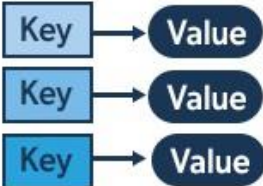


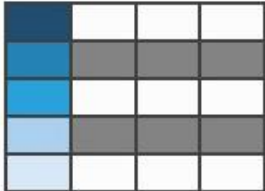
NoSQL

NoSQL

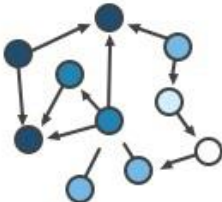
Key-Value



Column-Family



Graph



Document



¹ <https://media.geeksforgeeks.org/wp-content/uploads/20220405112418/NoSQLDatabases.jpg>

Table des matières

1. Introduction	3
2. Base de données clé-valeur	4
2.1. Fonctionnement	4
2.2. Fonctionnalités	4
2.3. Avantages.....	5
2.4. Cas d'utilisation.....	5
3. Base de données en mémoire	7
3.1. Fonctionnement	Erreur ! Signet non défini.
3.2. Avantages.....	7
3.3. Cas d'utilisation.....	7
4. Base de données orientées graphe	9
4.1. Fonctionnement	9
4.2. Avantages.....	9
4.3. Cas d'utilisation.....	10
5. Bases de données orientées document	12
5.1. Fonctionnement	12
5.2. Opérations de base	12
5.3. Avantages.....	12
5.4. Cas d'utilisation.....	13
6. Avantages et inconvénients NOSQL	14
7. NoSQL vs SQL.....	15
7.2. Différences fondamentales.....	15
7.2. Choix en fonction des besoins	Erreur ! Signet non défini.
Bibliographie.....	17

1. Introduction

Dans cette lecture individuelle, nous allons voir les bases de données NoSQL. J'ai choisi cette thématique car nous avons dû ce type de base de données dans un projet.

Le terme « NoSQL » est un acronyme pour « Not only SQL ». Une base de données NoSQL est une base de données non relationnelle.

Les systèmes NoSQL sont compatibles avec un variété étendue de technologies permettant le stockage de différents types de données, qu'elles soient structurées, non-structurées, semi-structurées ou polymorphiques.

Caractéristiques du NoSQL

La caractéristique principale des bases de données NoSQL est qu'elles ne reposent pas sur le modèle relationnel et ne présentent pas les données sous forme de colonnes fixes. Dans le NoSQL, la normalisation des données ou le mappage relationnel ne sont pas nécessaires. Il est possible de créer des interactions sans recourir à des langages de requêtes complexes.

Une autre caractéristique est l'absence des schémas ou leur flexibilité. La création de schémas de données n'est pas obligatoire, et les données de différentes structures peuvent coexister dans un même système.

Les bases de données NoSQL offrent une interface d'utilisation simple pour le stockage et la requête de données. Des APIs permettent de manipuler les données avec diverses méthodes de sélection. Les protocoles, basés sur le texte, reposent principalement sur HTTP REST avec JSON. En général, un langage de requête NoSQL est utilisé.

Une dernière caractéristique du NoSQL est sa nature distribuée. Cela signifie que les bases de données peuvent fonctionner de manière distribuée, offrant ainsi des capacités d'auto-évolutivité et de reprise sur panne.

2. Base de données clé-valeur

La bases de données orientée clé-valeur utilise une méthode clé-valeur simple pour stocker des données. Elle stocke les données sous forme de paires clé-valeur où une clé sert d'identifiant unique.

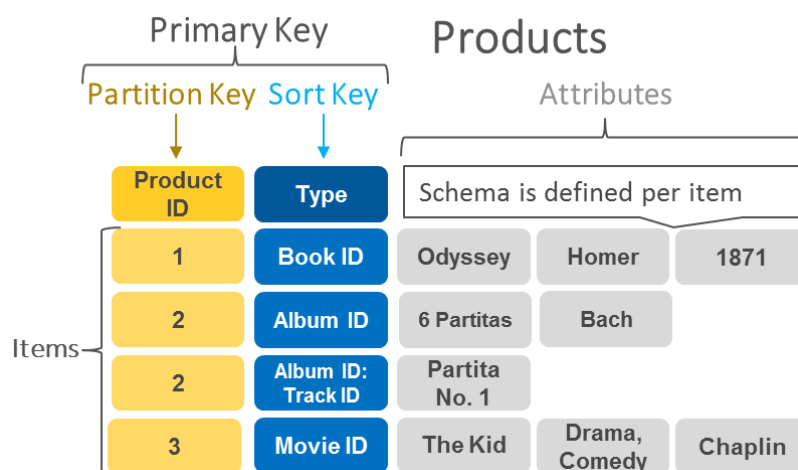
Les clés et les valeurs peuvent avoir différentes formes, des objets simples aux objets complexes.

2.1. Fonctionnement

Ce type de base de données fonctionne en organisant toutes ces données sous forme d'un ensemble de paires clé-valeur. La clé peut être vu comme la question et la valeur la réponse.

Dans l'exemple ci-dessous, la clé primaire est composée de deux clés, Product ID et Type. La combinaison de ces deux clés forme une clé primaire unique, qui correspond à une valeur unique dans la base de données.

²Dans cet exemple, chaque élément possède son propre schéma, ce qui rend le stockage clé-valeur très flexible pour les données de structures variées.



2.2. Fonctionnalités

En fonction de la solution que nous choisissons, il peut y avoir différentes fonctionnalités :

- Pas besoin de jointure de tables

Ce type de base de données n'as pas besoin d'effectuer de jointures de table. Leur flexibilité permet le regroupement de toutes les informations dans une seule table.

- Clés triées

Le stockage clé-valeur peut faire du tri des clés afin de stocker les données systématiquement et pour implémenter le partitionnement. Voici des exemples de tri :

- o Alphabétiquement ou numériquement

² <https://d1.awsstatic.com/product-marketing/DynamoDB/PartitionKey.8dd0530a7f6d66d101f31de30db515564f4cf28a.png>

- Chronologiquement
 - Par taille de données
- Support pour des données complexes
Ce type de base de données prend en charge des objets complexes, tels que des tableaux, des dictionnaires imbriqués, des images, des vidéos...
 - Partitionnement
Il s'agit de la façon dont nous distribuons les différentes données entre les nœuds. Des nombreuses bases de données proposent des options de partitionnement par défaut. Par exemple, il est possible de partitionner les clés numériques en groupes de 1000.

2.3. Avantages

Ce type de base de données NoSQL a différents avantages :

- Capacité de mise à l'échelle
De nombreuses bases de données clé-valeur fournissent un support intégré pour les fonctionnalités de mise à l'échelle avancées. Elles mettent à l'échelle horizontalement et distribuent automatiquement les données sur les serveurs afin de réduire les goulots d'étranglement sur un seul serveur.
- Simplicité d'utilisation
Ce type de base de données suit le paradigme orienté objet permettant aux développeurs de mapper des objets directement du monde réel à des objets logiciels. Cela rend le stockage clé-valeur plus intuitif à utiliser pour les développeurs.
- Performances hautes
Elles traitent des opérations de lecture-écriture constantes avec des appels au serveur peu onéreux. Une latence améliorée et un temps de réponse réduit offrent des meilleures performances à grande échelle. Elles sont basées sur des structures simples à table unique plutôt que sur plusieurs tables interdépendantes.

2.4. Cas d'utilisation

Voici quelques exemples de cas d'utilisation de base de données clé-valeur :

- Gestion de session
Dans une application orientée session, comme une application web, une session démarre lorsqu'un utilisateur se connecte et se termine lorsqu'il se déconnecte ou lorsque la session expire. Pendant cette période, l'application stocke tous les attributs de la session utilisateur dans la mémoire principale ou une base de données. Ces données peuvent inclure des informations sur le profil de l'utilisateur, des messages, des données personnalisées, des thèmes, des recommandations...

Chaque session utilisateur est identifiée par un identifiant unique. Les données de session sont interrogées uniquement par une clé primaire, ce qui rend une base de données clé-valeur rapide idéale dans ce contexte. En général, les frais généraux par

page liés aux bases de données clé-valeur sont inférieurs à ceux associés aux bases de données relationnelles.

- Panier d'achat

Un site Web d'e-commerce peut recevoir des milliards de commandes en quelques secondes pendant la période des achats de Noël. Les bases de données clé-valeur peuvent gérer la mise à l'échelle de grandes quantités de données et de grands volumes de changements d'état tout en répondant aux besoins de millions d'utilisateurs simultanés grâce à un traitement et à un stockage distribués. Le stockage de données clé-valeur intègre également une capacité de redondance, lui permettant de gérer la perte de nœuds de stockage.

- Mise en cache

Une base de données clé-valeur peut être utilisée pour stocker temporairement des données afin de les récupérer plus rapidement. Par exemple, les applications de réseaux sociaux peuvent stocker des données fréquemment consultées, comme le contenu des fils d'actualité. Les systèmes de mise en cache de données en mémoire utilisent également le stockage clé-valeur pour accélérer les réponses des applications.

3. Base de données en mémoire

Une base de données en mémoire est une solution personnalisée qui s'appuie principalement sur la mémoire vive (RAM) pour stocker et accéder aux données. Cela permet d'obtenir des temps de réponse minimaux en éliminant la nécessité d'accéder à des unités de disque standard (SSD). Les bases de données en mémoire sont particulièrement adaptées aux applications nécessitant des temps de réponse de l'ordre de la microseconde, comme les classements de jeux, le stockage de sessions et l'analyse de données en temps réel.

3.1. Avantages

Voici les différents avantages d'une base de données en mémoire :

- **Faible latence**
Les bases de données en mémoire offrent des latences prévisibles, quelle que soit l'échelle. Elles présentent une latence de lecture de quelques microsecondes, une latence d'écriture de quelques millisecondes, et un débit élevé.
Cela permet aux entreprises de prendre des décisions basées sur des données en temps réel, comme le calcul en mémoire des données des capteurs des véhicules autonomes pour une réponse quasi instantanée lors d'un freinage d'urgence.
- **Débit élevé**
Les bases de données en mémoire sont réputées pour leur haut débit, mesuré en opérations de lecture ou d'écriture par période donnée. Elles offrent une performance exceptionnelle pour le traitement rapide des données.
- **Grande capacité de mise à l'échelle**
La mise à l'échelle en écriture et en lecture est possible sans affecter les performances, permettant à la base de données de rester en ligne tout en répondant aux fluctuations de la demande.

3.2. Cas d'utilisation

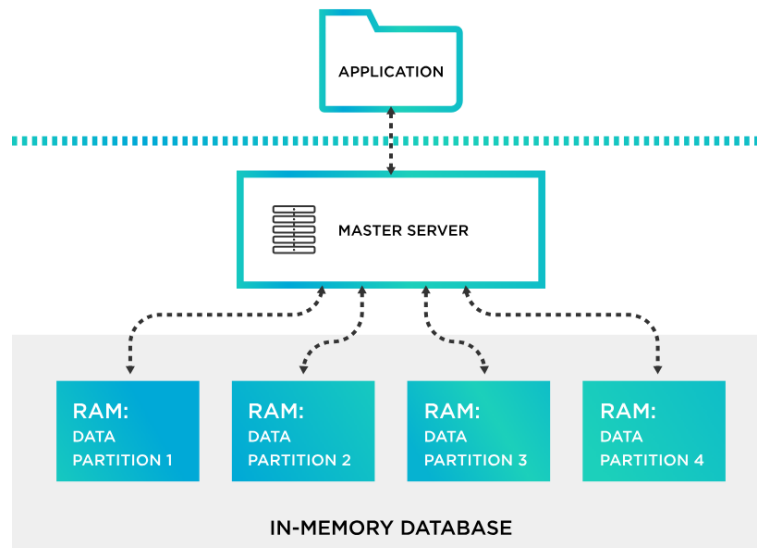
Voici quelques exemples de cas d'utilisation de base de données en mémoire :

- **Mise en cache**
Les caches sont des zones de stockage à haut débit qui stockent un sous-ensemble de données généralement transitoires, augmentant ainsi les performances en réduisant le besoin d'accéder à la couche de stockage sous-jacente plus lente. Cela permet un accès plus rapide aux données mises en cache.
- **Classement de jeux**
Les bases de données en mémoire fournissent rapidement des résultats de tri et peuvent mettre à jour le classement en temps réel pour des jeux rassemblant des millions de joueurs.

- Enchères en temps réel

Idéales pour les enchères en temps réel, ces bases permettent d'ingérer, de traiter et d'analyser des données avec une latence inférieure à la milliseconde, indispensable lors des transactions publicitaires en ligne.

3



³ https://www.tibco.com/sites/tibco/files/media_entity/2021-05/in-memory-database-diagram.svg

4. Base de données orientées graphe

Une base de données orientée graphe est une structure organisée de données qui met l'accent sur les relations entre différentes entités de données. Elle utilise la théorie mathématique des graphes pour représenter les connexions entre les données, offrant ainsi une alternative aux bases de données relationnelles. Contrairement à ces dernières, qui utilisent des structures de table rigides, les bases de données orientées graphe stockent les données sous la forme d'un réseau d'entités et de relations. Cette approche conduit généralement à des performances améliorées et à une flexibilité accrue, particulièrement adaptée à la modélisation de scénarios réels.

4.1. Fonctionnement

Qu'est-ce qu'un graphe ?

Graphe :

Un graphe, dans le contexte des bases de données orientées graphe, est composé de nœuds et d'arêtes.

Nœuds :

Les nœuds sont des sommets qui stockent des objets de données, chacun pouvant avoir un nombre illimité et des types variés de relations.

Arêtes :

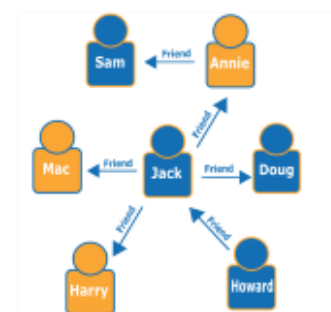
Les arêtes représentent les relations entre les nœuds, pouvant décrire des liens de type "one-to-many" ou "many-to-many". Chaque arête possède un nœud initial, un nœud final, un type et une direction.

Propriétés :

Chaque nœud peut avoir des propriétés ou des attributs le décrivant, et dans certains cas, les arêtes peuvent également avoir des propriétés. Les graphes avec des propriétés sont souvent appelés graphes de propriétés.

Exemple de graphe :

Un exemple concret de graphe peut illustrer un réseau social, montrant des personnes en tant que nœuds et leurs relations en tant qu'arêtes. Ces graphes permettent d'identifier les "amis des amis" d'une personne, par exemple, les amis des amis d'Howard.



⁴ <https://d1.awsstatic.com/graph%20network.1d238490c0b1b43bcc6e905607999f81c2db35df.png>

4.2. Avantages

Ce type de base de données NoSQL a différents avantages :

- Flexibilité
Les modèles de graphe peuvent être modifiés sans impact sur les fonctions existantes. Il n'est pas nécessaire de modéliser les domaines à l'avant, offrant ainsi une flexibilité précieuse.
- Efficacité
Les requêtes de graphe sont plus courtes et plus efficaces que celles de bases de données relationnelles, en particulier pour générer des rapports similaires. La traversée des relations entre les nœuds est rapide, car les relations sont préalablement stockées dans la base de données.
- Performances
Contrairement aux bases de données relationnelles, qui peuvent devenir moins optimales avec des relations plus complexes, les performances des bases de données orientées graphe s'améliorent considérablement lors de l'interrogation des relations, même avec un volume de données croissant.

4.3. Cas d'utilisation

Voici quelques exemples de cas d'utilisation de base de données orientées graphe:

- Détection de fraudes
Les bases de données orientées graphe sont efficaces pour détecter des schémas de fraude sophistiqués.
Par exemple, nous pouvons utiliser ce type de base de données pour suivre les transactions financières en temps presque réel. En formulant des requêtes de graphe rapides, nous pouvons détecter qu'un acheteur potentiel utilise la même adresse e-mail et la même carte de crédit enregistrées lors d'un précédent cas de fraude.
- Gestion des connaissances
Les bases de données orientées graphe proposent des techniques d'intégration de données, de données liées et de partage d'informations. Elles représentent des métadonnées ou des concepts de domaine complexes dans un format standardisé et fournissent une sémantique riche pour le traitement du langage naturel.
Par exemple, les algorithmes de machine learning font la distinction entre la forêt amazonienne et la marque Amazon grâce à des modèles de graphe.
- Optimisation de routage
Les problèmes d'optimisation de routage impliquent l'analyse d'un jeu de données et la recherche des valeurs les plus adaptées à un scénario particulier.
Par exemple, nous pouvons utiliser une base de données orientée graphe pour rechercher les éléments suivants :

- l'itinéraire le plus court d'un point A à un point B sur une carte en tenant compte des différents chemins ;
- le bon employé pour un quart de travail spécifique en analysant les disponibilités, les emplacements et les compétences ;
- les meilleures machines pour les opérations en tenant compte de paramètres tels que le coût et la durée de vie de l'équipement.

Les requêtes de graphe peuvent analyser ces situations beaucoup plus rapidement, car elles peuvent compter et comparer le nombre de liens entre deux nœuds.

5. Bases de données orientées document

Une base de données orientée document est une catégorie de bases de données NoSQL conçue pour stocker et interroger des données sous la forme de documents, souvent au format JSON (JavaScript Object Notation). Ce format, lisible par les humains et les machines, offre une structure souple, semi-structurée et hiérarchique, permettant aux applications de s'adapter aux besoins changeants au fil du temps.

5.1. Fonctionnement

Les bases de données document stockent les données sous forme de paires clé-valeur au format JSON. Vous pouvez lire et écrire des documents JSON dans les bases de données par programmation.

Structure des documents JSON

Le format JSON représente les données de trois manières :

- Clé-valeur
Enregistre les données entre des accolades, avec la clé sous forme de chaîne et la valeur qui peut être tout type de donnée. Par exemple : {« année » :2013}.
- Table
Un ensemble de données ordonnée de valeurs délimitées par des crochets, séparées par des virgules. Par exemple : {« fruit » :[« pomme », « mangue »]}.
- Objets
Un ensemble de paire clé-valeur, permettant des structures imbriquées. Par exemple : {"adresse": {"pays": "USA", "état": "Texas"}}.

5.2. Opérations de base

Les bases de données document permettent des opérations telles que la création, la lecture, la mise à jour et la suppression de documents complets. Ces opérations sont effectuées par la programmation à l'aide d'une API ou d'un langage de requête.

5.3. Avantages

Ce type de base de données NoSQL a différents avantages :

- Facilité de développement :
Les développeurs peuvent travailler efficacement avec des documents JSON, souvent mappés à des objets dans de nombreux langages de programmation. Cela permet une flexibilité accrue dans la création et la mise à jour de documents directement depuis le code, accélérant le développement des applications.
- Schéma flexible :
Une base de données document permet de créer plusieurs documents avec différents champs au sein d'un même recueil. Cela peut être pratique lorsque vous stockez des

données non structurées telles que des e-mails ou des publications sur les réseaux sociaux.

Certaines de ces bases de données proposent également une validation de schéma pour imposer des restrictions si nécessaire.

- Performance à l'échelle :

La capacité intégrée de distribution permet de mettre à l'échelle horizontalement sur plusieurs serveurs sans compromettre les performances. Les bases de données document assurent également la tolérance aux pannes et la disponibilité grâce à la réplication intégrée.

5.4. Cas d'utilisation

Voici quelques exemples de cas d'utilisation de base de données orientée document :

- Gestion de contenu

Les applications de gestion de contenu, telles que les blogs et les plateformes vidéo, bénéficient de la souplesse des base de données document. Chaque entité peut être stockée en tant que document unique, facilitant ainsi la mise à jour en fonction des évolutions des exigences.

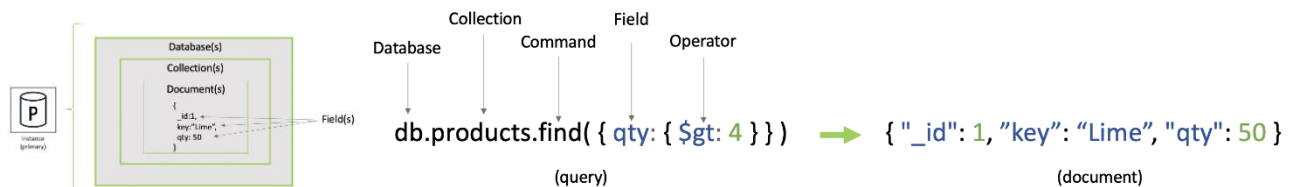
- Catalogues

Les bases de données document sont idéales pour stocker des informations sur des catalogues, comme dans le cas d'une application d'e-commerce. Les attributs différents des produits peuvent être décrits dans un seul document, améliorant la gestion et la vitesse de lecture.

- Gestion de capteurs

Dans le contexte de l'Internet des objets (IoT), les bases de données document sont pratiques pour stocker des données des capteurs. Elles permettent un stockage rapide des données sans nécessité de nettoyage préalable, et la mise à l'échelle est simple selon les besoins d'analyse, avec la possibilité de supprimer des documents entiers après les analyses effectuées.

56



⁵ <https://d1.awsstatic.com/AWS%20Databases/JSON%20document%20database.64fe2a382abc8ca2b8743f0e3b5af553a33f3fb0.png>

⁶ <https://d1.awsstatic.com/AWS%20Databases/JSON%20database%20query.725e49a724fff7c4acb904b82f36db0db31e99d2.png>

6. Avantages et inconvénients NOSQL

Les bases de données NoSQL offrent de nombreux avantages, mais présentent également des inconvénients. Elles sont particulièrement adaptées au stockage et à l'analyse du Big Data, éliminant ainsi le risque de point de défaillance unique.

Avantages

- Stockage et analyse Big Data
Elles sont idéales pour traiter des vastes ensembles de données, notamment dans le contexte du Big Data.
- Facilité de réplication
La réplication est simplifiée, assurant la disponibilité des données même en cas de défaillance d'un nœud.
- Pas de défaillance unique
La distribution des données élimine le risque de défaillance unique, améliorant ainsi la résilience du système.
- Performances élevées et scalabilité horizontale
Les performances sont élevées, et la scalabilité horizontale permet d'ajouter facilement des ressources pour répondre à la croissance.
- Support pour données structurées et non structurées
Les données structurées et non structurées sont prises en charge de manière équivalente.
- Compatible avec plusieurs langages de programmation courants
Compatible avec les principaux langages de programmation, offrant ainsi une flexibilité dans le choix des technologies.

Inconvénients

- Capacités de requêtes limitées
La capacité de requêtes de ce type de base de données est parfois limitée par rapport aux bases de données traditionnelles.
- Moins adapté aux données relationnelles
Elle est moins performante pour des données relationnelles, ce qui peut être un inconvénient selon les cas d'utilisation.
- Absence de règles de standardisation
Les bases de données NoSQL manquent de règles de standardisation, ce qui peut entraîner différents défis d'interopérabilité.
- Difficulté à maintenir des clés uniques
Cela devient complexe au fur et à mesure que le volume de données augmente.
- Options open source moins populaires en entreprise
Les options open source ne sont pas toujours les préférées dans les entreprises, ce qui peut limiter l'adoption dans certains environnements.

7. NoSQL vs SQL

Voici un tableau récapitulatif des caractéristiques principales des base de données SQL et des base de données NoSQL :

SQL	NoSQL
Signifie « Structured Query Language », en français «	Acronyme pour “Not Only SQL”
Système de gestion de base de données relationnelle	Système de gestion de base de données non relationnelle
Adapté aux données structurées avec un schéma prédéfini	Adapté aux données non-structurées et semi-structurées
Les données sont stockées dans des tables avec des colonnes et des lignes	Les données sont stockées dans des collections ou dans des documents
Prise en charge des jointures et des requêtes complexes	Ne prends pas en charge les jointures ni les requêtes complexes
Nécessite une mise à l'échelle verticale pour gérer des grands volumes de données	La mise à l'échelle horizontale est possible pour gérer des grands volumes de données
Exemples : MySQL, PostgreSQL, SQL Server, Microsoft SQL	Exemples : MongoDB, Cassandra, Couchbase, Amazon DynamoDB

7.2. Différences fondamentales

Voici les différences fondamentales entre ces deux types de base de données :

- Architecture de base de données
Leur plus grand différence est que les bases de données SQL sont relationnelles, tandis que les bases de données NoSQL sont non relationnelles.
Les bases de données relationnelles stockent des données sous forme de ligne et des tables, tandis que les base de données NoSQL utilise un modèle de stockage basé sur le type de données qu'elle stocke.
- Schémas de base de données et langages de requête
Les bases de données SQL utilisent un langage de requête structuré et ont un schéma prédéfini pour définir et manipuler les données. SQL est un langage idéal pour les requêtes complexes.
Au contraire, les bases de données NoSQL ont des schémas dynamiques pour les données non structurées et stockent les données de plusieurs manières, par exemple orientée colonnes, orientée documents, basé sur des graphiques.
- Mise à l'échelle de la base de données
Les bases de données SQL sont évolutives de manière verticale. Cela signifie que nous pouvons augmenter la charge sur un seul serveur, par exemple, en ajoutant plus de CPU, de RAM ou de capacité SSD.
Les bases de données NoSQL sont évolutives de manière horizontale. Nous pouvons gérer un trafic plus élevé de données via un processus appelé sharding, qui ajoute plus de serveurs à notre base de données. La mise à l'échelle horizontale a une capacité globale plus grande que la mise à échelle verticale.

- Cas d'utilisation

Les bases de données SQL sont meilleures pour les transactions multi-lignes, tandis que NoSQL est plus adapté pour les données non structurées comme les documents ou le JSON. Les bases de données SQL sont également couramment utilisées pour les systèmes hérités construits autour d'une structure relationnelle.

Vous pourriez utiliser une base de données SQL pour des applications orientées utilisateur avec plusieurs opérations de jointure. Le schéma SQL vous aidera à établir des propriétés ACID et à améliorer la compatibilité des données. Ces bases de données sont également utiles pour trouver rapidement les données dont vous avez besoin pour accomplir une tâche.

Vous pourriez utiliser une base de données NoSQL pour des applications avec des données dynamiques sans opérations de jointure. NoSQL est également mieux adapté aux applications avec des ensembles de données manquants qui n'affecteront pas l'efficacité commerciale.

Bibliographie

Mark SmallCombe (2023). SQL vs NoSQL : 5 critical differences

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/in-memory/>

Rédaction de AWS (sans date). Présentation de NoSQL

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/>

Rédaction de AWS (sans date). Qu'est-ce qu'une base de données clé-valeur ?

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/key-value/>

Rédaction de AWS (sans date). Qu'est-ce qu'une base de données document ?

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/document/>

Rédaction de AWS (sans date). Qu'est-ce qu'une base de données orientée graphe ?

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/graph/>

Rédaction de AWS (sans date). Qu'est-ce qu'une base de données en mémoire ?

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/in-memory/>

Rédaction de DataScientist (sans date). NoSQL : Tout savoir sur les bases de données non relationnelles

Consulté le 4 janvier 2024, à l'adresse <https://aws.amazon.com/fr/nosql/in-memory/>