

**MongoDB**



**TP2 Réalisé par Amal TAOUS**

**Examiné par :Mr SAMIR Youcef**

**Vérification et Installation de Docker**

1. **Vérification de Docker sur ma machine :**

**J’exécute cette commande dans le terminal**

docker --version

Si Docker n’est pas installé, on peut le télécharger depuis [Docker](https://www.docker.com/) et suivre les instructions en fonction de notre système d’exploitation.

**Configuration et lancement de MongoDB dans Docker**

1. **Lancer MongoDB dans un conteneur Docker** :

J’exécute la commande suivante pour démarrer MongoDB :

docker run --name mongodb -d -p 27017:27017 -v $(pwd)/data:/data/db mongo:latest

* + **--name mongodb** : Donne un nom au conteneur.
  + **-d** : Lance le conteneur en arrière-plan.
  + **-p 27017:27017** : Mappe le port 27017 du conteneur au port 27017 de la machine hôte.
  + **-v $(pwd)/data:/data/db** : Monte un répertoire local (data) pour persister les données.

1. **Vérification de l’etat du conteneur MongoDB:**

docker ps

puisque tout est bien configuré, une ligne indiquant que le conteneur Mongodb est actif

**Préparation des données**

1. **Je télécharge le fichier films.json.** et je le place dans mon repertoire de travail

**Importation des données dans MongoDB**

1. **Copier le fichier dans le conteneur Docker :**

docker cp films.json mongodb:/films.json

1. **Accéder au conteneur :**

On lance une session interactive dans le conteneur

docker exec -it mongodb bash

1. **Importater les données avec mongoimport :**

Une fois connecté , on exécute cette commande pour importer le fichier json dan mangodb :

mongoimport --db lesfilms --collection films --file /films.json --jsonArray

* + **--db lesfilms** : Crée ou sélectionne la base de données lesfilms.
  + **--collection films** : Crée ou sélectionne la collection films.
  + **--file /films.json** : Chemin vers le fichier JSON.
  + **--jsonArray** : Indique que les données sont un tableau JSON.

**Accéder à MongoDB**

1. **Se connecter au shell MongoDB :**

On peut accéder au shell mangodb directement depuis le conteneur avec cette commande

docker exec -it mongodb mongosh

1. **Sélectionnez la base de données lesfilms :**

use lesfilms

**Réaliser les requêtes**

Dans cette partie je vous présente l’ensemble des requêtes MongoDB demandées dans le TP :

**1. Vérifiez que les données sont importées :**pour vérifier que les données ont bien été importées, j’ai compté le nombre total de documents dan la collection fils en utilisant cette commande :

db.films.count()

Le nombre total de documents dans la collection fils est 278

**2. Examiner un document :**

db.films.findOne()

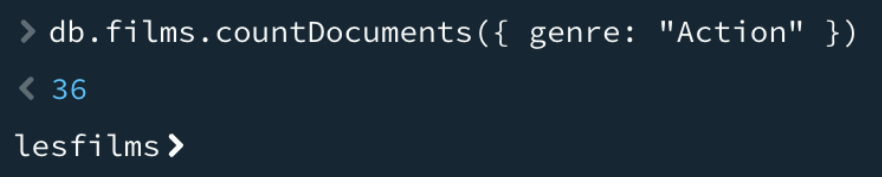
Cela affiche un document aléatoire de la collection pour comprendre sa structure.  
son but est d’examiner les champs disponibles ainsi que leur structure

**3. Afficher les films d’action :**

db.films.find({ genre: "Action" })

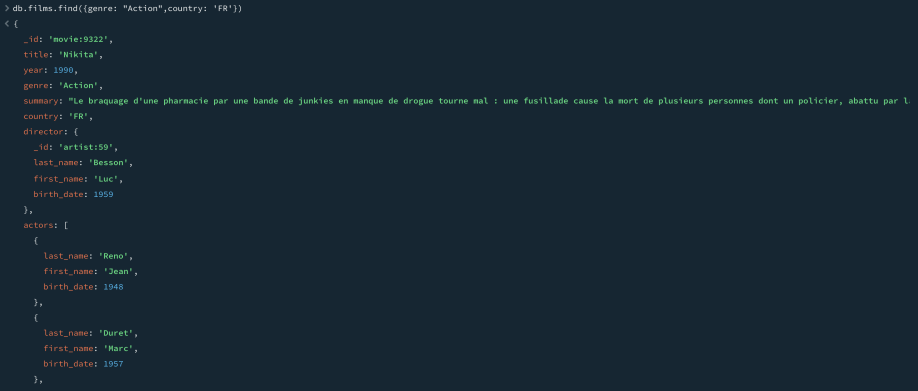
**4. Compter les films d’action :**

db.films.count({ genre: "Action" })



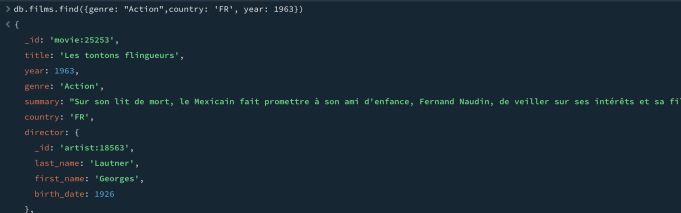
**5. Afficher les films d’action produits en France :**

db.films.find({ genre: "Action", country: "FR" })



**6. Afficher les films d’action produits en France en 1963 :**

db.films.find({ genre: "Action", country: "FR", year: 1963 })

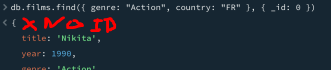


**7. Afficher les films d’action en France sans les notes (grades) :**

db.films.find({ genre: "Action", country: "FR" }, { grades: 0 })



**8. Afficher les films sans identifiants (\_id) :**

db.films.find({ genre: "Action", country: "FR" }, { \_id: 0, grades: 0 })  


**9. Afficher les titres et grades des films d’action en France :**

db.films.find({ genre: "Action", country: "FR" }, { \_id: 0, title: 1, grades: 1 })



**10. Afficher les films avec une note > 10 :**

db.films.find({ genre: "Action", country: "FR", "grades.note": { $gt: 10 } }, { \_id: 0, title: 1, grades: 1 })

le résultats de cette commande st un peu non correcte vu que ca m’a renvoyé des films avec des notes inferieur à 10.  
donc pour afficher que des films ayant que des notes supérieures à 10,on doit exclure explicitement les films contenant des notes inferieures ou égales à 10.

**11. Afficher les films ayant toutes les notes > 10 :**

db.films.find({ genre: "Action", country: "FR", grades: { $not: { $elemMatch: { note: { $lte: 10 } } } } }, { \_id: 0, title: 1, grades: 1 })

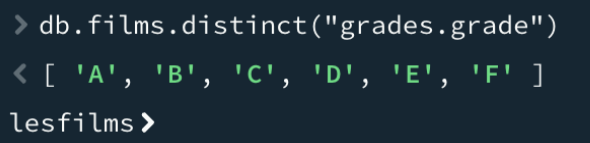
**12. Lister les genres :**

db.films.distinct("genre")



**13. Lister les grades :**

db.films.distinct("grades")



**14. Films avec des artistes spécifiques :**

db.films.find({ artists: { $in: ["artist:4", "artist:18", "artist:11"] } })

**15. Films sans résumé :**

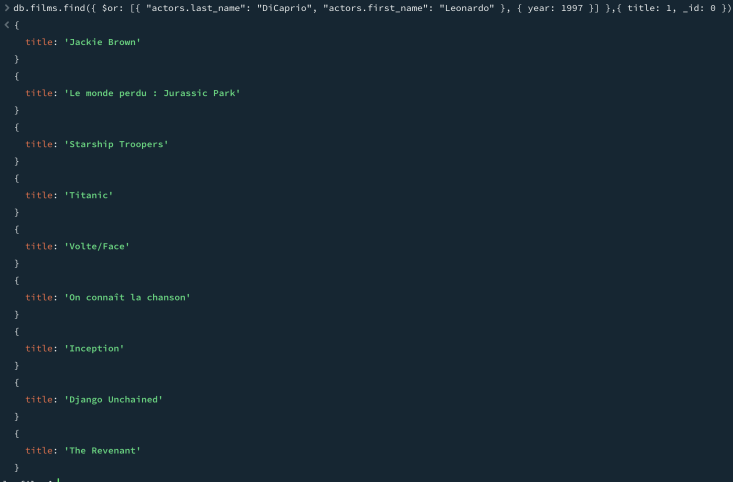
db.films.find({ summary: { $exists: false } })

**16. Films avec Leonardo DiCaprio en 1997 :**

db.films.find({ "actors.first\_name": "Leonardo", "actors.last\_name": "DiCaprio", year: 1997 })  


**17. Films avec Leonardo DiCaprio ou réalisé en 1997 :**

db.films.find({ $or: [{ "actors.first\_name": "Leonardo", "actors.last\_name": "DiCaprio"}, { year: 1997 }] })



**Finalement : Arrêter MongoDB**

Pour arrêter MongoDB :

docker stop mongodb

Pour supprimer le conteneur :

docker rm mongodb