CMD as admin

net stop MongoDB --> to stop Mongo Server

net start MongoDB --> start Mongo Server

USE MONGOSH SHELL

Search for help inside mongosh: help

show dbs

use db\_name

db.createCollection("collection\_name") 🡪 I can add more parameters, review the file validation.js from Sec3-Vid54

db.collectionName.drop() 🡪 delete my actual collection

In MongoDB

Create

insertOne(data, options)

insertMany(data, options)

Read

find(filter, options).pretty() --> <filter> if use {key: {$gt: value}},

means find every document with key greater than value

returns a cursor, NOT a list of documents

find().toArray() --> enables to show more than 20 documents, but it's not to optimal

find().forEach((variable\_name) => {printjson(variable\_name)})

--> allows to apply a JS function

findOne(filter, options) --> <filter> means put the atributte that makes that document unique

shows the first marching element

In here doesn't work .pretty()

Update

updateOne(filter, data, options) --> in <data> use {$set: {key: value}}

Adds the new information and keeps the oldest

updateMany(filter, data, options) --> if <filter> is {} blanck, means to select all the documents. Needs $set too

Adds the new information and keeps the other inf in the doc

update(filter, data, options) --> in this case i dont need to use $set.

But replace everything inside my document with the data object specified in update

It's better use updateOne or Many.

replaceOne(filter, data, options)

Delete

deleteOne(filter, options)

deleteMany(filter, options)

Any word with $ before. i.e. $set. Means it's a reserved operator or word

PROJECTION

Is a filter or a data transformation. It shows only the keys values i specific ask to show.

db.nameCollection.find({},{key\_to\_show: 1, another: 1})

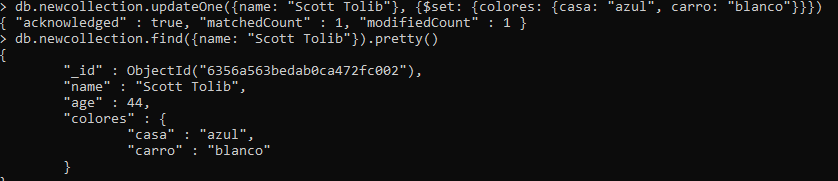
<1> means i want to show all with key\_to\_show. And the \_id.



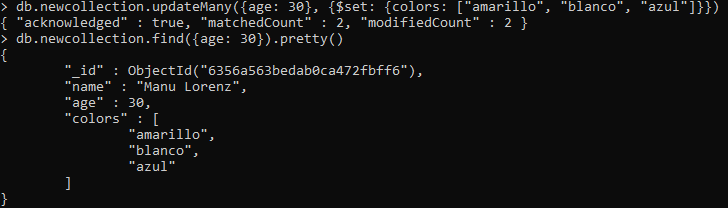
<0> means i exclude something. I use \_id: 0, to hide the id information in the projection



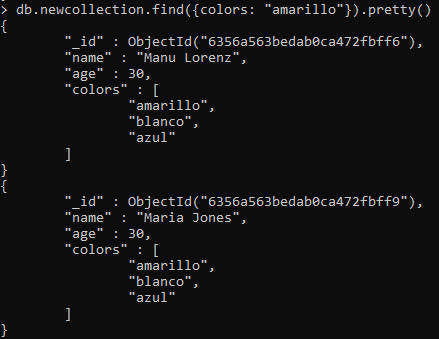
NESTED DOCUMENTS



NESTED LISTS







db.stats() 🡪 utility method, returns me stats about the db in use



db.patientData.updateOne({"firstName": "Julio"}, {$set: {"age": 28, "firstName": "Julio Andres", "lastName": "Aguilar Paz",

DATA RELATIONSHIPS

One to one embedded

One to one using references

One to many embedded

One to many using references

Many to many embedded

Many to many using references

Look.Up() for merging reference relations. This union use more performance than an embedded document. Cannot use always

([{ from: where the related documents live in, localField: where find the collection, foreingField: it’s a key, as: give an alias

CREATE A BLOG

Three entities:

**User**  id  
 name  
age  
email  
 post  
 comment

**Post** id  
 title  
 content  
 tags  
 embedded user not ok if I nest this. Bcs if I modified something from the user, I need to change all the post too. There’s a lot of duplicated data  
 embedded comments nesting this it’s ok

**Comments** id  
 content

Relation between entities:

User post 1:M

User comments 1:M

Post comments 1:M

