

Objectifs

- Maîtriser la modélisation NoSQL avec des références (relations entre collections).
- Savoir utiliser des requêtes avancées dans MongoDB.
- Explorer l'utilisation de `populate` pour charger dynamiquement les données liées.
- Répondre à des cas fonctionnels réalistes via des requêtes complexes.

Contexte et modélisation

La bibliothèque souhaite gérer :

- Des **livres** avec plusieurs auteurs, des catégories, et des informations d'édition.
- Des **auteurs**, chacun ayant potentiellement écrit plusieurs livres.
- Des **utilisateurs** qui peuvent emprunter plusieurs livres.
- Un historique des **emprunts** contenant la date de début, de retour, et une évaluation optionnelle.

Travail 1 : Proposez une modélisation MongoDB (collections et champs) qui répond à ces besoins, en distinguant les objets stockés par référence et ceux stockés en inclusion.

Indice : pensez aux références avec `_id` pour les relations de type « plusieurs à plusieurs ».

Initialisation de la base de données

1. Création des collections

Créer les collections suivantes : `auteurs`, `livres`, `utilisateurs`, `emprunts`.

Indice : utilisez des `ObjectId` pour référencer les auteurs dans les livres et les livres dans les emprunts.

2. Insertion de données

Insérer les exemples suivants :

- 3 auteurs (au moins un avec plusieurs livres)
- 4 livres (deux livres avec plusieurs auteurs)
- 2 utilisateurs
- 3 emprunts (dont un avec une date de retour dépassée)

Travail 2 : Ajouter ces documents dans les collections concernées, en respectant les liens entre les entités.

Requêtes fonctionnelles

Exercice 1 — Requêtes simples

- Lister tous les livres disponibles dans une catégorie donnée.
- Afficher les livres d'un auteur donné.
- Trouver les utilisateurs ayant effectué au moins un emprunt.

Indice : utilisez `find()` avec `$in` ou `$elemMatch` si nécessaire.

Exercice 2 — Requêtes multi-collections

- Pour un utilisateur donné, lister tous les livres qu'il a empruntés avec la date de retour prévue.
- Afficher tous les livres empruntés avec plus de 10 jours de retard.
- Trouver tous les auteurs dont au moins un livre est actuellement emprunté.

Indice : Utilisez une requête sur `emprunts` avec un filtre sur la date de retour, et combinez avec des références.

Exercice 3 — ‘populate’ (Mongoose ou requêtes manuelles)

- Charger tous les emprunts avec les informations complètes sur les livres et les utilisateurs.
- Charger les livres avec tous leurs auteurs détaillés.

Indice : En Mongoose, utilisez `.populate('auteurs')` ou `.populate('livre utilisateur')`.

Exercice 4 — Agrégation

- Calculer le nombre de livres empruntés par utilisateur.
- Calculer la note moyenne d'un livre basé sur les évaluations dans les emprunts.
- Classer les livres les plus empruntés.

Indice : Utilisez `$group`, `$lookup`, `$unwind` dans les pipelines.

Travail d'analyse

Travail 3 : Discussion

- Quelles données stocker en inclusion ? Lesquelles par référence ?
- Quels sont les avantages et inconvénients de l'approche NoSQL pour ce projet ?
- Quelles optimisations seraient nécessaires si la base contient 100 000 livres et 50 000 utilisateurs ?

Pour aller plus loin

- Implémenter une API REST connectée à cette base MongoDB (Node.js ou Symfony).
- Mettre en place une indexation personnalisée sur le champ `categorie`.
- Ajouter des règles de validation (ex. : pas d'emprunt sans disponibilité).