

# Comprendre le Sharding dans MongoDB

Djebabla Ammar

17 juin 2025

## Résumé

Ce document présente une explication approfondie du concept de **sharding** dans MongoDB, incluant son architecture, ses avantages et des cas d'utilisation pratiques. Un TP complet permet de mettre en œuvre ces concepts avec des exemples concrets de répartition de données.

## Table des matières

<b>1</b>	<b>Introduction au Sharding</b>	<b>2</b>
<b>2</b>	<b>Architecture détaillée</b>	<b>2</b>
2.1	Composants principaux . . . . .	2
2.2	Types de Sharding . . . . .	3
<b>3</b>	<b>Exemple détaillé : Sharding de 1000 clients</b>	<b>3</b>
3.1	Configuration initiale . . . . .	3
3.2	Découpage des données . . . . .	3
3.3	Insertion des données . . . . .	4
3.4	Vérification . . . . .	4
<b>4</b>	<b>TP : Répartition de données d'étudiants</b>	<b>5</b>
4.1	Objectifs pédagogiques . . . . .	5
4.2	Exercice complet . . . . .	5
4.2.1	Partie 1 : Configuration initiale . . . . .	5
4.2.2	Partie 2 : Insertion des données . . . . .	5
4.2.3	Partie 3 : Analyse . . . . .	5
<b>A</b>	<b>Annexe : Commandes utiles</b>	<b>6</b>

# 1 Introduction au Sharding

Le **sharding** est une technique fondamentale dans les bases de données distribuées qui permet de partitionner horizontalement les données sur plusieurs serveurs (appelés **shards**). Cette approche est particulièrement utile pour gérer de très grandes collections de données qui dépassent les capacités d'un seul serveur.



FIGURE 1 – Architecture typique d'un cluster MongoDB sharded

## 2 Architecture détaillée

### 2.1 Composants principaux

- **mongos** : Le routeur qui dirige les opérations de lecture/écriture vers les shards appropriés
- **Config Servers** : Stockent les métadonnées du cluster (namespace, plages de clés, etc.)
- **Shards** : Serveurs contenant les données partitionnées

## 2.2 Types de Sharding

1. **Sharding par plage (Range-based)** : Partitionnement basé sur des plages de valeurs
2. **Sharding par hachage (Hash-based)** : Utilisation d'une fonction de hachage pour distribuer les données
3. **Sharding par zone (Zone-based)** : Attribution de plages spécifiques à des shards particuliers

## 3 Exemple détaillé : Sharding de 1000 clients

### 3.1 Configuration initiale

```
1 // Activer le sharding pour la base de données
2 use admin
3 sh.enableSharding("testdb")
4
5 // Créer la collection shardée
6 use testdb
7 db.createCollection("clients")
8
9 // Choisir la clé de sharding
10 sh.shardCollection("testdb.clients", { client_id: 1 })
```

Listing 1 – Initialisation du sharding

#### Information

**Remarque** : Le choix de la clé de sharding est crucial. Une bonne clé doit avoir :

- Une cardinalité élevée
- Une distribution uniforme
- Des requêtes qui l'utilisent fréquemment

### 3.2 Découpage des données

```
1 // Découper la plage de données en 3 chunks
2 sh.splitAt("testdb.clients", { client_id: 333 })
3 sh.splitAt("testdb.clients", { client_id: 666 })
4
5 // Répartir les chunks sur les shards
6 sh.moveChunk("testdb.clients",
7   { client_id: 100 }, "shard1")
8 sh.moveChunk("testdb.clients",
9   { client_id: 500 }, "shard2")
10 sh.moveChunk("testdb.clients",
```

```
11 { client_id: 800 }, "shard3")
```

Listing 2 – Découpage en chunks

### 3.3 Insertion des données

```
1 // Insertion de 1000 documents
2 for (let i = 1; i <= 1000; i++) {
3   db.clients.insertOne({
4     client_id: i,
5     nom: "Client_" + i,
6     email: "client" + i + "@example.com",
7     date_creation: new Date(),
8     solde: Math.random() * 1000
9   })
10 }
```

Listing 3 – Insertion des documents

### 3.4 Vérification

```
1 use config
2
3 // Afficher la distribution des chunks
4 db.chunks.aggregate([
5   { $match: { ns: "testdb.clients" }},
6   { $group: {
7     _id: "$shard",
8     count: { $sum: 1 },
9     min: { $min: "$min.client_id" },
10    max: { $max: "$max.client_id" }
11   }
12 }
13 ])
```

Listing 4 – Vérification de la distribution

TABLE 1 – Répartition des données

Shard	Nombre de chunks	Plage min	Plage max
shard1	1	1	332
shard2	1	333	665
shard3	1	666	1000

## 4 TP : Répartition de données d'étudiants

### 4.1 Objectifs pédagogiques

- Comprendre la répartition automatique des données
- Maîtriser les commandes de gestion du sharding
- Analyser la distribution des données

### 4.2 Exercice complet

#### 4.2.1 Partie 1 : Configuration initiale

```
1 // 1. Créer la base de données et la collection
2 use etudiantsDB
3 sh.enableSharding("etudiantsDB")
4 db.createCollection("notes")
5
6 // 2. Activer le sharding sur la collection
7 sh.shardCollection("etudiantsDB.notes",
8   { etudiant_id: "hashed" })
9
10 // 3. Vérifier la configuration
11 sh.status()
```

#### 4.2.2 Partie 2 : Insertion des données

```
1 // 4. Insérer 9000 documents
2 for (let i = 1; i <= 9000; i++) {
3   db.notes.insertOne({
4     etudiant_id: i,
5     nom: "Etudiant_" + i,
6     note_math: Math.floor(Math.random() * 20),
7     note_physique: Math.floor(Math.random() * 20),
8     promotion: 2020 + (i % 4)
9   })
10 }
```

#### 4.2.3 Partie 3 : Analyse

```
1 // 5. Analyser la distribution
2 use config
3 db.chunks.aggregate([
4   { $match: { ns: "etudiantsDB.notes" } },
```

```
5   { $group: {
6     _id: "$shard",
7     count: { $sum: 1 },
8     min: { $min: "$min.etudiant_id" },
9     max: { $max: "$max.etudiant_id" }
10  }
11 }
12 ])
13
14 // 6. Vérifier l'équilibrage
15 db.adminCommand({ getBalancerState: 1 })
```

## Conclusion

Le sharding est une technique puissante pour gérer de grandes quantités de données dans MongoDB. Ce document a présenté :

- Les concepts fondamentaux du sharding
- L'architecture détaillée d'un cluster shardé
- Des exemples pratiques de configuration
- Un TP complet pour mettre en œuvre ces concepts

Pour aller plus loin, consultez la documentation officielle MongoDB sur le sharding : <https://docs.mongodb.com/manual/sharding/>

## A Annexe : Commandes utiles

```
1 // Voir l'état du sharding
2 sh.status()
3
4 // Activer/désactiver le balancer
5 sh.startBalancer()
6 sh.stopBalancer()
7
8 // Forcer le déplacement d'un chunk
9 sh.moveChunk("db.collection",
10 { field: value }, "shardName")
11
12 // Voir les statistiques de distribution
13 db.collection.getShardDistribution()
```

Listing 5 – Commandes de gestion du sharding