

TP n°1 Partie MongoDB

Partie 2 : Cette partie vise à vous familiariser avec le système de gestion de bases de données MongoDB.

Votre mission est de refaire les manipulations suivantes, de les expliquer, en rédigeant, comme pour la première partie, un document clair et concis résumant votre travail.

1) Installation de Mongo DB

référence :

<https://www.mongodb.com/docs/v7.0/tutorial/install-mongodb-on-debian/>

Résultat :

```
ing@ing:~$ sudo systemctl status mongod
● mongod.service - MongoDB Database Server
  Loaded: loaded (/lib/systemd/system/mongod.service; enabled; preset: enabled)
  Active: active (running) since Thu 2025-11-27 10:38:18 CET; 2s ago
    Docs: https://docs.mongodb.org/manual/
   Main PID: 14996 (mongod)
     Memory: 77.7M
       CPU: 175ms
      CGroup: /system.slice/mongod.service
              └─14996 /usr/bin/mongod --config /etc/mongod.conf

nov. 27 10:38:18 ing systemd[1]: Started mongod.service - MongoDB Database Server.
nov. 27 10:38:18 ing mongod[14996]: {"t":{"$date":"2025-11-27T09:38:18.082Z"}, "c": "initandlisten", "id": 1, "s": "Serving connection from 127.0.0.1:49352", "v": 1}
```

2) Se connecter au shell MongoDB

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (<https://www.mongodb.com/legal/privacy-policy>). You can opt-out by running the `disableTelemetry()` command.

The server generated these startup warnings when booting

2025-11-27T10:38:18.115+01:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See <http://dochub.mongodb.org/core/prodnotes-filesystem>

2025-11-27T10:38:18.226+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2025-11-27T10:38:18.226+01:00: For customers running MongoDB 7.0, we suggest changing the contents of the following sysfsFile

2025-11-27T10:38:18.226+01:00: `vm.max_map_count` is too low

test>

3) importer les données dans MongoDB :

```
ing@ing:~$ curl https://atlas-education.s3.amazonaws.com/sampleddata.archive -o sampleddata.archive
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total   Spent   Left  Speed
100  353M  100  353M    0     0  1146k      0  0:05:15  0:05:15 --::--  3260k
ing@ing:~$ mongorestore --archive=sampleddata.archive --port=27017
2025-11-27T10:47:30.034+0100      preparing collections to restore from
2025-11-27T10:47:30.039+0100      reading metadata for sample_guides.planets from
                                   archive 'sampledata.archive'
2025-11-27T10:47:30.039+0100      reading metadata for sample_training.inspections from archive 'sampledata.archive'
2025-11-27T10:47:30.039+0100      reading metadata for sample_training.companies from archive 'sampledata.archive'
2025-11-27T10:47:30.039+0100      reading metadata for sample_training.grades from archive 'sampledata.archive'
2025-11-27T10:47:30.040+0100      reading metadata for sample_geospatial.shipwrecks from archive 'sampledata.archive'
2025-11-27T10:47:30.040+0100      reading metadata for sample_mflix.comments from
```

```

mongosh mongodb://1...      x      ing@ing: ~      x      ing@ing: ~      x      ▾
changing the contents of the following sysfsFile
2025-11-27T10:38:18.226+01:00: vm.max_map_count is too low
-----


test> show dbs
admin              40.00 KiB
config             60.00 KiB
Local              40.00 KiB
sample_airbnb      52.47 MiB
sample_analytics   9.44 MiB
sample_geospatial  980.00 KiB
sample_guides      40.00 KiB
sample_mflix       94.62 MiB
sample_restaurants 5.92 MiB
sample_supplies    968.00 KiB
sample_training    40.44 MiB
sample_weatherdata 2.39 MiB
test>

```



```

...
switched to db sample_mflix
sample_mflix> show collections
comments
embedded_movies
movies
sessions
theaters
users
sample_mflix>

```

Partie 1 – Filtrer et projeter les données

1. Afficher les 5 films sortis depuis 2015

Commande

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

Interprétation

La condition `{ year: { $gte: 2015 } }` sélectionne tous les films dont l'année de production est supérieure ou égale à 2015.

La méthode `.limit(5)` réduit l'affichage aux 5 premiers résultats.

2. Trouver tous les films dont le genre est "Comedy"

Commande

```
sample_mflix> db.movies.find({ genres: "Comedy" })
[ {
  _id: ObjectId('573a1390f29313caabcd4803'),
  plot: 'Cartoon figures announce, via comic strip balloons, that they will move - and move they do, in a wildly exaggerated style.',
  genres: [ 'Animation', 'Short', 'Comedy' ],
  runtime: 7,
  cast: [ 'Winsor McCay' ],
  num_mflix_comments: 0,
  poster: 'https://m.media-amazon.com/images/M/MV5BYzg2NjNhNTctMjUxMi00ZWU4LWI3ZjYtNTI0NTQxNThjZTk2XkEyXkFqcGdeQXVyNzg5OTk2OA@@._V1_SY1000_SX677_AL_.jpg',
  title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics',
  fullplot: 'Cartoonist Winsor McCay agrees to create a large set of drawings that will be photographed and made into a motion picture. The job requires plenty of drawing supplies, and the cartoonist must also overcome some mishaps caused by an assistant. Finally, the work is done, and everyone can see the result'
}
```

Interprétation

Le champ genres est un tableau.

Chercher { genres: "Comedy" } signifie : « retourner tous les films dont la liste des genres contient l'élément Comedy ».

3. Afficher les films sortis entre 2000 et 2005

Commande

```
sample_mflix> db.movies.find(
...   { year: { $gte: 2000, $lte: 2005 } },
...   { title: 1, year: 1, _id: 0 }
... ).pretty()
...
[ {
  title: 'Kate & Leopold', year: 2001 },
{ title: 'Crime and Punishment', year: 2002 },
{ year: 2001, title: 'Glitter' },
{ year: 2000, title: 'In the Mood for Love' },
{ year: 2003, title: 'The Manson Family' },
{ title: 'The Dancer Upstairs', year: 2002 },
{ year: 2000, title: 'State and Main' },
{ title: 'April Captains', year: 2000 },
{ year: 2000, title: 'Chicken Run' },
{ year: 2005, title: 'Fantastic Four' },
{ year: 2002, title: 'Frida' },
{ year: 2001, title: 'From Hell' } ]
```

Interprétation

La condition combine deux opérateurs :

- \$gte : 2000 : année supérieure ou égale à 2000
- \$lte : 2005 : année inférieure ou égale à 2005

La projection { title: 1, year: 1 } affiche uniquement ces deux champs.

4. Afficher les films de genres “Drama” ET “Romance”

Commande

```
sample_mflix> db.movies.find(
...   { genres: { $all: ["Drama", "Romance"] } },
...   { title: 1, genres: 1, _id: 0 }
... )
...
[
  {
    genres: [ 'Drama', 'Romance', 'War' ],
    title: 'The Four Horsemen of the Apocalypse'
  },
  {
    genres: [ 'Drama', 'Romance' ],
    title: 'A Woman of Paris: A Drama of Fate'
  },
  {
    genres: [ 'Drama', 'Romance', 'Thriller' ],
    title: 'He Who Gets Slapped'
  },
]
```

Interprétation

L'opérateur `$all` impose que les deux genres soient présents dans le tableau `genres`.
Le document doit donc contenir à la fois “Drama” et “Romance”.

5. Afficher les films qui n'ont pas le champ `rated`

Commande

```
sample_mflix> db.movies.find(
...   { rated: { $exists: false } },
...   { title: 1, _id: 0 }
... )
...
[ {
  title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics',
},
{ title: 'Gertie the Dinosaur' },
{ title: 'In the Land of the Head Hunters' },
{ title: 'The Perils of Pauline' },
{ title: 'Civilization' },
{ title: 'The Poor Little Rich Girl' },
{ title: 'Wild and Woolly' },
{ title: 'The Blue Bird' },
{ title: 'The Saphead' },
```

Interprétation

L'opérateur `$exists: false` sélectionne tous les documents dans lesquels le champ `rated` n'est pas défini.

La projection affiche uniquement le titre.

Partie 2 – Agrégation (requêtes simples)

6. Afficher le nombre de films par année

```
sample_mflix> db.movies.aggregate([
...   { $group: { _id: "$year", total: { $sum: 1 } } },
...   { $sort: { _id: 1 } }
... ])
...
[
  { _id: 1896, total: 2 }, { _id: 1903, total: 1 },
  { _id: 1909, total: 1 }, { _id: 1911, total: 2 },
  { _id: 1913, total: 1 }, { _id: 1914, total: 3 },
  { _id: 1915, total: 2 }, { _id: 1916, total: 2 },
  { _id: 1917, total: 2 }, { _id: 1918, total: 1 },
  { _id: 1919, total: 1 }, { _id: 1920, total: 4 },
  { _id: 1921, total: 5 }, { _id: 1922, total: 3 },
  { _id: 1923, total: 2 }, { _id: 1924, total: 6 },
  { _id: 1925, total: 3 }, { _id: 1926, total: 6 },
  { _id: 1927, total: 4 }, { _id: 1928, total: 8 }
]
Type "it" for more
```

Interprétation :

- **\$group** regroupe les documents par année (`_id: "$year"`).
- **total: { \$sum: 1 }** compte combien de films il y a pour chaque année.
- **\$sort: { _id: 1 }** trie le résultat par année croissante.

7. Moyenne des notes IMDb par genre

```

sample_mflix> db.movies.aggregate([
...   { $unwind: "$genres" },
...   { $group: { _id: "$genres", moyenne: { $avg: "$imdb.rating" } } },
...   { $sort: { moyenne: -1 } }
... ])
...
[
  { _id: 'Film-Noir', moyenne: 7.397402597402598 },
  { _id: 'Short', moyenne: 7.377574370709382 },
  { _id: 'Documentary', moyenne: 7.365679824561403 },
  { _id: 'News', moyenne: 7.252272727272728 },
  { _id: 'History', moyenne: 7.1696100917431185 },
  { _id: 'War', moyenne: 7.128591954022989 },
  { _id: 'Biography', moyenne: 7.087984189723319 },
  { _id: 'Talk-Show', moyenne: 7 },
  { _id: 'Animation', moyenne: 6.89669603524229 },
  { _id: 'Music', moyenne: 6.883333333333334 },

```

Interprétation :

Sépare les genres, calcule la moyenne des notes IMDb pour chaque genre, puis trie du meilleur au moins bon.

8. Nombre de films par pays

```

sample_mflix> db.movies.aggregate([
...   { $unwind: "$countries" },
...   { $group: { _id: "$countries", total: { $sum: 1 } } },
...   { $sort: { total: -1 } }
... ])
...
[
  { _id: 'USA', total: 10921 },
  { _id: 'UK', total: 2652 },
  { _id: 'France', total: 2647 },
  { _id: 'Germany', total: 1494 },
  { _id: 'Canada', total: 1260 },
  { _id: 'Italy', total: 1217 },
  { _id: 'Japan', total: 786 },
  { _id: 'Spain', total: 675 },
  { _id: 'India', total: 564 },
  { _id: 'Australia', total: 470 },
  { _id: 'Sweden', total: 402 },

```

Interprétation courte :

Compte le nombre de films associés à chaque pays et affiche les pays du plus représenté au moins représenté.

9. Top 5 réalisateurs

```
sample_mflix> db.movies.aggregate([
...   { $unwind: "$directors" },
...   { $group: { _id: "$directors", total: { $sum: 1 } } },
...   { $sort: { total: -1 } },
...   { $limit: 5 }
... ])
...
[
  { _id: 'Woody Allen', total: 40 },
  { _id: 'Martin Scorsese', total: 32 },
  { _id: 'Takashi Miike', total: 31 },
  { _id: 'Steven Spielberg', total: 29 },
  { _id: 'Sidney Lumet', total: 29 }
]
sample_mflix> █
```

Interprétation :

Compte combien de films a réalisé chaque réalisateur et affiche les cinq réalisateurs les plus présents dans la base.

10. Films triés par note IMDb

```
sample_mflix> db.movies.aggregate([
...   { $sort: { "imdb.rating": -1 } },
...   { $project: { title: 1, "imdb.rating": 1 } }
... ])
...
[
  {
    _id: ObjectId('573a1393f29313caabccdbed'),
    title: 'La nao capitana',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13b3f29313caabd3c7ac'),
    title: 'Landet som icke èr',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13b8f29313caabd4d540'),
```

Interprétation :

Trie tous les films par note IMDb décroissante et affiche seulement leur titre et note.

Partie 3 – Mises à jour

11. Ajouter un champ `etat`

```
sample_mflix> db.movies.updateOne(
...   { title: "Jaws" },
...   { $set: { etat: "culte" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
sample_mflix> █
```

Interprétation courte :

Ajoute (ou met à jour) le champ `etat` avec la valeur "culte" pour le film dont le titre est "Jaws".

12. Incrémenter les votes IMDb de 100

```
sample_mflix> db.movies.updateOne(  
...   { title: "Inception" },  
...   { $inc: { "imdb.votes": 100 } }  
... )  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
sample_mflix> █
```

Interprétation :

Augmente la valeur du champ `imdb.votes` de 100 pour le film "Inception".

13. Supprimer le champ `poster` pour tous les films

```
sample_mflix> db.movies.updateMany(  
...   {},  
...   { $unset: { poster: "" } }  
... )  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 21349,  
  modifiedCount: 18044,  
  upsertedCount: 0  
}  
sample_mflix>
```

Interprétation :

Supprime le champ `poster` dans tous les documents de la collection `movies`.

14. Modifier le réalisateur

```
sample_mflix> db.movies.updateOne(  
...   { title: "Titanic" },  
...   { $set: { directors: ["James Cameron"] } }  
... )  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
sample_mflix>
```

Interprétation :

Remplace le champ `directors` du film "Titanic" par un tableau contenant "`James Cameron`" comme unique réalisateur.

Partie 4 – Requêtes complexes

15. Afficher les films dont le titre commence par “Star”

```
sample_mflix> db.movies.find({ title: /^Star/ }, { title: 1 })
[
  { _id: ObjectId('573a1395f29313caabce10cc'), title: 'Stars' },
  { _id: ObjectId('573a1396f29313caabce37ff'), title: 'Star!' },
  {
    _id: ObjectId('573a1396f29313caabce4248'),
    title: 'Start the Revolution Without Me'
  },
  { _id: ObjectId('573a1396f29313caabce57f7'), title: 'Stardust' },
  {
    _id: ObjectId('573a1397f29313caabce68f6'),
    title: 'Star Wars: Episode IV - A New Hope'
  },
  { _id: ObjectId('573a1397f29313caabce7509'), title: 'Starcrash' },
  { _id: ObjectId('573a1397f29313caabce750b'), title: 'Starting Over' },
  {
    _id: ObjectId('573a1397f29313caabce7546'),
    title: 'Star Trek: The Motion Picture'
}
```

Interprétation courte :

Recherche les films dont le titre commence par “Star” en utilisant une expression régulière.

16. Afficher les films avec plus de 2 genres

```
sample_mflix> db.movies.find(
...   { $where: "this.genres.length > 2" },
...   { title: 1, genres: 1 }
... )
...
[
  {
    _id: ObjectId('573a1390f29313caabcd4803'),
    genres: [ 'Animation', 'Short', 'Comedy' ],
    title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics'
  },
  {
    _id: ObjectId('573a1390f29313caabcd50e5'),
    genres: [ 'Animation', 'Short', 'Comedy' ],
    title: 'Gertie the Dinosaur'
  },
  {
```

Interprétation courte :

Sélectionne les films qui possèdent plus de deux genres.

17. Afficher les films de Christopher Nolan

```
sample_mflix> db.movies.find(
...   { directors: "Christopher Nolan" },
...   { title: 1, year: 1, "imdb.rating": 1 }
... )
...
[
  {
    _id: ObjectId('573a139df29313caabcf8dd4'),
    imdb: { rating: 7.6 },
    year: 1998,
    title: 'Following'
  },
  {
    _id: ObjectId('573a13a0f29313caabd05acc'),
    imdb: { rating: 8.5 },
    year: 2000,
    title: 'Memento'
  }
]
```

Interprétation :

Affiche les films où Christopher Nolan est dans la liste des réalisateurs.

Partie 5 – Indexation

19. Créer un index sur year

```
sample_mflix> db.movies.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
sample_mflix>
```

Interprétation courte :

Crée un index croissant sur le champ `year` pour accélérer les requêtes qui filtrent par année.

20. Vérifier les index existants

```
sample_mflix> db.movies.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
sample_mflix>
```

Interprétation courte :

Affiche la liste de tous les index définis sur la collection `movies`.

21. Comparer deux requêtes avec et sans index (explain)

```
sample_mflix> db.movies.find({ year: 1995 }).explain("executionStats")
...
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'sample_mflix.movies',
    indexFilterSet: false,
    parsedQuery: { year: { '$eq': 1995 } },
    queryHash: '108EB0D0',
    planCacheKey: '108EB0D0',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: { year: { '$eq': 1995 } },
      ...
    }
  }
}
```

Interprétation courte :

Affiche le plan d'exécution de la requête (nombre de documents examinés, temps d'exécution) pour voir l'impact de l'index sur `year`.