

# TP2: Prise en main de MongoDB, un serveur de gestion de bases de données (NoSQL) documentaires et distribuées

Malek SLOKOM

November 2024

## 1 Introduction

MongoDB est une base de données NoSQL documentaire, largement utilisée pour la gestion de données non relationnelles. Contrairement aux bases de données relationnelles, MongoDB stocke les données sous forme de documents JSON, ce qui offre une flexibilité et une performance accrues pour manipuler des données complexes.

L'objectif de ce TP est de prendre en main MongoDB à travers une série d'opérations de manipulation de données, en partant de l'importation d'une collection jusqu'à l'exécution de requêtes complexes.

## 2 Importation des données

```
(base) malekslokom@Maleks-MacBook-Pro ~ % mongoimport --db lesfilms --collection films --file ~/desktop/3ING/noSQL/films.json --jsonArray
2024-11-29T15:12:09.551+0100    connected to: mongodb://localhost/
2024-11-29T15:12:09.644+0100    278 document(s) imported successfully. 0 document(s) failed to import.
(base) malekslokom@Maleks-MacBook-Pro ~ %
```

Cette commande permet d'importer les données stockées dans un fichier JSON dans une collection MongoDB (films) d'une base de données nommée lesfilms. C'est une étape essentielle pour initialiser une base de données avec des informations.

## 3 Vérification des données

Pour vérifier que les données ont bien été importées, nous avons compté le nombre total de documents dans la collection films en utilisant la commande suivante :

```
>_MONGOSH

> use lesfilms
< switched to db lesfilms

> db.films.countDocuments()
< 278

lesfilms>
```

Le nombre total de documents dans la collection films est : 278

## 4 Exploration de la structure des documents

Pour comprendre la structure des documents de la collection, nous avons utilisé la commande suivante :

```
>_MONGOSH
> db.films.findOne()
< {
  _id: 'movie:24',
  title: 'A History of Violence',
  year: 2005,
  genre: 'Drama',
  summary: 'Max Swell, un père de famille à la vie paisiblement tranquille, abat dans un réflexe de légitime défense son agresseur dans un restaurant. Il devient alors un persona',
  country: 'US',
  director: {
    _id: 'artist:224',
    last_name: 'Cramenberg',
    first_name: 'David',
    birth_date: 1943
  },
  actors: [
    {
      last_name: 'Halle',
      first_name: 'Marta',
      birth_date: 1991
    },
    {
      last_name: 'Hurtenson',
      first_name: 'Riggen',
      birth_date: 1958
    },
    {
      last_name: 'McLaws',
      first_name: 'Ashdon',
      birth_date: 1978
    },
    {
      last_name: 'Gibson',

```

Cette commande retourne un seul document de la collection, ce qui nous permet d'examiner les champs disponibles ainsi que leur structure.

## 5 Requêtes spécifiques

### 5.1 Liste des films d'action

Pour afficher tous les films appartenant au genre "Action", nous avons utilisé la commande suivante :

```
>_MONGOSH
}
> db.films.find({genre: "Action"})
< {
  _id: 'movie:24',
  title: 'Kill Bill : Volume 1',
  year: 2003,
  genre: 'Action',
  summary: 'Au cours d'une cérémonie de mariage en plein désert, un commando fait irruption dans la chapelle et tire sur les convives. Laissée pour morte, la Mariée enceinte retrou',
  country: 'US',
  director: {
    _id: 'artist:138',
    last_name: 'Tarantino',
    first_name: 'Quentin',
    birth_date: 1963
  },
  actors: [
    {
      last_name: 'Thurman',
      first_name: 'Uma',
      birth_date: 1970
    },
    {
      last_name: 'Liu',
      first_name: 'Lucy',
      birth_date: 1968
    },
    {
      last_name: 'Carradine',
      first_name: 'David',
      birth_date: 1936
    },
  ],
}
```

## 5.2 Nombre de films d'action

Pour connaître le nombre total de films d'action dans la collection, nous avons utilisé la commande suivante :

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

## 5.3 Films d'action produits en France

Pour identifier les films d'action produits en France, nous avons utilisé la commande suivante :

```
>_MONGODB  
> 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  
    },  
    {  
      last_name: 'Karyo',  
      first_name: 'Tchéky',  
      birth_date: 1953  
    },  
    {  
      last_name: 'Parillaud',
```

Cette commande combine deux critères pour afficher uniquement les films correspondant à un genre et à un pays spécifiques.

## 5.4 Films d'action produits en France en 1963

Pour cibler les films d'action produits en France en 1963, nous avons utilisé la commande suivante :

```
> MONGOSH
> 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 fille Patricia. Fernand découvre alors qu'il se",
  country: 'FR',
  director: {
    _id: 'artist:18563',
    last_name: 'Lautner',
    first_name: 'Georges',
    birth_date: 1926
  },
  actors: [
    {
      last_name: 'Ventura',
      first_name: 'Lino',
      birth_date: 1919
    },
    {
      last_name: 'Frank',
      first_name: 'Horst',
      birth_date: 1929
    },
    {
      last_name: 'Dalban',
      first_name: 'Robert',
      birth_date: 1903
    },
    {
      last_name: 'Bliez',

```

## 5.5 Films d'action sans les grades

Pour afficher les films d'action produits en France tout en excluant les grades des résultats, nous avons utilisé la commande suivante :

```
> MONGOSH
> 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, 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
    },
    {
      last_name: 'Karyo',
      first_name: 'Tchéky',
      birth_date: 1953
    },
    {
      last_name: 'Parillaud',

```

Cette commande est utile lorsque certains champs ne sont pas nécessaires dans les résultats et permettent de simplifier l'affichage.

## 5.6 Films d'action sans identifiants

Pour afficher les films d'action sans inclure leurs identifiants (`_id`), nous avons utilisé la commande suivante :

```

> db.films.find({ genre: "Action", country: "FR" }, { _id: 0 })
< {
  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
    },
    {
      last_name: 'Karyo',
      first_name: 'Tchéky',
      birth_date: 1953
    },
    {
      last_name: 'Parillaud',
      first_name: 'Anne',

```

Cette commande permet de supprimer l'affichage par défaut du champ `_id`, souvent non nécessaire dans les analyses.

## 5.7 Titres et grades des films d'action

Pour n'afficher que les titres et les grades des films d'action réalisés en France, nous avons utilisé la commande suivante :

```

> 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,
      grade: 'D'
    },
    {
      note: 30,
      grade: 'E'
    }
  ]
}

```

## 5.8 Titres et notes supérieures à 10

Pour afficher les titres et grades des films ayant obtenu une note supérieure à 10, nous avons utilisé la commande suivante:

```
>_MONGOSH
> db.film.find( { genre: "Action", country: "FR", "grades.note": { $gt: 10 } }, { 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,
      grade: 'D'
    },
    {
      note: 30,
      grade: 'E'
    }
  ]
}
```

Nous avons remarqué que certains films retournés contenaient des notes inférieures à 10. Cela s'explique par la manière dont MongoDB évalue les tableaux dans les documents.

Dans les documents MongoDB, le champ grades est un tableau contenant plusieurs objets, et MongoDB applique la condition `$gte: 10` à chaque élément du tableau. Si au moins un élément satisfait la condition, MongoDB inclut tout le document dans les résultats, même si d'autres éléments du tableau ne respectent pas la condition.

- **Solution 1:**

Pour afficher que des films ayant que des notes supérieures à 10, on doit préciser que les notes obtenues sont toutes supérieures à 40.

```
>_MONGOSH
> db.film.find( { genre: "Action", country: "FR", "grades.note": { $gt: 40 } }, { 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: 'Les tontons flingueurs',
  grades: [
    {
      note: 35,
      grade: 'C'
    },
    {
      note: 48,
      grade: 'F'
    }
  ]
}
```

- **Solution 2:** Pour garantir que seules les films ayant toutes leurs notes supérieures à 10 soient affichés, nous devons exclure explicitement les films contenant des notes inférieures ou égales à 10. Cela peut être fait en utilisant une combinaison d'opérateurs comme \$not et \$lte.

```
> db.films.find( { genre: "Action", country: "FR", "grades.note": { $not: { $lte: 10 } } }, { 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: 'Les tontons flingueurs',
    grades: [
      {
        note: 35,
        grade: 'C'
      },
      {
        note: 48,
        grade: 'F'
      }
    ]
  }
]
```

## 6 Requêtes supplémentaires

### 6.1 Genres distincts

Pour obtenir la liste des genres distincts dans la collection, nous avons utilisé la commande suivante :

```
> 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>
```

Cette commande permet de connaître la variété des genres disponibles dans la base, ce qui peut être utile pour des analyses globales.

## 6.2 Les grades attribués

Pour extraire une liste unique des grades présents dans les documents du champ grades, nous avons utilisé la commande suivante :

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

## 6.3 Films qui n'ont pas de résumé

Pour afficher les films qui n'ont pas de résumé, nous avons utilisé la commande suivante :

```
> db.films.find({ summary: { $exists: false } }, { title: 1, _id: 0 })
<
lesfilms>
```

## 6.4 Films de Leonardo DiCaprio en 1997

Pour afficher les films dans lesquels joue Leonardo DiCaprio en 1997, nous avons utilisé la commande suivante :

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

## 6.5 Films avec Leonardo DiCaprio ou réalisés en 1997

Pour afficher les films réalisés en 1997 ou joués par Leonardo DiCaprio, nous avons utilisé la commande suivante :

```
> 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'
}
lesfilms>
```