

---

---

## Commandes MongoDB

---

---

### 1. Operations CRUD

#### a) Opérations sur la base de donnée

Syntaxe	Sémantique de la commande
show dbs	Connaître les Bases stockées dans le répertoire géré par le serveur
db	Afficher le nom de la base actuelle
use <nomBase>	Basculer à une autre base, si cette dernière n'est pas créée elle le sera à la première création de collection
db.dropDatabase()	Effacer la base courante, la base est effacée sur disque mais continue d'être référencée

Tableau 1: Commandes MongoDB des opérations CRUD (bases de données)

#### b) b Opérations sur les collections

Syntaxe	Sémantique de la commande
db.createCollection(<nomCollection>);	Créer une collection
show collections ;	Lister les collection
db.<nomCollection>.rename(<NouveauNom>)	Renommer la collection
db.<nomCollection>.drop( )	Supprimer la collection

Tableau 2: Commandes MongoDB des opérations CRUD sur les collections

#### c) Opérations sur les documents d'une collection

Syntaxe	Sémantique de la commande
db.<nomCollection>.insert({<attribut1>:<valeur1>,<attribut2>:<valeur2>,...})	Insérer un document dans la collection, le paramètre de la fonction insert doit être un document
db.<nomCollection>.insertMany([{<attribut1>:<valeur1>,...},{<attribut2>:<valeur2>,...}])	Insérer plusieurs documents dans la collection
db.<nomCollection>.find()	Afficher les données des documents

db.<nomCollection>.update({<attribut1>:<valeur1>,...},{<attribut1>:<valeur1>,...})	Le premier paramètre de la commande update permet de sélectionner le document selon les valeurs des couples attributs :valeurs, le deuxième permet de modifier en écrasant le document par les nouvelles données en gardant le même ID du document modifié.
db.collection.updateMany()	Ajouter et/ou modifier un champ à plusieurs documents
<b>Opérateurs sur les attributs du Update</b>	
db.<nomCollection>.update({},{\$set:{<attribut1>:<valeur1>,...}})	Modifier et/ou rajouter un champ au document (opérateur \$set)
db.<nomCollection>.update({},{\$unset:{<attribut1>:""}})	Supprimer un attribut.
db.<nomCollection>.update({},{\$rename:{<attribut1>:<nouveauNom1>,...}})	Renommer un attribut.
<b>Opérateurs sur l'attribut tableau du Update</b>	
db.<nomCollection>.update({},{\$set:{<attribut>:<valeur>}})	<attribut> est un champ de type array il faudra utiliser la notation pointée pour représenter l'indice du tableau et modifier un élément EXP : attribut.0 est le premier élément du tableau.
db.<nomCollection>.update({},{\$addToSet:{<attribut1>:<valeur1>,...}})	L'opérateur \$addToSet permet d'ajouter sans doublon
db.<nomCollection>.update({},{\$push:{<attribut1>:<valeur1>,...}})	Rajouter les blègements à la fin du tableau
db.<nomCollection>.update({},{\$pull:{<attribut1>:<value1   condition>,...}})	Supprimer les valeurs d'un tableau
db.<nomCollection>.remove()	Pour supprimer tous les documents de la collection db.collection.remove(), sinon il faudra spécifier un filtre entre ()

Tableau 3: Commandes Mongodb des opérations CRUD sur les documents d'une collection

## 2. Interrogation des données

L'interrogation des données se fait à l'aide de la fonction find() de la commande :

```
db.<collectionName>.find ( { ... } , { ... } )
```

La commande contient deux arguments :

- le premier argument est la condition de sélection/filtrage où seuls les documents satisfaisant cette condition seront retenus ;
- Le second argument décrit la projection sur les documents retenus

c'est l'équivalent en sql :

```
SELECT <attributs du deuxième augment> FROM collectionName  
WHERE <conditions du premier paramètre>
```

Un ensemble d'opérateurs (voir tableau4) peuvent être utilisés dans la projection ou sélection pour écrire les conditions de la requête.

Opérateur	Sémantique de l'opérateur
<b>Logiques</b>	
\$and, \$or	Et logique, ou logique
\$not, \$nor	NOT et NOR logique
<b>Comparaisons</b>	
\$eq, \$ne	==, !=
\$gt, \$gte, \$lt, \$lte	>, >=, <, <=
<b>Test sur éléments</b>	
\$exists	Existence d'un champ
\$type	Teste le type d'un champ
<b>Évaluation</b>	
\$mod	Calcule et teste le résultat d'un modulo
\$regex	Recherche d'expression régulière
\$text	Analyse d'un texte
<b>Tableaux</b>	
\$all	Test sur le contenu
\$elemMatch	Limite le contenu d'un tableau à partir des résultats de la requête pour ne contenir que le premier élément correspondant à la condition \$elemMatch.
\$size	Taille du tableau

Tableau 4: Liste des opérateurs (requête de sélection/projection) de la commande find

### 3 Framework d'agrégation

Depuis la version 2.2, MongoDB propose un framework d'agrégation qui permet de :

- réaliser des opérations complexes d'analyse ;
- récupérer et manipuler les données en pipeline dans mongoDB sans utiliser des traitements complexes en batch avec Map/Reduce ??

```
db.<collectionName>.aggregate([<Étape1>], <Étape2>], <Étape3>]..])
```

Dans chaque étape, un opérateur (tableau 5) est utilisé

Opérateur	Sémantique
\$Group	Permet de regrouper les enregistrements selon l'_id (qui peut être re- défini), et d'appliquer des fonctions de groupe (\$sum, \$min, \$avg,etc.) aux autres attributs projetés.
\$match	Permet de sélectionner/filtrer les document à passer au pipeline, selon une ou des conditions.
\$sort	Permet de trier les documents et de les transmettre de façon ordonnée au pipe- line
\$project	Permet de mettre en forme les documents avec les champs existants ou de nouveaux champs, avant de les passer au pipeline
\$unwind	Permet de réorganiser les champs d'un tableau, avec un document pour chaque élément du tableau.
\$out	Permet de créer une collection avec le résultat d'un pipeline
\$lookup	Permet de faire une "jointure" entre des objets de deux collections

Tableau 5: Quelques opérateurs du framework d'agrégation