

Rapport de TP : Prise en Main de MongoDB

Houda FILALI

30 janvier 2025

1 Introduction

Qu'est-ce que MongoDB ?

MongoDB est un système de gestion de bases de données (SGBD) de type NoSQL, conçu pour stocker et gérer des données sous forme de documents JSON. Contrairement aux bases de données relationnelles qui organisent les données sous forme de tables avec des lignes et des colonnes, MongoDB utilise une structure flexible où chaque enregistrement est un document stocké au format BSON (Binary JSON). Cette approche permet de manipuler des données semi-structurées de manière efficace, tout en offrant une scalabilité optimale.

Pourquoi choisir MongoDB ?

- **Flexibilité** : Contrairement aux bases SQL qui imposent un schéma rigide, MongoDB permet d'avoir des documents avec des champs dynamiques.
- **Scalabilité** : Il est conçu pour gérer de grandes quantités de données et s'adapte facilement aux systèmes distribués.
- **Performance** : Les requêtes sont rapides grâce à l'indexation et aux recherches optimisées sur les documents.
- **Facilité d'utilisation** : L'interface en ligne de commande et les pilotes pour différents langages de programmation rendent son utilisation intuitive.

Objectif du TP

Ce TP vise à explorer les fonctionnalités de MongoDB en exécutant diverses requêtes sur une collection de films. Nous verrons comment :

- Installer et configurer MongoDB avec Docker.
- Importer et manipuler des données dans une base MongoDB.
- Effectuer des requêtes avancées pour filtrer et récupérer des informations spécifiques.

À la fin de ce TP, nous aurons une meilleure compréhension des bases de données NoSQL et de leur utilisation dans des environnements modernes.

2 Installation et Configuration de MongoDB avec Docker

L'installation de MongoDB via Docker permet une mise en place rapide et sans configuration complexe. Pour démarrer un conteneur MongoDB, utilisez la commande suivante :

```
docker run --name mongodb -d -p 27017:27017 mongo
```

Une fois le conteneur en cours d'exécution, nous pouvons interagir avec la base de données MongoDB en ouvrant un terminal interactif MongoDB :

```
docker exec -it mongodb mongosh
```

3 Importation des Données

Nous allons importer une collection de films dans la base de données `lesfilms` en utilisant `mongoimport`, un outil permettant d'insérer des documents à partir d'un fichier JSON :

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

4 Vérification de l'Importation

4.1 Comment vérifier que les données ont bien été importées ?

Question

Comment s'assurer que la base de données `lesfilms` contient bien les films importés ?

Réponse : Nous listons les bases de données disponibles en exécutant :

```
show dbs
```

Sortie attendue :

```
admin      40.00 KiB
config     108.00 KiB
lesfilms   400.00 KiB
local      40.00 KiB
```

La présence de `lesfilms` confirme que l'importation a réussi.

4.2 Comment afficher la structure d'un document ?

Question

Comment afficher la structure d'un film stocké dans la base de données ?

Réponse : Pour examiner un document type, nous utilisons la commande suivante :

```
db.films.findOne()
```

Exemple de sortie :

```
{
  _id: 'movie:11',
  title: 'La Guerre des toiles ',
  year: 1977,
  genre: 'Aventure',
  summary: "Il y a bien longtemps, dans une galaxie tr s lointaine...",
  country: 'US',
  director: {
    _id: 'artist:1',
    last_name: 'Lucas',
    first_name: 'George',
    birth_date: 1944
  },
  actors: [
    { last_name: 'Hamill', first_name: 'Mark', birth_date: 1951 },
    { last_name: 'Ford', first_name: 'Harrison', birth_date: 1942 },
    { last_name: 'Fisher', first_name: 'Carrie', birth_date: 1956 },
    { last_name: 'Cushing', first_name: 'Peter', birth_date: 1913 }
  ],
  grades: [
    { note: 93, grade: 'C' },
    { note: 68, grade: 'C' },
    { note: 50, grade: 'E' },
    { note: 29, grade: 'D' }
  ]
}
```

Ce document représente un film stocké dans la base de données **lesfilms**. Il contient :

- Un **id** unique identifiant le film.
- Un **title** (titre) et une **year** (année de sortie).
- Un **genre** et un **summary** (résumé) décrivant le film.
- Un champ **director** contenant les informations du réalisateur.
- Une liste **actors** détaillant les acteurs ayant joué dans le film.
- Une liste **grades** contenant les notes attribuées au film.

4.3 Liste et Nombre de Films d'Action

Question

Comment obtenir la liste complète des films appartenant au genre **Action** ?

Réponse : Nous pouvons récupérer tous les films d'action enregistrés dans la base de données en utilisant la requête suivante :

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

La sortie peut être trop longue pour être affichée en intégralité, mais voici quelques exemples de films obtenus :

```
{ _id: 'movie:1', title: 'Kill Bill : Volume 1', year: 2003, genre: 'Action' }
{ _id: 'movie:2', title: 'Gladiator', year: 2000, genre: 'Action' }
{ _id: 'movie:3', title: 'Minority Report', year: 2002, genre: 'Action' }
{ _id: 'movie:4', title: 'Terminator', year: 1984, genre: 'Action' }
```

Question

Comment compter le nombre total de films d'action dans la base de données ?

Réponse : Nous utilisons la méthode `countDocuments()` pour obtenir le nombre total de films appartenant au genre **Action** :

```
db.films.countDocuments({ genre: "Action" })
```

Sortie attendue :

```
36
```

Ainsi, la base de données contient **36 films d'action**.

4.4 Films d'Action Produits en France

Question

Comment filtrer les films d'action réalisés en France ?

Réponse : Nous appliquons un filtre supplémentaire sur le pays de production (`country: "FR"`) en plus du genre :

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

Exemple de résultat :

```
{
  _id: 'movie:4034',
  title: "L'Homme de Rio",
  year: 1964,
  genre: 'Action',
  country: 'FR',
  director: {
    last_name: 'de Broca',
    first_name: 'Philippe'
  }
}
```

Nous obtenons ainsi une liste de films d'action ayant été produits en France.

4.5 Films d'Action Produits en France en 1963

Question

Comment récupérer uniquement les films d'action français sortis en 1963 ?

Réponse : Nous ajoutons une condition supplémentaire pour filtrer uniquement les films d'action réalisés en France et sortis en 1963 :

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

Exemple de sortie :

```
{
  _id: 'movie:25253',
  title: 'Les tontons flingueurs',
  year: 1963,
  genre: 'Action',
  country: 'FR',
  director: {
    last_name: 'Lautner',
    first_name: 'Georges'
  }
}
```

Ce résultat indique que **Les tontons flingueurs** est un film d'action français sorti en 1963.

4.6 Affichage Sélectif des Attributs

Question

Comment afficher uniquement les films d'action produits en France sans inclure leurs notes (**grades**) ?

Réponse : Nous utilisons une projection qui exclut le champ **grades** en lui attribuant la valeur 0 :

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

Exemple de sortie :

```
{
  _id: 'movie:4034',
  title: "L'Homme de Rio",
  year: 1964,
  genre: 'Action',
  country: 'FR',
  summary: "...",
  director: { last_name: 'de Broca', first_name: 'Philippe' },
  actors: [ ... ]
}
```

Ainsi, toutes les informations du film sont affichées, sauf les notes.

4.7 Exclusion des Identifiants

Question

Comment exclure également les identifiants des documents (**_id**) dans les résultats ?

Réponse : Pour masquer les identifiants (`_id`), nous les excluons explicitement dans la projection :

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

Exemple de sortie :

```
{
  title: "Nikita",
  year: 1990,
  genre: 'Action',
  country: 'FR',
  summary: "...",
  director: { last_name: 'Besson', first_name: 'Luc' },
  actors: [ ... ]
}
```

Cela permet d'obtenir uniquement les informations utiles sans les identifiants techniques.

4.8 Affichage des Titres et Grades des Films d'Action en France

Question

Comment afficher uniquement les titres des films et leurs notes (`grades`) ?

Réponse : Nous appliquons une projection qui inclut uniquement les champs `title` et `grades` :

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

Exemple de sortie :

```
{
  title: "Les tontons flingueurs",
  grades: [
    { note: 35, grade: 'C' },
    { note: 48, grade: 'F' },
    { note: 64, grade: 'C' },
    { note: 42, grade: 'F' }
  ]
}
```

Seuls les titres et les notes des films sont affichés.

4.9 Films Notés Supérieurs à 10

Question

Comment afficher uniquement les films ayant obtenu au moins une note supérieure à 10 ?

Réponse : Nous utilisons l'opérateur `$gt` pour filtrer les films ayant des notes supérieures à 10 :

```
db.films.find({ genre: "Action", country: "FR", "grades.note": { $gt: 10 } })
```

Exemple de sortie :

```
{
  _id: 'movie:9322',
  title: "Nikita",
  year: 1990,
  genre: 'Action',
  country: 'FR',
  summary: "...",
  director: { last_name: 'Besson', first_name: 'Luc' },
  actors: [ ... ],
  grades: [
    { note: 88, grade: 'F' },
    { note: 36, grade: 'C' },
    { note: 74, grade: 'D' },
    { note: 62, grade: 'D' }
  ]
}
```

4.10 Films ayant uniquement des notes supérieures à 10

Question

Comment afficher uniquement les films où **toutes** les notes sont strictement supérieures à 10 ?

Réponse : Nous utilisons `$not` combiné avec `$elemMatch` pour exclure les films ayant au moins une note inférieure ou égale à 10 :

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

Exemple de sortie :

```
[
  {
    title: 'Nikita',
    grades: [
      { note: 88, grade: 'F' },
      { note: 36, grade: 'C' },
      { note: 74, grade: 'D' },
      { note: 62, grade: 'D' }
    ]
  },
  {
    title: 'Les tontons flingueurs',
    grades: [
      { note: 35, grade: 'C' },
      { note: 48, grade: 'F' },
      { note: 64, grade: 'C' },
      { note: 42, grade: 'F' }
    ]
  }
]
```

Seuls les films où **toutes** les notes sont supérieures à 10 sont affichés.

4.11 Liste des Genres de Films Disponibles

Question

Comment récupérer la liste des genres distincts présents dans la base de données ?

Réponse : Nous utilisons la fonction `distinct()` :

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

Résultat :

```
[
  'Action',
  'Aventure',
  'Com die',
  'Drama',
  'Fantastique',
  'Guerre',
  'Horreur',
  'Mystery',
  'Romance',
  'Science-Fiction',
  'War',
  'Adventure',
  'Comedy',
  'Crime',
  'Drame',
  'Fantasy',
  'Histoire',
  'Musique',
  'Myst re',
  'Science Fiction',
  'Thriller',
  'Western'
]
```


4.12 Liste des Notes et Grades Distincts

Question

Comment obtenir toutes les notes distinctes et leurs grades associés ?

Réponse : Nous utilisons la fonction `distinct()` pour récupérer les valeurs uniques de la clé `grades` :

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

Extrait des résultats :

```
[
  { note: 1, grade: 'A' }, { note: 2, grade: 'C' },
  { note: 3, grade: 'D' }, { note: 4, grade: 'E' },
  { note: 5, grade: 'F' }, { note: 6, grade: 'B' },
  { note: 7, grade: 'C' }, { note: 8, grade: 'D' },
  { note: 9, grade: 'E' }, { note: 10, grade: 'A' },
  { note: 11, grade: 'B' }, { note: 12, grade: 'C' },
  { note: 13, grade: 'D' }, { note: 14, grade: 'E' },
  { note: 15, grade: 'F' }, { note: 16, grade: 'A' },
  { note: 17, grade: 'B' }, { note: 18, grade: 'C' },
  { note: 19, grade: 'D' }, { note: 20, grade: 'E' },
  ...
]
```

Cette requête permet d'extraire toutes les combinaisons uniques de notes et de grades présents dans la base.

4.13 Films Réalisés par Certains Réalisateurs

Question

Comment récupérer tous les films réalisés par certains réalisateurs spécifiques ?

Réponse : Nous utilisons l'opérateur `$in` pour filtrer les films réalisés par une liste de réalisateurs ayant des identifiants donnés :

```
db.films.find({ "director._id": { $in: ["artist:240", "artist:108", "artist:1"] } }).pretty()
```

Extrait des résultats :

```
[
  {
    _id: 'movie:11',
    title: 'La Guerre des toiles ',
    year: 1977,
    genre: 'Aventure',
    director: { _id: 'artist:1', last_name: 'Lucas', first_name: 'George' }
  },
  {
    _id: 'movie:120',
    title: 'The Lord of the Rings: The Fellowship of the Ring',
    year: 2001,
    genre: 'Adventure',
    director: { _id: 'artist:108', last_name: 'Jackson', first_name: 'Peter' }
  },
  {
    _id: 'movie:62',
    title: "2001 l'Odyssée de l'espace",
    year: 1968,
    genre: 'Science-Fiction',
    director: { _id: 'artist:240', last_name: 'Kubrick', first_name: 'Stanley' }
  }
]
```

4.14 Films Sans Résumé

Question

Comment afficher uniquement les films n'ayant pas de résumé ?

Réponse : Nous utilisons l'opérateur `$exists` pour filtrer les films où le champ `summary` est absent :

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

Cette requête renvoie tous les films pour lesquels la clé `summary` n'est pas définie.

4.15 Recherche de Films Avec un Acteur Spécifique

Question

Comment rechercher les films dans lesquels **Leonardo DiCaprio** a joué en 1997 ?

Réponse : Nous filtrons sur le prénom et le nom de l'acteur, ainsi que sur l'année de sortie du film :

```
db.films.find({ "actors.first_name": "Leonardo", "actors.last_name": "DiCaprio",  
year: 1997 }).pretty()
```

Résultat :

```
[  
  {  
    _id: 'movie:597',  
    title: 'Titanic',  
    year: 1997,  
    genre: 'Drame',  
    director: { _id: 'artist:2710', last_name: 'Cameron', first_name: 'James' }  
  }  
]
```

4.16 Films Correspondant à Plusieurs Critères

Question

Comment rechercher les films soit avec **Leonardo DiCaprio**, soit réalisés en 1997 ?

Réponse : Nous utilisons l'opérateur logique **\$or** pour rechercher les films qui remplissent l'un des critères :

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

Extrait des résultats :

```
[  
  { title: 'Jackie Brown', year: 1997, genre: 'Crime' },  
  { title: 'Titanic', year: 1997, genre: 'Drame' },  
  { title: 'Inception', year: 2010, genre: 'Action', actors: ['Leonardo DiCaprio'] },  
  { title: 'The Revenant', year: 2015, genre: 'Western', actors: ['Leonardo DiCaprio'] }  
]
```

5 Conclusion

Ce TP nous a permis de découvrir MongoDB, son installation avec Docker et ses principales fonctionnalités. Nous avons exploré diverses requêtes permettant :

- D'extraire des informations spécifiques (films d'un genre donné, films d'un réalisateur, etc.).
- De filtrer les résultats en fonction de certains critères (notes supérieures à une valeur, présence d'un acteur...).
- D'utiliser des opérateurs avancés (**\$in**, **\$or**, **\$exists**) pour optimiser nos recherches.

MongoDB offre une grande flexibilité pour manipuler et interroger des données non structurées, ce qui le rend particulièrement adapté aux applications modernes nécessitant une gestion dynamique des données.