

Licenciatura en Sistemas de Información
Bases de Datos NSQL

PRACTICA #4

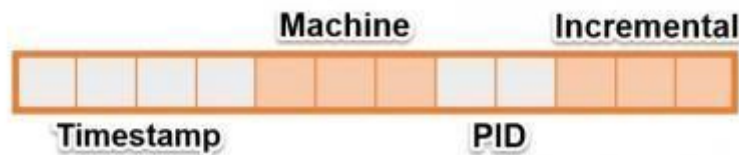
Parte #1 – Introducción a MongoDB

La estructura de almacenamiento de MongoDB está compuesta de 4 elementos básicos, Bases de Datos, Colecciones, Documentos y Campos; y 2 componentes avanzados índices y cursores. Una vez montado el contenedor de MongoDB en Docker, pasemos a ejecutar algunas actividades básicas (en el siguiente link se encuentran toda la documentación de mongo <https://docs.mongodb.com/manual/reference/>):

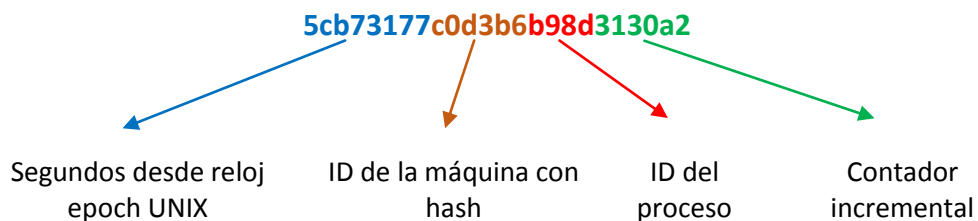
1. Primero veamos la ayuda de mongo `db.help()`, no dará ayuda de los comandos básicos.
2. Obtenga la versión del servido con el comando `db.version()`.
3. Obtenga la estadísticas de la base de datos pruebas con `db.stats()`.
4. Luego veamos las bases de datos que tenemos creadas con el comando: `show dbs`.
5. Ahora utilice el comando `use db_prueba`, para cambiarse a la base de datos `db_prueba`, si la misma no está creada, mongo la creará automáticamente.
6. Obtenga el nombre de las colecciones que están en dicha base de datos, utilizando el comando `db.getCollectionNames()`.
7. Ahora inserte un nuevo documento "elemento" en la colección lenguajes (si la colección no está crea se crea automáticamente al igual que la base de datos), usando el comando `db.lenguajes.insert({'nombre':'python', 'version':'3.8.2'})`.
8. Ahora vuelva a ejecutar el punto 6 y debería obtener el nombre de la nueva colección.
9. Luego inserte un nuevo documento con distinta estructura de campos, por ejemplos `db.lenguajes.insert({'nombre':'java', 'extension':'.jar', 'simbolo':'taza de café'})`.
10. Para tener algunos datos más para poder trabajar pegue lo siguiente en la terminal,
`db.lenguajes.insert({'nombre':'python', 'version':'2.7.9', 'proyectos':13, 'frameworks':['flask', 'django']});`
`db.lenguajes.insert({'nombre':'javascript', 'version':'4.5', 'extension':'.js', 'frameworks':['react', 'angular', 'vue.js', 'express.js']});`
`db.lenguajes.insert({'nombre':'pascal', 'version':'?', 'proyectos': 0});`
`db.lenguajes.insert({'nombre':'HTML', 'version':'5'});`
`db.lenguajes.insert({'nombre':'C++', 'version':'3.4.6', 'simbolo':'C+', 'extension':['.h', '.cpp']});`
`db.lenguajes.insert({'nombre':'Android', 'version':'Lollipop', 'extension':'.apk'});`
`db.lenguajes.insert({'nombre':'Arduino', 'version':'1.8.7', 'extension':'.ino', 'observaciones':'el lenguaje en el fondo es java :) '});`
`db.lenguajes.insert({'nombre':'C'});`
`db.lenguajes.insert({'nombre':'Rubi', 'simbolo':'Rubi rojo'});`
`db.lenguajes.insert({'nombre':'GO', 'version':'1.10', 'año':2009});`
11. Para acceder a las distintas colecciones puede hacerlo de las siguientes maneras `db.getCollection('nombre')`, `db['nombre']` o `db.nombre`.
12. Obtengamos la cantidad de documentos de una colección usando el comando `db.coleccion.count()`.
13. Ahora veamos cómo obtener todos los documentos de una colección, para esto utilizamos `db.lenguajes.find({})`, el comando `find({})` nos permitirá realizar consultas muy potentes sobre las colecciones.

Licenciatura en Sistemas de Información Bases de Datos NSQL

14. Formatee los datos de salida para que sean más legibles usando `db.lenguajes.find({}).pretty()`.
15. Observe que todos los documentos creados tienen un atributo extra que se llama `_id` con el siguiente formato "5cb73177c0d3b6b98d3130a2". Este objectid tiene es un identificador único para cada documento y sigue el siguiente formato:



Por lo tanto nuestro `_id` queda compuesto de la siguiente manera:



Comencemos con consultas básicas utilizando el comando `find({})`

16. Obtengamos los lenguajes cuyo nombre sea python de la siguiente manera `db.lenguajes.find({nombre: 'python'})`. Pruebe con otros nombres y campos.
17. Observe que si ejecuta el siguiente comando `db.lenguajes.find({extension: '.ino'})` no todos los documentos tienen el campo extensión y sin embargo funciona sin problemas.
18. Si queremos consultar por dos campos utilizando un `and` lo podemos hacer de la siguiente manera en forma implícita `db.lenguajes.find({nombre: 'python ', version: '3.4.6'})`.
19. Si queremos consultar por dos campos utilizando un `or` lo podemos hacer de la siguiente manera de forma explícita con el operador `$or` como se observa a continuación `db.lenguajes.find({$or : [{nombre: 'python'}, {version: '3.4.6'}])`.
20. Ahora que determinar todos los documentos que tienen el campo versión para esto usaremos el operador `$exists` de la siguiente manera `db.lenguajes.find({version : {$exists : true}})`.
21. Ahora liste los que no tienen el campo extensión.
22. Además tenemos los operadores `$lt` (less than), `$lte` (less than equal), `$gt` (greater than), `$gte` (greater than equal), `$ne` (not equal), `$eq` (equal), `$not` (not), `$in` (value in array) y `$nin` (value not in array), veamos un poco como funcionan: obtengamos los lenguajes que tiene más de 7 proyectos con el comando `db.lenguajes.find({proyectos : {$gt : 7}})`, ahora obtenga los que tiene 1 o menos proyectos. Determinar si algún lenguaje tiene el framework Vue.js `db.lenguajes.find({frameworks : {$in : ['vue.js']})` ahora determinar si alguno tiene el framework angular o flask.
23. Ahora supongamos que solo queremos ver el nombre de los lenguajes para eso debemos filtrar los atributos que queremos mostrar de la siguiente manera

Licenciatura en Sistemas de Información Bases de Datos NSQL

- db.lenguajes.find({}, {nombre:1}), si queremos mostrar otros campo lo seguimos agregando y aumentando su número db.lenguajes.find({}, {nombre:1, version:2}).
24. Como pudo observar en el ejercicio anterior por defecto se muestra el campo `_id` si lo queremos quitar debemos hacerlo de la siguiente manera `db.lenguajes.find({}, {_id:0, nombre:1, version:2})`.
 25. Ahora veamos como actualizar un valor par a esto utilizaremos el operador `$set`, por ejemplo si queremos modificar la versión de Arduino a "1.8.9" debemos hacerlo de la siguiente manera con el comando `update db.lenguajes.update({nombre: 'Arduino'}, {$set : {version: '1.8.9'}})`, luego verifique que se haya actualizado.
 26. Ahora supongamos que queremos indicar la cantidad de proyectos que tenemos en Arduino, pero este documento no tiene este campo como hacemos para agregarlo; esto lo podemos solucionar con el siguiente comando `db.lenguajes.update({nombre: 'Arduino'}, {$set : {proyecto: 7}})`.
 27. Ahora nos dimos cuenta de que agregamos mal el nombre del campo era proyectos, lo podemos modificar con el comando `$rename` de la siguiente manera `db.lenguajes.update({nombre: 'Arduino'}, {$rename : {'proyecto': 'proyectos'}})`.
 28. Ahora queremos filtrar todos los documentos que en el campo nombre contengan una u para esto usaremos el operador `$regex` de la siguiente manera `db.lenguajes.find({nombre: {$regex : 'u'}})`.
 29. Ahora queremos filtrar todos los documentos que en el campo nombre contengan empiecen con A (recuerde que es case sensitive) de la siguiente manera `db.lenguajes.find({nombre: {$regex : '^A'}})`.
 30. Ahora queremos filtrar todos los documentos que en el campo nombre terminan con d de la siguiente manera `db.lenguajes.find({nombre: {$regex : 'd$'}})`.
 31. Ahora necesitamos agregar un dato a una lista para esto usamos `$push` de la siguiente manera `db.lenguajes.update({nombre: 'javascript'}, {$push : {frameworks: 'nodejs'}})`.
 32. Si queremos eliminar un valor de una lista podemos utilizar el operador `$pop` de la siguiente manera `db.lenguajes.update({nombre: 'nodejs'}, {$pop : {frameworks: -1}})`, esto elimina el primero elemento de la lista, elimine el ultimo cambiándole el valor a 1.
 33. Ahora tenemos que ordenar los datos ascendente de la consulta para eso usaremos el comando `sort()` de la siguiente manera `db.lenguajes.find({}, {_id:0, nombre:1, version:2}).sort({nombre : 1})`.
 34. Ahora liste de manera descendente cambiando el valor 1 por -1.
 35. Ahora ordene de acuerdo a dos campos por nombre de manera ascendente y por versión de manera descendente de la siguiente manera `db.lenguajes.find({}, {_id:0, nombre:1, version:2}).sort({nombre : 1, version:-1})`.

Licenciatura en Sistemas de Información Bases de Datos NSQL

Parte #2 – En una galaxia muy lejana...

Ahora vamos a importar dos archivos JSON para poder trabajar en mongodb, los archivos son episodios.json y personajes.json para poder importarlos debemos ejecutar los siguientes comandos (puede encontrar más datos en <https://swapi.co/>):

1. Debemos copiar el archivo al contenedor en la carpeta temporal
docker cp /home/walter/Descargas/personajes.json mongodb:/tmp/personajes.json
2. Ahora importe el archivo a la base de datos con el siguiente comando
docker exec -it mongodb mongoimport --db star_wars --collection personajes --drop --file /tmp/personajes.json
3. Ahora acomodar los datos de los valores del campo films de cada personajes dejando solo "Episodio ¿ - nombre del episodio", especie y homeworld
4. Elimine los campos vehicles, starships, created, edited
5. cambie el nombre del campo url a id_personaje y deje solo el número el personaje

Ahora resuelva las siguientes consultas:

6. Cuantos documentos hay en cada colección
7. Muestre el texto de la introducción del episodio III
8. Liste todos los personajes de manera ascendente solo mostrar su nombre, edad, año de nacimiento, mundo origen
9. Muestre toda la información de Darth Vader y Kylo Ren
10. Muestre los personajes hayan nacido antede del año 880 BBY (Before Battle of Yavin)
11. Muestre los personajes de menos de 80 cm de alto y hasta 40 kilos
12. Liste todos los personajes que comiencen con la letra C de manera descendente
13. Todos los personajes que contengan la letra a en su nombre
14. Todos los personajes que aparecen en el episodio III
15. Todos los personajes que aparecen en el episodio III y VI
16. Los personajes que aparecen en todos los episodio
17. Todos los personajes que son de la especie 2
18. Los episodios en los que aparecen los Luke Skywalker, C-3PO, R2-D2, Han Solo, Rey, Chewbacca
19. Muestre los datos del personaje 79, 85 y 88
20. Listar los personajes que cominezan con la letra P y F