Mongo DB - CRUD
(Create, Read, Update, Delete)

Para todos -> db.getCollection(“ “)

- INSERCCIÓN

Esta instrucción ***insertará un documento*** *con un campo “_id” en caso de no existir, y el campo que se está incluyendo adicionalmente.*

insert({“titulo”: “El Quijote”})

Cuando es necesario ***insertar un conjunto de documentos*** *se pueden pasar como parámetro un array con el conjunto de documentos que deben ser insertados. Esto funciona siempre y cuando se quieran insertar todos en la* ***misma colección.***

insert([***{“_id”: 0}, {“_id”: 1}, {“_id”: 2}*** **])**

- BORRADO

El método ***remove*** ***elimina todos los documentos de una colección****, pero no elimina la colección ni la metainformación acerca de ella.*

remove({ })

Permite además eliminar todos los documentos que encajen con una condición dada:

- Eliminar todos los documentos de la colección “Persona” dónde el valor para el campo “salida” es cierto

  db.getCollection(“Persona”).remove({“salida”: true})

A veces si se va a borrar todos los documentos, es más rápido borrar toda la colección:

drop( )

- ACTUALIZACIÓN

*El método **update** toma dos parámetros:*

- *Condición de búsqueda*
- *Conjunto de cambios a realizar*

El tipo de actualización más simple consiste en reemplazar un documento por otro. Por ejemplo, si se quiere cambiar el siguiente documento:

```
{
        "nombre" : "Juan",
        "amigos" : 32,
        "enemigos" : 2
 }
```

Y se quiere crear un campo "relaciones" que englobe a los campos "amigos" y "enemigos", se podría realizar con un update:

```
var juan = db.getCollection("Persona").findOne( {"nombre": "juan"} );

juan.relaciones = {"amigos": juan.amigos, "enemigos": juan.enemigos};
juan.PrimerNombre = juan.nombre

delete juan.amigos
delete juan.enemigos
delete juan.nombre

db.getCollection("Persona").update( {"nombre": "Juan"}, juan );
```

IMPORTANTE LOS PUNTOS Y COMAS

OPERADORES DE MODIFICACIÓN:

- ***$inc:*** *Permite cambiar el valor numérico de una clave que ya existe incrementando su valor por el especificado junto al operador, o bien puede crear una clave que no existía inicializándola al valor dado.*

Ejemplo:
{
"URL": "www.ejemplo.es",
"visitas": 35
}

Cada vez que se visite la página se actualizarán las visitas en uno:

update(*{"URL": "www….."}, {"$inc": {"visitas": 1} }*)

También se podría decrementar usando números negativos.
Si "visitas" no hubiese existido, se crea como un nuevo campo.

- ***$set y $unset:*** *Este valor establece un valor para un campo dado, y si el campo dado no existe entonces lo crea. Es útil para modificar el esquema de un documento o añadir claves definidas por el usuario.*

  Ejemplo:

  Añadir un campo sobre un libro favorito a un usuario con _id = 2 :

  update({"_id": 2}, {"$set": *{"libroFavorito": "Guerra y Paz"}* })

  También se podría cambiar el valor de "libroFavorito" por un array ó cambiar el valor de un documento embebido.

  "$unset" *se encarga de eliminar campos de un documento.*

- ***$push:*** *Añade elementos al final del array si existe o bien crea uno nuevo si no existe.*

  update({"Titulo": "posts de un blog"}, {"$push": {"comentarios:
  {"nombre": "juan",
  "email": "juan@ejemplo.com"
  "contenido": "buen post" }
  } })

- ***$each:*** *Se usa junto con* ***"$push"*** *para añadir más de un elemento al array:*

  update(… , { "$push": *{ "horas":* { "$each": [23, 24, 12] } *}* })

- ***$sort:*** *Permite ordenar los elementos indicando el campo de ordenación y el criterio en forma de 1 (ascendente) o -1 (descendente). Debe aparecer junto con un* ***$each.*** *Se escribe despues del array de $each, seguido de una coma:*

  update(… , { "$push": *{ "horas":* { "$each": [23, 24, 12], $sort: {"campo": 1} } } })

  Añadir a un array un elemento si no existe previamente: ***$ne***

  update({"autores citados": *{"$ne": "Pepe"}* }, {"$push": {"autores citados": "Pepe"} })

- ***$pop:*** *Borra elementos de un array, tratando a este como si fuera una cola o una pila. Si toma el valor 1, borra elementos del final. Si toma el valor -1, los borra del principio.*

  update({"_id": 1}, {"$pop": *{scores : 1}* })

  Modificar un elemento en concreto del array: Acceder a él mediante su posición:
  Si tenemos un array llamado comentarios y queremos acceder al primer elemento se accedería mediante: ***comentarios.0***