

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

Lo que sucede al ejecutar el método drop() sobre una colección o una base de datos es que se elimina por completo.

No es recomendable hacerlo en producción, obviamente, al menos que tengamos un backup y podamos recuperar la información en caso de que la volvamos a necesitar.

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

Al ejecutar el método skip() sobre una colección es indicarle a la misma que se saltee la cantidad de documentos que le indicamos por parámetro.

Es decir, si tenemos `países_db.países.find()`, ese comando en MongoDB nos traerá todos los países que estén en dicha colección, pero si ponemos `skip()` al final de la sentencia, con algún parametro numerico, lo que hará es saltar la cantidad de países como el parametro lo indica.

Por ejemplo: `países_db.países.find().skip(2)` saltará los primeros dos países.

`países_db.países.find().skip(-1)` saltará el último país y así sucesivamente dependiendo el parámetro indicado.

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

Teniendo insertado lo siguiente en una colección llamada test:

```
db.test.insert("correo" : "correo1_1@gmail.com")
```

```
db.test.insert("correo" : "correo2_2@gmail.com")
```

```
db.test.insert("correo" : "correo3_3@gmail.es")
```

Podemos realizar búsquedas por patrones, para ello tenemos un equivalente a LIKE de SQL:

```
db.test.find( { correo : /@/ } )
```

O también podemos olvidarnos de los backslash y utilizar regex (Regular expression) que deja más claro que estamos utilizando expresiones regulares.

```
db.test.find( { correo : { $regex : "@" } } )
```

Con esto vamos a buscar todos los correos que contengan una arroba entre caracteres.

```
db.test.find( { correo : { $regex : ".com$" } } )
```

Nos enseñará todos los correos que acaben en el patrón puesto anteriormente. Hay que tener en cuenta que el símbolo \$ se utiliza como delimitador, es decir dirá si atrás puede haber más caracteres o no.

Así sucesivamente dependiendo de la expresión regular que tengamos.

5.12. Describa como se realiza un backup de la base de datos mongo países_db.

Respaldo y recuperación de base de datos:

Respaldo:

Abrimos una nueva terminal para realizarle un respaldo:

```
mongodump --db tp3_laboratorio
```

Nos realiza el respaldo de la base de datos tp3_laboratorio en el directorio que tenía la terminal.

Todo se guardará en BSON.

Si perdiéramos nuestra base de datos, haciendo por ejemplo:

```
db.dropDatabase()
```

La recuperaríamos haciendo uso del respaldo de la siguiente manera:

```
mongorestore --db tp3_laboratorio dump/tp3_laboratorio
```

El último parametro (dump/tp3_laboratorio) es la ruta donde está el backup

Vamos a mongo:

```
show databases;
```

Veremos que se recuperó la base de datos tp3_laboratorio con sus colecciones.

Respaldo y recuperación de una colección:

Respaldo:

```
mongodump --collection nombreDeLaColeccion --db tp3_laboratorio
```

Nos respalda la coleccion indicada de la base de datos tp3_laboratorio.

Recuperación:

```
mongorestore --collection nombreDeLaColeccion --db tp3_laboratorio  
dump/tp3_laboratorio/nombreDeLaColeccion.bson
```