

Rapport de LabWork 7 - MongoDB

Réalisé par : LAABID ABDESSAMAD

Introduction

Ce rapport présente les résultats du LabWork 7 portant sur l'utilisation de MongoDB, une base de données NoSQL orientée document. L'objectif principal était de se familiariser avec l'environnement MongoDB à travers son interface utilisateur MongoDB Compass et son shell de commande, puis d'explorer des collections de données en écrivant des requêtes pour extraire des informations spécifiques.

MongoDB stocke les données sous forme de documents JSON, une structure de données composée de paires clé/valeur, permettant de représenter des informations complexes et hiérarchiques. Contrairement aux bases de données relationnelles traditionnelles qui utilisent des tables, MongoDB utilise des collections pour stocker ces documents.

Installation et Configuration

L'installation de MongoDB 8.0 Community Edition sur Ubuntu 24.04 a été effectuée en suivant ces étapes:

1. Importer la clé publique :
 - a. Installez gnupg et curl s'ils ne sont pas déjà disponibles :

```
applus@applus:/mnt/c/Users/applus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo apt-get install gnupg curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17.2).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 97 not upgraded.
```

- b. Importer la clé publique GPG de MongoDB

```
applus@applus:/mnt/c/Users/applus/Documents/sda2/Traitement Paralleles/tps/tp7$ curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
--dearmor
```

2. Créer le fichier de liste :

```
applus@applus:/mnt/c/Users/applus/Documents/sda2/Traitement Paralleles/tps/tp7$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse
```

3. Recharger la base de données du package :

```
applus@applus:/mnt/c/Users/applus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 InRelease
Reading package lists... Done
```

4. Installer le serveur communautaire MongoDB :

```
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo apt-get install -y mongodb-org
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org-database mongodb-org-database-tools-extra mongodb-org-mongos mongodb-org-server mongodb-org-shell
  mongodb-org-tools
The following NEW packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org mongodb-org-database mongodb-org-database-tools-extra mongodb-org-mongos mongodb-org-server
  mongodb-org-shell mongodb-org-tools
0 upgraded, 9 newly installed, 0 to remove and 97 not upgraded.
Need to get 186 MB of archives.
```

5. Démarrer MongoDB :

```
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo systemctl start mongod
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo systemctl enable mongod
```

6. Vérifiez que MongoDB a démarré avec succès :

```
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ sudo systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-05-20 15:59:48 +01; 55s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 2979 (mongod)
    Memory: 195.9M ()
    CGroup: /system.slice/mongod.service
            └─2979 /usr/bin/mongod --config /etc/mongod.conf

May 20 15:59:48 apus systemd[1]: Started mongod.service - MongoDB Database Server.
May 20 15:59:48 apus mongod[2979]: {"t":{"$date":"2025-05-20T14:59:48.807Z"},"s":"I", "c":"CONTROL", "id":7484500, "ctx":"main", "msg":"Environment varia
```

Premiers Tests avec MongoDB Shell

Après l'installation, les tests initiaux ont été effectués via le shell MongoDB pour vérifier le bon fonctionnement du système :

```
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ mongosh
Current Mongosh Log ID: 682c99bc8b470f85d7c59f34
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1
Using MongoDB:      8.0.9
Using Mongosh:      2.5.1

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2025-05-20T15:59:48.914+01:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/pr
  odnotes-filesystem
  2025-05-20T15:59:49.423+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2025-05-20T15:59:49.423+01:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
  2025-05-20T15:59:49.423+01:00: We suggest setting the contents of sysfsFile to 0.
  2025-05-20T15:59:49.423+01:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
  -----

test> show dbs
admin    40.00 KiB
config  12.00 KiB
local   40.00 KiB
test> use mydb
switched to db mydb
mydb> db.persons.insertOne({
...   nome: "aplus",
...   master: "SDA"
... })
{
  acknowledged: true,
  insertedId: ObjectId('682c9a060b470f85d7c59f35')
}
mydb> show collections
persons
mydb> db.persons.find()
[
  {
    _id: ObjectId('682c9a060b470f85d7c59f35'),
    nome: 'aplus',
    master: 'SDA'
  }
]
```

L'identificateur unique (_id) a été automatiquement généré par MongoDB pour chaque document créé.

Exercice 1: Requêtes sur la Collection "restaurants"

La collection "restaurants" a été importée dans la base de données avec ce script de javascript :

```
aplust@aplust:/mnt/c/Users/aplust/Documents/sda2/Traitement Paralleles/tps/tp7$ cat temp.js
db.getSiblingDB("restaurantsDB").restaurants.insertMany(
  [{name:"ToorSeaFoodrestaurant", tag:["seafood", "expensive"], orderNeeded:false, maxPeople:100, review:4.3, cost:"medium", location:{type:"Point", coordinates:[45.0664, 7.6609]}}, {contact:{phone:"+3908763456", facebook:"ToorSeaFood"}},
  {name:"PandaParadise", tag:["chinese", "japanese"], orderNeeded:false, maxPeople:50, review:4.7, cost:"low", location:{type:"Point", coordinates:[45.0671, 7.6627]}}, {contact:{phone:"+395487634998", facebook:"PandaP"}},
  {name:"Stagione", tag:["italian", "pizza"], orderNeeded:false, maxPeople:10, review:3.8, cost:"low", location:{type:"Point", coordinates:[45.0698, 7.6634]}}, {contact:{phone:"+390223456245"}},
  {name:"Takitai", tag:["tai", "spicy"], orderNeeded:true, maxPeople:150, review:4.2, cost:"high", location:{type:"Point", coordinates:[45.0609, 7.6568]}}, {contact:{phone:"+39087673456", facebook:"TakiTai", twitter:"TakiTaiOfficial"}},
  {name:"MishiSushi", tag:["japanese", "unlimitedoffering"], orderNeeded:false, maxPeople:75, review:3.9, cost:"low", location:{type:"Point", coordinates:[45.0587, 7.6612]}}, {contact:{}},
  {name:"IlDivinPanino", tag:["casual", "goodforkids"], orderNeeded:false, maxPeople:10, review:4.6, cost:"low", location:{type:"Point", coordinates:[45.0645, 7.6608]}}, {contact:{phone:"+393319416860", website:"ildivinpanino.it"}},
  {name:"IlTempo", tag:["italian", "cosy"], orderNeeded:true, maxPeople:100, review:4.2, cost:"high", location:{type:"Point", coordinates:[45.0661, 7.6544]}}, {contact:{phone:"+390772376563"}},
  {name:"Smartbar", tag:["bar", "morningcoffee"], orderNeeded:false, maxPeople:15, review:4.5, cost:"low", location:{type:"Point", coordinates:[45.0643, 7.6494]}}, {contact:{phone:"+390986756234"}},
  {name:"Mcdonalds", tag:["fastfood"], orderNeeded:false, maxPeople:70, review:3.9, cost:"medium", location:{type:"Point", coordinates:[45.0696, 7.6503]}}, {contact:{facebook:"McdonaldTorino", website:"mcdonalds.com"}},
  {name:"OldNavyHamburger", tag:["hamburger", "fastfood"], orderNeeded:false, maxPeople:100, review:4.5, cost:"medium", location:{type:"Point", coordinates:[45.0772, 7.6740]}}, {contact:{phone:"+396763452345", facebook:"OldNavyHamburger"}}]);
```

```
aplust@aplust:/mnt/c/Users/aplust/Documents/sda2/Traitement Paralleles/tps/tp7$ mongosh --quiet < temp.js
test> db.getSiblingDB("restaurantsDB").restaurants.insertMany(
...  [{name:"ToorSeaFoodrestaurant", tag:["seafood", "expensive"], orderNeeded:false, maxPeople:100, review:4.3, cost:"medium", location:{type:"Point", coordinates:[45.0664, 7.6609]}}, {contact:{phone:"+3908763456", facebook:"ToorSeaFood"}},
...  {name:"PandaParadise", tag:["chinese", "japanese"], orderNeeded:false, maxPeople:50, review:4.7, cost:"low", location:{type:"Point", coordinates:[45.0671, 7.6627]}}, {contact:{phone:"+395487634998", facebook:"PandaP"}},
...  {name:"Stagione", tag:["italian", "pizza"], orderNeeded:false, maxPeople:10, review:3.8, cost:"low", location:{type:"Point", coordinates:[45.0698, 7.6634]}}, {contact:{phone:"+390223456245"}},
...  {name:"Takitai", tag:["tai", "spicy"], orderNeeded:true, maxPeople:150, review:4.2, cost:"high", location:{type:"Point", coordinates:[45.0609, 7.6568]}}, {contact:{phone:"+39087673456", facebook:"TakiTai", twitter:"TakiTaiOfficial"}},
...  {name:"MishiSushi", tag:["japanese", "unlimitedoffering"], orderNeeded:false, maxPeople:75, review:3.9, cost:"low", location:{type:"Point", coordinates:[45.0587, 7.6612]}}, {contact:{}},
...  {name:"IlDivinPanino", tag:["casual", "goodforkids"], orderNeeded:false, maxPeople:10, review:4.6, cost:"low", location:{type:"Point", coordinates:[45.0645, 7.6608]}}, {contact:{phone:"+393319416860", website:"ildivinpanino.it"}},
...  {name:"IlTempo", tag:["italian", "cosy"], orderNeeded:true, maxPeople:100, review:4.2, cost:"high", location:{type:"Point", coordinates:[45.0661, 7.6544]}}, {contact:{phone:"+390772376563"}},
...  {name:"Smartbar", tag:["bar", "morningcoffee"], orderNeeded:false, maxPeople:15, review:4.5, cost:"low", location:{type:"Point", coordinates:[45.0643, 7.6494]}}, {contact:{phone:"+390986756234"}},
...  {name:"Mcdonalds", tag:["fastfood"], orderNeeded:false, maxPeople:70, review:3.9, cost:"medium", location:{type:"Point", coordinates:[45.0696, 7.6503]}}, {contact:{facebook:"McdonaldTorino", website:"mcdonalds.com"}},
...  {name:"OldNavyHamburger", tag:["hamburger", "fastfood"], orderNeeded:false, maxPeople:100, review:4.5, cost:"medium", location:{type:"Point", coordinates:[45.0772, 7.6740]}}, {contact:{phone:"+396763452345", facebook:"OldNavyHamburger"}}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('682c9d0f37a08e8070c59f35'),
    '1': ObjectId('682c9d0f37a08e8070c59f36'),
    '2': ObjectId('682c9d0f37a08e8070c59f37'),
    '3': ObjectId('682c9d0f37a08e8070c59f38'),
    '4': ObjectId('682c9d0f37a08e8070c59f39'),
    '5': ObjectId('682c9d0f37a08e8070c59f3a'),
    '6': ObjectId('682c9d0f37a08e8070c59f3b'),
    '7': ObjectId('682c9d0f37a08e8070c59f3c'),
    '8': ObjectId('682c9d0f37a08e8070c59f3d'),
    '9': ObjectId('682c9d0f37a08e8070c59f3e')
  }
}
test> aplust@aplust:/mnt/c/Users/aplust/Documents/sda2/Traitement Paralleles/tps/tp7$
```

Requêtes et Résultats

1. Trouver tous les restaurants dont le prix est "medium" :

```

restaurantsDB> db.restaurants.find({"cost":"medium"})
[
  {
    _id: ObjectId('682c9d0f37a08e8070c59f35'),
    name: 'ToorSeafoodrestaurant',
    tag: [ 'seafood', 'expensive' ],
    orderNeeded: false,
    maxPeople: 100,
    review: 4.3,
    cost: 'medium',
    location: { type: 'Point', coordinates: [ 45.0664, 7.6609 ] },
    contact: { phone: '+390223456245', facebook: 'ToorSeaFood' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3d'),
    name: 'Mcdownloads',
    tag: [ 'fastfood' ],
    orderNeeded: false,
    maxPeople: 70,
    review: 3.9,
    cost: 'medium',
    location: { type: 'Point', coordinates: [ 45.0696, 7.6503 ] },
    contact: { Facebook: 'McdownloadTorino', website: 'mcdownloads.com' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3e'),
    name: 'OldNavyHamburgar',
    tag: [ 'hamburger', 'fastfood' ],
    orderNeeded: false,
    maxPeople: 100,
    review: 4.5,
    cost: 'medium',
    location: { type: 'Point', coordinates: [ 45.0772, 7.674 ] },
    contact: { phone: '+396763452345', facebook: 'OldNavyHamburgar' }
  }
]

```

Cette requête filtre les restaurants par leur catégorie de prix.

2. Trouver tous les restaurants dont l'avis est supérieur à 4 et le prix est moyen ou bas:

```

restaurantsDB> db.restaurants.find({review:{$gt:4}, $or:[{cost:"medium"},{cost:"low"}]})
[
  {
    _id: ObjectId('682c9d0f37a08e8070c59f35'),
    name: 'ToorSeafoodrestaurant',
    tag: [ 'seafood', 'expensive' ],
    orderNeeded: false,
    maxPeople: 100,
    review: 4.3,
    cost: 'medium',
    location: { type: 'Point', coordinates: [ 45.0664, 7.6609 ] },
    contact: { phone: '+390223456245', facebook: 'ToorSeaFood' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f36'),
    name: 'PandaParadise',
    tag: [ 'chinese', 'japanese' ],
    orderNeeded: false,
    maxPeople: 50,
    review: 4.7,
    cost: 'low',
    location: { type: 'Point', coordinates: [ 45.0671, 7.6627 ] },
    contact: { phone: '+395487634998', facebook: 'PandaP' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3a'),
    name: 'ilDivinPanino',
    tag: [ 'casual', 'goodforkids' ],
    orderNeeded: false,
    maxPeople: 10,
    review: 4.6,
    cost: 'low',
    location: { type: 'Point', coordinates: [ 45.0645, 7.6608 ] },
    contact: { phone: '+393319416860', website: 'ildivinpanino.it' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3c'),
    name: 'Smartbar',
    tag: [ 'bar', 'morningcoffee' ],
    orderNeeded: false,
    maxPeople: 15,
    review: 4.5,
    cost: 'low',
    location: { type: 'Point', coordinates: [ 45.0643, 7.6494 ] },
    contact: { phone: '+390986756234' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3e'),
    name: 'OldNavyHamburger',
    tag: [ 'hamburger', 'fastfood' ],
    orderNeeded: false,
    maxPeople: 100,
    review: 4.5,
    cost: 'medium',
    location: { type: 'Point', coordinates: [ 45.0772, 7.674 ] },
    contact: { phone: '+396763452345', facebook: 'OldNavyHamburger' }
  }
]

```

Cette requête combine une condition sur la note (`review`) avec une condition OR sur le coût.

3. Trouver tous les restaurants pouvant contenir plus de 5 personnes et dont le tag est "italien" ou "japonais" avec prix moyen/élevé, OU dont le tag n'est ni "italien" ni "japonais" avec avis supérieur à 4,5:

```

restaurantsDB> db.restaurants.find({
...   $and: [
...     {maxPeople: {$gt: 5}},
...     {$or: [
...       {$and: [
...         {tags: {$in: ["italian", "japanese"]}},
...         {cost: {$in: ["medium", "high"]}}
...       ]},
...       {$and: [
...         {tags: {$nin: ["italian", "japanese"]}},
...         {review: {$gt: 4.5}}
...       ]}
...     ]}
...   ]
... })
...
[
  {
    _id: ObjectId('682c9d0f37a08e8070c59f36'),
    name: 'PandaParadise',
    tag: [ 'chinese', 'japanese' ],
    orderNeeded: false,
    maxPeople: 50,
    review: 4.7,
    cost: 'low',
    location: { type: 'Point', coordinates: [ 45.0671, 7.6627 ] },
    contact: { phone: '+395487634998', facebook: 'PandaP' }
  },
  {
    _id: ObjectId('682c9d0f37a08e8070c59f3a'),
    name: 'IlDivinPanino',
    tag: [ 'casual', 'goodforkids' ],
    orderNeeded: false,
    maxPeople: 10,
    review: 4.6,
    cost: 'low',
    location: { type: 'Point', coordinates: [ 45.0645, 7.6608 ] },
    contact: { phone: '+393319416860', website: 'ildivinpanino.it' }
  }
]

```

Cette requête complexe combine plusieurs opérateurs logiques et filtres.

4. Calculer l'évaluation moyenne de tous les restaurants:

```

restaurantsDB> db.restaurants.aggregate([{$group:{$_id:null, review_avg:{$avg:"$review"}}}])
[ { _id: null, review_avg: 4.26 } ]
restaurantsDB>

```

Cette requête utilise l'agrégation pour calculer la moyenne des notes.

5. Compter le nombre de restaurants dont l'évaluation est supérieure à 4,5 et peut contenir plus de 5 personnes:

```

restaurantsDB> db.restaurants.aggregate([{$group:{$_id:null, review_avg:{$avg:"$review"}}}])
[ { _id: null, review_avg: 4.26 } ]
restaurantsDB> db.restaurants.aggregate([
...   {$match: {review: {$gt: 4.5}, maxPeople: {$gt: 5}}},
...   {$group: { _id: null, count: {$sum: 1}}}
... ])
...
[ { _id: null, count: 2 } ]
restaurantsDB>

```

Cette requête utilise un pipeline d'agrégation avec un filtre (\$match) suivi d'un regroupement (\$group).

6. Trouver combien de restaurants se trouvent à moins de 500 mètres du point [45.0623, 7.6627]:

```
// Création de l'index géospatial
```

```
restaurantsDB> db.restaurants.createIndex({location: "2dsphere"})
location_2dsphere
restaurantsDB> |
```

```
// Requête géospatiale
```

```
restaurantsDB> db.restaurants.find({
...   location: {
...     $near: {
...       $geometry: {
...         type: "Point",
...         coordinates: [45.0623, 7.6627]
...       },
...       $maxDistance: 500
...     }
...   }
... }).count()
...
3
```

Cette requête utilise les capacités géospatiales de MongoDB pour trouver des restaurants à proximité d'un point donné.

Exercice 2 : Requêtes sur la Collection "movies"

Pour cet exercice, nous avons travaillé sur une collection de films qui j'import avec le command suivant :

```
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ mongoimport --db moviesDB --collection movies --type json --file movies.json
2025-05-20T16:32:48.070+0100    connected to: mongodb://localhost/
2025-05-20T16:32:50.012+0100    23539 document(s) imported successfully. 0 document(s) failed to import.
aplus@aplus:/mnt/c/Users/aplus/Documents/sda2/Traitement Paralleles/tps/tp7$ |
```

Requêtes et Résultats

1. Rechercher tous les films sortis en 1893:

```
moviesDB> db.movies.find({ year: 1893 })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William M.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  }
]
moviesDB> |
```

2. Rechercher tous les films qui ont une durée supérieure à 120 minutes:

```
moviesDB> db.movies.find({ "runtime": { $gt: 120 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd5293'),
    plot: 'Young Pauline is left a lot of money when her wealthy uncle dies. However, her uncle's secretary has been named as her guardian until she marries, at which time she will officially take ...',
    genres: [ 'Action' ],
    runtime: 199,
    cast: [ 'Pearl White', 'Crane Wilbur', 'Paul Panzer', 'Edward José' ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BMzgxDk1Mzk2M15BM15BanBnXkFtZTgwMDg0NzkzMjE@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Perils of Pauline',
    fullplot: 'Young Pauline is left a lot of money when her wealthy uncle dies. However, her uncle's secretary has been named as her guardian until she marries, at which time she will officially take possession of her inheritance. Meanwhile, her "guardian" and his confederates constantly come up with schemes to get rid of Pauline so that he can get his hands on the money himself.',
    languages: [ 'English' ],
    released: ISODate('1914-03-23T00:00:00.000Z'),
    directors: [ 'Louis J. Gasnier', 'Donald MacKenzie' ],
    writers: [
      'Charles W. Goddard (screenplay)',
      'Basil Dickey (screenplay)',
      'Charles W. Goddard (novel)',
      'George B. Seitz',
      'Bertram Millhauser'
    ],
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-09-12 00:01:18.647000000',
    year: 1914,
    imdb: { rating: 7.6, votes: 744, id: 4465 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 2.8, numReviews: 9 },
      production: 'Pathé Frères',
      lastUpdated: ISODate('2015-09-11T17:46:19.000Z')
    }
  },
  {
    _id: ObjectId('573a1390f29313caabcd548c'),
    plot: 'The Civil War divides friends and destroys families, but that's nothing compared to the anarchy in the black-ruled South after the war.',
    genres: [ 'Drama', 'History', 'Romance' ],
```

3. Retrouver tous les films qui n'ont pas été évalués (UNRATED ou NOT RATED):

```
moviesDB> db.movies.find({ $or: [{ rated: "UNRATED" }, { rated: "NOT RATED" }] })
[
  {
    _id: ObjectId('573a1390f29313caabcd4135'),
    plot: 'Three men hammer on an anvil and pass a bottle of beer around.',
    genres: [ 'Short' ],
    runtime: 1,
    cast: [ 'Charles Kayser', 'John Ott' ],
    num_mflix_comments: 1,
    title: 'Blacksmith Scene',
    fullplot: 'A stationary camera looks at a large anvil with a blacksmith behind it and one on either side. The smith in the middle draws a heated metal rod from the fire, places it on the anvil, and all three begin a rhythmic hammering. After several blows, the metal goes back in the fire. One smith pulls out a bottle of beer, and they each take a swig. Then, out comes the glowing metal and the hammering resumes.',
    countries: [ 'USA' ],
    released: ISODate('1893-05-09T00:00:00.000Z'),
    directors: [ 'William K.L. Dickson' ],
    rated: 'UNRATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-26 00:03:50.133000000',
    year: 1893,
    imdb: { rating: 6.2, votes: 1189, id: 5 },
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3, numReviews: 184, meter: 32 },
      lastUpdated: ISODate('2015-06-28T18:34:09.000Z')
    }
  },
  {
    _id: ObjectId('573a1390f29313caabcd4323'),
    plot: 'A young boy, oppressed by his mother, goes on an outing in the country with a social welfare group where he dares to dream of a land where the cares of his ordinary life fade.',
    genres: [ 'Short', 'Drama', 'Fantasy' ],
    runtime: 14,
    rated: 'UNRATED',
    cast: [
      'Martin Fuller',
      'Mrs. William Bechtel',
      'Walter Edwin',
      'Ethel Jewett'
    ],
    num_mflix_comments: 2,
```

Cette requête utilise l'opérateur \$or pour combiner deux conditions.

4. Retrouver tous les films qui ont une note IMDb supérieure à 7:


```

moviesDB> db.movies.find({ "imdb.rating": { $gt: 7 } })
[
  {
    _id: ObjectId('573a1390f29313caabcd42e8'),
    plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
    genres: [ 'Short', 'Western' ],
    runtime: 11,
    cast: [
      'A.C. Abadie',
      'Gilbert M. 'Broncho Billy' Anderson',
      'George Barnes',
      'Justus D. Barnes'
    ],
    poster: 'https://m.media-amazon.com/images/M/MV5BMTU3NjE5NzYtYTUyNS00MDVhLWlWYjgtMmYwYWIxZDZyNzU2XkEyXkFqcGdeQXVyNzQzNzQxNzI@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Great Train Robbery',
    fullplot: "Among the earliest American cinema - notable as the first film that presented a narrative story to tell - it depicts a group of cowboy outlaws who hijack a train and its passengers. They are then pursued by a Sheriff's posse. Several scenes have color included - all hand tinted.",
    languages: [ 'English' ],
    released: ISODate('1903-12-01T00:00:00.000Z'),
    directors: [ 'Edwin S. Porter' ],
    rated: 'TV-G',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-13 00:27:59.177000000',
    year: 1903,
    imdb: { rating: 7.4, votes: 9847, id: 439 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.7, numReviews: 2559, meter: 75 },
      fresh: 6,
      critic: { rating: 7.6, numReviews: 6, meter: 100 },
      rotten: 0,
      lastUpdated: ISODate('2015-08-08T19:16:10.000Z')
    }
  },
  {
    _id: ObjectId('573a1390f29313caabcd4323'),
    plot: 'A young boy, oppressed by his mother, goes on an outing in the country with a social welfare group where he dares to dream of a land where the car

```

Cette requête montre comment accéder à des champs imbriqués dans les documents JSON.

5. Retrouver les 5 meilleurs films avec les notes IMDb les plus élevées:

```

moviesDB> db.movies.aggregate([
...   { $sort: { "imdb.rating": -1 } },
...   { $limit: 5 }
... ])
[
  {
    _id: ObjectId('573a13b8f29313caabd4d540'),
    plot: "The remarkable love story inspired by the lives of artists Lili Elbe and Gerda Wegener. Lili and Gerda's marriage and work evolve as they navigate Lili's groundbreaking journey as a transgender pioneer.",
    genres: [ 'Biography', 'Drama' ],
    runtime: 120,
    cast: [
      'Alicia Vikander',
      'Eddie Redmayne',
      'Amber Heard',
      'Matthias Schoenaerts'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BMjA0NjA4NjE2N1S0M1S0anBnXkFtZTgweNzIxNTY2NjE@._V1_SY1000_SX677_AL_.jpg',
    title: 'The Danish Girl',
    fullplot: "Copenhagen, early 1920s. Danish artist, Gerda Wegener, painted her own husband, Einar Wegener (Eddie Redmayne), as a lady in her painting. When the painting gained popularity, Einar started to change his appearance into a female appearance and named himself Lili Elbe. With his feminism passion and Gerda's support, Einar - or Elbe attempted first-ever male to female sex reassignment surgery, a decision that turned into a massive change for their marriage, that Gerda realized her own husband is no longer a man or the person she married before. A childhood friend of Einar, art-dealer Hans Axgil (Matthias Schoenaerts), shows up and forms a complex love triangle with the couple.",
    languages: [ 'English' ],
    released: ISODate('2015-11-27T00:00:00.000Z'),
    directors: [ 'Tom Hooper' ],
    writers: [ 'David Ebershoff (novel)', 'Lucinda Coxon (screenplay)' ],
    awards: { wins: 0, nominations: 1, text: '1 nomination.' },
    lastupdated: '2015-09-01 01:17:41.067000000',
    year: 2015,
    imdb: { rating: '', votes: '', id: 810819 },
    countries: [ 'USA', 'UK' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 4.1, numReviews: 300 },
      lastUpdated: ISODate('2015-09-10T17:15:10.000Z')
    }
  },
]

```

Cette requête utilise un pipeline d'agrégation pour trier les films par note décroissante et limiter les résultats aux 5 premiers.

6. Trouver la note IMDb moyenne des films pour chaque catégorie de notation:

```

moviesDB> db.movies.aggregate([
...   { $group: { _id: "$rated", averageRating: { $avg: "$imdb.rating" } } }
... ])
...
[
  { _id: 'TV-14', averageRating: 6.886516853932584 },
  { _id: null, averageRating: 6.82571922334045 },
  { _id: 'NOT RATED', averageRating: 7.0212703101920235 },
  { _id: 'M', averageRating: 7.013513513513513 },
  { _id: 'TV-MA', averageRating: 7.364999999999999 },
  { _id: 'R', averageRating: 6.4568460982658955 },
  { _id: 'PASSED', averageRating: 7.305494505494505 },
  { _id: 'NC-17', averageRating: 6.760526315789473 },
  { _id: 'X', averageRating: 6.5552631578947365 },
  { _id: 'Approved', averageRating: 7.040000000000001 },
  { _id: 'APPROVED', averageRating: 7.167369901547117 },
  { _id: 'PG-13', averageRating: 6.405897546276367 },
  { _id: 'Not Rated', averageRating: 7.7 },
  { _id: 'AO', averageRating: 5.3999999999999995 },
  { _id: 'GP', averageRating: 7.040909090909091 },
  { _id: 'G', averageRating: 6.6280922431865825 },
  { _id: 'TV-PG', averageRating: 6.951315789473684 },
  { _id: 'UNRATED', averageRating: 6.97190412782956 },
  { _id: 'TV-Y7', averageRating: 6.033333333333334 },
  { _id: 'TV-G', averageRating: 6.4576271186440675 }
]
Type "it" for more

```

Cette requête regroupe les films par catégorie de notation et calcule la moyenne des notes IMDb pour chaque groupe.

7. Calculer la durée moyenne des films sortis dans chaque pays:

```

moviesDB> db.movies.aggregate([
...   { $match: { type: 'movie' } },
...   { $unwind: '$countries' },
...   { $group: { _id: '$countries', averageRuntime: { $avg: '$runtime' } } }
... ])
...
[
  { _id: 'Hong Kong', averageRuntime: 106.25526315789473 },
  { _id: 'Ghana', averageRuntime: 113 },
  { _id: 'Zimbabwe', averageRuntime: 113 },
  { _id: 'Cyprus', averageRuntime: 123.2 },
  { _id: 'Côte d'Ivoire', averageRuntime: 90 },
  { _id: 'Iran', averageRuntime: 96.87962962962963 },
  { _id: 'Vietnam', averageRuntime: 103.9090909090909 },
  { _id: 'Paraguay', averageRuntime: 105.66666666666667 },
  { _id: 'USA', averageRuntime: 101.34581858979891 },
  { _id: 'Republic of Macedonia', averageRuntime: 108.3076923076923 },
  { _id: 'Cameroon', averageRuntime: 103.8 },
  { _id: 'Kuwait', averageRuntime: 96 },
  { _id: 'Pakistan', averageRuntime: 93.33333333333333 },
  { _id: 'Liechtenstein', averageRuntime: 114.33333333333333 },
  { _id: 'Singapore', averageRuntime: 95.75 },
  { _id: 'Vanuatu', averageRuntime: 100 },
  { _id: 'Iraq', averageRuntime: 116.81818181818181 },
  { _id: 'Poland', averageRuntime: 100.70967741935483 },
  { _id: 'Uzbekistan', averageRuntime: 81 },
  { _id: 'Bulgaria', averageRuntime: 104.45454545454545 }
]
Type "it" for more
moviesDB>

```

Cette requête filtre d'abord pour ne garder que les films, puis "déplie" le tableau des pays avant de calculer la durée moyenne par pays.

8. Trouver le genre de films le plus répandu dans la base de données:

```

moviesDB> db.movies.aggregate([
...   { $unwind: '$genres' },
...   { $group: { _id: '$genres', count: { $sum: 1 } } },
...   { $sort: { count: -1 } },
...   { $limit: 1 }
... ])
...
[ { _id: 'Drama', count: 13789 } ]
moviesDB>

```

Cette requête utilise \$unwind pour transformer chaque élément du tableau des genres en un document distinct, puis compte les occurrences de chaque genre.

9. Trouver le top 10 des directeurs de films en mentionnant les titres de leurs films réalisés:

```
moviesDB> db.movies.aggregate([
...   { $unwind: '$directors' },
...   { $group: {
...     _id: '$directors',
...     movieCount: { $sum: 1 },
...     movies: { $push: { title: '$title', year: '$year' } }
...   } },
...   { $sort: { movieCount: -1 } },
...   { $limit: 10 }
... ])
[
...
{
  _id: 'Woody Allen',
  movieCount: 40,
  movies: [
    { title: 'Take the Money and Run', year: 1969 },
    { title: 'Bananas', year: 1971 },
    { title: 'Sleeper', year: 1973 },
    { title: 'Love and Death', year: 1975 },
    { title: 'Annie Hall', year: 1977 },
    { title: 'Interiors', year: 1978 },
    { title: 'Manhattan', year: 1979 },
    { title: 'Stardust Memories', year: 1980 },
    { title: 'A Midsummer Night's Sex Comedy', year: 1982 },
    { title: 'Zelig', year: 1983 },
    { title: 'Broadway Danny Rose', year: 1984 },
    { title: 'The Purple Rose of Cairo', year: 1985 },
    { title: 'Hannah and Her Sisters', year: 1986 },
    { title: 'Radio Days', year: 1987 },
    { title: 'Another Woman', year: 1988 },
    { title: 'Crimes and Misdemeanors', year: 1989 },
    { title: 'New York Stories', year: 1989 },
    { title: 'Alice', year: 1990 },
    { title: 'Husbands and Wives', year: 1992 },
    { title: 'Shadows and Fog', year: 1991 },
    { title: 'Manhattan Murder Mystery', year: 1993 },
    { title: 'Bullets Over Broadway', year: 1994 },
  ]
}
```

Cette requête avancée regroupe les films par réalisateur et conserve la liste des films pour chaque réalisateur.

10. Trouver le film le plus ancien primé (avec un award):

```
moviesDB> db.movies.find({ "awards.wins": { $gt: 0 } }).sort({ year: 1 }).limit(1)
[
  {
    _id: ObjectId('573a13a3f29313caabd0d5a4'),
    plot: 'An athlete swings Indian clubs.',
    genres: [ 'Documentary', 'Short' ],
    runtime: 1,
    title: 'Newark Athlete',
    num_mflix_comments: 3,
    countries: [ 'USA' ],
    fullplot: 'A young man stands before the camera holding a club in each hand, horizontal to the ground. He raises the heads of the two clubs in unison, b
y rotating the clubs without lifting his arms. The film then shows the same footage over again, at different speeds.',
    languages: [ 'English' ],
    directors: [ 'William K.L. Dickson' ],
    rated: 'NOT RATED',
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-03 00:57:26.680000000',
    year: 1891,
    imdb: { rating: 4.9, votes: 827, id: 241763 },
    type: 'movie',
    tomatoes: { lastUpdated: ISODate('2012-09-30T00:00:00.000Z') }
  }
]
moviesDB> |
```

Cette requête filtre les films ayant reçu au moins un prix et les trie par année croissante.

Conclusion

Ce LabWork a permis de se familiariser avec MongoDB, un système de gestion de base de données NoSQL orienté document. Les principales compétences acquises sont:

1. Installation et configuration de MongoDB sur un système Ubuntu

2. **Manipulation de documents** avec le shell MongoDB
3. **Utilisation de l'interface graphique** MongoDB Compass
4. **Création de requêtes complexes** utilisant filtres, opérateurs logiques et agrégations
5. **Exploitation des fonctionnalités avancées** comme les requêtes géospatiales et les opérations sur les tableaux

MongoDB se distingue des bases de données relationnelles traditionnelles par sa flexibilité dans la structure des données et ses performances élevées pour certains types d'opérations, notamment les lectures à grande échelle. Son modèle de données basé sur des documents JSON facilite le développement d'applications modernes, particulièrement celles utilisant JavaScript.

Les exercices réalisés ont démontré la puissance du langage de requête de MongoDB, notamment pour:

- Effectuer des filtres complexes sur les données
- Calculer des statistiques sur les collections
- Manipuler des structures de données imbriquées
- Réaliser des requêtes géospatiales
- Analyser des collections volumineuses

Cette expérience pratique constitue une base solide pour l'utilisation de MongoDB dans des projets futurs, en particulier pour des applications manipulant des données non structurées ou semi-structurées à grande échelle.