

Pràctica UF4

Bases de Dades objecte-relacionals



Elaborat per

MERITXELL LLANAS GIRONÈS

ÍNDEX

1. Què són les bases de dades NoSQL?	2
2. Tipus de bases de dades NoSQL	2
3. Diferències, avantatges i inconvenients entre les bases de dades SQL i les noSQL	3
4. Aplicació d'ús: mongoDB	4
5. Com crear una BBDD	5
6. Omplir la BBDD	5
7. CRUD	8
7.1. Crear / Inserir nous Documents (Create)	8
7.2. Mostrar / Llegir / Buscar Documents (Read)	9
7.3. Modificar / Actualitzar Documents (Update)	11
7.4. Eliminar Documents (Delete)	13
7.5. Eliminar Collections (Delete)	14
7.6. Eliminar BD (Delete)	15
8. Webgrafia	15

Fes un treball de recerca sobre les bases de dades noSQL. Explica breument quines existeixen i quines són les diferències, avantatges i inconvenients entre les bases de dades SQL i les noSQL.

1. Què són les bases de dades NoSQL?

Les bases de dades NoSQL no utilitzen el llenguatge SQL com a preferència per a interactuar amb les dades i, a més a més, són sistemes d'emmagatzematge d'informació que no compleixen amb l'esquema entitat-relació.

Significa que no és una base de dades relacional, ja que estructura les dades de forma flexible, sense encasellar-les dins de taules jerarquizades.

2. Tipus de bases de dades NoSQL

Actualment, hi ha 5 tipus de BBDD NoSQL disponibles:



BBDD de clau-valor

Són altament divisibles i permeten un elevat nivell d'escalat horitzontal. Ideals per treballar amb jocs, tecnologia publicitària i IoT.



BBDD de documents

Les dades es representen com un objecte de tipus JSON perquè és un model de dades eficient i intuïtiu per als desenvolupadors. Aquest tipus de model funciona bé amb catàlegs, perfils d'usuari i sistemes d'administració de continguts que evolucionen amb el temps.



BBDD de gràfics

Faciliten la creació i execució d'aplicacions que funcionen amb conjunts de dades altament connectades. Ideals per treballar amb xarxes socials, motors de recomanacions, detecció de frau i gràfics de coneixement.



BBDD en memòria

Ideals per a aplicacions que treballin amb taules de classificació, botigues de sessió i anàlisi en temps real que requereixen temps de resposta de microsegons i que poden tenir grans pics de trànsit en qualsevol moment.



BBDD de cerca

Cercadors que generen registres per ajudar els desenvolupadors a solucionar problemes. Proporcionen visualitzacions en temps real i anàlisi de dades generades per màquines en indexar, agregar i buscar registres i mètriques semiestructurades.

3. Diferències, avantatges i inconvenients entre les bases de dades SQL i les noSQL

Les BBDD NoSQL s'adapten perfectament a moltes aplicacions modernes, com a dispositius mòbils, web i jocs, que requereixen bases de dades flexibles, escalables, d'alt rendiment i altament funcionals per proporcionar excel·lents experiències d'usuari.

COMPARATIVA

Flexibilitat: les **BBDD NoSQL** generalment ofereixen esquemes flexibles que permeten un desenvolupament més ràpid i més iteratiu, fent-les ideals per a dades semiestructurades i no estructurades.

- Les **BBDD Relacionals** tenen una estructura fortament marcada per a tots els registres d'una taula: han de tenir les mateixes columnes i mateix tipus de dades associades a cadascuna.
- Per contra, les **BBDD NoSQL** emmagatzemen les dades organitzades en documents, i aquests poden tenir diferents camps i de diferent tipus de dades.

Escalabilitat: les **BBDD NoSQL** generalment estan dissenyades per escalar usant clústers distribuïts de maquinari enlloc d'escalar afegint servidors cars i sòlids, com és el cas de les **BBDD SQL**.

Rendiment: les **BBDD NoSQL** estan optimitzades per a models de dades específics i patrons d'accés que permeten un alt rendiment.

- **SQL** permet combinar de forma eficient diferents taules per extreure'n **informació relacionada**, mentre que **NoSQL** no ho permet o molt limitadament.
- **NoSQL** permet **distribuir grans quantitats d'informació**; mentre que **SQL** facilita distribuir bases de dades relacionals.

Altament funcional: les **BBDD NoSQL** proporcionen APIs altament funcionals i tipus de dades que estan dissenyades específicament per a cadascun dels seus models de dades respectius.

- Les BBDD SQL s'utilitzen en múltiples aplicacions de tota mena, les BBDD NoSQL s'utilitzen principalment per al Big Data.
- Les BBDD SQL proporcionen consistència a les dades (integritat). Tot i això, les NoSQL, en buscar rapidesa, no se centren en aquesta característica.

A continuació, crea una base de dades amb **mongoDB**. Insereix contingut i documenta com treballes amb aquest contingut (buscar, insertar, modificar, eliminar).

4. Aplicació d'ús: mongoDB

Per realitzar l'explicació pràctica de com es treballa en una BBDD NoSQL, partiré d'un SGBD força estès, que es diu **mongoDB**. Aquesta BD treballa gestionant les dades a mode de **documents**.

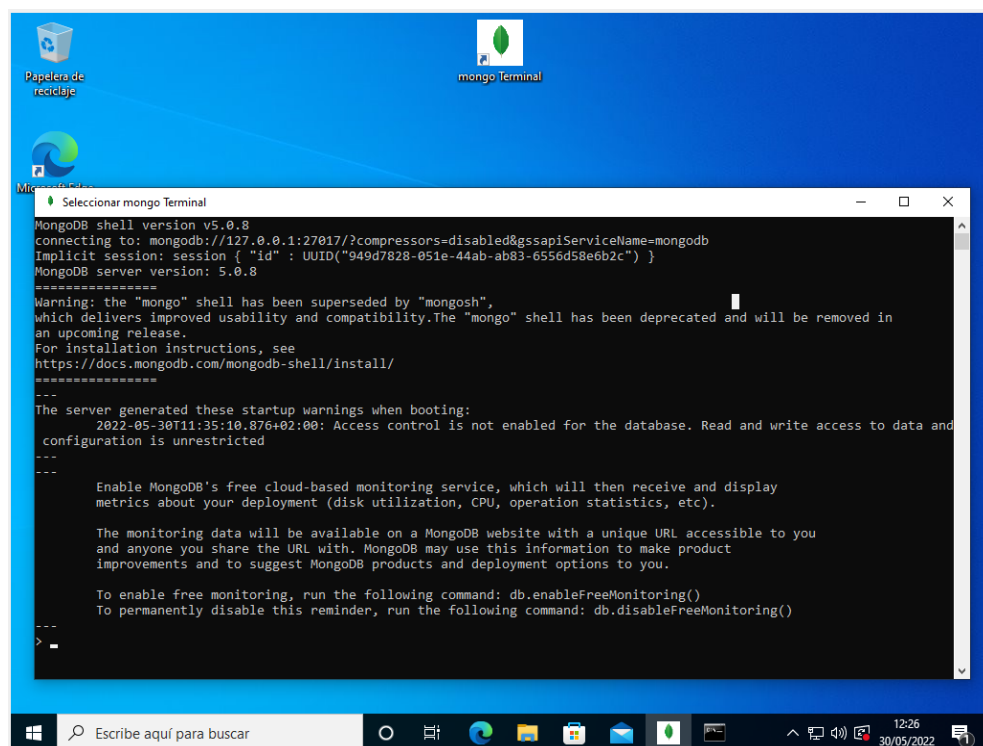
Concretament, treballaré amb la versió *MongoDB 5.0 Community Edition on LTS*.

Com a exemple, es procedirà a implementar un sistema de gestió d'exemplars de llibres per a una biblioteca.

MongoDB disposa de dos modes de treball: mitjançant comandes (via terminal) i mitjançant un aplicatiu d'escriptori anomenat *MongoDB Compass*.

Treballaré tot el tutorial fent ús del terminal natiu del programa: *mongoDB shell*.

Un cop instal·lat el programa, fem doble-clic sobre la icona ¹"*mongo Terminal*" que ens ha creat l'instal·lador, per obrir la terminal:



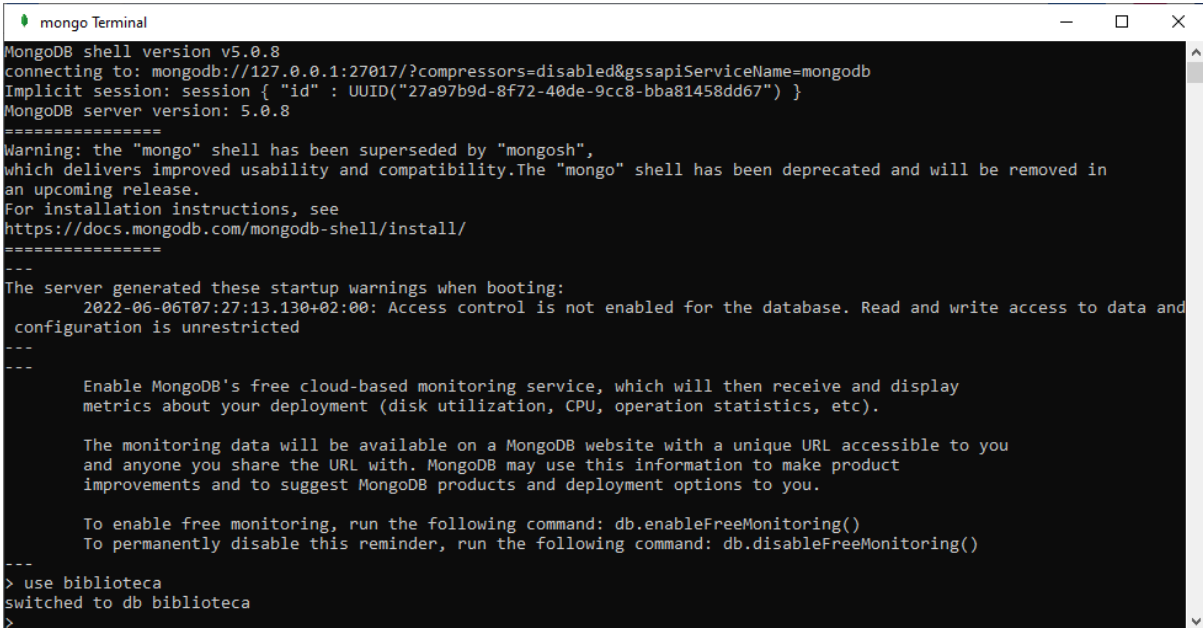
¹ Si no existeix aquest accés directe a l'escriptori, podem buscar el programa fent ús del cercador de programes del sistema operatiu on ens trobem.

5. Com crear una BBDD

El primer de tot, és crear la BBDD i definir-ne les diferents col·leccions. Posteriorment aniré afegint-hi els diversos documents que contindrà.

La següent comanda serveix per seleccionar la BBDD sobre la que volem treballar, però també permet precrear-la, ja que si encara no existeix no la materialitzarà fins que no fem una inserció d'un *Document*:

```
use biblioteca
```



```
mongo Terminal
MongoDB shell version v5.0.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("27a97b9d-8f72-40de-9cc8-bba81458dd67") }
MongoDB server version: 5.0.8
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2022-06-06T07:27:13.130+02:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use biblioteca
switched to db biblioteca
>
```

6. Omplir la BBDD

La implementació de les dades passa per definir *collections* (col·leccions) que contindran un o més documents a dins.

- Les Collections (NoSQL BBDD) equivalen a les taules (SQL BBDD, relacionals). Agrupen Documents.
- Els Documents (NoSQL BBDD) equivalen als registres (SQL BBDD, relacionals).

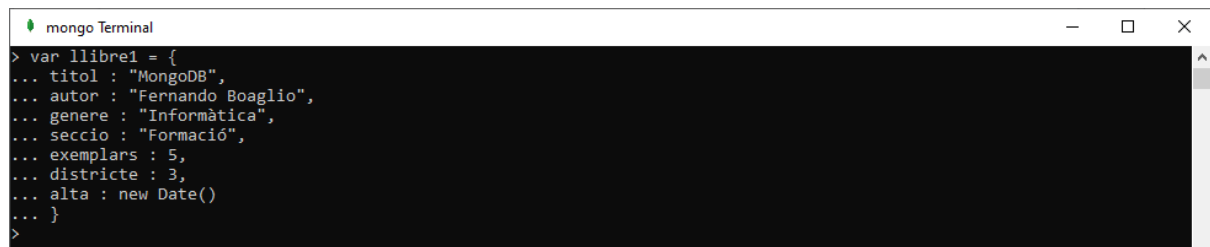
Es poden definir primer les *collections*, però com passa a l'hora de crear una BBDD, només es crearan quan inserim algun document en elles. Podríem utilitzar la comanda següent per forçar-ne la creació:

```
db.createCollection(nom_collection, opcions)
```

Així que començaré per crear un *Document*, que a l'hora definirà una *Collection* i em permetrà crear la BD.

Els documents es guarden dins d'objectes JSON. Per crear-ne un, l'hem de definir dins d'una variable seguint una estructura de **clau / valor**, separats per comes:

```
var llibre1 = {  
  titol : "MongoDB",  
  autor : "Fernando Boaglio",  
  genere : "Informàtica",  
  seccio : "Formació",  
  exemplars : 5,  
  districte : 3,  
  alta : new Date()  
}
```

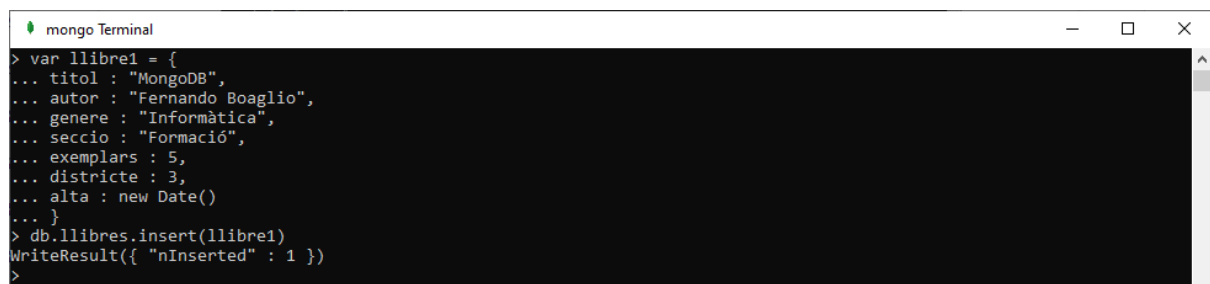


```
mongo Terminal  
> var llibre1 = {  
... titol : "MongoDB",  
... autor : "Fernando Boaglio",  
... genere : "Informàtica",  
... seccio : "Formació",  
... exemplars : 5,  
... districte : 3,  
... alta : new Date()  
... }  
>
```

Per inserir el Document a la BD que hem definit prèviament, escrivim:

```
db.llibres.insert(llibre1)
```

db: defineix la bd on estem treballant, definida al pas (1)
llibres: defineix la *Collection* on volem guardar el Document que acabem de definir.
insert: és l'acció que volem aplicar al *Document*
(llibre1): és el *Document* que hem definit dins la variable llibre1.



```
mongo Terminal  
> var llibre1 = {  
... titol : "MongoDB",  
... autor : "Fernando Boaglio",  
... genere : "Informàtica",  
... seccio : "Formació",  
... exemplars : 5,  
... districte : 3,  
... alta : new Date()  
... }  
> db.llibres.insert(llibre1)  
WriteResult({ "nInserted" : 1 })  
>
```

Si escric:

```
db.llibres.find();
```

Veig el document que acabo d'inserir a la meua Collection. La comanda `.find()` equival a un `SELECT * FROM llibres;` d'una BD SQL (relacional).

```
mongo Terminal
> db.llibres.find();
{ "_id" : ObjectId("629dbafa428d784390ac94dc"), "titol" : "MongoDB", "autor" : "Fernando Boaglio", "genere" : "Informàtica", "seccio" : "Formació", "exemplars" : 5, "districte" : 3, "alta" : ISODate("2022-06-06T08:28:22.892Z") }
>
```

Per visualitzar la informació d'una manera més estructurada i visual, es pot afegir la comanda `.pretty()`:

```
mongo Terminal
> db.llibres.find().pretty();
{
  "_id" : ObjectId("629dbafa428d784390ac94dc"),
  "titol" : "MongoDB",
  "autor" : "Fernando Boaglio",
  "genere" : "Informàtica",
  "seccio" : "Formació",
  "exemplars" : 5,
  "districte" : 3,
  "alta" : ISODate("2022-06-06T08:28:22.892Z")
}
```

Si escric:

```
show collections;
```

Comprovo que la Collection usuaris s'ha creat correctament:

```
mongo Terminal
> show collections;
llibres
>
```

El mateix per verificar si existeix la BD llibreria. Escriure:

```
show databases;
```

```
show dbs;
```

ò bé, per saber el nom de la BD actual:

```
db
```

```
mongo Terminal
> show databases;
admin      0.000GB
biblioteca 0.000GB
config     0.000GB
local      0.000GB
missatges  0.000GB
test       0.000GB
> show dbs;
admin      0.000GB
biblioteca 0.000GB
config     0.000GB
local      0.000GB
missatges  0.000GB
test       0.000GB
> db
biblioteca
>
```

Si s'han escrit totes les comandes bé, tot s'haurà creat correctament.

7. CRUD

Per portar la gestió de les dades emmagatzemades dins una BBDD, s'aplica un concepte anomenat **CRUD** (acrònim de l'anglès: Create, Read, Update i Delete), que engloba les quatre funcions principals que podem aplicar a qualsevol BBDD.

7.1. Crear / Inserir nous Documents (Create)

Podem inserir-los de forma individual o múltiple, segons definim una o més variables. Abans hem vist com inserir un sol *Document*, ara provarem una inserció múltiple definint varies variables per a cadascun:

```
var llibre2 = {  
  titol : "Cálculo I",  
  autor : "Fernando Boaglio",  
  genere : "Matemàtiques",  
  seccio : "Formació",  
  exemplars : 1,  
  districte : 6,  
  alta : new Date()  
}  
  
var llibre3 = {  
  titol : "Para Pensar Mejor",  
  autor : "Miguel de Guzman Ozamiz",  
  genere : "Matemàtiques",  
  seccio : "Formació",  
  exemplars : 8,  
  districte : 2,  
  alta : new Date()  
}
```

Inserim les dades dins la *Collection*:

```
db.llibres.insert( [llibre2, llibre3 ] )
```

```
mongo Terminal
> var exemplar2 = {
... titol : "Cálculo I",
... autor : "Mariano Soler Dorda",
... genere: "Matemàtiques",
... seccio : "Formació",
... alta : new Date()
... }
> var exemplar3 = {
... titol : "Para Pensar Mejor",
... autor : "Miguel de Guzman Ozamiz",
... genere : "Matemàtiques",
... seccio : "Formació",
... alta : new Date()
... }
> db.exemplars.insert([exemplar2, exemplar3])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
>
```

7.2. Mostrar / Llegir / Buscar Documents (Read)

Com ja s'ha comentat anteriorment, es poden visualitzar els Documents que continguin una Collection mitjançant la comanda `.find()`:

a. Per cerques genèriques, escriure:

```
db.llibres.find(); // SELECT * FROM llibres;
```

Com hem vist abans, per visualitzar la informació d'una manera més estructurada, afegim `.pretty()` a la comanda `.find()`:

```
db.llibres.find().pretty();
```

```
mongo Terminal
> db.llibres.find().pretty();
{
  "_id" : ObjectId("629dbafa428d784390ac94dc"),
  "titol" : "MongoDB",
  "autor" : "Fernando Boaglio",
  "genere" : "Informàtica",
  "seccio" : "Formació",
  "exemplars" : 5,
  "districte" : 3,
  "alta" : ISODate("2022-06-06T08:28:22.892Z")
}
{
  "_id" : ObjectId("629dbe68428d784390ac94dd"),
  "titol" : "Cálculo I",
  "autor" : "Fernando Boaglio",
  "genere" : "Matemàtiques",
  "seccio" : "Formació",
  "exemplars" : 1,
  "districte" : 6,
  "alta" : ISODate("2022-06-06T08:37:33.892Z")
}
{
  "_id" : ObjectId("629dbe68428d784390ac94de"),
  "titol" : "Para Pensar Mejor",
  "autor" : "Miguel de Guzman Ozamiz",
  "genere" : "Matemàtiques",
  "seccio" : "Formació",
  "exemplars" : 8,
  "districte" : 2,
  "alta" : ISODate("2022-06-06T08:43:50.436Z")
}
>
```

b. Per cerques específiques, separar les dades en dos blocs, seguint la següent estructura:

```
db.llibres.find(
{
  // bloc 1: indiquem condicions de cerca/filtratge
  genere : "Matemàtiques",
  autor:
  {
    $eq : "Fernando Boaglio"
  }
},
{
  // bloc 2: indiquem camps a mostrar
  titol : 1,      // mostra el títol del llibre
  autor: 1,       // mostra l'autor
  _id : 0         // especifiquem expressament que NO mostri l'id
}
)
```

```
mongo Terminal
> db.llibres.find( { genere : "Matemàtiques", autor : { $eq : "Fernando Boaglio" } }, { titol : 1, autor : 1, _id : 0 } ) ^
.pretty();
{ "titol" : "Cálculo I", "autor" : "Fernando Boaglio" }
>
```

Aquesta estructura equivaldria en SQL a indicar un seguit de camps a mostrar i/o a afegir una condició de tipus WHERE a la nostra consulta principal:

(SQL) Equivaldria a fer:

```
SELECT genere, titol FROM llibres WHERE autor='Fernando Boaglio';
```

7.3. Modificar / Actualitzar Documents (Update)

Per actualitzar dades, s'utilitza la comanda `.update()`.

Tal i com passa a l'hora de crear *Documents*, també es poden modificar de forma individual o múltiple.

A. Per modificar de forma genèrica diversos *Documents*, escriure:

```
db.llibres.update(  
  {}, // bloc 1: serveix per especificar una condició  
  {  
    // bloc 2: Indica camps a actualitzar  
    $set : {  
      estoc : 12,          // actualitzo l'estoc total d'exemplars  
      zona : "Terrassa" // afegeixo el camp 'zona'  
    }  
  },  
  {  
    // bloc 3: aplica sense restriccions a tots els documents que trobi  
    multi : true  
  }  
)
```

(SQL) Equivaldria a fer:

```
UPDATE llibres SET estoc=12;  
ALTER TABLE llibres ADD zona VARCHAR(50);  
UPDATE llibres SET zona='Terrassa';
```

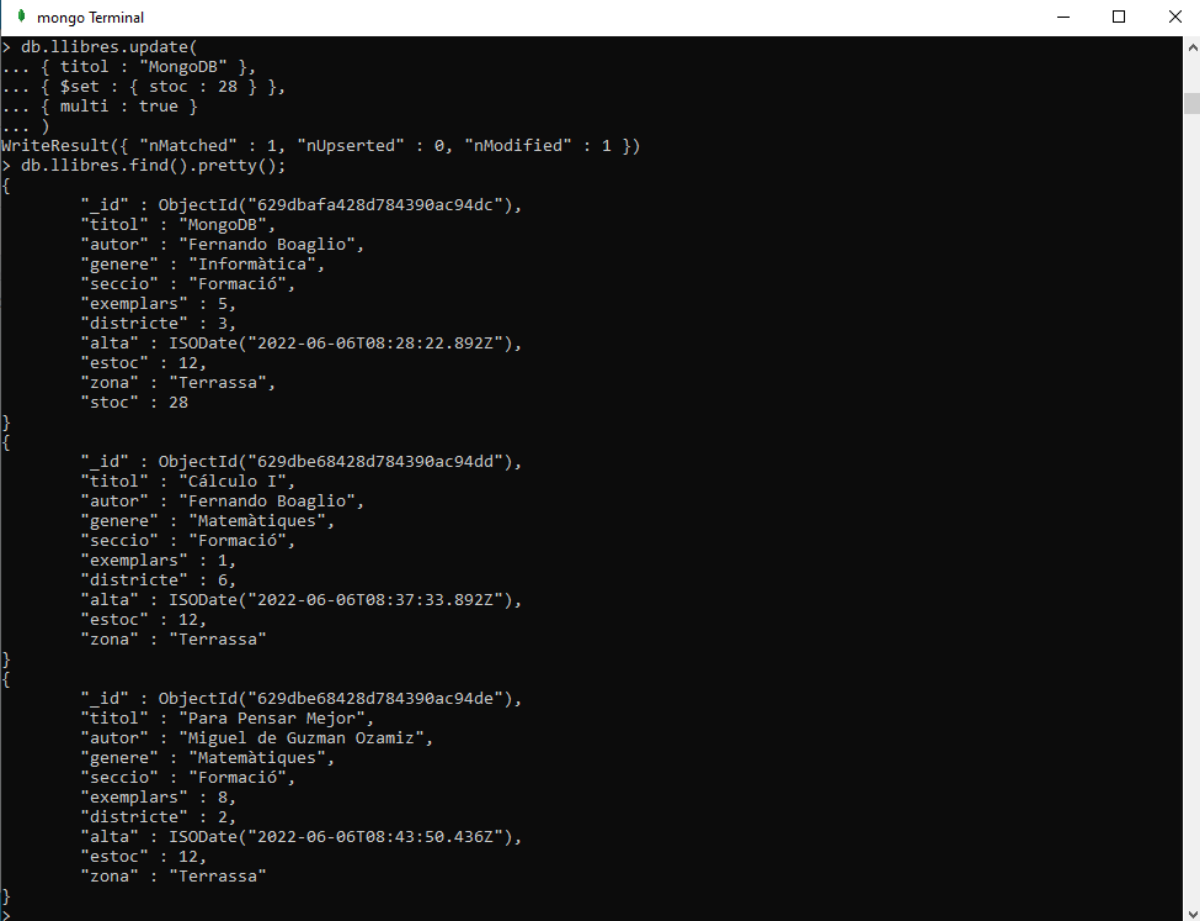
```
mongo Terminal
> db.llibres.update(
... {},
... { $set : {
...     estoc : 12, zona : "Terrassa"
...     }
... },
... { multi : true }
... )
WriteResult({ "nMatched" : 3, "nUpserted" : 0, "nModified" : 3 })
> db.llibres.find().pretty();
{
  "_id" : ObjectId("629dbafa428d784390ac94dc"),
  "títol" : "MongoDB",
  "autor" : "Fernando Boaglio",
  "gènere" : "Informàtica",
  "secció" : "Formació",
  "exemplars" : 5,
  "districte" : 3,
  "alta" : ISODate("2022-06-06T08:28:22.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
{
  "_id" : ObjectId("629dbe68428d784390ac94dd"),
  "títol" : "Cálculo I",
  "autor" : "Fernando Boaglio",
  "gènere" : "Matemàtiques",
  "secció" : "Formació",
  "exemplars" : 1,
  "districte" : 6,
  "alta" : ISODate("2022-06-06T08:37:33.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
{
  "_id" : ObjectId("629dbe68428d784390ac94de"),
  "títol" : "Para Pensar Mejor",
  "autor" : "Miguel de Guzman Ozamiz",
  "gènere" : "Matemàtiques",
  "secció" : "Formació",
  "exemplars" : 8,
  "districte" : 2,
  "alta" : ISODate("2022-06-06T08:43:50.436Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
}
```

B. Per fer-ho de forma individual, cal especificar una condició:

```
db.llibres.update(
{
  // bloc 1: serveix per especificar una condició
  títol : "MongoDB"
},
{
  // bloc 2: Indica camps a actualitzar
  $set : {
    estoc : 28 // actualitzo l'estoc total d'exemplars
  }
},
{
  // bloc 3: aplica sense restriccions a tots els documents que trobi
  multi : true
}
)
```

(SQL) Equivaldria a fer:

```
UPDATE llibres SET estoc=28 WHERE titol='MongoDB';
```



```
mongo Terminal
> db.llibres.update(
... { titol : "MongoDB" },
... { $set : { stoc : 28 } },
... { multi : true }
... )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.llibres.find().pretty();
{
  "_id" : ObjectId("629dbafa428d784390ac94dc"),
  "titol" : "MongoDB",
  "autor" : "Fernando Boaglio",
  "genere" : "Informàtica",
  "seccio" : "Formació",
  "exemplars" : 5,
  "districte" : 3,
  "alta" : ISODate("2022-06-06T08:28:22.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa",
  "stoc" : 28
}
{
  "_id" : ObjectId("629dbe68428d784390ac94dd"),
  "titol" : "Cálculo I",
  "autor" : "Fernando Boaglio",
  "genere" : "Matemàtiques",
  "seccio" : "Formació",
  "exemplars" : 1,
  "districte" : 6,
  "alta" : ISODate("2022-06-06T08:37:33.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
{
  "_id" : ObjectId("629dbe68428d784390ac94de"),
  "titol" : "Para Pensar Mejor",
  "autor" : "Miguel de Guzman Ozamiz",
  "genere" : "Matemàtiques",
  "seccio" : "Formació",
  "exemplars" : 8,
  "districte" : 2,
  "alta" : ISODate("2022-06-06T08:43:50.436Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
>
```

7.4. Eliminar Documents (Delete)

```
db.llibres.remove( { autor : "Miguel de Guzman Ozamiz" } )
// Elimina tots els llibres que de l'autor especificat
```

(SQL) Equivaldria a fer:

```
DELETE FROM llibres WHERE autor="Miguel de Guzman Ozamiz";
```

```
mongo Terminal
> db.llibres.remove( { autor : "Miguel de Guzman Ozamiz" } )
WriteResult({ "nRemoved" : 1 })
> db.llibres.find().pretty();
{
  "_id" : ObjectId("629dbafa428d784390ac94dc"),
  "titol" : "MongoDB",
  "autor" : "Fernando Boaglio",
  "genere" : "Informàtica",
  "seccio" : "Formació",
  "exemplars" : 5,
  "districte" : 3,
  "alta" : ISODate("2022-06-06T08:28:22.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa",
  "stoc" : 28
}
{
  "_id" : ObjectId("629dbe68428d784390ac94dd"),
  "titol" : "Cálculo I",
  "autor" : "Fernando Boaglio",
  "genere" : "Matemàtiques",
  "seccio" : "Formació",
  "exemplars" : 1,
  "districte" : 6,
  "alta" : ISODate("2022-06-06T08:37:33.892Z"),
  "estoc" : 12,
  "zona" : "Terrassa"
}
>
```

```
db.llibres.remove( {} )
// Elimina tots els documents de tipus llibre
```

(SQL) Equivaldria a fer:
DELETE FROM llibres;

```
mongo Terminal
> db.llibres.remove({})
WriteResult({ "nRemoved" : 2 })
> db.llibres.find();
>
```

7.5. Eliminar Collections (Delete)

```
db.llibres.drop()
// Elimina la col·lecció llibres
```

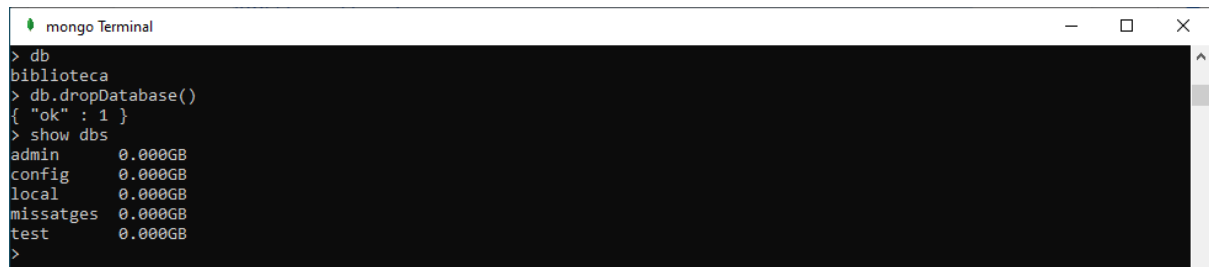
(SQL) Equivaldria a fer:
DROP TABLE llibres;

```
mongo Terminal
> show collections;
llibres
> db.llibres.drop()
true
> show collections;
>
```

7.6. Eliminar BD (Delete)

```
db.dropDatabase()  
// Elimina la base de dades actual (biblioteca)
```

(SQL) Equivaldria a fer:
DROP DATABASE biblioteca;



```
mongo Terminal  
> db  
biblioteca  
> db.dropDatabase()  
{ "ok" : 1 }  
> show dbs  
admin      0.000GB  
config     0.000GB  
local      0.000GB  
missatges  0.000GB  
test       0.000GB  
>
```

8. Webgrafia

[1] Amazon. ¿Qué es NoSQL?. Recuperat 5 de juny de 2022, des de <https://aws.amazon.com/es/nosql/>

[2] Jorge Rubio. (2021). MongoDB. Recuperat 5 de juny de 2022, des de <https://git.copernic.cat/jorge.rubio/m02/blob/master/M02UF4.adoc>