

## 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](#) 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.

## 2. Vérification de l'état du conteneur MongoDB:

```
docker ps
```

puisque tout est bien configuré, une ligne indiquant que le conteneur Mongoddb 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
```

## 2. Accéder au conteneur :

On lance une session interactive dans le conteneur

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

### 3. 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
```

### 2. 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" })
```

```
> db.films.countDocuments({ genre: "Action" })
< 36
lesfilms >
```

## 5. Afficher les films d'action produits en France :

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

```
> db.films.find({genre: "Action",country: 'FR'})
< {
  _id: 'movie:9322',
  title: 'Nikita',
  year: 1990,
  genre: 'Action',
  summary: "Le braquage d'une pharmacie par une bande de junkies en manque de drogue tourne mal : une fusillade cause la mort de plusieurs personnes dont un policier, abattu par l.",
  country: 'FR',
  director: {
    _id: 'artist:59',
    last_name: 'Besson',
    first_name: 'Luc',
    birth_date: 1959
  },
  actors: [
    {
      last_name: 'Reno',
      first_name: 'Jean',
      birth_date: 1948
    },
    {
      last_name: 'Duret',
      first_name: 'Marc',
      birth_date: 1957
    }
  ],
}
```

## 6. Afficher les films d'action produits en France en 1963 :

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

```
> db.films.find({genre: "Action",country: 'FR', year: 1963})
< {
  _id: 'movie:25253',
  title: 'Les tontons flingueurs',
  year: 1963,
  genre: 'Action',
  summary: "Sur son lit de mort, le Mexicain fait promettre à son ami d'enfance, Fernand Naudin, de veiller sur ses intérêts et sa fi",
  country: 'FR',
  director: {
    _id: 'artist:18563',
    last_name: 'Lautner',
    first_name: 'Georges',
    birth_date: 1926
  },
}
```

## 7. Afficher les films d'action en France sans les notes (grades) :

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

```
> db.films.find({ genre: "Action", country: "FR" }, { grades: 0 })
< {
  _id: 'movie:9322',
  title: 'Nikita',
  year: 1990,
  genre: 'Action',
  summary: "Le braquage d'une pharmacie par une bande de junkies en manque de drogue tourne mal : une fusillade cause la mort de plusieurs personnes dont un policier",
  country: 'FR',
  director: {
    _id: 'artist:59',
    last_name: 'Besson',
    first_name: 'Luc',
    birth_date: 1959
  },
  actors: [
    {
      last_name: 'Reno',
      first_name: 'Jean',
      birth_date: 1948
    }
  ]
}
```

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

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

```
> db.films.find({ genre: "Action", country: "FR" }, { _id: 0 })
< { X NO ID
  title: 'Nikita',
  year: 1990,
  genre: 'Action'
```

## 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 })
```

```

> db.films.find({ genre: "Action", country: "FR" }, { title: 1, grades: 1, _id: 0 })
< {
  title: 'Nikita',
  grades: [
    {
      note: 88,
      grade: 'F'
    },
    {
      note: 36,
      grade: 'C'
    },
    {
      note: 74,
      grade: 'D'
    },
    {
      note: 62,
      grade: 'D'
    }
  ]
}
{
  title: "L'Homme de Rio",
  grades: [
    {
      note: 4,

```

## 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")

```
> db.films.distinct("genre")
< [
  'Action',          'Adventure',
  'Aventure',        'Comedy',
  'Comédie',         'Crime',
  'Drama',           'Drame',
  'Fantastique',     'Fantasy',
  'Guerre',          'Histoire',
  'Horreur',         'Musique',
  'Mystery',         'Mystère',
  'Romance',         'Science Fiction',
  'Science-Fiction', 'Thriller',
  'War',             'Western'
]
lesfilms>
```

### 13. Lister les grades :

```
db.films.distinct("grades")
```

```
> db.films.distinct("grades.grade")
< [ 'A', 'B', 'C', 'D', 'E', 'F' ]
lesfilms>
```

### 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 })
> db.films.find({ "actors.last_name": "DiCaprio", "actors.first_name": "Leonardo", year: 1997 }, { title: 1, _id: 0 })
< {
  title: 'Titanic'
}
```

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

```
db.films.find({ $or: [{ "actors.first_name": "Leonardo", "actors.last_name": "DiCaprio" }, { year: 1997 }] })
```

```
> db.films.find({ $or: [{ "actors.last_name": "DiCaprio", "actors.first_name": "Leonardo" }, { year: 1997 }] }, { title: 1, _id: 0 })
< {
  title: 'Jackie Brown'
}
{
  title: 'Le monde perdu : Jurassic Park'
}
{
  title: 'Starship Troopers'
}
{
  title: 'Titanic'
}
{
  title: 'Volte/Face'
}
{
  title: 'On connaît la chanson'
}
{
  title: 'Inception'
}
{
  title: 'Django Unchained'
}
{
  title: 'The Revenant'
}
```

## Enfin : Arrêter MongoDB

Pour arrêter MongoDB :

```
docker stop mongodb
```

Pour supprimer le conteneur :

```
docker rm mongodb
```