TP1 – MongoDB : Prise en main & Manipulation

Base de Données NoSQL

Haiballa ElVarougou

Table des matières

1 Installation et Import des Données

1.1 Installation de MongoDB

Sous Linux (Ubuntu)

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

Sous Docker

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

1.2 Import du dataset `sample_mflix`

MongoDB fournit un jeu de données éducatif contenant films, utilisateurs et commen- taires.

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

mongorestore --archive=sampledata.archive \
mongodb://user:password@adresse:27017
```

2 Requêtes MongoDB – dataset sample_mflix

2.1 Films sortis depuis 2015

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

2.2 Films du genre Comedy

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

2.3 Films entre 2000 et 2005

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

2.4 Films Drama & Romance

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

2.5 Films sans champ `rated`

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

2.6 Nombre de films par année

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

2.7 Moyenne IMDb par genre

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

3 Manipulation des Documents

3.1 Ajouter un champ

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

3.2 Incrémenter un champ

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

3.3 Supprimer un champ

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

4 Indexation

4.1 Créer un index sur l'année

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

4.2 Voir les index existants

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

4.3 Analyse explain()

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

4.4 Supprimer un index

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

4.5 Index composé

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

5 Dataset Allociné

5.1 Importation

```
mongoimport \
--db allocine \
--collection films \
--file films.json \
--jsonArray
```

5.2 Requêtes principales

Films d'action

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

Films d'action français

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

Films de 1963

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

Films avec note > 10

```
db.films.find({
  "grades.note": { $gte: 10 }
})
```

6 Conclusion

Ce TP m'a permis de :

— comprendre la structure d'une base de données MongoDB,

— manipuler les requêtes `find()`, `update()`, `aggregate()`,

— importer des datasets volumineux,

— analyser les performances grâce aux index.

MongoDB se révèle particulièrement adapté aux données semi-structurées et aux opérations analytiques flexibles.