

## **INTRODUCTION**

Le monde des bases de données offre une multitude d'options pour stocker et gérer des données. Deux approches principales se démarquent : les bases de données NoSQL et les bases de données relationnelles (SQL).

Les bases de données relationnelles, souvent désignées sous le terme SQL (Structured Query Language), ont dominé le paysage des bases de données pendant des décennies. Elles sont basées sur le modèle de données relationnel et utilisent des tables avec des relations prédéfinies pour stocker les données.

En revanche, les bases de données NoSQL (Not Only SQL) ont émergé plus récemment et proposent une alternative aux bases de données relationnelles. Elles adoptent une approche plus flexible et adaptée aux besoins des applications modernes.

Dans cette présentation, nous allons comparer deux représentants populaires : MongoDB, une base de données NoSQL, et SQL, représentant les bases de données relationnelles.

Nous allons explorer les fonctionnalités, les différences de modèle de données, les performances et d'autres aspects pertinents pour vous aider à comprendre les avantages et les cas d'utilisation de chacun de ces systèmes.

Prêt à plonger dans cette comparaison entre NoSQL (MongoDB) et SQL ? Continuez avec notre première diapositive sur les fonctionnalités de MongoDB.

## **NoSQL (MongoDB)**

MongoDB est un système de gestion de base de données NoSQL populaire. Il utilise un modèle de données centrées sur les documents, ce qui signifie que les données sont transférées sous forme de documents JSON (JavaScript Object Notation).

Une des fonctionnalités clés de MongoDB est sa structure flexible. Contrairement aux bases de données relationnelles, MongoDB n'impose pas de schéma prédéfini. Cela signifie que vous pouvez stocker des documents avec des structures différentes au sein de la même collection.

MongoDB offre également une évolutivité horizontale. Vous pouvez facilement ajouter de nouveaux serveurs pour gérer une charge élevée et répartir les données. Cela permet une extensibilité plus facile par rapport aux bases de données relationnelles traditionnelles.

Une autre caractéristique importante de MongoDB est sa haute disponibilité et sa tolérance aux pannes. Il prend en charge la réplication, ce qui signifie que vous pouvez avoir plusieurs copies de vos données réparties sur différents serveurs pour garantir la même disponibilité en cas de défaillance d'un serveur.

Enfin, MongoDB est également adapté pour stocker de grandes quantités de données non intégrées, telles que des données de capteurs, des journaux d'événements ou des données provenant de sources diverses.

## **SQL**

Passons maintenant aux bases de données relationnelles, souvent associées à SQL (Structured Query Language). Elles utilisent un modèle de données tabulaires avec des relations prédéfinies entre les tables.

Contrairement à MongoDB, les bases de données relationnelles ont un schéma rigide. Vous devez définir la structure de vos tables et les relations entre elles avant de pouvoir stocker les données.

Les bases de données relationnelles sont conçues pour garantir l'intégrité des données et maintenir la cohérence en utilisant des contraintes et des vérifications. Elles prennent en charge les transactions ACID (Atomicité, Cohérence, Isolation, Durabilité), offrant une garantie de cohérence des données même en cas d'échec ou de panne.

SQL est également puissant en termes de requêtes complexes. Vous pouvez utiliser des opérations telles que les jointures, les regroupements et les tris pour interroger et analyser vos données de manière sophistiquée.

Les bases de données relationnelles sont souvent utilisées dans des domaines tels que la finance, les applications de gestion d'inventaire et les applications d'entreprise qui entraînent des transactions fiables et des relations complexes entre les données.

## **COMPARAISON MONGODB VS SQL**

**Schéma :** MongoDB n'est pas un schéma fixe, ce qui permet une flexibilité dans la structure des données. SQL est un schéma rigide basé sur des tables et des relations prédéfinies.

**Flexibilité :** MongoDB permet une structure flexible sans schéma défini, tandis que SQL exige un schéma rigide.

**Évolutivité :** MongoDB offre une évolutivité horizontale facile grâce à la réplication et à la distribution de données, tandis que SQL nécessite une gestion plus complexe pour l'évolutivité.

**Performance :** MongoDB est efficace pour les opérations de lecture et d'écriture massives, tandis que SQL excelle dans les requêtes complexes et les agrégations de données.

**Types de données :** MongoDB prend en charge des types de données flexibles tels que les tableaux et les objets imbriqués, tandis que SQL a des types de données prédéfinis.

**Transactions :** SQL prend en charge les transactions ACID (Atomicité, Cohérence, Isolation, Durabilité), tandis que MongoDB propose des transactions multi-documents mais avec des compromis sur l'isolement et la cohérence.

## **CONCLUSION**

Récapitulez brièvement les différences de clés entre MongoDB (NoSQL) et SQL (bases de données relationnelles). Le choix entre les deux dépend des besoins spécifiques du projet, tels que la flexibilité du schéma, l'évolutivité, les performances et les exigences transactionnelles. MongoDB convient particulièrement aux applications nécessitant une évolutivité horizontale et une gestion de données non modulées, tandis que SQL est souvent préféré pour les systèmes nécessitant des transactions ACID et des requêtes complexes.