



## **Base De Données NoSQL**

Réalisé par : Meziane Sarah  
INFOA3-25/26

# MANUEL D'INSTALLATION ET UTILISATION DE MONGODB AVEC DOCKER

## Introduction

MongoDB est une base de données NoSQL orientée documents.

Elle manipule les données sous forme de documents JSON et offre des commandes permettant de rechercher, mettre à jour, transformer et optimiser l'accès aux informations.

---

## Étapes du processus:

### 1. Création du conteneur MongoDB

Créer le conteneur MongoDB avec un volume vers le dossier courant :

```
$ docker run -d --name mongodb -p 27017:27017 -v "${PWD}:/data" mongo
```

Vérifier que le conteneur est en cours d'exécution :

```
docker ps
```

### 2. Télécharger les données (sur la machine hôte)

Depuis PowerShell :

```
curl https://atlas-education.s3.amazonaws.com/sampleddata.archive -o sampledata.archive
```

### 3. Copier le fichier dans le conteneur

```
docker cp ./sampledata.archive mongodb:/data
```

#### 4. . Accéder au conteneur

```
docker exec -it mongodb bash
```

#### 5. Restaurer la base dans MongoDB

Dans le conteneur :

```
cd /data
mongorestore --archive=sampleddata.archive --port=27017
```

#### 6. Lancer le shell MongoDB

```
mongosh
```

-Changer de base :

```
use sample_mflix
```

-Lister les collections :

```
show collections
```

---

## Commandes MongoDB utiles

### 1. Lecture des données – Commande find()

La lecture d'informations dans MongoDB s'appuie sur la méthode `find()`, qui permet de sélectionner des documents en fonction de critères.

#### Principes généraux

- Le premier paramètre sert à définir les **conditions de filtrage** (équivalent du WHERE).
- Le second paramètre permet de **choisir les champs à afficher**, pratique pour contrôler la forme du résultat.
- La commande retourne un **curseur**, sur lequel on peut appliquer :
  - `sort()` pour trier les résultats,

- limit() pour restreindre le nombre d'éléments,
  - skip() pour ignorer une partie du résultat,
  - pretty() pour un affichage lisible.
- 

## 2. Mise à jour des données – updateOne() et updateMany()

La mise à jour en MongoDB repose sur deux commandes permettant de modifier partiellement un document sans le remplacer complètement.

### Principes généraux

- updateOne() : modifie un seul document correspondant au filtre.
- updateMany() : applique la modification à l'ensemble des documents sélectionnés.

Les modifications utilisent des opérateurs comme :

- **\$set** pour ajouter ou mettre à jour un champ,
- **\$inc** pour modifier une valeur numérique,
- **\$unset** pour supprimer un champ.

## 3. Agrégations – Pipeline aggregate()

Les traitements plus avancés passent par la commande aggregate(), qui permet d'enchaîner plusieurs transformations au sein d'un pipeline.

### Principes généraux

Une agrégation est un tableau d'étapes exécutées successivement :

- **\$match** : filtrage avancé,
- **\$project** : sélection et création de champs dérivés (par ex. formater ou calculer une valeur),
- **\$group** : regroupement de documents permettant d'obtenir des statistiques,
- **\$sort** : tri après transformation,

- **\$unwind** : décomposition d'un tableau pour traiter ses éléments individuellement,
- **\$limit** : restriction du nombre de résultats.

## 4. Indexation – Optimisation des performances

Les index permettent d'accélérer les requêtes en évitant d'examiner tous les documents d'une collection.

### Principes généraux

- Un index se crée sur un ou plusieurs champs.
- Les index permettent d'améliorer la vitesse de find() et parfois aggregate().
- L'impact d'un index se vérifie avec explain(), notamment via :
  - **le nombre total de documents parcourus**,
  - **le temps d'exécution**.

---

### Arrêter/Supprimer le conteneur

docker stop mongodb

docker rm mongodb