uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Database

Lourdes Moreno López
Paloma Martínez Fernández
José Luis Martínez Fernández
Rodrigo Alarcón García

Lab demo 2 (Topic MongoDB (3.3))

1. Download the dataset of "primer-dataset.json" (source: https://github.com/OpenKitten/Mongo-Assets/blob/master/primer-dataset.json) and save the file in 'C:\data\db\'

Each document is a restaurant with information: its identifier, location, the neighborhood, the type of cuisine, the name and a set of documents embedded on the scores received on different dates


```
{
        "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"
    }
```


2. Import of the "restaurants" collection.
Open the shell windows command line, and you execute (see Figure1, Figure 2 and Figure 3):

Execute:

```
Mongoimport --host localhost --port 27017 -d test -c restaurants
C:\data\db\primer-dataset.json
```

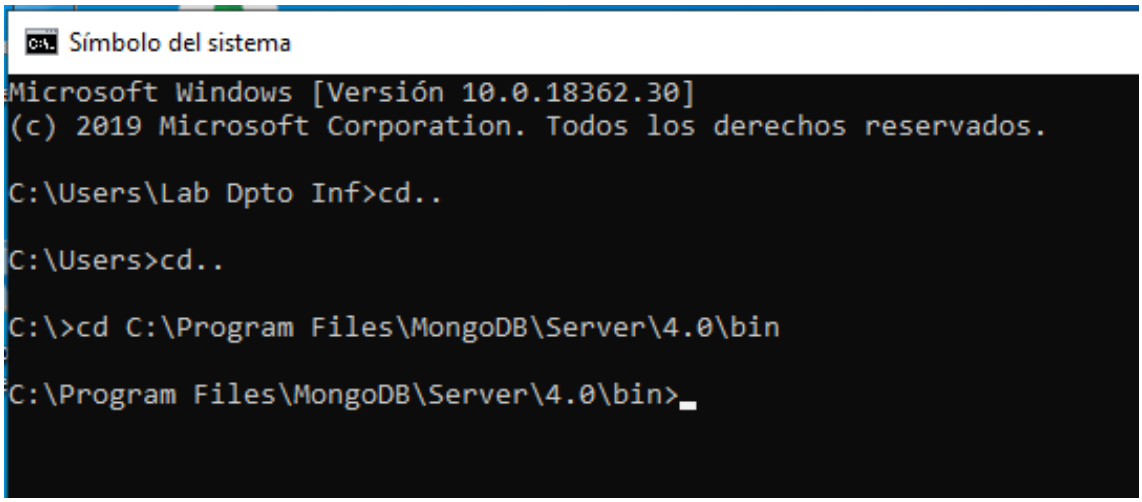

Figure 1

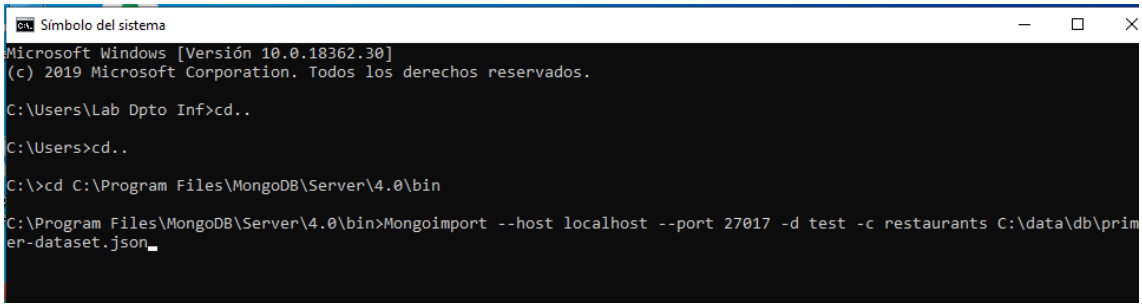

Figure 2

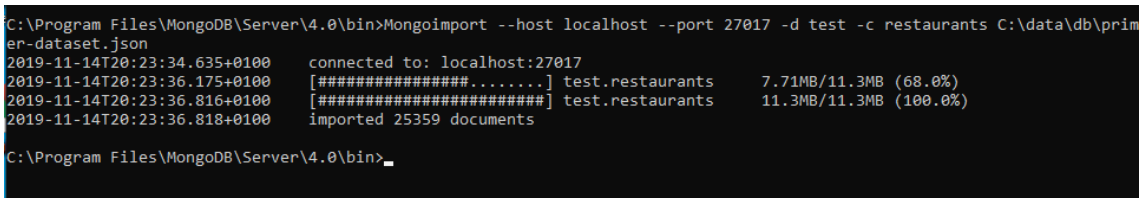

Figure 3

3. Open mongoDB with shell or Robo3T, and execute the following operations.

All documents in the collection:

```
db.restaurants.find()
```

Restaurants district of Manhattan:

```
db.restaurants.find( { "borough": "Manhattan" } )
```

To access a field in an embedded document, the notation "." is used. Restaurants with zip code equal to 10075

```
db.restaurants.find( { "address.zipcode": "10075" } )
```

Use of "Greater than" ($gt): Restaurants with a score greater than 30.

```
db.restaurants.find( { "grades.score": { $gt: 30 } } )
```

Use of "Less than" ($lt): Restaurants scoring less than 10

```
db.restaurants.find( { "grades.score": { $lt: 10 } } )
```

AND: conditions separated by commas: Italian restaurants in the zip code "10075"

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

OR: Using the operator $or: Restaurants with Italian cuisine or that your zip code is 10075

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

Sorting results: "sort" method with the sort fields and "1" for ascending and "-1" for descending: List of restaurants sorted ascendingly by borough, and by zip code

```
db.restaurants.find().sort( { "borough": 1, "address.zipcode": 1 } )
```

Update a field (use $set operator, if it does not exist the field creates it): Update the "cuisine" field with "American (New)" for the restaurant with name "Juni"

```
db.restaurants.update(
    { "name" : "Juni" },
    {
      $set: { "cuisine": "American (New)" }
    }
)
```

Update an embedded field (use $ set operator, if the field does not exist, create it): Update an embedded field (use $ set operator, if the field does not exist, create it)

```
db.restaurants.update(
  { "restaurant_id" : "41156888" },
  { $set: { "address.street": "East 31st Street" } }
)
```

Replace a document

```
db.restaurants.update(
   { "restaurant_id" : "41704620" },
   {
     "name" : "Vella 2",
     "address" : {
              "coord" : [ -73.9557413, 40.7720266 ],
              "building" : "1480",
              "street" : "2 Avenue",
              "zipcode" : "10075"
     }
   }
)
```

Remove () to delete documents from a collection according to the condition: Delete restaurants whose borough is Manhattan

```
db.restaurants.remove( { "borough": "Manhattan" } )
```

Use justOne to erase only one: Remove only one restaurant from the borough of Queens.

```
db.restaurants.remove( { "borough": "Queens" }, { justOne: true } )
```

Delete all documents in the collection:

```
db.restaurants.remove( { } )
```

To erase everything:

```
db.restaurants
```