SQL e NoSQL : Quelle est la différence et lequel choisir ?

* juillet 30, 2021
* [Technologies](https://www.saagie.com/fr/blog/category/technologies/)

***Vous connaissez un peu le SQL, mais avez entendu parler du NoSQL et vous ne savez pas lequel choisir pour vos projets ?***

***L’une des***[***bonnes pratiques à adopter dans un projet data science***](https://www.saagie.com/fr/blog/5-pratiques-a-adopter-dans-un-projet-de-data-science/)***est de bien choisir ces technologies dès le début du projet pour qu’il n’y ait pas de frein à celui-ci. Nous allons donc voir les particularités du SQL et du NoSQL.***

**1. Définitions du SQL et du NoSQL**

**A. SQL, le langage des données structure**

**Le SQL, qui signifie Structured Query Language, est un langage informatique normalisé permettant de communiquer avec une base de données.**

Les bases de données relationnelles ont longtemps été les plus populaires. Il s’agit de bases de données dans lesquelles l’information est organisée avec des tableaux à deux dimensions nommées « tables ». Les lignes correspondent aux enregistrements. Chaque enregistrement contient un groupe d’informations – les attributs – relatives à un sujet. La plupart des systèmes de gestion de bases de données relationnelles reconnaissent le SQL.

Plus concrètement, le SQL a pour but de stocker, de manipuler et de retrouver ces données. Il permet également d’effectuer des requêtes, de mettre à jour les données ou de les réorganiser, de créer et de modifier le schéma et la structure d’un système de base de données.

**B. NoSQL, le SQL des bases de données non relationnelles**

**NoSQL signifie à la fois « Not only SQL », car certains langages NoSQL comprennent le langage SQL en plus de leur propre capacité, et « Non-relationnel » parce qu’il ne peut pas stocker facilement des données relationnelles.**

La particularité des bases de données NoSQL est qu’elles n’utilisent pas le modèle relationnel. Il n’y a donc pas de tableau avec des caractéristiques et nombres d’attributs fixes. Les schémas sont donc absents ou flexibles. Cela permet de regrouper des données ayant des structures différentes. Les bases de données NoSQL peuvent également être distribuées, c’est-à-dire qu’elles peuvent être stockées sur plusieurs systèmes, plusieurs serveurs par exemple. Pour approfondir le sujet, nous vous conseillons l’article [NoSQL : Tout comprendre sur les bases de données non relationnelles](https://datascientest.com/nosql).

**2. Avantages et inconvénients du SQL et du NoSQL**

**A. SQL et NoSQL solutions différentes au théorème de CAP**

**Le théorème de Brewer**, aussi appelé théorème de **CAP**, a été formalisé en 2000 par Eric A. Brewer. Il s’intéresse aux trois propriétés fondamentales des bases de données :

* **Cohérence** (**C**onsistency en anglais) : la donnée a toujours un seul état visible à un instant précis, peu importe le nombre de copies ;
* **Disponibilité** (**A**vailability en anglais) : la donnée est disponible tant que le système tourne ;
* **Distribution** (**P**artition tolerance en anglais) : toute requête fournit un résultat correct, peu importe le nombre de serveurs.

***Selon le théorème de CAP, dans toute base de données, vous ne pouvez respecter au plus que 2 propriétés parmi la cohérence, la disponibilité et la distribution.***

Il faut donc prioriser les besoins du projet et choisir la technologie adaptée au bon couple de propriétés.

Pour le couple **CA** (cohérence et disponibilité), on utilisera des langages SQL comme **Oracle, MySQL, SQLServer**.

Pour le couple **CP** (cohérence et distribution), on choisira des langages NoSQL comme **HBase, BigTable, MongoDB\* et CosmosDB\*.**

Pour le couple **AP** (disponibilité et distribution), on privilégiera des langages NoSQL comme **Elasticsearch, Spark, Neo4j, OrientDB, FlockDB, Redis, SimpleDB\*, Memcached\*, DynamoDB\*, CouchBase\*, Cassandra\*.**

**\*Les langages suivis d’un astérisque ont la capacité de pouvoir passer de CP à AP ou inversement.**

**B. Cas d’usages du SQL et du NoSQL**

[La data est un champ en constante évolution](https://www.saagie.com/fr/blog/data-evolution-constante/) à la fois par la quantité de données et leur diversité. La création du NoSQL après le SQL répond donc à ces changements.

**Le SQL est à privilégier lorsque les données sont structurées et que leurs relations sont fondamentales.** Si les bases de données sont complexes, ce système est principalement choisi.

Au contraire, si les données ne sont pas structurées ou changent de format avec le temps, le NoSQL sera pertinent. **Les bases de données NoSQL sont plus adaptées lorsque l’on manipule de très larges volumes de données dont les relations entre celles-ci ne sont pas particulièrement importantes.** Lors des augmentations de volumes de données, deux choix sont possibles :

* **la scalabilité verticale**: augmentation de capacité sur un serveur existant,
* **la scalabilité horizontale** : augmentation du nombre de serveurs.

**Le SQL permet uniquement la scalabilité verticale alors que le NoSQL autorise à la** **fois la scalabilité verticale et horizontale, car il est distribué**. On comprend donc rapidement la difficulté à laquelle peut faire face le SQL en cas de très large volume de données.

Le NoSQL est quant à lui plus limité dans la façon de faire des recherches. On accède à ces données par leur identifiant, nommé « clé primaire ». Il est donc facile de retrouver une donnée, mais trouver l’ensemble des données supérieures à une valeur, par exemple, devient beaucoup plus complexe. Le SQL n’a lui pas de soucis avec ce genre de besoin.

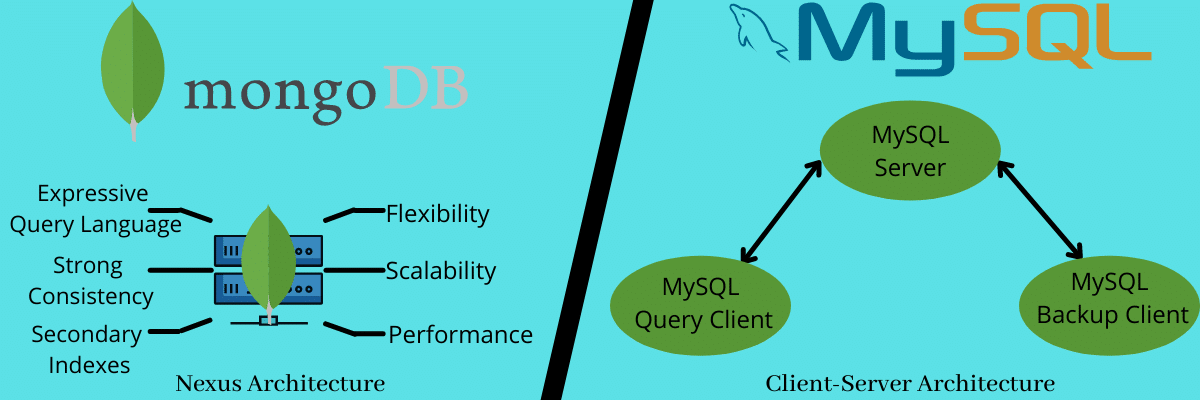
Pour vous aider à utiliser le bon langage lors de vos projets, voici [13 bonnes ressources pour apprendre SQL et NoSQL](https://geekflare.com/fr/learn-sql-nosql/).

***Le SQL et le NoSQL resteront tous les deux utilisés à l’avenir. Chaque langage a ses avantages. En revanche, les inconvénients peuvent amener, en fonction du besoin, à utiliser l’autre langage.***

## Comparaison des SGBD MongoDB et MySQL

Voici quelques paramètres sur la base desquels nous allons comparer MongoDB vs MySQL.

### Architecture

[](https://kinsta.com/wp-content/uploads/2021/11/MongoDBvsMySQL_-Architecture.png)Architecture MongoDB vs MySQL

L’architecture constitue la base de tout système et établit le cadre dans lequel toutes les caractéristiques et fonctionnalités peuvent être introduites. Il est donc important de comparer l’architecture de MongoDB et de MySQL et de les comprendre de près afin de déterminer quel sera le meilleur choix pour votre application.

#### MongoDB

Il a pour philosophie de conception l’architecture Nexus, qui combine les fonctionnalités des bases de données relationnelles. Il peut répondre aux besoins des applications modernes en offrant une grande évolutivité, une disponibilité globale et un schéma flexible. Par conséquent, il est assez facile d’apporter des modifications à sa conception.

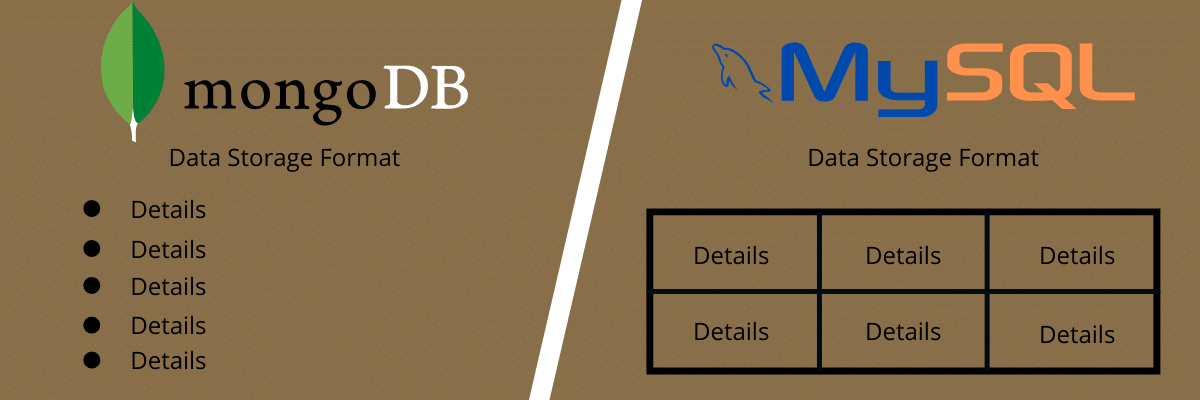
En outre, MongoDB comprend des pilotes officiels pour les principaux environnements de développement comme [AWS](https://kinsta.com/fr/blog/panne-aws/), Azure et Kubernetes, les langages de programmation comme JavaScript, Python, etc. et les frameworks comme Django.

#### MySQL

MySQL, quant à lui, comprend une architecture client-serveur avec un stockage optimisé pour offrir de hautes performances et le multi-threading. Sa documentation présente également quelques techniques d’optimisation des performances qui portent sur la configuration plutôt que sur le réglage fin des mesures SQL.

**Gagnant**: C’est une égalité.

### Format de stockage des données

[](https://kinsta.com/wp-content/uploads/2021/11/Data-Storage-Format.png)MongoDB vs MySQL : Format de stockage des données

#### MongoDB

Le format de stockage des données dans MongoDB peut ressembler à ceci :

{

Account Number: 1234567890

First Name: "Jon"

Last Name: "Doe"

Branch Name: "Los Angeles"

}

Vous pouvez voir que le format est de type JSON, et que vous pouvez facilement apporter des modifications aux données, c’est-à-dire que vous pouvez ajouter des données supplémentaires, supprimer certaines informations et modifier des données sans problème. L’exemple ci-dessus montre qu’il n’y a pas de schéma fixe pour la base de données, ce qui introduit plus de flexibilité.

#### MySQL

Le format de stockage des données dans MySQL, d’autre part, ressemble à quelque chose comme ceci :

| **Account Number** | **First Name** | **Last Name** | **Branch Name** |
| --- | --- | --- | --- |
| 12345678901 | Jon | Doe | Los Angeles |
| 12345678902 | Jane | Doe | Seattle |

Le tableau ci-dessus montre comment MySQL organise les données sous forme de lignes et de colonnes. Il possède une structure propre et rigide qu’il est difficile de modifier par rapport à MongoDB. C’est parce que vous ne pouvez pas introduire une ligne ou une colonne spate ; vous devez faire la conception de telle sorte que pour chaque ligne, il y a une colonne et vice versa ; sinon, cela violera le schéma. Mais, dans MongoDB, vous pouvez facilement modifier les données.

Il existe une légère différence de terminologie entre MongoDB et MySQL :

| **MongoDB** | **MySQL** |
| --- | --- |
| Collection | Table |
| Document | Row |
| Champ | Colonne |
| Liaison et incorporation | Joindre |
| Sharding | Partitionnement |
| RepISet | Réplication |

**Gagnant**: Le format de stockage des données de MongoDB est plus facile à modifier.

### Flexibilité des schémas

La base de données que vous choisissez doit offrir la flexibilité de modifier la conception ou le schéma de votre base de données en fonction des besoins variables. Sinon, cela devient vraiment ennuyeux lorsqu’il y a ne serait-ce qu’un léger changement dans les exigences.

Alors, découvrons MongoDB et MySQL en fonction de la flexibilité de leurs schémas.

#### MongoDB

MySQL offre un schéma flexible qui permet aux utilisateurs de modifier la conception en fonction des exigences, notamment pour les applications Big Data. Il vous permet de combiner et de stocker facilement différents types de données et de modifier le schéma de manière dynamique sans temps d’arrêt. Vous pouvez stocker plusieurs documents dans une collection même sans aucune relation entre eux, car il s’agit d’un système de base de données non relationnel. Il utilise des documents de type JSON ayant des schémas optionnels.

Cependant, il ne dispose pas de transactions ni de jointures ; vous devez donc optimiser fréquemment le schéma en fonction de la manière dont l’application accède aux données.

#### MySQL

Dans MySQL, vous devez définir clairement les colonnes et les tables avant de stocker des données avec des lignes et des colonnes. Ici, chaque champ comprend une ligne et une colonne. Cela signifie que le stockage des données ne vous donne pas beaucoup de flexibilité comme dans MongoDB. Cela signifie également un processus de déploiement et de développement plus lent.

Mais si vous avez un schéma fixe pour vos applications, MySQL est la meilleure solution. Il vous offrira une meilleure cohérence des données sans avoir à modifier sans cesse le schéma ou à perdre du temps à le faire. Mais là encore, si vous avez des besoins changeants, MongoDB peut être une meilleure option pour vous.

**Gagnant**: De toute évidence, MongoDB offre une plus grande flexibilité de schéma.

### Langage de requête utilisé

Il est essentiel de savoir quelle base de données utilise quel langage de requête. Cela vous aidera à comprendre lequel peut être plus pratique pour vous, au lieu de vous embrouiller après l’avoir installé.

#### MongoDB

MongoDB utilise le langage de requête MongoDB (MQL) au lieu de [SQL](https://kinsta.com/fr/blog/injections-sql/). Il est expressif et riche et prend en charge les fonctions CRUD, qui vous permettent de créer, lire, mettre à jour et supprimer des données. En outre, il facilite également l’agrégation de données, les requêtes géospatiales et la recherche de texte.

Si vous voulez faire une requête sur des données, vous devez définir des documents dont les propriétés correspondent à celles des résultats attendus. Cela signifie que vous devez effectuer des opérations de requête pour récupérer des données dans la base de données, comme **db.collection.find()**. MongoDB exécute les requêtes en utilisant généralement des opérateurs liés à l’aide de JSON. En outre, il prend en charge les requêtes OR et AND.

Cependant, MongoDB n’utilise jamais d’opérations de jointure, et il ne dispose d’aucun autre opérateur équivalent.

#### MySQL

D’autre part, MySQL utilise SQL comme les autres bases de données relationnelles. Il peut rassembler des données provenant de différentes tables en prenant en charge la fonctionnalité de jointure. C’est ce qui rend une base de données relationnelle comme MySQL « relationnelle » Cette opération vous permet de lier les données de plusieurs tables dans une requête.

Cela dit, SQL a un :

* Data Definition Language (DDL) pour créer, supprimer et modifier des tables
* Data Transaction Language (DTL) avec des opérations telles que commit et rollback
* Data Manipulation Language (DML) avec des options telles que l’insertion, la suppression et la mise à jour de rangées
* Data Control Language (DCL) avec des commandes de révocation et d’octroi

**Gagnant**: C’est une égalité.

### Performance et vitesse

[](https://kinsta.com/wp-content/uploads/2021/11/Performance-and-Speed.png)MongoDB vs MySQL : Performance et vitesse

Les performances et la vitesse sont des éléments que vous ne pouvez jamais ignorer lorsque vous choisissez une base de données. Vous devez savoir à quoi vous attendre avec telle ou telle base de données et à quelles fins. Et pour les professionnels occupés comme les développeurs et les administrateurs, chaque seconde est essentielle.

Par conséquent, vous devez choisir une base de données qui peut offrir de meilleures performances pour soutenir votre productivité et non l’inverse. Comparons donc leur vitesse et leurs performances.

Étant donné que MongoDB et MySQL ont des approches de stockage de données différentes, il est un peu difficile d’évaluer leurs performances. Vous pouvez comparer deux bases de données SQL à l’aide de certains repères standard, mais il est difficile de faire de même avec des bases de données non relationnelles.

Nous allons toutefois comparer MongoDB et MySQL sur la base des opérations courantes et de leurs performances en présence de volumes de données plus importants.

#### MongoDB

Comme MongoDB stocke un grand volume de données non structurées et suit une approche de stockage basée sur les documents, il est relativement plus rapide que MySQL. Cela signifie que MongoDB stocke les données dans un seul document pour une entité et permet une lecture ou une écriture plus rapide des données. Des fonctionnalités comme la réplication peuvent y contribuer. Ses performances sont également meilleures lorsqu’il s’agit d’objets grâce à son stockage d’objets de type Jason.

En outre, MongoDB n’implique jamais de verrouillage du fournisseur, ce qui vous donne la liberté d’améliorer les performances en utilisant des alternatives si vous n’êtes pas satisfait d’un service.

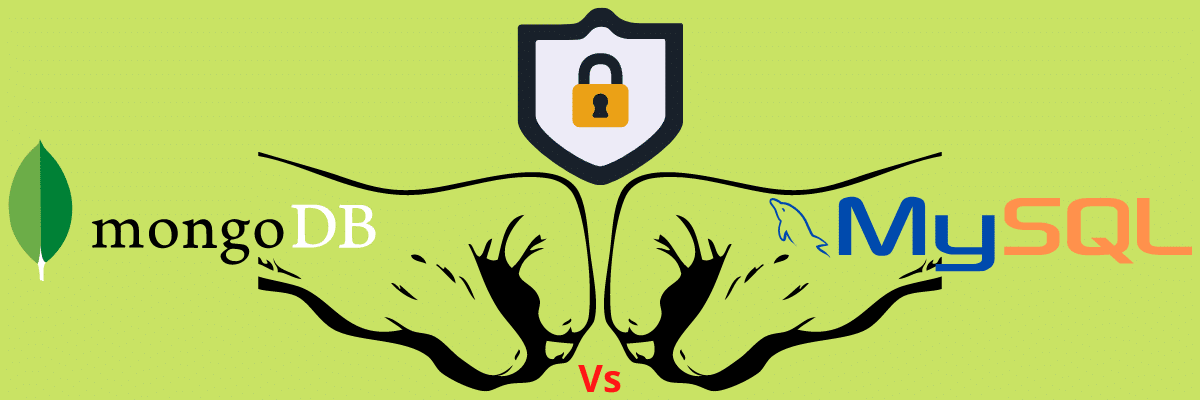
#### MySQL

MySQL peut présenter des performances lentes lorsqu’il s’agit de traiter un énorme volume de données. C’est parce qu’il stocke les tables d’une manière normalisée. Et si vous voulez modifier des données ou les extraire, vous devez passer par de nombreuses tables pour écrire et lire des données, ce qui augmente les charges du serveur et affecte ses performances. Mais vous pouvez opter pour MySQL pour les opérations transactionnelles.

Il nécessite une structure de données définie pour pouvoir ajouter des données dans la base de données. Par conséquent, il n’est pas adapté si vous souhaitez stocker des données non structurées. Et parfois, il devient également difficile de concevoir un schéma approprié lorsqu’il s’agit de données complexes.

**Gagnant**: MongoDB est plus rapide et plus performant.

### Sécurité

[](https://kinsta.com/wp-content/uploads/2021/11/Security.png)MongoDB vs MySQL : La sécurité

La sécurité est toujours l’un des principaux critères de comparaison entre deux systèmes, compte tenu de la multiplication des [cyber-attaques](https://kinsta.com/fr/blog/captcha-wordpress/) dans le monde. Il est donc important de comparer MongoDB et MySQL afin de déterminer lequel des deux est le plus sûr pour vos applications.

#### MongoDB

MongoDB exploite les contrôles d’accès basés sur les rôles avec des autorisations flexibles pour les utilisateurs et les appareils. Chaque utilisateur se voit attribuer un rôle en fonction duquel il reçoit des autorisations spécifiques pour accéder aux données et effectuer des opérations. Par exemple, les utilisateurs comme les employés de haut niveau ont des niveaux d’autorisation plus élevés, leurs privilèges sont donc plus importants.

C’est une façon de sécuriser votre base de données afin qu’aucun utilisateur non autorisé ou aucune attaque ne puisse accéder à votre base de données et l’exploiter. En outre, MongoDB facilite également le cryptage [TLS (Transport Layer Security)](https://kinsta.com/fr/blog/tls-1-3/) et un protocole de sécurité appelé [SSL (Secure Sockets Layer)](https://kinsta.com/fr/base-de-connaissances/tls-vs-ssl/) pour une sécurité accrue. Vous pouvez également écrire des documents chiffrés dans des collections de données avec une clé maîtresse pour obtenir un chiffrement des données au repos.

#### MySQL

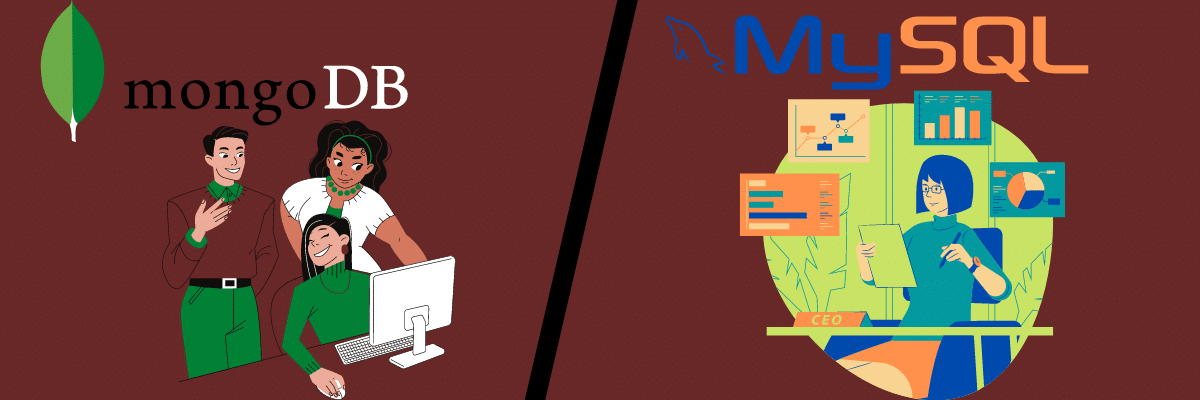
D’autre part, MySQL dispose de contrôles d’accès basés sur les privilèges. Il prend également en charge les facilités de cryptage comme MongoDB avec un modèle d’authentification similaire, comprenant l’autorisation, l’authentification et l’audit. Vous pouvez accorder des rôles et des privilèges aux utilisateurs, leur permettant d’obtenir des autorisations d’accès pour les ensembles de données et les opérations. En outre, vous pouvez également demander TLS et SSL pour plus de sécurité.

Dans sa récente mise à jour, MySQL a également inclus la prise en charge du double mot de passe afin de garantir une plus grande sécurité pour l’accès aux données.

Bien que les deux semblent se livrer une compétition au coude à coude en matière de sécurité, MySQL est considéré comme plus sûr. La raison réside dans son architecture et son schéma rigides, qui offrent une meilleure cohérence et fiabilité des données.

**Gagnant**: MySQL est plus sûr (être rigide n’est pas aussi mauvais).

### Évolutivité

[](https://kinsta.com/wp-content/uploads/2021/11/Scalability.png)MongoDB vs MySQL : Évolutivité

À mesure que votre application se développe avec une base d’utilisateurs et un trafic accrus, vous devez être en mesure de la faire évoluer en douceur pour répondre aux demandes changeantes. Et si votre application n’évolue pas, les clients peuvent avoir une mauvaise expérience en l’utilisant avec des plantages fréquents, des décalages et des temps d’arrêt. Personne n’aime utiliser de tels systèmes, et ils peuvent partir sur d’autres alternatives, si ce n’est pas maintenant, du moins plus tôt.

Il est donc important de prendre soin de l’évolutivité des applications, et la base de données que vous choisissez peut affecter cette évolutivité. Comparons donc MongoDB et MySQL en fonction de l’évolutivité qu’ils offrent.

#### MongoDB

MongoDB est extrêmement évolutif, ce qui est l’une des principales raisons pour lesquelles elle est utilisée dans les sites web en pleine croissance, les systèmes CMS et les boutiques eCommerce. Il peut évoluer horizontalement en utilisant un processus appelé sharding. Un shard est une partie d’une base de données, et le sharding est une technique de distribution des données sur plusieurs collections et machines. Il vous permet de déployer des systèmes avec des opérations à haut débit et de grands ensembles de données.

Avec une plus grande évolutivité, vous pouvez facilement créer plusieurs grappes de serveurs en ajoutant de plus en plus de serveurs dans votre base de données en fonction de vos besoins. Il vous permet de stocker une réplique de vos ensembles de données dans un cluster sharded avec des performances d’écriture et de lecture plus élevées pour prendre en charge des applications de différentes échelles. Il vous permet également de garantir la sauvegarde de vos données et de ne jamais les perdre en cas de cyber-attaques ou de catastrophes.

Sans oublier que la division de la charge et des ensembles de données en différents serveurs vous permet également de les exécuter à un coût moindre par rapport à un seul serveur avec toutes les données qui nécessite un matériel haut de gamme et coûteux. Un autre avantage du sharding dans MongoDB est qu’il maximise votre espace disque et offre un équilibrage de charge dynamique.

En outre, MongoDB prend en charge le sharding basé sur la plage ou le partitionnement des données, ainsi que le routage transparent des requêtes et la distribution automatique du volume de données.

#### MySQL

En ce qui concerne MySQL, l’évolutivité est limitée. Il vous offre deux choix pour la mise à l’échelle de votre application : la création de répliques en lecture ou la mise à l’échelle verticale. Il permet la réplication des données et la mise à l’échelle verticale par le biais du clustering pour aider à améliorer l’évolutivité et les performances de l’application via différents types de synchronisation.

La différence notable ici est que MySQL offre une mise à l’échelle verticale, tandis que MongoDB offre une mise à l’échelle horizontale avec plus de flexibilité. Maintenant, l’évolutivité verticale signifie que le système vous permet d’augmenter la charge en augmentant les spécifications du CPU ou de la RAM dans un seul serveur avec une limite supérieure.

Si vous souhaitez effectuer une réplication, c’est facile avec les réplications en lecture. Cela vous permet de créer des copies en lecture seule de votre base de données et de les ajouter à différents serveurs, mais avec des limitations – l’une étant sur le nombre total de répliques que vous pouvez ajouter. En raison de cette limitation, vous pouvez rencontrer des problèmes concernant les applications qui lisent et écrivent régulièrement pour vos bases de données (ou qui sont lourdes en écriture).

Bien que la réplication multi-main soit introduite dans MySQL, sa mise en œuvre est encore limitée par rapport aux fonctionnalités que vous obtenez dans MongoDB. Elle peut ajouter plus d’échelle d’écriture, mais uniquement pour des applications distinctes ; chacune d’entre elles pourrait écrire sur plusieurs mains et obtenir l’échelle.

De plus, MySQL n’implique aucune implémentation standard pour le sharding. Même s’il propose deux méthodes de sharding – MySQL Fabric (framework de sharding) et le sharding automatique – les gens les déploient rarement en raison des nombreux obstacles et limitations. C’est pourquoi des entreprises comme Facebook utilisent leur propre framework de sharding.

Si vous exploitez le sharding pour l’évolutivité, veillez à choisir la bonne clé de sharding car une mauvaise clé peut entraîner une inflexibilité du système. Par exemple, la modification de la clé de sharding peut avoir un effet négatif sur une application, les transactions nodales et l’emplacement. De plus, des problèmes comme la cohérence des données peuvent apparaître si les changements de sharding sont incomplets.

Ainsi, lorsque vous utilisez MySQL, vous devez prendre soigneusement les bonnes décisions pour les changements de schéma et le mappage entre les partitions de données, les clés de sharding, les nœuds et les bases de données.

**Gagnant**: MongoDB offre une meilleure évolutivité, tandis que MySQL présente de nombreuses limitations et peut entraîner des incohérences et des problèmes si certains processus ne sont pas effectués correctement, comme expliqué précédemment.