

Maîtrisez les bases de données NoSQL

🕒 15 hours 📶 Medium

License 

Last updated on 6/4/21



Choisissez votre famille NoSQL



Au fond, c'est quoi le NoSQL ?

Depuis les années 70, la base de données relationnelle était l'incontournable référence pour gérer les données d'un système d'information. Toutefois, face aux 3V (*Volume, Velocity, Variety*), le relationnel peut difficilement lutter contre cette vague de données. Le NoSQL s'est naturellement imposé dans ce contexte en proposant une nouvelle façon de gérer les données, sans reposer sur le paradigme relationnel, d'où le **"Not Only SQL"**. Cette approche propose de relâcher certaines contraintes lourdes du relationnel pour favoriser la distribution (structure des données, langage d'interrogation ou la cohérence).

Dans un contexte bases de données, il est préférable d'avoir un langage de haut niveau pour interroger les données plutôt que tout exprimer en [Map/Reduce](#). Toutefois, avoir un langage de trop haut niveau comme SQL ne facilite pas la manipulation. Et c'est en ce sens que l'on peut parler de "Not Only SQL", d'autres solutions peuvent être proposées pour résoudre le problème de distribution. Ainsi, le NoSQL est à la fois une autre manière d'interroger les données, mais aussi de les stocker.

Les besoins de stockage et de manipulation dans le cadre d'une base de données sont variables et dépendent principalement de l'application que vous souhaitez intégrer. Pour cela, différentes familles de bases NoSQL existent : *Clé/Valeur, colonnes, documents, graphes*. Chacune de ces familles répond à des besoins très spécifiques que nous allons développer par la suite, avec pour but *in fine*, de vous permettre de choisir **votre** solution NoSQL.

Les clés-valeurs

Le but de la famille clé-valeur est l'efficacité et la simplicité. Un système clé-valeur agit comme une énorme table de hachage distribuée sur le réseau. Tout repose sur le couple Clé/Valeur. La clé identifie la donnée de manière unique et permet de la gérer. La valeur contient n'importe quel type de données.

Le fait d'avoir n'importe quoi implique qu'il n'y ait ni schéma, ni structure pour le stockage. D'un point de vue de bases de données, il n'y a pas la possibilité d'exploiter ni de contrôler la structure des données et de fait, pas de langage (SQL = **Structured** Query Language). En soit ce n'est pas un problème si vous savez ce que vous cherchez (la clé) et que vous manipulez directement la valeur.



Les seules opérations de type CRUD peuvent être utilisées :

<

Immergez vos données dans le NoSQL

>

▶ 1. Choisissez votre famille NoSQL


2. Maîtrisez le théorème de CAP

3. Passez à l'échelle


4. Choisissez votre base de données NoSQL

📋 Quiz: Savez-vous vraiment ce qu'est le NoSQL ?


Created by



CentraleSupélec


Grande École d'ingénieurs :
cycle ingénieur, Master et
École Doctorale, Mastère
Spécialisé et formation
continue



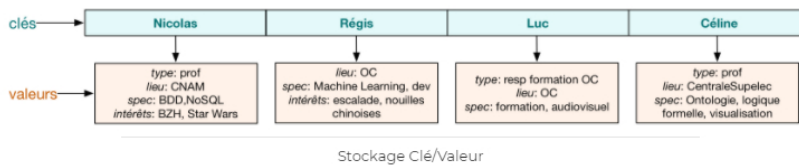
OpenClassrooms, Leading
E-Learning Platform in
Europe







- Create (key,value)
- Read (key)
- Update (key,value)
- Delete (key)



Qui s'intéresse à des fonctionnalités aussi simples ?

- **Redis** (VMWare) : Vodafone, Trip Advisor, Nokia, Samsung, Docker
- **Memcached** (Danga) : LiveJournal, Wikipédia, Flickr, Wordpress
- **Azure Cosmos DB** (Microsoft) : Real Madrid, Orange tribes, MSN, LG, Schneider Electric
- **SimpleDB** (Amazon)

Quels types d'applications pourraient correspondre à cette solution ? Voici quelques idées qui pourraient orienter votre choix : *détection de fraude en temps réel, IoT, e-commerce, gestion de cache, transactions rapides, fichiers de logs, chat.*

Des lignes vers les colonnes

Traditionnellement, les données sont représentées en ligne, représentant l'ensemble des attributs. Le stockage orienté colonne change ce paradigme en se focalisant sur chaque attribut et en les distribuant. Il est alors possible de focaliser les requêtes sur une ou plusieurs colonnes, sans avoir à traiter les informations inutiles (les autres colonnes).

Stockage orienté lignes					Stockage orienté colonnes					
id	type	lieu	spec	intérêts	id	type	id	lieu	id	spec
Nicolas	prof	CNAM	BDD, NoSQL	BZH, Star Wars	Nicolas	prof	Céline	CentraleSupélec	Nicolas	BDD
Régis		OC	Machine Learning, Dev	escalade, nouilles chinoises	Céline	prof	Nicolas	CNAM	Nicolas	NoSQL
Luc	resp formation OC	OC	formation, audiovisuel		Régis		Régis	OC	Régis	Machine Learning
Céline	prof	CentraleSupélec	Ontologie, logique formelle, visualisation		Luc	resp formation OC	Luc	OC	Régis	Dev
							Luc	formation	Luc	audiovisuel
							Céline	Ontologie	Céline	logique formelle
							Céline	visualisation	Céline	visualisation

Stockage orienté colonnes

Cette solution est très adaptée pour effectuer des traitements sur des colonnes comme les agrégats (comptage, moyennes, co-occurrences...). D'une manière plus concrète, elle est adaptée à de gros calculs analytiques. Toutefois, cette solution est beaucoup moins appropriée pour la lecture de données spécifiques comme pour les clés/valeurs.

Qui fait de l'orienté-colonnes ?

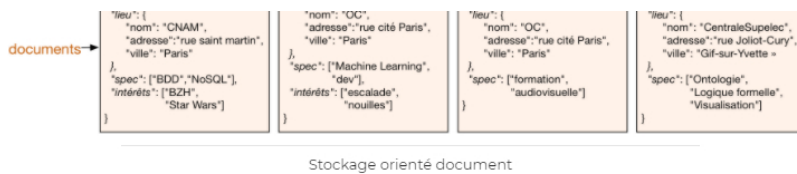
- **BigTable** (Google)
- **HBase** (Apache, Hadoop)
- **Spark SQL** (Apache)
- **Elasticsearch** (elastic) -> moteur de recherche

Quelques exemples d'applications : *Comptage* (vote en ligne, compteur, etc), *journalisation*, *recherche de produits dans une catégorie*, *reporting à large échelle*.

La gestion de documents

Les bases orientées documents ressemblent sans doute le plus à ce que l'on peut faire dans une base de données classique pour des requêtes complexes. Le but de ce stockage est de manipuler des documents contenant des informations avec une structure complexe (types, listes, imbrications). Il repose sur le principe du clé/valeur, mais avec une extension sur les champs qui composent ce document.





L'avantage de cette solution est d'avoir une approche structurée de chaque valeur, formant ainsi un document. De fait, ces solutions proposent des langages d'interrogation riches permettant de faire des manipulations complexes sur chaque attribut du document (et sous-documents) comme dans une base de données traditionnelles, tout en passant à l'échelle dans un contexte distribué.

Quelles solutions pour gérer vos documents :

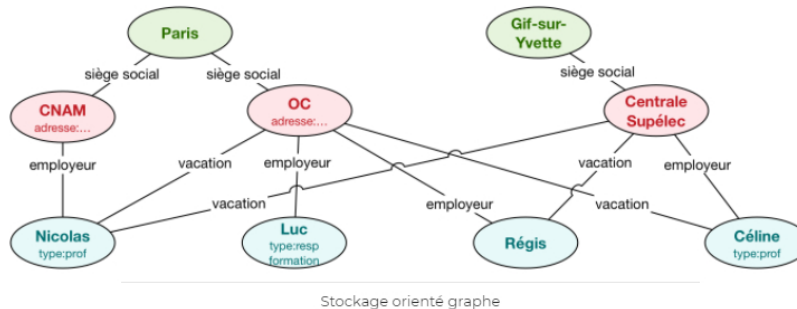
- **MongoDB** (*MongoDB*) : ADP, Adobe, Bosch, Cisco, eBay, Electronic Arts, Expedia, Foursquare
- **CouchBase** (*Apache, Hadoop*) : AOL, AT&T, Comcast, Disney, PayPal, Ryanair
- **DynamoDB** (Amazon) : BMW, Dropcam, Duolingo, Supercell, Zynga
- **Cassandra** (*Facebook -> Apache*) : NY Times, eBay, Sky, Pearson Education

Cassandra est souvent considérée comme une solution orientée colonnes. C'est une solution orientée documents appelée "wide-column store", dans le sens où le document est structuré comme en relationnel. La confusion vient du fait que l'on y définit des colonnes et des familles de colonnes dans le schéma, ce qui est différent du stockage de "colonnes".

Quelques exemples d'utilisation : *gestion de contenu (bibliothèques numériques, collections de produits, dépôts de logiciels, collections multimédia, etc.), framework stockant des objets, collection d'événements complexes, gestion des historiques d'utilisateurs sur réseaux sociaux.*

Et les graphes ?

Les trois premières familles NoSQL n'adressent pas le problème de corrélations entre les éléments. Prenons l'exemple d'un réseau social : dans certains cas, il devient très complexe de calculer la distance entre deux personnes non directement connectées. Et c'est ce type d'approche que résolvent les bases orientées Graphe.



Dans la base orientée graphe, les données stockées sont : les nœuds, les liens et des propriétés sur ces nœuds et ces liens. Les requêtes que l'on peut exprimer sont basées sur la gestion de chemins, de propagations, d'agrégations, voire de recommandations. Toutefois, contrairement aux solutions précédentes la distribution des nœuds sur le réseau n'est pas triviale.

Quelles bases de données gèrent de gros graphes ?

- **Neo4j** : eBay, Cisco, UBS, HP, TomTom, The National Geographic Society
- **OrientDB** (*Apache*) : Comcast, Warner Music Group, Cisco, Sky, United Nations, VErisign
- **FlockDB** (*Twitter*) : Twitter

Quelques idées d'applications reposant sur des bases orientées graphes : *réseaux sociaux (recommandation, plus court chemin, cluster...), réseaux SIG (routes, réseau électrique, fret...), web social (Linked Data).*

[← MAÎTRISEZ LES BASES DE DONNÉES
NOSQL](#)[MAITRISEZ LE THÉORÈME DE CAP →](#)

Teachers



Régis Behmo

Expert en machine learning, développeur fullstack, grimpeur invétéré et gros, très gros amateur de nouilles chinoises.



Nicolas Travers

Maître de Conférences en informatique au CNAM Spécialités : Bases de données, Optimisation BDD, SQL, NoSQL

OPENCLASSROOMS

[What we do](#)[Apprenticeship](#)[Path experience](#)[Our blog](#)

BUSINESS SOLUTIONS

[Business](#)

CONTACT

[FAQ](#)

LEARN MORE

[Work at OpenClassrooms](#)[Become a mentor](#) [Our store](#)[Terms of use](#)[Privacy policy](#)[Accessibility](#)[English](#) ▼