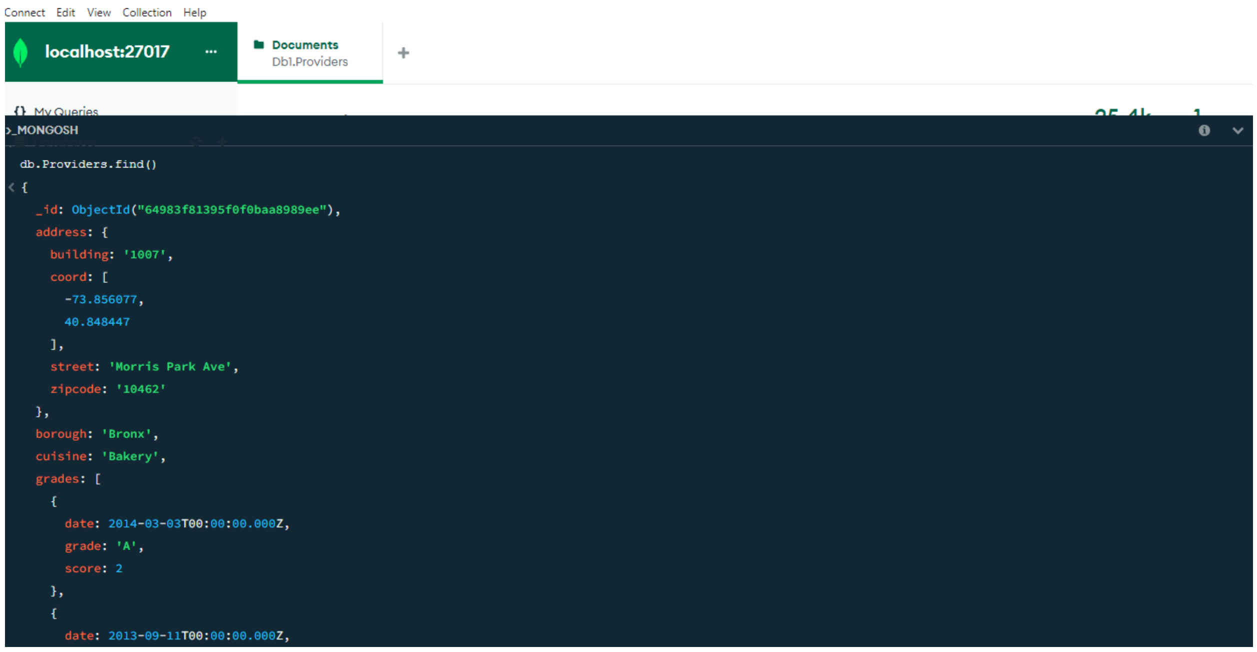
1. Result:



```
_id: ObjectId('64983f81395f0f0baa8989ee')
▸ address: Object
  borough: "Bronx"
  cuisine: "Bakery"
▸ grades: Array
  name: "Morris Park Bake Shop"
```

```
>_MONGOSH
  })
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("6498406b288c75cabeebeb9e")
    }
  }
```

2. Command: db.Providers1.find()



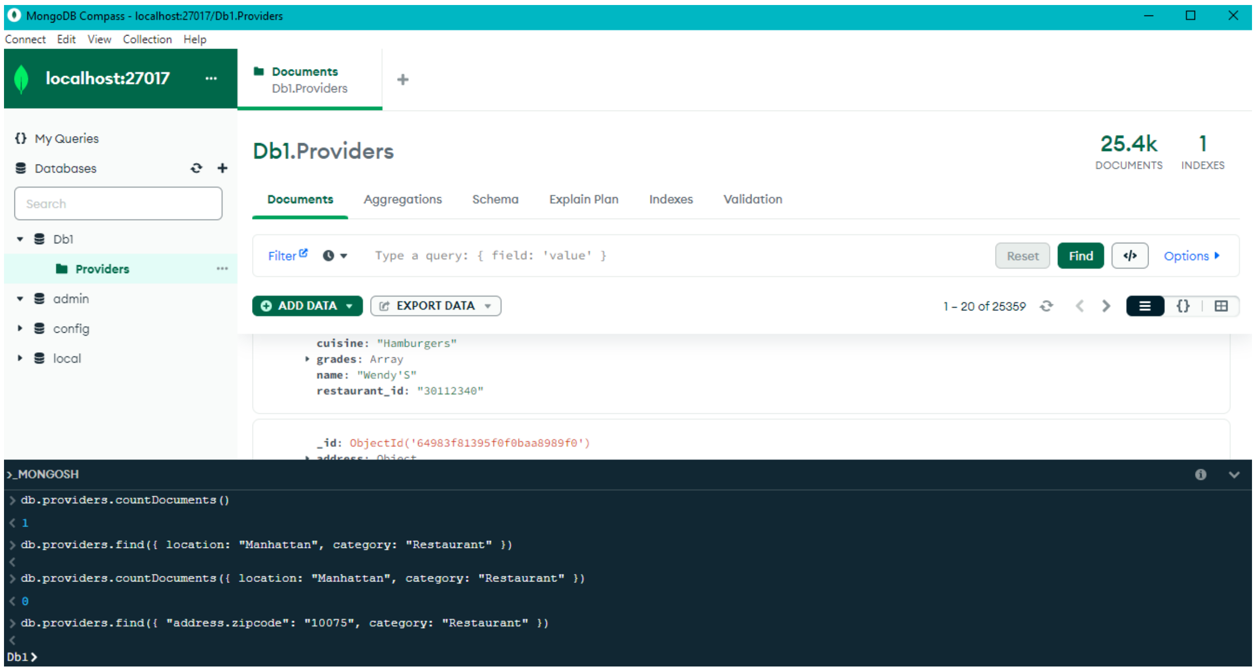
```
>_MONGOSH
  db.Providers.find()
< {
    _id: ObjectId("64983f81395f0f0baa8989ee"),
    address: {
      building: '1007',
      coord: [
        -73.856077,
        40.848447
      ],
      street: 'Morris Park Ave',
      zipcode: '10462'
    },
    borough: 'Bronx',
    cuisine: 'Bakery',
    grades: [
      {
        date: 2014-03-03T00:00:00.000Z,
        grade: 'A',
        score: 2
      },
      {
        date: 2013-09-11T00:00:00.000Z,
```

3. Command: db.providers.countDocuments() (*provided in screenshot below for 3 - 6*)

4. db.providers.find({ location: "Manhattan", category: "Restaurant" })

5. db.providers.countDocuments({ location: "Manhattan", category: "Restaurant" })

6. db.providers.find({ "address.zipcode": "10075", category: "Restaurant" })

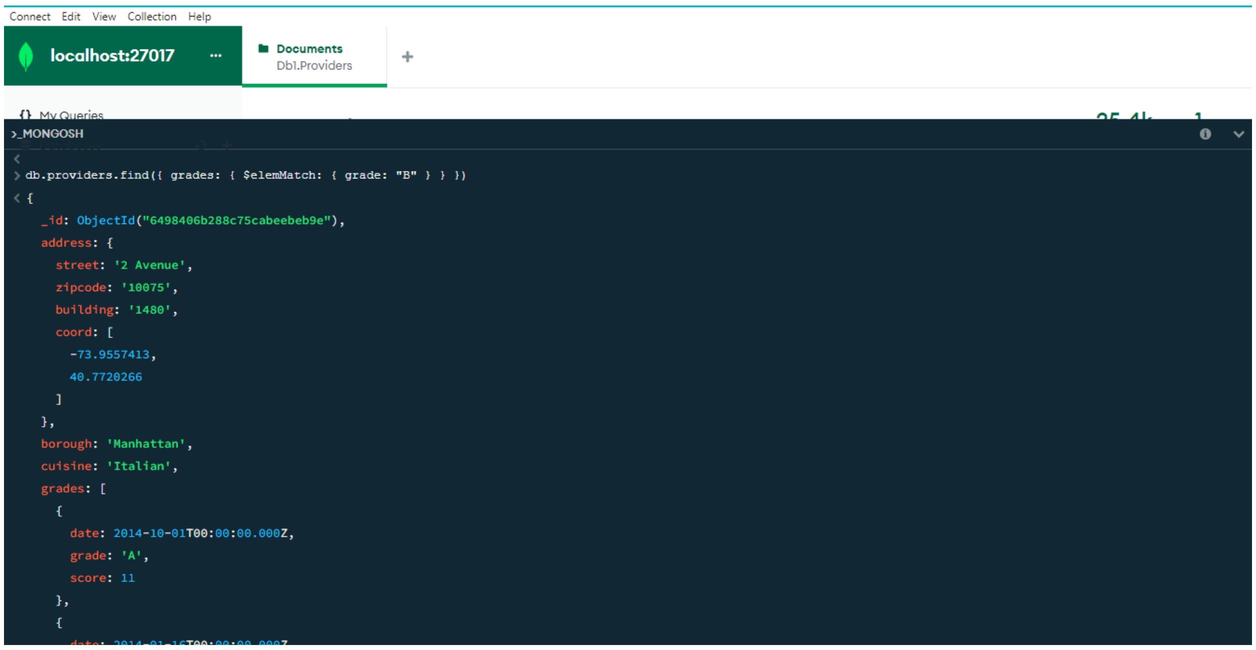


```
    cuisine: "Hamburgers"
  ▸ grades: Array
    name: "Wendy'S"
    restaurant_id: "30112340"

    _id: ObjectId('64983f81395f0f0baa8989f0')
```

```
>_MONGOSH
> db.providers.countDocuments()
< 1
> db.providers.find({ location: "Manhattan", category: "Restaurant" })
<
> db.providers.countDocuments({ location: "Manhattan", category: "Restaurant" })
< 0
> db.providers.find({ "address.zipcode": "10075", category: "Restaurant" })
<
Db1>
```

7. Command: db.providers.find({ grades: { $elemMatch: { grade: "B" } } })

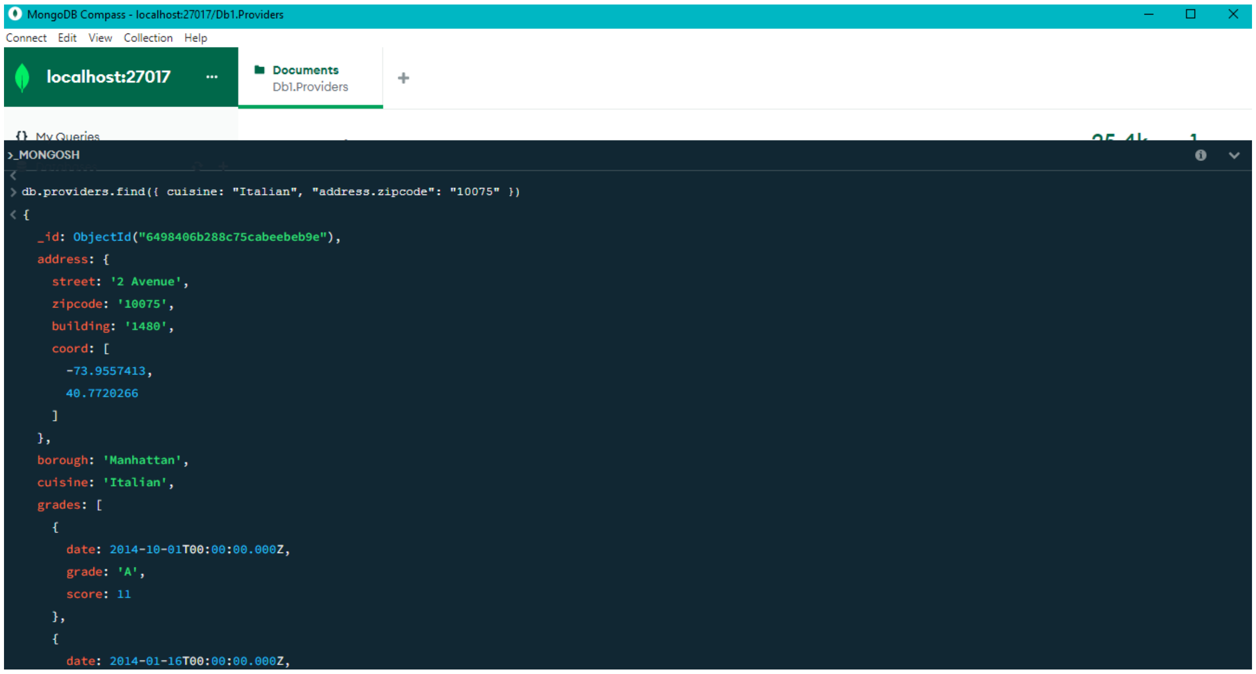


```
>_MONGOSH
<
> db.providers.find({ grades: { $elemMatch: { grade: "B" } } })
< {
    _id: ObjectId("6498406b288c75cabeebeb9e"),
    address: {
      street: '2 Avenue',
      zipcode: '10075',
      building: '1480',
      coord: [
        -73.9557413,
        40.7720266
      ]
    },
    borough: 'Manhattan',
    cuisine: 'Italian',
    grades: [
      {
        date: 2014-10-01T00:00:00.000Z,
        grade: 'A',
        score: 11
      },
      {
```

8. Command: db.providers.find({ score: { $gt: 30 } })

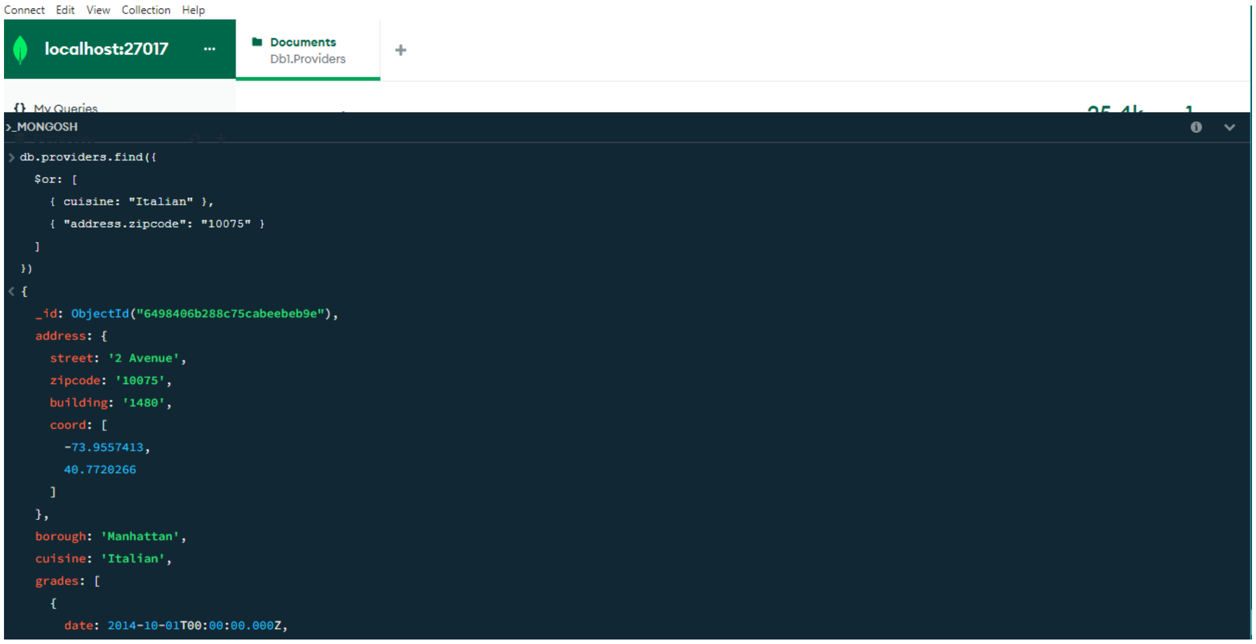
db.providers.find({ score: { $lt: 10 } })

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


MongoDB Compass - localhost:27017/Db1.Providers
Connect Edit View Collection Help
localhost:27017
Documents
Db1.Providers
My Queries
>_MONGOSH

```
> db.providers.find({ cuisine: "Italian", "address.zipcode": "10075" })
< {
    _id: ObjectId("6498406b288c75cabeebeb9e"),
    address: {
      street: '2 Avenue',
      zipcode: '10075',
      building: '1480',
      coord: [
        -73.9557413,
        40.7720266
      ]
    },
    borough: 'Manhattan',
    cuisine: 'Italian',
    grades: [
      {
        date: 2014-10-01T00:00:00.000Z,
        grade: 'A',
        score: 11
      },
      {
        date: 2014-01-16T00:00:00.000Z,
```

10.



```
> db.providers.find({
    $or: [
      { cuisine: "Italian" },
      { "address.zipcode": "10075" }
    ]
  })
< {
    _id: ObjectId("6498406b288c75cabeebeb9e"),
    address: {
      street: '2 Avenue',
      zipcode: '10075',
      building: '1480',
      coord: [
        -73.9557413,
        40.7720266
      ]
    },
    borough: 'Manhattan',
    cuisine: 'Italian',
    grades: [
      {
        date: 2014-10-01T00:00:00.000Z,
```