TP1 – Introduction à MongoDB

Base de Données NoSQL

NICOLAS Ethan

G5SI2
2025–2026

Table des matières

1 Installation et import des données

1.1 Installation de MongoDB

Linux (Debian/Ubuntu)

```
sudo apt update
sudo apt install -y mongodb
mongo
```

macOS (Homebrew)

```
brew tap mongodb/brew
brew install mongodb-community
brew services start mongodb-community
mongosh
```

Docker

```
docker run -d --name mongo -p 27017:27017 mongo
docker exec -it mongo mongosh
```

1.2 Téléchargement et import du jeu `sample_mflix`

MongoDB fournit un jeu de données éducatif contenant films, utilisateurs et commentaires.

Téléchargement

```
curl https://atlas-education.s3.amazonaws.com/sampledata.archive \
     -o sampledata.archive
```

Import via `mongorestore`

Les fichiers sont au format BSON, donc `mongoimport` est inadapté.

```
mongorestore --archive=sampledata.archive mongodb://user:password@<ip_address:port>
```

Connexion à la base

```
mongosh
use sample_mflix
```

2 Partie 2 | Quelques requêtes pour comprendre le fonctionnement

1. Films sortis depuis 2015

```
db.movies.find({ year: { $gte: 2015 } }).limit(5)
```

Sélection par comparaison : `$gte` garde les années >= 2015. Limite à 5 documents.

2. Films du genre Comedy

```
db.movies.find({ genres: "Comedy" })
```

Le champ `genres` est un tableau. La valeur simple correspond à « tableau contenant cet élément ».

3. Films entre 2000 et 2005

```
db.movies.find({year: {$gte: 2000, $lte: 2005}}, {title: 1, year: 1})
```

Projection pour n'afficher que deux champs.

4. Genres Drama ET Romance

```
db.movies.find({genres: {$all: ["Drama", "Romance"]}}, {title:1, genres:1})
```

`$all` impose la présence simultanée des deux valeurs dans le tableau.

5. Films sans champ rated

```
db.movies.find({rated: {$exists: false}}, {title:1})
```

`$exists:false` filtre les documents ne possédant pas la clé.

6. Nombre de films par année

```
db.movies.aggregate([
  {$group: {_id: "$year", total: {$sum: 1}}},
  {$sort: {_id: 1}}
])
```

`$group` regroupe par année et compte. `$sort` trie.

7. Moyenne IMDb par genre

```
db.movies.aggregate([
  {$unwind: "$genres"},
  {$group: {_id: "$genres", moyenne: {$avg: "$imdb.rating"}}},
  {$sort: {moyenne: -1}}
])
```

`$unwind` transforme le tableau en lignes séparées.

8. Nombre de films par pays

```
db.movies.aggregate([
  {$unwind: "$countries"},
  {$group: {_id: "$countries", total: {$sum: 1}}},
  {$sort: {total: -1}}
])
```

9. Top 5 réalisateurs

```
db.movies.aggregate([
  {$unwind: "$directors"},
  {$group: {_id: "$directors", total: {$sum: 1}}},
  {$sort: {total: -1}},
  {$limit: 5}
])
```

10. Films triés par note IMDb

```
db.movies.aggregate([
  {$sort: {"imdb.rating": -1}},
  {$project: {title: 1, "imdb.rating": 1}}
])
```

11. Ajouter un champ

```
db.movies.updateOne({title: "Jaws"}, {$set: {etat: "culte"}})
```

12. Incrémenter un champ

```
db.movies.updateOne(
    {title: "Inception"},
    {$inc: {"imdb.votes": 100}}
)
```

13. Supprimer un champ

```
db.movies.updateMany({}, {$unset: {poster: ""}})
```

14. Modifier le réalisateur

```
db.movies.updateOne(
  {title: "Titanic"},
  {$set: {directors: ["James Cameron"]}}
)
```

15. Films les mieux notés par décennie

```
db.movies.aggregate([
  {$match: {"imdb.rating": {$exists: true}}},
  {$project: {
    title: 1,
    decade: {$subtract: ["$year", {$mod: ["$year", 10]}]},
    "imdb.rating": 1
  }},
  {$group: {_id: "$decade", maxRating: {$max: "$imdb.rating"}}},
  {$sort: {_id: 1}}
])
```

16. Titres commençant par “Star”

```
db.movies.find({title: /^Star/}, {title: 1})
```

17. Films avec plus de 2 genres

```
db.movies.find(
  {$where: "this.genres.length > 2"},
  {title: 1, genres: 1}
)
```

18. Films de Christopher Nolan

```
db.movies.find(
  {directors: "Christopher Nolan"},
  {title: 1, year: 1, "imdb.rating": 1}
)
```

19. Créer un index sur year

```
db.movies.createIndex({year: 1})
```

20. Voir les index

```
db.movies.getIndexes()
```

21. Comparer avec explain()

```
db.movies.find({year: 1995}).explain("executionStats")
```

Comparer `totalDocsExamined` et `executionTimeMillis` :
Avec index : 1 ms, 372 documents scannés
Sans index : 12 ms, 21379 documents scannés

22. Supprimer l'index

```
db.movies.dropIndex({year: 1})
```

23. Index composé year + rating

```
db.movies.createIndex({year: 1, "imdb.rating": -1})
```

3 Partie 3 | Prise en main

```
mongoimport \
  --host <ip_address> \
  --port <port> \
  --username <username> \
  --password <password> \
  --db allocine \
  --collection films \
  --file films.json \
  --jsonArray
```

1. Compter les documents

```
db.films.countDocuments()
> 278
```

2. Afficher un film

```
db.films.findOne()
> {
  _id: 'movie:75',
  title: 'Mars Attacks!',
  year: 1996,
  genre: 'Comedy',
  summary: "'We come in peace' is not what those green men from Mars mean when they invade our
  country: 'US',
  director: {
    _id: 'artist:510',
    last_name: 'Burton',
    first_name: 'Tim',
    birth_date: 1958
  },
  actors: [
    {
      last_name: 'Nicholson',
      first_name: 'Jack',
      birth_date: 1937
    },
    ...
    {
      last_name: 'Steiger',
      first_name: 'Rod',
      birth_date: 1925
    }
  ],
  grades: [
```

```
    {
      note: 43,
      grade: 'C'
    },
    ...
    {
      note: 24,
      grade: 'E'
    }
   ]
  }
```

3. Films d'actions

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

4. Films d'actions

```
db.films.find({genre : "Action"}).size()
> 36
```

5. Films d'actions produit en France

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

6. Films d'actions produit en France en 1963

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

7. Films d'actions produit en France sans grades

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

8. Films d'actions produit en France sans ID

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

9. Films d'actions produit en France avec titre et grades

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

10. Films d'actions produit en France avec titre et grades, une des grades $>$ 10

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

11. Films d'actions produit en France avec titre et grades, all grades $>$ 10

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

12. Genres

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

13. Grades

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

15. Pas de résumé

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

16. DiCaprio en 1997

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

17. DiCaprio ou 1997

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