

# Devoir – Projet MongoDB : “LibDB”

Groupe 3 – Cindy Tchuiesseu & Joseph Rambert

EFREI M1 Data Engineering

## Description Générale

La base de données **libdb** gère des informations sur des **bibliothèques** et leurs **livres**.

- **Collections principales :**
  - **libraries** : décrit chaque bibliothèque (nom, ville, année de création, budget, sections internes).
  - **books** : contient les livres disponibles (titre, auteur, genre, année de publication, nombre d'exemplaires, bibliothèque associée).
- **Collection additionnelle :**
  - **cities** : relie les bibliothèques à leur ville via une référence city\_ref.

## Connexion MongoDB Atlas :

mongodb+srv://User:User@cluster0.ucqgbv9.mongodb.net/

La base est visualisée avec **MongoDB Compass** et hébergée sur **MongoDB Atlas**.

Les fichiers de données (fichiers **.json**) utilisés pour ce projet sont disponibles en ligne sur le dépôt GitHub suivant :

[https://github.com/LucasRambert4/MongoDB\\_Data.git](https://github.com/LucasRambert4/MongoDB_Data.git)

Ce dépôt contient l'ensemble des jeux de données nécessaires à la mise en place et aux tests de la base MongoDB.

## Étape 1 – Création des JSON

Deux collections créées :

1. **books** (fichier books.json)  
Contient plus de 50 livres français, répartis par genre :
  - Roman, Philosophie, Poésie, Théâtre, Satire, Aventure, Autobiographie
  - Champs : \_id, title, author, genre, publication\_year, copies, library\_name
2. **libraries** (fichier libraries.json)  
50 bibliothèques dans différentes villes françaises avec documents imbriqués :
  - Champs : \_id, name, city, established\_year, sections, budget\_euros, city\_ref.
  - sections = tableau d'objets { name, books\_count }

## Étape 2 – Requêtes FIND

1. Trouver un document par **\_id** :

```
db.books.find({ _id: ObjectId("690301d8827bc6bdb73d3504") })
```

2. Trouver un document par **titre** :

```
db.books.find({ title: "Les Misérables" })
```

3. Trier par **genre** et **année décroissante** :

```
db.books.find().sort({ genre: 1, publication_year: -1 })
```

4. Afficher seulement le **titre** des 3 premiers livres :

```
db.books.find({}, { title: 1, _id: 0 }).limit(3)
```

5. Livres avec plus de 5 copies :

```
db.books.find({ copies: { $gt: 5 } })
```

6. Livres du **genre Roman** avec plus de 4 copies :

```
db.books.find({ $and: [ { genre: "Roman" }, { copies: { $gt: 4 } } ] })
```

7. Titres commençant par “L” :

```
db.libraries.aggregate([  
  { $match: { name: { $regex: "L", $options: "i" } } },  
  { $sample: { size: 3 } }  
])
```

## Étape 3 – Mises à jour (UPDATE)

1. Modifier un livre (le premier du genre “Roman”) :

```
db.books.updateOne({ genre: "Roman" }, { $set: { copies: 10 } })
```

2. Incrémenter toutes les copies de +2 :

```
db.books.updateMany({}, { $inc: { copies: 2 } })
```

3. Ajouter une section à une bibliothèque :

```
db.libraries.updateOne(  
  { name: "Bibliothèque Centrale" },  
  { $push: { sections: { name: "Histoire", books_count: 200 } } }  
)
```

4. Renommer un champ :

```
db.books.updateMany({}, { $rename: { "copies": "available_copies" } })
```

5. Supprimer tous les livres :

```
db.books.deleteMany({})
```

#### **Étape 4 – Références (Referred)**

1. Création de la collection **cities** :

```
db.cities.insertMany([  
  { "city_name": "Paris", "region": "Île-de-France" },  
  { "city_name": "Lyon", "region": "Auvergne-Rhône-Alpes" },  
  { "city_name": "Marseille", "region": "Provence-Alpes-Côte d'Azur" },  
  { "city_name": "Bordeaux", "region": "Nouvelle-Aquitaine" },  
  { "city_name": "Toulouse", "region": "Occitanie" },  
  { "city_name": "Nice", "region": "Provence-Alpes-Côte d'Azur" },  
  { "city_name": "Nantes", "region": "Pays de la Loire" },  
  { "city_name": "Lille", "region": "Hauts-de-France" },  
  { "city_name": "Rennes", "region": "Bretagne" },  
  { "city_name": "Grenoble", "region": "Auvergne-Rhône-Alpes" }  
])
```

2. Ajout d'un champ calculé city\_id :

```
db.cities.updateMany({}, [  
  { $set: { city_id: { $strLenCP: "$city_name" } } }  
])
```

3. Jointure entre bibliothèques et villes :

```
db.libraries.aggregate([  
  {  
    $lookup: {  
      from: "cities",  
      localField: "city_ref",  
      foreignField: "city_id",  
      as: "city_info"  
    }  
  }  
])
```

### **Étape 5 – Les Index**

1. Index simple sur le titre :

```
db.books.createIndex({ title: 1 })
```

2. Index composé :

```
db.books.createIndex({ genre: 1, publication_year: -1 })
```

3. Index sur la ville :

```
db.libraries.createIndex({ city: 1 })
```

#### **Justifications :**

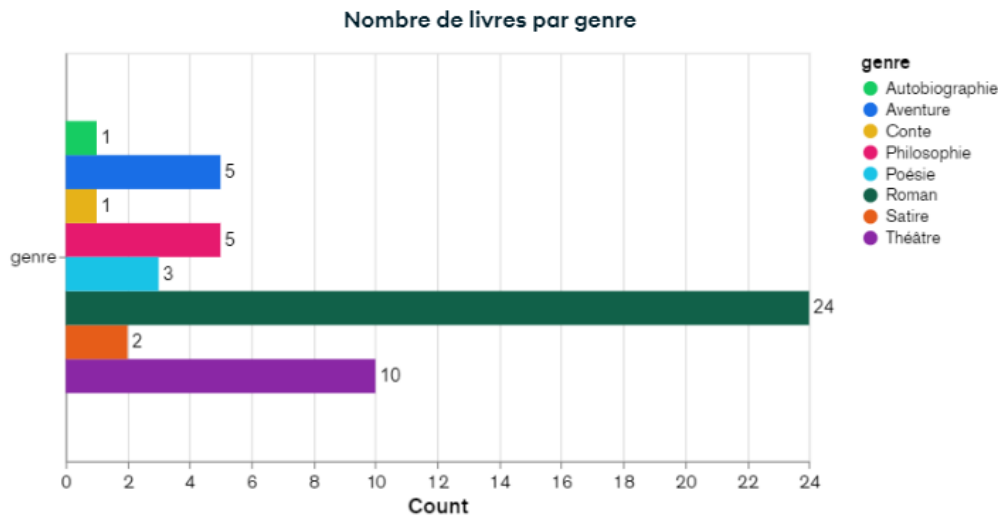
- title : accélère la recherche par titre.
- genre + publication\_year : optimise les tris et filtres combinés.
- city : améliore la recherche de bibliothèques par ville.

## Étape 6 – MongoDB Charts

Deux visualisations :

### 1. Nombre de livres par genre

→ Diagramme à barres basé sur books.genre



### 2. Budgets des bibliothèques par ville

→ Diagramme circulaire basé sur libraries.city et budget\_euros

