MONGODB: NOSQL DATABASES

Exercises

Alberto Fernández Merchán

1. Exercises in MongoDB

First of all, we must use the database “ExercisesMongoDB”.

> use ExercisesMongoDB

1. Create a collection called "products" and populate it with these document:

{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" }

{ item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" }

{ item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" }

{ item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" }

{ item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }

```
> db.products.insertMany([{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
                          { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
                          { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
                          { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
                         { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }])
```

2. Perform the following CRUD operations:
   2.1. Get all documents in the collection: > **db.products.find()**

   2.2. Get all the information of the product "planner": > **db.products.find({item:"planner"})**

   2.3. Get all the information of the products that have a quantity (qty) greater than 60:
   > **db.products.find({qty: {$gt: 60}})**

   2.4. Obtain all the information of the products that have a quantity (qty) between 75 and 100 (both inclusive): > **db.products.find({qty: {$gte: 75,$lte: 100}})**

   2.5. Get all the information of the products whose size is saved in centimeters:
   > **db.products.find({"size.uom":"cm"})**

   2.6. Obtain the name and status of products with "D" status: > **db.products.find({status:"D"}, {item:1,status:1,_id:0})**

   2.7. Change the status of the product "postcard" to the value "C":
   > **db.products.updateOne({item:"postcard"},{$set:{status:"C"}})**

   2.8. Change notebook product dimensions to (10, 14):
   > **db.products.updateOne({item:"notebook"},{$set:{"size.h":10,"size.w":14}})**

   2.9. Move to "cm" all products "in":
   > **db.products.updateMany({"size.uom":"in"},{$set:{"size.uom":"cm"}})**

   2.10. Remove products with "A" status: > **db.products.deleteMany({status:"A"})**

3. Create a collection called "products2" and populate it with these documents.

{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [14, 21] },

{ item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [14, 21] },

{ item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [14, 21] },

{ item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [22.85, 30] },

{ item: "postcard", qty: 45, tags: ["blue"], dim_cm: [10, 15.25] }

3.1. Creating a new collection: > **db.createCollection("products2")**

3.2. Inserting documents: > db.products2.insertMany([
{ item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [14, 21] },
{ item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [14, 21] },
{ item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [14, 21] },
{ item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [22.85, 30] },
{ item: "postcard", qty: 45, tags: ["blue"], dim_cm: [10, 15.25] }])

4. Perform the following CRUD operations:

4.1. Get products that have the "blank" and "network" labels (only those tags and in that order):
> **db.products2.find({tags:["blank","red"]})**

4.2. Get products that have the labels "network" and "blank" (in any order):
> **db.products2.find({tags:{$all: ["blank","red"]}})**

4.3. Get products that have the "plain" label:
> **db.products2.find({tags:{$all:["plain"]}})**

4.4. Obtain products that have at least one dimension less than 12:
> **db.products2.find({dim_cm:{$lt:12}})**

4.5. Obtain products whose dimensions meet, at least, be greater than 15 or less than 20:
> **db.products2.find({dim_cm:{$elemMatch: {$gt:15,$lt: 20}}})**

4.6. Obtain products that have at least a dimension greater than 15 and less than 18:
> **db.products2.find({dim_cm:{$gt:15,$lt: 18}})**

4.7. Obtain products whose second dimension is 14:
> **db.products2.find({"dim_cm.1":14})**

4.8. Obtain products whose first dimension is greater than 20:
> **db.products2.find({"dim_cm.0":{$gt: 20}})**

4.9. Get products that have a single label:
> **db.products2.find({tags:{$size:1}})**

4.10. Add the "orange" label to the "journal" product:
> **db.products2.updateOne({item:"journal"},{$push: {tags: "orange"}})**

4.11. Remove the "plain" label from the "paper" product:
> **db.products2.updateOne({item:"paper"},{$pull: {tags: "plain"}})**

4.12. Increase by 10 the quantities of products that have more than 50 units:
> **db.products2.updateMany({qty:{$gt: 50}},{$inc: {qty: 10}})**