

[a4paper,11pt]article [utf8]inputenc [T1]fontenc [french]babel listings xcolor geometry tcolorbox
hyperref margin=2.5cm

MongoDB — TP avancé

Lab 3 : Création et utilisation des index

Djebabla Ammar

Objectifs pédagogiques

- Comprendre les différents types d'index disponibles dans MongoDB.
- Apprendre à les créer, à les exploiter et à mesurer leur impact.
- Analyser les performances d'une requête grâce à `.explain("executionStats")`.

Notions clés à connaître

Index simple

Un index sur un seul champ permet d'accélérer les recherches sur ce champ uniquement.

```
db.products.createIndex({ name: 1 })
```

Remarque : 1 indique un tri ascendant, -1 un tri descendant.

Index composé

Un index combiné sur plusieurs champs. Il est utile lorsque les requêtes filtrent ou trient selon plusieurs champs.

```
db.products.createIndex({ name: 1, price: -1 })
```

Index textuel

Permet d'effectuer des recherches textuelles sur du contenu (full-text search). Il faut que le champ soit de type texte.

```
db.products.createIndex({ description: "text" })
```

Index géospatial

Utilisé pour interroger des positions géographiques avec des coordonnées GPS.

```
db.places.createIndex({ location: "2dsphere" })
```

Index wildcard

Permet d'indexer dynamiquement tous les sous-champs d'un objet, souvent utilisé pour des documents non structurés.

```
db.products.createIndex({ "specs.$**": 1 })
```

Analyse de performance avec .explain()

La commande .explain("executionStats") permet d'obtenir des informations détaillées sur le plan d'exécution d'une requête.

```
db.products.find({ name: "Clavier" }).explain("executionStats")
```

À observer :

- executionTimeMillis : durée d'exécution de la requête.
- totalDocsExamined : nombre de documents parcourus.
- stage : indique si l'index est utilisé (IXSCAN) ou non (COLLSCAN).

Lab 3 : Expérimenter les index

Étape 1 — Préparation de la collection

Créer une collection products avec les champs suivants :

- name : nom du produit
- price : prix
- description : texte libre
- location : tableau de coordonnées [longitude, latitude]
- specs : objet avec des clés dynamiques (par ex. color, dpi, etc.)

Insérer au moins 10 documents variés.

Étape 2 — Création des index

Créer les types d'index suivants sur la collection products :

1. Index simple sur name
2. Index composé sur name et price
3. Index textuel sur description
4. Index géospatial sur location
5. Index wildcard sur specs

Afficher les index créés :

```
db.products.getIndexes()
```

Étape 3 — Comparaison de performances

Effectuer une recherche sur un champ indexé (name par exemple) :

- D'abord sans forcer l'utilisation de l'index.
- Ensuite avec `.hint()` pour imposer l'usage d'un index.

```
// Requête sans index forcé
db.products.find({ name: "Clavier" })
  .hint({})
  .explain("executionStats")

// Requête avec index forcé
db.products.find({ name: "Clavier" })
  .hint({ name: 1 })
  .explain("executionStats")
```

Comparer les résultats entre les deux requêtes :

- Temps d'exécution
- Nombre de documents examinés
- Type de scan (COLLSCAN ou IXSCAN)

À retenir

- Les index améliorent les performances des requêtes, mais alourdissent les insertions et mises à jour.
- Il faut bien comprendre les schémas de requêtes pour choisir les bons index.
- Tous les index ne sont pas utiles : un index mal utilisé peut ralentir MongoDB.