

NoSQL

Introduction au NoSQL

Qu'est-ce que le NoSQL ?

Le terme "NoSQL" signifie "Not Only SQL". Il désigne une famille de systèmes de gestion de bases de données conçus pour répondre aux limites du SQL:

- **Scalabilité horizontale:** Répartition des données sur plusieurs serveurs (sharding).
- **Flexibilité des modèles de données:** Pas de schéma fixe. Les données peuvent être stockées sous forme de documents, de clés-valeurs, de colonnes ou de graphes.
- **Performances accrues**

Pourquoi le NoSQL ?

Limites des bases de données relationnelles (SQL)

Les bases relationnelles (MySQL, PostgreSQL ou Oracle), reposent sur un schéma fixe. Bien qu'elles soient robustes et adaptées à de nombreux cas, elles montrent certaines limites:

- **Scalabilité verticale difficile:** Une base relationnelle est conçue pour fonctionner sur un serveur unique, ce qui rend sa mise à l'échelle coûteuse (puissance du matériel, mémoire...)
- **Inadéquation aux données non structurées:** Images, vidéos, logs ou autres données ne s'intègrent pas bien dans des bases SQL.

- **Rigidité du schéma:** Le modèle relationnel impose un schéma rigide qui peut rendre les modifications complexes lorsque les besoins changent rapidement.
- **Problèmes de performance (volumes massifs de données):** Les requêtes complexes, les jointures nombreuses ou les traitements en temps réel peuvent devenir coûteux en ressources.
- **Emergence du besoin:** Avec la montée en puissance d'internet, des réseaux sociaux, du big data et des objets connectés, le besoin de traiter des quantités massives de données hétérogènes, souvent non structurées, est apparu. Les bases NoSQL ont émergé pour répondre à ces nouveaux défis.

Historique du NoSQL

- **1960s-1970s:**

Premières bases non relationnelles comme IMS (IBM).

- **1970s:**

Développement des bases relationnelles (modèle SQL par Edgar Codd).

- **1990s:**

Explosion des bases relationnelles grâce à leur standardisation.

- **2000s:**
 - Apparition de systèmes NoSQL modernes pour répondre aux défis des grandes entreprises du web (Google, Amazon, Facebook).
 - Publication du **BigTable** de Google (2006) et du **Dynamo** d'Amazon (2007), posant les bases des concepts actuels.
- **Aujourd'hui:** Large adoption des bases NoSQL dans les entreprises pour des cas d'usages spécifiques, souvent en complément des bases SQL.

Les principaux types de bases NoSQL

Au fil du temps, quatre grands types de BDD NoSQL ont émergé:

- Bases documentaires,
- Bases clé-valeur,
- Bases en colonnes larges,
- Bases de données graphes.

Les bases de données multimodèles gagnent également en popularité aujourd'hui, car elles permettent de gérer plusieurs types de données.

Bases orientées document:

Une base de données orientée document stocke les données sous forme de documents similaires aux objets JSON (BSON). Chaque document contient des paires clé-valeur dont les valeurs peuvent être de différents types.

Caractéristiques principales:

- **Modèle de données flexible:** Idéal pour les données semi-structurées ou non structurées.
- **Structures imbriquées:** Permettent de représenter facilement des relations complexes ou des données hiérarchisés.

Exemples technologiques: MongoDB, Couchbase.

Bases clé-valeur

Une base clé-valeur est une forme simplifiée de base de données où chaque élément contient une **clé unique** associée à une valeur.

Utilisé pour:

- La gestion des **caches**
- Les **sessions utilisateur**

Caractéristiques principales:

- **Haute performance:** lecture et écriture très rapide.
- **Simplicité:** Adaptées à des données simples ou temporaires

Exemples technologiques: Amazon DynamoDB, Redis.

Bases en colonnes larges

Les bases en colonnes larges stockent les données dans des **tables**, **lignes** et **colonnes dynamiques**.

Contrairement aux bases relationnelles traditionnelles, ces bases permettent à des lignes différentes d'avoir un nombre variable de colonnes.

Avantages:

- **Compression des colonnes:** Réduit l'espace de stockage.
- **Efficacité pour les données éparses:** Adaptées aux grands ensembles de données avec des colonnes larges.

Exemples technologiques: Apache Cassandra, HBase.

Bases de données graphe

Une base de données graphe organise les données sous forme de **noeuds** et d'**arêtes**.

- **Noeuds:** Représente des entités (personnes, objets, lieux).
- **Arêtes:** Représente les relations entre les noeuds.

Elles conviennent particulièrement aux ensembles de données très connectés où les relations ou modèles ne sont pas immédiatement évidents, comme les réseaux sociaux ou la gestion des dépendances.

Exemples technologiques: Neo4j, Amazon Neptune.

MongoDB peut également traiter des graphes via la commande
`$graphLookup`

Bases multimodèles

Les bases de données multimodèles prennent en charge plusieurs types de modèles de données (document, graphe, clé-valeur, etc.) dans une seule instance. Cela permet aux dévs de choisir le modèle le plus adapté à chaque cas d'usage au sein d'une application.

Exemples technologiques: CosmoDB, ArangoDB

Les problématiques du NoSQL

- **Pas de standardisation:** Chaque solution NoSQL a ses propres commandes, API et concepts.
- **Manque de support pour les transactions complexes:** Peu adapté à des systèmes nécessitant des relations rigoureuses entre les données.
- **Complexité:** Peut nécessiter des connaissances spécifiques pour configurer et maintenir.

Quand utiliser le NoSQL ?

- **Réseaux Sociaux:** Gestion des connexions et des relations entre utilisateurs.
- **E-commerce:** Catalogues produits avec des structures de données variées.
- **Big Data:** Analyse de grandes quantités de données non structurées.
- **IoT:** Stockage et analyse des données des capteurs en temps réel.

Merci pour votre attention

Des questions ?

