**Laboratorio de computación 4: TP2 individual teoría MongoDB**

**Alumno :** Damián Agüero

6-Describa que sucede al ejecutar el método drop() sobre una colección y sobre una base de datos.

El método drop() se utiliza en MongoDB para eliminar una colección o una base de datos completa. Se puede ejecutar sobre una colección y sobre una base de datos:

* Sobre una colección: elimina completamente esa colección y todos sus documentos asociados. Después de llamar a drop() en una colección, la colección ya no existe en la base de datos y no se puede acceder a ella. Los índices y los metadatos asociados con esa colección también se eliminan.

Por ejemplo, con la colección "paises", al ejecutar “coleccion\_paises.drop()” , eliminaríamos completamente la colección "paises" de la base de datos.

* Sobre una base de datos: elimina completamente toda la base de datos y todas las colecciones que contiene. Después de llamar a drop() en una base de datos, la base de datos ya no existe en el servidor MongoDB y todos sus datos se pierden permanentemente.

Por ejemplo, con la base de datos "paises\_db", al ejecutar “db.drop\_database()”, eliminaríamos completamente la base de datos "paises\_db" y todas sus colecciones.

9-Describa que sucede al ejecutar el método skip() sobre una colección. Ejemplifique con la colección países.

El método skip() omite un número específico de documentos al consultar una colección. Permite "saltar" los primeros documentos según la cantidad especificada y recuperar los siguientes documentos de la colección.

Cuando se ejecuta skip(n) sobre una colección en MongoDB, la consulta resultante omite los primeros n documentos de la colección y devuelve el resto de los documentos. Esto es útil al paginar resultados u omitir documentos específicos al realizar una consulta.

Por ejemplo podemos usar el siguiente código en Python para “skipear” los primero 3 países e imprimir por consola los datos obtenidos de los países que les suceden.

from pymongo import MongoClient

cliente = MongoClient('localhost', 27017)

db = cliente['paises\_db']

coleccion\_paises = db['paises']

documentos = coleccion\_paises.find().skip(3)

# Itera sobre los documentos obtenidos y los imprime

for documento in documentos:

print(documento)

10-Describa y ejemplifique como el uso de expresiones regulares en Mongo puede reemplazar el uso de la cláusula LIKE de SQL.

En MongoDB, las expresiones regulares se utilizan para realizar consultas basadas en patrones de texto en los campos de los documentos. Esto es similar al uso de la cláusula LIKE en SQL para buscar valores que coincidan con un patrón específico en una columna de una tabla. Sin embargo, la sintaxis y la funcionalidad pueden diferir ligeramente.

Supongamos que tenemos una colección llamada empleados en MongoDB, que contiene documentos con información sobre los empleados. Queremos buscar todos los empleados cuyo nombre comience con "J".

En SQL, utilizaríamos la cláusula LIKE de la siguiente manera:

SELECT \* FROM empleados WHERE nombre LIKE 'J%';

En MongoDB, podemos lograr lo mismo utilizando expresiones regulares:

empleados\_con\_J = coleccion\_empleados.find({"nombre": {"$regex": "^J"}})

for empleado in empleados\_con\_J:

print(empleado)

Utilizamos $regex en la consulta para indicar que queremos aplicar una expresión regular al campo "nombre".

El patrón ^J coincide con cualquier nombre que comience con la letra "J".

La consulta recuperará todos los documentos donde el campo "nombre" comience con "J".

11-Investigue createIndex()

createIndex() es un método en MongoDB que se utiliza para crear un índice en una colección específica. Los índices en MongoDB son estructuras de datos que mejoran la eficiencia de las consultas al permitir que MongoDB recupere y ordene los documentos de manera más rápida.

Cuando se ejecuta createIndex(), se debe especificar el campo o los campos de la colección a indexar. Se pueden crear índices simples o compuestos, dependiendo de las necesidades.

Por ejemplo, para crear un índice en la colección "paises" en MongoDB con el campo "nombre" como índice:

coleccion\_paises.create\_index("nombre")

12-Describa como se realiza un backup de la base de datos mongo países\_db.

Para realizar un backup de la BD “países\_db” hay dos métodos:

* Utilizando este comando en el cmd:

mongodump --db=paises\_db out=C:\XXXXX\XXXXX\XXXXXX\backup

Se debe especificar el nombre de la base de datos a respaldar, en este caso se crea un backup de la base de datos "países\_db" en el directorio especificado.

* Utilizando un Snapshot de los Datos:

Podemos utilizar las herramientas proporcionadas por el sistema de archivos (como ZFS o LVM) para crear un snapshot del volumen o del almacenamiento donde se encuentran los archivos de datos de MongoDB y luego copiar los archivos de datos del snapshot a un lugar seguro como una unidad de almacenamiento externa o una ubicación de red.