University of Minho
Department of Computer Science

FE03

Course: MSc in Informatics/ MSc in Bioinformatics
U.C.: NoSQL Databases

Worksheet 03 - PL04	
Teacher:	António Abelha / Cristiana Neto
Theme:	Introduction to MongoDB
Class:	Laboratory Practice
Academic Year:	2023-2024 – 2nd Semester
Duration of the lesson:	2 hours

1. Customers

Using mongoshell:

```
# docker exec -it <id-container-mongo> bash
# mongosh
```

[1] List all databases after installing the container with the MongoDB image.

[2] Create a database called "*customers*".

[3] Verify the creation of the database.

[4] Create a collection named "customers".

[5] Validate the creation of the collection.

[6] Create a client with the following characteristics:

```
first_name: "John", last_name: "Do", age: 30
```

[7] Introduce 2 clients into the collection with a single statement created with the following characteristics:

```
first_name: "Steven", last_name:"Williams", gender: "male"
first_name: "Mary", last_name: "Troy", age: 19
```

[8] Introduce one more customer with the following characteristics:

```
first_name: "Ric", last_name: "Foe", address: {street: "4 main st", city: "Boston"}
```

[9] Create a client with the following characteristics:

```
first_name: "Ana", last_name: "During", degree:["phD","Msc"],
address: {street: "4 Square Garden", city: "New York"}, age: 32
```

[10] Create a client with the following characteristics:

```
first_name: "Natalia", last_name: "Will", age: 44, gender: "female"
```

[11] List all customers.

[12] Perform an update to the client 'Ric', put age 45.

[13] Find all customers who have 'Will' in their last name.

[14] Perform an update to the client 'Steven', put age 35.

[15] Check if the age of the client 'Ana' is over 30 and if yes increase the age by 10 years.

[16] Client 'Ric' wants his age removed from the database.

[17] Search for a customer with the first name: "Jimmy" and update, or create, if it does not exist, with the following characteristics:

```
first_name: "Jimmy", last_name: "Connors", age: 25, gender: male
```

[18] Search for all customers over the age of 25.

[19] Search for all male customers.

[20] Delete the client whose first name is "Mary".

[21] Find customers with the name "Ana" or "Ric".

2. restaurants.json

Import the "restaurants.json" using Compass for a collection *restaurants*.

Consider:

```
{
  "address": {
    "building": "351",
    "coord": {
      "type": "Point",
      "coordinates": [
        -73.98513559999999,
        40.7676919
      ]
    },
    "street": "West   57 Street",
    "zipcode": "10019"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grids": [
     { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
     { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
     { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
     { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
     { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

[1] List all documents in the Restaurants collection.

[2] List only the *restaurante_id*, *Name*, *Borough,* and *Cuisine fields* for all documents in the collection.

[3] List the *restaurante_id*, *Name*, *Borough,* and *Cuisine fields* for all documents in the collection, but delete the _id field.

[4] List the *restaurante_id*, *Name*, *Borough,* and *ZipCode fields* for all documents in the collection, but exclude the _id field.

[5] List the restaurants that are located in the *"Bronx"* borough.

[6] List the first 5 restaurants that are located in the *"Bronx"* borough.

[7] list the 5 restaurants after the first 5 (6th-10th) that are located in the *"Bronx"* borough.

University of Minho
Department of Computer Science

[8] List all restaurants that have at least a score greater than 90.

[9] List all restaurants that have a score greater than 80 but less than 100.

[10] List all restaurants that are located at a latitude (*coordinates.0*) lower than -95.754168.

[11] List all restaurants whose cuisine type is not "American", whose *score* is greater than 70, and whose latitude (*address .coord.0*) is less than -65.754168, using the operator $and.

[12] List all restaurants whose cuisine type is not "American", their *score* is greater than 70, and their latitude (*address .coord.0*) is less than -65.754168.

[13] List all restaurants whose cuisine is not "American" and that have achieved an "A" grade rating but do not belong to the "Brooklyn" *borough*. It should be presented according to the type of cuisine in descending order.

[14] List all restaurants that belong to the *Bronx borough* and whose cuisine is either "American" or "Chinese".

[15] List all restaurants that contain street information (address.street).

[16] List all restaurants in ascending order by type of cuisine and descending by borough.

[17] List the restaurant_id, name, address, and geographic location (*coord*) for restaurants whose second element of the geographic location array (*coord*) is greater than 42 and even 52.

[18] List the restaurants (*restaurante_id*, *name*, *borough*, *cuisine*) that did not achieve a score greater than 10.

[19] List all restaurants (*restaurante_id*, *name*, *borough,* and *cuisine*) that do not belong to the *borough* of "Staten Islan", or "Queens", or "Bronx", or "Brooklyn".

[20] Update the "name" field for a specific restaurant with the restaurante_id equal to "30191841" to a new name, e.g. "NewName Restaurant".