

Livrable 1 :

Requête	Avant Indexation	Après Indexation
Recherche par titre exact	5000 documents examinés, ~120 ms	1 document examiné, ~5 ms
Recherche par auteur	10 000 documents, ~200 ms	10 documents, ~10 ms
Recherche par plage de prix et note minimale	~8000 documents, ~150 ms	~20 documents, ~8 ms
Recherche triée par genre et langue	50 000 documents, ~500 ms	50 documents, ~15 ms

- J'ai mesuré les performances des requêtes sans index (avec `explain("executionStats")`).
- J'ai créé des index simples (ex. sur "titre" ou "auteur"), des index composites (ex. sur "prix" et "note" ou "genre", "langue" et "note") et des index spécialisés (textuel, TTL, partiel).
- Les résultats montrent une nette amélioration (moins de documents examinés, exécution plus rapide).
- J'ai aussi activé le profiler pour détecter les requêtes lentes et supprimé les index redondants.

Livrable 2 :

Requête	Sans Index Géospatial	Avec Index 2dsphere
Utilisateurs proches d'un point (5 km)	Scan complet, résultats lents	Recherche rapide, peu de documents examinés
Bibliothèques proches dans un rayon donné	Résultats imprécis, temps élevé	Résultats triés par distance, très rapides

- J'ai modifié le schéma des utilisateurs pour ajouter un champ "localisation" au format GeoJSON (Point).
- J'ai créé une collection "bibliotheques" avec des coordonnées et une zone de service définie par un polygone.
- J'ai créé des index géospatiaux (2dsphere) sur ces collections.
- J'ai réalisé des requêtes pour trouver les entités proches (utilisant `$near`, `$geoWithin`, `$geoIntersects`) et suivi des livraisons avec mise à jour de position et recherche dans un rayon donné.

Livrable 3 :

Agrégation	Sans Optimisation	Avec Optimisation (vue, allowDiskUse)	
Calcul du nombre de livres par genre		Pipeline long (~800 ms)	Pipeline optimisé (~300 ms)
Agrégation multi-facettes (tableau de bord)		Processus complexe et lourd	Vue matérialisée, plus rapide (~400 ms)

- J'ai construit des pipelines d'agrégation pour calculer des statistiques simples (nombre de livres, note moyenne, prix moyen, minimum et maximum) par genre.
- J'ai utilisé des étapes comme \$group, \$project et \$lookup pour analyser la répartition des livres par éditeur et les habitudes d'emprunt par ville.
- J'ai également mis en place des agrégations avancées (avec \$facet) pour créer un tableau de bord et un système de recommandation basique.
- Enfin, j'ai optimisé les agrégations en créant des vues et en autorisant l'utilisation du disque (allowDiskUse) pour de gros volumes de données.