

Chapitre III

Bases de données NoSQL

- Introduction
- Caractéristiques
- NoSQL vs SQL
- Types des BD NoSQL
- Flexibilité des BD NoSQL
- FireBase

BD NoSQL

Introduction

- Un terme apparu en 2009 avec plusieurs interprétations possibles:
 - No SQL
 - Not Only SQL
- N'utilise pas de modèle relationnel, et donc pas de langage SQL
- Conçu pour être exécuté dans un cluster
- Tendence à être Open Source
- Généralement sans schéma et donc permettant de stocker n'importe quelle donnée dans n'importe quelle « ligne »

BD NoSQL

Caractéristiques

- Un SGBD qui n'est pas structuré en un table
- Son élément de base n'est pas en tuple
- Possède un langage de requête non uniformisé (propre à chaque BD). Souvent en format JSON avec une API REST
- Caractérisée par une dénormalisation de données où certains enregistrement sont en partie ou entièrement dupliqués
- Possède différents type (choix selon vos besoins)

NoSQL vs SQL

SQL

- Forte connaissance
- Transactionnel
- Très mature
- Adoption facile

Mais...après l'arrivé de Big data?

- **Big data:** ensemble de données tellement volumineux qu'il est difficile de les manipuler avec des SGBD classiques
- **Scalabilité:** Capacité d'un système à maintenir ses fonctionnalités et ses performances en cas de forte demande

NoSQL vs SQL

SQL : Problème de l'atomicité

- Une opération atomique ne permet pas des accès entrelacés
- La BDD est verrouillée pendant l'écriture

Solution?

- *Réduire la durée de verrouillage ou supprime le verrouillage de la BDD*

NoSQL vs SQL

SQL : Problème de la consistance

- Toutes les transactions doivent être totalement commises ou totalement annulées
- Tous les nœuds d'un cluster doivent avoir les mêmes données: en modifiant une données on doit la mettre à jour sur tous les nœuds
- Respect des schémas

Solution?

- *Plus besoin d'avoir exactement les mêmes données sur toutes les instances*
- *Mais...on synchronise quand le peut*
- *Plus de contraintes sur les champs (schemaless)*

NoSQL vs SQL

SQL : Problème de la durabilité

- Avant de répondre au client, on doit être sûr que les données ont été écrites sur le disque

Solution?

- *Ne pas écrire sur le disque directement, ou ne pas attendre la confirmation de l'écriture*

NoSQL vs SQL

SQL : Autres limitations

- Service réparti sur plusieurs sites/continents
- Accès concurrent de plusieurs centaines de milliers d'utilisateurs
- Stockage d'objets avec prise en compte de l'héritage
- Reconnaissance automatique des images, ou traitement automatique de textes/données

NoSQL vs SQL

NoSQL : Avantages

- Les données sont stockées dans des documents
- Données non structurées (ne doivent pas respecter un schéma unifié)
- NoSQL support les jointures
- N'encourage pas la normalisation
- Accepte la répétition/redondance des données
- Fournit un support excellent pour la scalabilité et la disponibilité de données

NoSQL vs SQL

	SQL	NoSQL
Types	Un seul (BD SQL)	Clé & valeur, orienté colonne, orienté document, orienté graphe
Histoire	Développé dans les années 1970	Développé dans les années 2000 pour résoudre les limitations de SQL
Exemples	MySQL, Postgres, Oracle	MongoDB, HBase, CouchDB, DynamoDB, etc
Schémas	Structures et types de données fixes	Dynamic
Manipulation	Select, Insert, Update	API orienté objet

Types BD No SQL

1. Clé/Valeur

- Les BDD NoSQL les plus simples
- Chaque élément est une paire clé-valeur
- **Description:** Stocke les informations sous forme de clé/valeur. Où la valeur peut être de n'importe quel type.
- **Utilisation:** Stockage des sessions utilisateurs ou la mise en cache d'un site.
- **Opération:** Création d'une paire clé-valeur, suppression, accès à une valeur à l'aide de la clé, etc.
- On peut définir la durée de vie d'une paire clé-valeur.
- Stockage en RAM pour accélérer les temps de lecture.
- **Implémentation:** Redis.

Types BD No SQL

1. Clé/Valeur

Des exemples de paires clé-valeur avec des types différents.

Clé	Valeur
pays.id-42	{"id" :42,"name" :"Chad"}
statistiques.nombre-visiteurs	1337
configuration.periode-gratuite	false
articles.categories-sport.latest	[22, 45, 67, 200, 87]

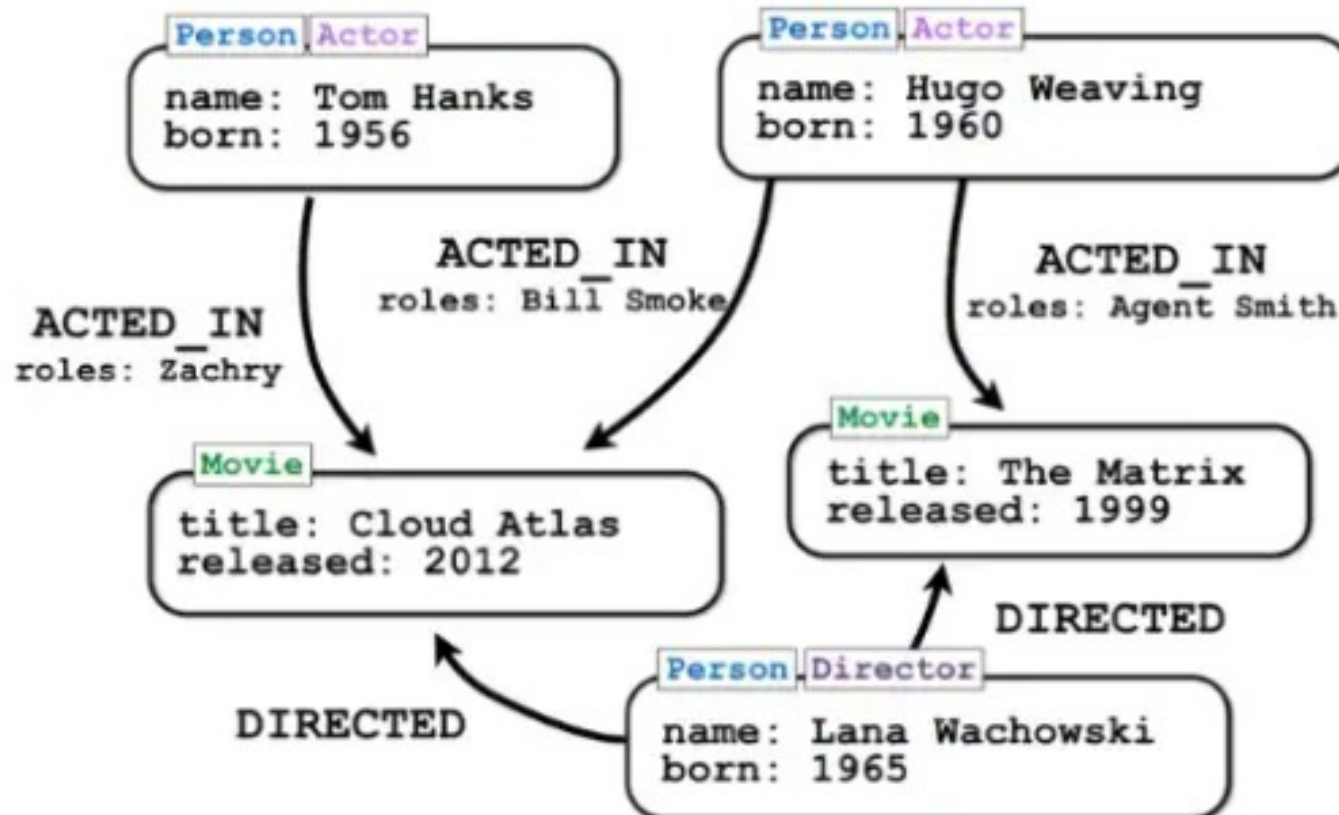
Types BD No SQL

2. Orientée graphe

- **Description:** Présente les données sous forme des nœuds et des relations.
- **Utilisation:** Utilisé principalement pour les données fortement reliées.
- **Opération:** Traitement très particulier de l'information.
- BD rarement utilisée pour le stockage.
- **Implémentation:** Neo4j.

Types BD No SQL

2. Orientée graphe



Types BD No SQL

3. Orientée colonne

- **Description:** ressemble au principe de SGBDR, mais les colonnes peuvent varier d'une ligne à une autre.
- **Utilisation:** Analyse et traitement massifs des données.
- **Implémentation:** Hbase, Cassandra.

Types BD No SQL

4. Orientée document

- **Description:** Aucun schéma fixe, un document peut contenir n'importe quel type d'information.
- **Utilisation:** base de données principalement utilisées pour le stockage.
- **Opération:** Ces données sont représentées sous forme d'objets XML ou JSON
- Accès facile (REST API)
- **Implémentation:** MongoDB, FireBase.

Firebase

- Une plateforme développée pour la création des applications mobiles et des sites web
- Une solution complète
 - Base de données (Firestore ou Realtime)
 - Cloud functions
 - Stockage
 - Hosting
 - Machine Learning
 - Analytics
 - Crashlytics
 - Remote Config
 - Messagerie
 - etc.

Firestore

- RealTime DB : données = Objets JSON
- Cloud Firestore : collections et documents
- Collection ≠ liste/tableau
- Avantage: scalabilité
- Collection de documents permet:
 - Recherche/requêtes
 - Pagination
 - Indexation