

1. Introduction : Qu'est une base de données et les différents types de bases de données existantes (bases de données relationnelles et non relationnelles)

Une base de données est un système informatique qui permet de stocker, organiser, consulter et modifier des données de manière efficace. Il existe plusieurs types de bases de données, chacune ayant des caractéristiques et des utilisations spécifiques. Les deux principaux types de bases de données sont les bases de données relationnelles et les bases de données non relationnelles.

Les bases de données relationnelles utilisent une structure de données organisée en tables, avec des relations définies entre les tables. Les données sont normalisées, c'est-à-dire qu'elles sont divisées en plusieurs tables pour éviter les doublons et les incohérences. Les requêtes SQL (Structured Query Language) sont utilisées pour interroger les données. Les bases de données relationnelles sont souvent utilisées pour les applications qui nécessitent des données très structurées et une forte intégrité des données.

Les bases de données non relationnelles, également appelées bases de données NoSQL, ont une structure de données plus flexible que les bases de données relationnelles. Les données peuvent être stockées sous différents formats (clés-valeurs, documents, graphes, etc.) et il n'y a pas de relations définies entre les données. Les requêtes NoSQL sont utilisées pour interroger les données. Les bases de données non relationnelles sont généralement utilisées pour les applications qui nécessitent une grande scalabilité et une forte performance pour gérer des volumes élevés de données en temps réel.

2. Bases de données relationnelles : caractéristiques, définition de clés primaires et étrangères et l'utilisation de requêtes SQL pour interroger les données.

Les bases de données relationnelles sont caractérisées par :

1. Normalisation des données : Les bases de données relationnelles utilisent la normalisation pour éviter les doublons et les incohérences dans les données. Les données sont divisées en plusieurs tables et des relations sont définies entre les tables pour garantir l'intégrité des données.
2. Clés primaires et étrangères : Les tables des bases de données relationnelles ont des clés primaires et étrangères qui permettent de relier les différentes tables entre elles. La clé primaire est un identifiant unique qui permet de distinguer les différents enregistrements d'une table, tandis que la clé étrangère est utilisée pour établir une relation entre une table et une autre.
3. Requêtes SQL : Les bases de données relationnelles utilisent le langage SQL (Structured Query Language) pour interroger les données. SQL permet de réaliser des opérations CRUD (Create, Read, Update, Delete) sur les données, ainsi que des requêtes plus avancées pour récupérer des données spécifiques ou effectuer des calculs sur les données.
4. Structure de données : Les bases de données relationnelles sont organisées en tables qui ont des colonnes et des lignes. Les tables ont des relations définies entre elles pour garantir l'intégrité des données et faciliter les requêtes.
5. Schéma : Les bases de données relationnelles ont un schéma prédéfini qui décrit la structure de la base de données, y compris les tables, les colonnes, les types de données et les relations entre les tables.
6. Intégrité des données : Les bases de données relationnelles garantissent l'intégrité des données en utilisant des contraintes telles que les clés primaires et étrangères pour éviter les données incohérentes ou incomplètes.

<https://www.programiz.com/sql/online-compiler/>

7. Bases de données non relationnelles : caractéristiques, notamment la flexibilité de la structure des données, la scalabilité, et l'utilisation de requêtes NoSQL pour interroger les données.

Les bases de données non relationnelles (ou NoSQL) sont caractérisées par :

1. Flexibilité de la structure des données : Les bases de données non relationnelles ne sont pas soumises aux contraintes de la normalisation des données et de la définition de clés primaires et étrangères. Cela permet une plus grande flexibilité dans la structure des données, adaptée aux besoins de l'application.
2. Scalabilité : Les bases de données non relationnelles sont généralement plus faciles à scaler horizontalement, c'est-à-dire en ajoutant de nouveaux serveurs pour gérer une charge croissante, plutôt que verticalement, c'est-à-dire en ajoutant plus de puissance à un serveur existant.
3. Requêtes NoSQL : Les bases de données non relationnelles utilisent des requêtes NoSQL qui sont différentes des requêtes SQL utilisées pour les bases de données relationnelles. Les requêtes NoSQL sont souvent plus simples et directes, mais moins expressives que les requêtes SQL.
4. Structure de données : Les bases de données non relationnelles n'ont pas de structure de données fixe, les données sont habituellement stockées sous forme de documents, de graphiques ou de clés-valeurs.
5. Schéma : Les bases de données non relationnelles n'ont pas besoin d'un schéma prédéfini, les données peuvent être ajoutées dynamiquement sans avoir à définir la structure de la base de données.
6. Intégrité des données : Les bases de données non relationnelles peuvent ne pas garantir l'intégrité des données de la même manière que les bases de données relationnelles. Cela peut être un avantage ou un inconvénient en fonction des besoins de l'application.

8. Comparaison : différences et les similitudes entre les bases de données relationnelles et non relationnelles.

Les différences :

1. Structure des données : Les bases de données relationnelles suivent une structure rigoureuse de normalisation des données, avec des tables définies par des clés primaires et étrangères, tandis que les bases de données non relationnelles sont plus flexibles dans la structure des données, adaptées aux besoins de l'application.
2. Requêtes : Les bases de données relationnelles utilisent des requêtes SQL pour interroger les données, tandis que les bases de données non relationnelles utilisent des requêtes NoSQL qui sont souvent plus simples, mais moins expressives.
3. Scalabilité : Les bases de données relationnelles sont généralement plus difficiles à scaler horizontalement, tandis que les bases de données non relationnelles sont plus faciles à scaler horizontalement.
4. Intégrité des données : Les bases de données relationnelles garantissent l'intégrité des données de manière stricte, tandis que les bases de données non relationnelles peuvent ne pas garantir l'intégrité des données de la même manière.

Les similitudes :

1. Stockage des données : Les deux types de bases de données stockent des données pour une utilisation ultérieure.
2. Indexation : Les deux types de bases de données permettent l'indexation des données pour une recherche plus rapide.

Avantages des bases de données relationnelles :

- Les bases de données relationnelles sont très fiables et garantissent l'intégrité des données grâce à la normalisation et à la définition de clés primaires et étrangères.
- Les requêtes SQL sont puissantes et permettent des requêtes complexes pour interroger les données
- Les bases de données relationnelles sont bien adaptées pour les applications qui nécessitent des relations de données claires et définies.

Inconvénients des bases de données relationnelles :

- Les bases de données relationnelles sont plus difficiles à scaler horizontalement.

- Les bases de données relationnelles peuvent être trop rigides pour certaines applications qui nécessitent une flexibilité accrue dans la structure des données

9. Utilisation : exemples concrets d'utilisation des bases de données relationnelles et non relationnelles dans des contextes différents.

Les bases de données relationnelles (SQL) et les bases de données non relationnelles (NoSQL) sont utilisées dans des contextes différents en fonction de leurs caractéristiques et de leurs avantages.

Exemples d'utilisation des bases de données relationnelles :

- Système de gestion de la base de données d'une entreprise : Les bases de données relationnelles sont souvent utilisées pour stocker les données financières, les informations sur les employés et les données de clients. Ces types de données ont souvent des relations claires et définies entre elles, ce qui convient parfaitement aux bases de données relationnelles.
- Système de gestion de la base de données d'une bibliothèque : Les bases de données relationnelles peuvent être utilisées pour stocker les informations sur les livres, les auteurs et les emprunteurs. Les relations entre ces types de données sont claires, ce qui convient parfaitement aux bases de données relationnelles.

Exemples d'utilisation des bases de données non relationnelles :

- Système de gestion de la base de données d'un site de commerce électronique : Les bases de données non relationnelles sont généralement utilisées pour stocker les informations sur les produits, les clients et les commandes. Ces types de données ont habituellement des relations complexes et peuvent varier en fonction des besoins de l'application, ce qui convient parfaitement aux bases de données non relationnelles.
- Système de gestion de la base de données d'un réseau social : Les bases de données non relationnelles peuvent être utilisées pour stocker les informations sur les utilisateurs, les publications et les relations entre les utilisateurs. Les relations entre ces types de données sont souvent complexes et peuvent varier en fonction des besoins de l'application, ce qui convient parfaitement aux bases de données non relationnelles.

- Système de gestion de la base de données d'un jeu en ligne : Les bases de données non relationnelles peuvent être utilisées pour stocker les informations sur les joueurs, les objets, les scores. Les relations entre ces types de données sont souvent complexes et peuvent varier en fonction des besoins de l'application, ce qui convient parfaitement aux bases de données non relationnelles.

10. Conclusion : le choix de la base de données en fonction des besoins de l'application.

Le choix de la base de données dépend des besoins de l'application et de la façon dont les données seront utilisées. Les bases de données relationnelles (SQL) et les bases de données non relationnelles (NoSQL) ont chacun leurs propres avantages et inconvénients, et il est important de comprendre ces différences pour choisir la base de données la plus appropriée pour l'application.

Les bases de données relationnelles (SQL) sont idéales pour les applications nécessitant des relations claires et définies entre les données, telles que les systèmes de gestion de la base de données d'une entreprise ou d'une bibliothèque. Les bases de données relationnelles sont également utiles pour les applications nécessitant des transactions atomiques, telles que les systèmes de gestion de comptes bancaires. Les bases de données relationnelles sont souvent plus performantes pour les requêtes de sélection, de mise à jour et de suppression de données.

Les bases de données non relationnelles (NoSQL) sont idéales pour les applications nécessitant des relations complexes et variables entre les données, telles que les systèmes de gestion de la base de données d'un site de commerce électronique ou d'un réseau social. Les bases de données non relationnelles sont également utiles pour les applications nécessitant une grande scalabilité et une haute disponibilité, telles que les systèmes de gestion de la base de données d'un jeu en ligne. Les bases de données non relationnelles sont souvent plus performantes pour les requêtes d'insertion et de lecture de données.

En fin de compte, le choix de la base de données dépend de la nature des données et de la façon dont elles seront utilisées dans l'application. Il est important de comprendre les avantages et les inconvénients de chaque type de base de données pour choisir celle qui convient le mieux aux besoins de

l'application. Il est également possible d'utiliser des approches hybrides qui combinent les avantages des bases de données relationnelles et non relationnelles pour répondre aux besoins de l'application.

NoSQL : Tout comprendre sur les bases de données non relationnelles

Le NoSQL est un type de bases de données, dont la spécificité est d'être non relationnelles. Ces systèmes permettent le stockage et l'analyse du Big Data.

À l'heure du Big Data, les bases de données relationnelles ne sont plus adaptées. Pour prendre en charge les immenses volumes de données, les stocker et les analyser, il est impératif de s'en remettre à de nouvelles solutions.

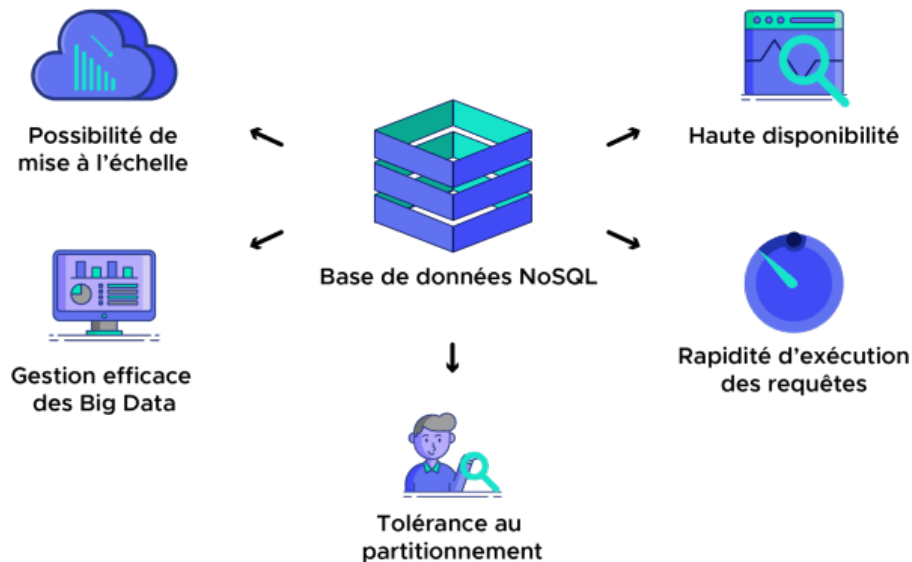
Une base de données NoSQL est une base de données « non relationnelle ». Il est possible d'y stocker des données sous une forme non structurée, sans suivre de schéma fixe. Les jointures ne sont plus nécessaires, et le scaling est facilité.

On utilise notamment les bases de données NoSQL pour les Data Stores distribués aux besoins élevés en capacité de stockage. Ainsi, NoSQL est utilisé pour le **Big Data** et les **applications web en temps réel**. Les géants de la technologie comme **Twitter**, **Facebook** ou **Google** collectent chaque jour plusieurs terabytes de données sur leurs utilisateurs.

Le terme NoSQL signifie en fait "Not Only SQL" (pas seulement SQL). En effet, les bases de données relationnelles utilisent la syntaxe SQL pour le stockage et l'analyse de données.

Ce n'est pas le cas d'une database non-relationnelle. Les systèmes NoSQL sont compatibles avec une large variété de technologies permettant le stockage de données structurées, non structurées, semi-structurées ou polymorphiques.

Qu'est ce qu'une Base de données NoSQL ?



L'histoire de NoSQL

Le terme et le concept NoSQL furent inventés en 1998 par Carl Stroz, afin de désigner sa base de données relationnelle légère et open source. Ce concept a ensuite été adopté et popularisé par les GAFAM tels que Google, Facebook ou Amazon confrontés à d'immenses volumes de données. Les bases de données relationnelles étaient devenues trop lentes.

Plutôt que de mettre à jour leur équipement informatique pour accroître les performances des RDBMS (Relational Database Management System), les géants de la technologie ont choisi de distribuer la charge sur de multiples serveurs hôtes. C'est la méthode dite du "scaling out". Les bases de données NoSQL sont idéales pour le scaling-out, puisqu'elles sont non relationnelles.

En l'an 2000, la base de données graphique **Neo4j** fut lancée. Ce fut ensuite le tour de la **Google Bigtable**, en 2004, puis **CouchDB** en 2005. L'histoire des bases de données NoSQL fut aussi marquée par **Amazon Dynamo** en 2007.

Puis, en 2008, Facebook rend open source la base de données non-relationnelle qu'elle utilise en interne : [Cassandra](#). Cet outil devient la référence des databases NoSQL, et remet le terme NoSQL sous le feu des projecteurs en lui donnant son sens et sa popularité actuelle.

Les caractéristiques de NoSQL

La principale particularité des bases de données NoSQL est qu'elles ne suivent pas le modèle relationnel et ne présentent **pas de tableaux sous forme de colonnes fixes**. Ces bases de données ne nécessitent pas de normalisation de données ou de mapping relationnel. Il est possible d'interagir sans utiliser de langages de requête complexe.

Une autre particularité est **l'absence ou la flexibilité des schémas**. Il n'est pas nécessaire de définir de schéma des données, et les données de différentes structures peuvent donc être regroupées sur un même système.

Les bases de données non relationnelles se distinguent aussi par une **interface simple d'utilisation** pour le stockage et la requête de données.

La dernière caractéristique d'une base de données NoSQL est d'être **distribuée**. De multiples bases NoSQL peuvent être exécutées de façon distribuée, offrant des capacités d'auto-scaling et de fail-over. Le concept ACID peut être délaissé au profit de l'élasticité et des performances.

Avantages et inconvénients de NoSQL

NoSQL présente de nombreux avantages, mais aussi des inconvénients. Ces bases de données sont idéales pour **le stockage et l'analyse Big Data**, et évitent aussi un point de défaillance unique.

Elles facilitent la **réplication**, et ne nécessitent pas de couche de caching séparée. Les **performances sont élevées**, et une

scalabilité horizontale est possible. Les databases NoSQL peuvent prendre en charge les données structurées ou non structurées de la même manière.

Les bases de données NoSQL ne nécessitent pas non plus de serveur dédié à hautes performances. Elles sont compatibles avec les principaux langages de programmation.

L'implémentation est plus simple qu'avec les RDBMS. Le schéma flexible peut être altéré facilement sans interruption.

Néanmoins, ce type de base de données présente aussi des points faibles. On peut citer **l'absence de règles de standardisation et les capacités de requêtes limitées**. Les capacités de bases de données traditionnelles, comme la **consistance** lorsque de multiples transactions sont effectuées simultanément, **peuvent aussi manquer**.

Par ailleurs, il devient **difficile de maintenir des valeurs uniques en guise de clés** lorsque le volume de données augmente. Ce modèle ne fonctionne pas aussi bien pour les données relationnelles. La courbe d'apprentissage peut être difficile pour les nouveaux développeurs, et les options open source ne sont pas toujours populaires au sein des entreprises. De manière générale, les bases de données relationnelles et leurs outils sont plus matures, plus aboutis et donc plus adoptés.

Pourquoi utiliser NoSQL ?

Les bases de données NoSQL conviennent pour plusieurs cas d'usage. Elles sont adaptées pour **stocker et retrouver de larges volumes de données**. Elles conviennent aussi lorsque les relations entre les données ne sont pas spécialement importantes.

On peut aussi s'en servir si les **données changent au fil du temps et ne sont pas structurées**. Elles conviennent enfin quand le volume de données augmente en continu et que le **scaling régulier de la base de données est nécessaire** pour les prendre en charge.