3ème année cycle d'ingénieur - Spécialité Informatique.

TP2 - MongoDb

Réalisé par :
ER-RACHDI Ilias

Encadrés par :
M. YOUCEF

Année Scolaire : 2025/2026

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 informations sous forme de documents JSON, ce qui offre une grande flexibilité ainsi que de bonnes performances pour la manipulation de données complexes.

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

2 Importation des données

Pour importer les données dans MongoDB, nous avons utilisé la commande suivante :

```
mongoimport --db lesfilms --collection films \
  --file ~/desktop/3ING/noSQL/films.json --jsonArray
```

Cette commande permet d'importer les données stockées dans un fichier JSON dans une collection MongoDB nommée `films`, appartenant à une base de données appelée `lesfilms`.

3 Vérification des données

Afin de vérifier que les données ont bien été importées, nous avons compté le nombre total de documents dans la collection `films` avec la commande :

```
db.films.countDocuments()
```

Le nombre total de documents présents dans la collection `films` est : **278**.

4 Exploration de la structure des documents

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

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

Cette commande retourne un document de la collection, ce qui permet d'observer les champs disponibles ainsi que leur organisation (types, sous-documents, tableaux, etc.).

5 Requêtes spécifiques

5.1 Liste des films d'action

Pour afficher tous les films appartenant au genre « Action », nous avons exécuté la requête suivante :

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

5.2 Nombre de films d'action

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

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

5.3 Films d'action produits en France

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

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

Cette commande combine deux critères (`genre` et `country`) pour ne conserver que les films d'action français.

5.4 Films d'action produits en France en 1963

Pour cibler les films d'action produits en France en 1963, nous avons précisé la condition sur l'année :

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

5.5 Films d'action sans les grades

Si l'on souhaite afficher les films d'action produits en France tout en excluant le champ `grades` des résultats, on peut utiliser une projection :

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

Cette commande est utile lorsque certains champs ne sont pas nécessaires à l'analyse et que l'on souhaite simplifier l'affichage.

5.6 Films d'action sans identifiants

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

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

Cette projection permet de supprimer l'affichage par défaut du champ `_id`, qui n'est pas toujours utile dans les résultats.

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 :

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

5.8 Titres et notes supérieures à 10

Pour afficher les titres et les notes des films d'action français ayant obtenu au moins une note strictement supérieure à 10, nous avons commencé par la requête suivante :

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

Nous avons toutefois constaté que certains films retournés possédaient également des notes inférieures ou égales à 10. Cela s'explique par la façon dont MongoDB évalue les tableaux dans les documents.

En effet, dans nos documents, le champ `grades` est un tableau d'objets. MongoDB applique la condition `$gt: 10` à chaque élément du tableau : si au moins un élément satisfait la condition, le document entier est inclus dans le résultat, même si d'autres éléments ne respectent pas ce critère.

Solution 1

Une première solution consiste à augmenter la valeur du seuil afin d'exclure les films ayant des notes plus faibles, par exemple :

```
db.films.find(
  { genre: "Action", country: "FR", "grades.note": { $gt: 40 } },
  { title: 1, grades: 1, _id: 0 }
)
```

Solution 2

Pour garantir que seuls les films dont *toutes* les notes sont strictement supérieures à 10 sont affichés, on peut exclure explicitement les films contenant au moins une note inférieure ou égale à 10, 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 }
)
```

6 Requêtes supplémentaires

6.1 Genres distincts

Pour obtenir la liste des genres distincts présents dans la collection, nous avons exécuté :

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

Cette commande permet d'identifier la variété des genres disponibles dans la base et constitue un bon point de départ pour une analyse globale.

6.2 Grades attribués

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

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

6.3 Films qui n'ont pas de résumé

Pour afficher les films qui ne possèdent pas de résumé, nous avons utilisé la requête suivante :

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

6.4 Films de Leonardo DiCaprio en 1997

Pour trouver les films dans lesquels joue « Leonardo DiCaprio » en 1997, nous avons exécuté :

```
db.films.find(
  { actors: "Leonardo DiCaprio", year: 1997 },
  { title: 1, _id: 0 }
)
```

6.5 Films avec Leonardo DiCaprio ou réalisés en 1997

Enfin, pour lister les films réalisés en 1997 *ou* dans lesquels joue Leonardo DiCaprio, nous avons utilisé une requête avec l'opérateur logique `$or` :

```
db.films.find(
  {
    $or: [
      { actors: "Leonardo DiCaprio" },
      { year: 1997 }
    ]
  },
  { title: 1, _id: 0 }
)
```