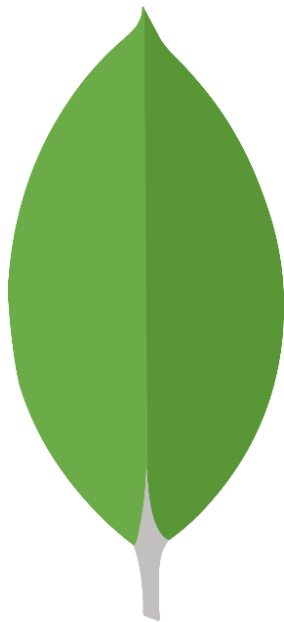


# Mémento

# MongoDB (NoSQL)

Version 1.3 (créé le 14/03/2024, modifié le XX/XX/XXXX)



NoSQL (No Structured Query Language) est un langage servant à exploiter des bases de données non-relationnelles.

MongoDB est un système de gestion de base de données orienté documents qui permet de manipuler des objets structurés au format JSON.



**Loric Informatique**

# Table des matières

|   |                                    |
|---|------------------------------------|
| 1. Prise en main.....   | 4                                  |
| 1.1. Outils nécessaires .....   | 4                                  |
| 1.2. Installer MongoDB .....  | 4                                  |
| 1.3. Démarrer MongoDB .....   | 4                                  |
| 1.4. Arrêter proprement MongoDB.....  | 4                                  |
| 2. Bases .....  | 5                                  |
| 2.1. Syntaxe dans le fichier json.....  | 5                                  |
| 2.2. Commentaires dans un fichier json .....  | <b>Erreur ! Signet non défini.</b> |
| 2.3. Les conditions .....   | 5                                  |
| 3. Requêtes d'initialisation de la base de données.....   | 5                                  |
| 3.1. Utiliser une base de données (ou la créer si elle n'existe pas) pour<br>réaliser des requêtes..... | 5                                  |
| 3.2. Afficher les bases de données.....   | 6                                  |
| 3.3. Supprimer une base de données.....   | 6                                  |
| 3.4. Afficher les collections d'une base de données.....  | 6                                  |
| 3.5. Supprimer une collection d'une base de données .....   | 6                                  |
| 3.6. Importer une collection (avec mongoimport).....  | 6                                  |
| 4. Requêtes de mise à jour de la collection .....   | 6                                  |
| 4.1. Insérer un document .....  | 6                                  |
| 4.2. Insérer plusieurs documents .....  | 7                                  |
| 4.3. Mettre à jour un document.....   | 7                                  |
| 4.3.1. Mise à jour simple.....  | 7                                  |
| 4.3.2. Faire incrémenter un document.....   | 7                                  |
| 4.3.3. Mettre à jour un tableau.....  | 7                                  |
| 4.3.4. Supprimer un champ.....  | 9                                  |

|  |                                    |
|--|------------------------------------|
| 4.4. Mettre à jour plusieurs documents.....                | 9                                  |
| 4.4.1. Mise à jour simple .....                            | 9                                  |
| 4.4.2. Faire incrémenter plusieurs documents.....          | 9                                  |
| 4.4.3. Mettre à jour un tableau.....                       | 9                                  |
| 4.4.4. Supprimer un champ.....                             | 10                                 |
| 4.4.5. Mises à jour multiples .....                        | 11                                 |
| 4.5. Supprimer un document.....                            | 11                                 |
| 4.6. Supprimer plusieurs documents .....                   | 11                                 |
| 5. Requêtes de sélection de données .....                  | 11                                 |
| 5.1. find.....   | 11                                 |
| 5.1.1. Afficher toute la collection.....                   | 11                                 |
| 5.1.2. Afficher certains champs.....                       | 11                                 |
| 5.1.3. Afficher certains champs d'une liste .....          | 12                                 |
| 5.1.4. Afficher les documents respectant la condition..... | 12                                 |
| 5.1.5. Limiter le nombre de documents à afficher.....      | <b>Erreur ! Signet non défini.</b> |
| 5.2. distinct .....  | 12                                 |
| 5.3. aggregate.....  | 13                                 |
| 5.3.1. L'opérateur group .....                             | 13                                 |
| 5.3.2. L'opérateur match .....                             | 14                                 |
| 5.3.3. L'opérateur unwind.....                             | 14                                 |

# 1. Prise en main

## 1.1. Outils nécessaires

- MongoDB (contenant mongod)
- MongoSH (contenant mongosh)
- MongoDB Command Line Database Tools
- Compass (facultatif)

## 1.2. Installer MongoDB

Télécharger MongoDB, et créer un dossier nommé data dans le répertoire C:\ et ajouter un dossier db dans ce dossier.

Copier le contenu des dossiers bin de MongoSH et MongoDB Command Line Database Tools dans le dossier bin de MongoDB.

## 1.3. Démarrer MongoDB

Ouvrir mongod.exe dans une console pour démarrer le serveur MongoDB.

Ouvrir ensuite mongosh.exe dans une autre console pour démarrer le client au serveur. Appuyer sur entrée pour utiliser le serveur local. C'est dans cette console que le code pourra être exécuté.

## 1.4. Arrêter proprement MongoDB

Dans la console de mongosh.exe, tapez la commande :  
`db.shutdownServer()`

## 2. Bases

### 2.1. Syntaxe dans le fichier json

**Bientôt disponible**

### 2.2. Les conditions

| Condition                                 | Signe équivalent                   |
|---|------------------------------------|
| "champ" : valeur                          | ==                                 |
| "champ" : {\$eq : valeur}                 | ==                                 |
| "champ" : {\$lt : valeur}                 | <                                  |
| "champ" : {\$lte : valeur}                | <=                                 |
| "champ" : {\$gt : valeur}                 | >                                  |
| "champ" : {\$gte : valeur}                | >=                                 |
| "champ" : {\$ne : valeur}                 | !=                                 |
| "champ" : {\$type : type}                 | Champ d'un certain type            |
| "champ" : {\$exists : true}               | Champ existant                     |
| "champ" : {\$in : [valeur1, valeur2...]}  | Contient une de ces valeurs        |
| "champ" : {\$nin : [valeur1, valeur2...]} | Ne contient pas une de ces valeurs |
| \$and : [{condition1}, {condition2}...]   | condition1 et condition2           |
| \$or : [{condition1}, {condition2}...]    | condition1 ou condition2           |
| \$not : {condition}                       | !condition                         |

## 3. Requêtes d'initialisation de la base de données

### 3.1. Utiliser une base de données (ou la créer si elle n'existe pas) pour réaliser des requêtes

use maBaseDeDonnees

### 3.2. Afficher les bases de données

```
show dbs
```

### 3.3. Supprimer une base de données

```
use maBaseDeDonnees  
db.dropDatabase()
```

### 3.4. Afficher les collections d'une base de données

```
use maBaseDeDonnees  
show collections
```

### 3.5. Supprimer une collection d'une base de données

```
use maBaseDeDonnees  
db.maCollection.drop()
```

### 3.6. Importer une collection (avec mongoimport)

Ouvrir la console dans le dossier bin du MongoDB, et tapez :

```
mongoimport fichier.json /db:maBaseDeDonnees --jsonArray --  
collection maCollection
```

À venir : exportation des données

## 4. Requêtes de mise à jour de la collection

### 4.1. Insérer un document

```
db.maCollection.insertOne({"champ1" : valeur1, "champ2" : valeur2...})
```

Si aucune valeur n'est renseignée pour le champ `_id`, la valeur est générée automatiquement (sous forme d'un Objet et non d'un entier)

## 4.2. Insérer plusieurs documents

```
db.maCollection.insertMany([{"champ1" : valeur1, "champ2" :  
valeur2...}, {"champ1" : valeur3, "champ2" : valeur4...}])
```

Attention : si un document ne peut pas être inséré, il est ignoré et les documents suivants sont quand même insérés.

## 4.3. Mettre à jour un document

### 4.3.1. Mise à jour simple

```
db.maCollection.updateOne({condition}, {$set : {"champ1" :  
valeur1}})
```

### 4.3.2. Faire incrémenter un document

```
db.maCollection.updateOne({condition}, {$inc : {"champ1" :  
valeur1}})
```

### 4.3.3. Mettre à jour un tableau

#### 4.3.3.1. Pour un rang du tableau

```
db.maCollection.updateOne({condition}, {$set : {"tableau.rang" :  
valeur1}})
```

#### 4.3.3.2. Pour ajouter une valeur dans le tableau sans redondance

```
db.maCollection.updateOne({condition}, {$addToSet : {"tableau" :  
valeur1}})
```

#### 4.3.3.3. Pour ajouter une valeur dans le tableau avec redondance

```
db.maCollection.updateOne({condition}, {$push : {"tableau" :  
valeur1}})
```

#### 4.3.3.4. Pour ajouter plusieurs valeurs dans le tableau (avec ou sans redondance)

```
db.maCollection.updateOne({condition}, {$addToSet : {"tableau" :  
{$each : [valeur1, valeur2...]}})
```

#### 4.3.3.5. Pour supprimer une valeur dans le tableau sans redondance

```
db.maCollection.updateOne({condition}, {$pull : {"tableau" :  
valeur1}})
```

#### 4.3.3.6. Pour mettre à jour une occurrence du tableau des valeurs (respectant les conditions)

```
db.maCollection.updateOne({condition}, {$inc : {"tableau.$" :  
valeur1}})
```

#### 4.3.3.7. Pour mettre à jour toutes les occurrences du tableau des valeurs

```
db.maCollection.updateOne({condition}, {$inc : {"tableau.$[]" :  
valeur1}})
```



#### 4.3.4. Supprimer un champ

```
db.maCollection.updateOne({condition}, {$unset : "champ1"})
```

### 4.4. Mettre à jour plusieurs documents

#### 4.4.1. Mise à jour simple

```
db.maCollection.updateMany({condition}, {$set : {"champ1" :  
valeur1}})
```

#### 4.4.2. Faire incrémenter plusieurs documents

```
db.maCollection.updateMany({condition}, {$inc : {"champ1" :  
valeur1}})
```

#### 4.4.3. Mettre à jour un tableau

##### 4.4.3.1. Pour un rang du tableau

```
db.maCollection.updateMany({condition}, {$set : {"tableau.rang" :  
valeur1}})
```

##### 4.4.3.2. Pour ajouter une valeur dans le tableau sans redondance

```
db.maCollection.updateOne({condition}, {$addToSet : {"tableau" :  
valeur1}})
```

##### 4.4.3.3. Pour ajouter une valeur dans le tableau avec redondance

```
db.maCollection.updateOne({condition}, {$push : {"tableau" :  
valeur1}})
```

#### 4.4.3.4. Pour ajouter plusieurs valeurs dans le tableau (avec ou sans redondance)

```
db.maCollection.updateOne({condition}, {$addToSet : {"tableau" :  
{$each : [valeur1, valeur2...]}}})
```

#### 4.4.3.5. Pour supprimer une valeur dans le tableau sans redondance

```
db.maCollection.updateOne({condition}, {$pull : {"tableau" :  
valeur1}})
```

#### 4.4.3.6. Pour mettre à jour une occurrence du tableau des valeurs (respectant les conditions)

```
db.maCollection.updateOne({condition}, {$inc : {"tableau.$" :  
valeur1}})
```

#### 4.4.3.7. Pour mettre à jour toutes les occurrences du tableau des valeurs

```
db.maCollection.updateOne({condition}, {$inc : {"tableau.$[]" :  
valeur1}})
```

#### 4.4.3.8. Pour mettre à jour toutes les occurrences du tableau des valeurs respectant la condition du tableau

```
db.maCollection.updateOne({condition}, {$inc :  
{"tableau.$[elem].champ1Tableau" : valeur1}}, {arrayFilters :  
[{"elem.champ2Tableau" : valeur2}]})
```

#### 4.4.4. Supprimer un champ

```
db.maCollection.updateMany({condition}, {$unset : "champ1"})
```

#### 4.4.5. Mises à jour multiples

```
db.maCollection.updateMany({condition}, [{ $set : {"champ1" :  
valeur1}, { $set : {"champ2" : valeur2}}...])
```

#### 4.5. Remplacer un document

#### 4.6. Supprimer un document

```
db.maCollection.deleteOne({condition})
```

#### 4.7. Supprimer plusieurs documents

```
db.maCollection.deleteMany({condition})
```

### 5. Requêtes de sélection de données

#### 5.1. find

La commande find permet d'interroger une base de données MongoDB et de renvoyer toutes les valeurs de la collection.

##### 5.1.1. Afficher toute la collection

```
db.maCollection.find({})
```

##### 5.1.2. Afficher certains champs

```
db.maCollection.find({}, {"_id" : 0, "champ1" : 1, "champ2" : 1...})
```

Par défaut, le champ `_id` est toujours affiché, pour le masquer, définir la visibilité à 0. Pour afficher un champ, définir la visibilité à 1.

### 5.1.3. Afficher certains champs d'une liste

```
db.maCollection.find({}, {"_id" : 0, "liste1.champ1" : 1, "champ2" : 1...})
```

### 5.1.4. Afficher les documents respectant la condition

```
db.maCollection.find({condition}, {"_id" : 0, "champ1" : 1, "champ2" : 1...})
```

### 5.1.5. Les curseurs

Syntaxe : `db.maCollection.find({condition}, {"_id" : 0, "champ1" : 1, "champ2" : 1...}).moncurseur()`

| Curseur                       | Description  |
|-------------------------------|--|
| <code>limit(n)</code>         | Limiter le nombre de documents à afficher                              |
| <code>skip(m).limit(n)</code> | Afficher les $n$ premiers documents à partir d'un rang $m$ non compris |
| <code>count()</code>          | Calculer le nombre de documents respectant la condition                |

## 5.2. distinct

La commande `distinct` permet d'afficher facilement différentes valeur que peut prendre un champ.

Commande : `db.maCollection.distinct("champ")`

## 5.3. aggregate

la commande aggregate permet de faire des requêtes plus avancées, notamment des calculs sur les attributs.

Commande : `db.maCollection.aggregate([ {opérateur1}, {opérateur2}...])`

### 5.3.1. L'opérateur group

#### 5.3.1.1. Grouper sur un champ

`$group : { _id : "$champ" }`

#### 5.3.1.2. Grouper sur un champ et faire des calculs

##### 5.3.1.2.1. Commande pour grouper un champ et faire des calculs

`$group : { _id : "$champ", nomChamp : {opérateurCalcul} }`

Note : Pour faire des calculs sur tous les documents, il est possible de faire :  
`$group : { _id : null, nomChamp : {opérateurCalcul} }`

##### 5.3.1.2.2. Les différents calculs possibles

| Opérateur de calcul            | Utilité  |
|--------------------------------|--|
| <code>\$sum : "\$champ"</code> | Calculer la somme des valeurs  |
| <code>\$avg : "\$champ"</code> | Calculer la moyenne des valeurs  |
| <code>\$min : "\$champ"</code> | Calculer la valeur minimale  |
| <code>\$max : "\$champ"</code> | Calculer la valeur maximale  |
| <code>\$count : {}</code>      | Calculer le nombre d'documents (à partir de la version 5.0 de mongoDB) |
| <code>\$sum : 1</code>         | Calculer le nombre d'documents (pour toutes les versions)              |

### 5.3.2. L'opérateur match

Cet opérateur permet d'utiliser une condition (avant ou après un regroupement).

Commande : `$match : {condition}`

### 5.3.3. L'opérateur unwind

Cet opérateur permet de transformer chaque élément d'un tableau en un document séparé contenant tous les champs.

Commande : `$unwind : "$tableau"`

#### 5.3.3.1. L'opérateur sort

Cet opérateur permet de trier les résultats dans l'ordre croissant (1) ou dans l'ordre décroissant (-1).

Commande : `$sort : {"champ1" : ordre1, "champ2" : ordre2...}`

#### 5.3.3.2. L'opérateur project

Cet opérateur permet de n'afficher que certains champs. C'est l'équivalent de la commande find.

Commande : `$project : {"_id" : 0, "champ1" : 1, "champ2" : 1...}`

## 6. Les vues

### 6.1. Créer une vue

`db.createView("maVue", "maCollection ", [pipeline])`

## 6.2. Afficher une vue

```
db.maVue.find(...)
```