NoSQL

Organisé par :
Fatma Amira

Qu'est-ce que NoSQL ?

Le NoSQL (Not only SQL) désigne une catégorie de base de données apparue en 2009 qui se différencie du modèle relationnel que l'on trouve dans des bases de données connues comme MySQL ou PostgreSQL. Ceci permet d'offrir une alternative au langage SQL.

pourquoi NoSQL ?

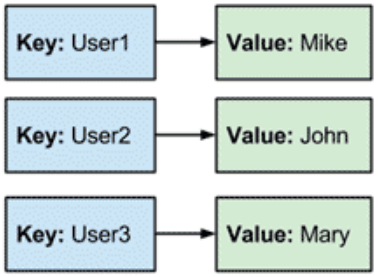
Le NoSQL est apparu afin de contrer la dominance des bases de données relationnelles dans le domaine de l'internet. En effet, un des problème récurrent des bases de données relationnelles est la perte de performance lorsque l'on doit traiter un très gros volume de données. De plus, la multiplication des architectures distribués a apporté le besoin de disposer de solution s'adaptant nativement aux mécanismes de réplication des données et de gestion de la charge.

Caractéristiques NoSQL

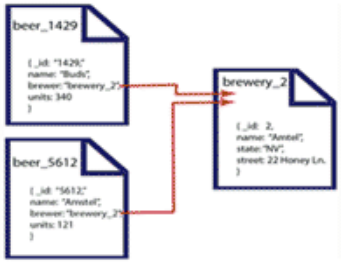
- **Le VOLUME :** des données créées double tous les 2 ans. IDC estime qu'en 2020 le volume atteindra 44 Zettabytes (1 ZB = 1 milliards de terabytes)
- **La VARIÉTÉ :** des types de données créées (Smartphones)
- **La VÉLOCITÉ :** avec laquelle les données changent est également très importante (Internet of Things)

Types de bases de données NoSQL

Clé-valeur



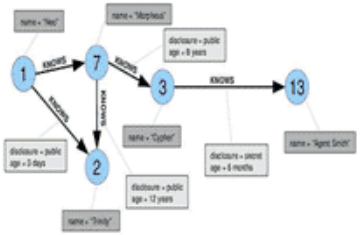
Document



Colonnes

Bases de données orientées aggrégats (BDOA)

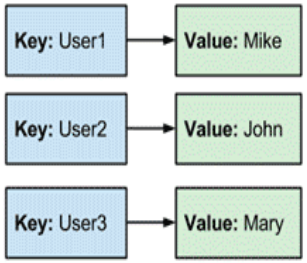
Graphes

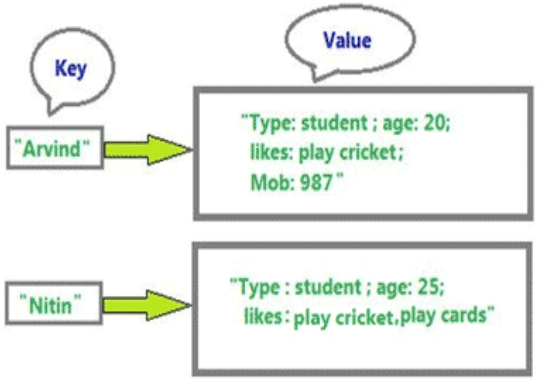


Bases de données orientées graphes (BDOG)

Type 1 : Entrepôts clé-valeur (ECV)

Les données sont stockées en clé-valeur : une clé plus un BLOB (dans lequel on peut mettre : nombre, date, texte, XML, photo, vidéo, structure objet).





Facilement scalable

Temps de réponse en écriture / lecture très bas

Mises à jour compliquées

Requêtes rudimentaires

Exemples d'implémentation

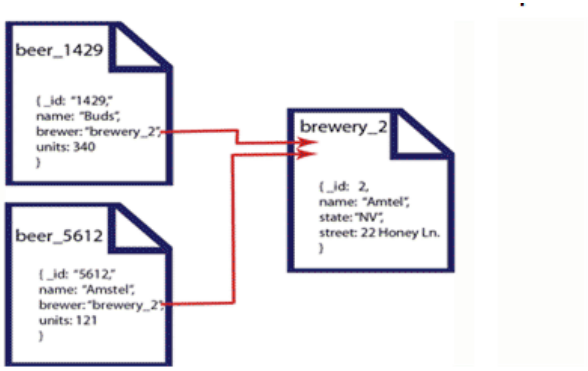
Amazon DynamoDB (Beta)

ORACLE BERKELEY DB 11g

redis

Type 2 : Bases orientées documents

Ces bases de données stockent des données semi-structurées : le contenu est formaté JSON ou XML, mais la structure n'est pas contrainte.



+

- Requétage plus complet
- Flexibilité
- Evolutif au cours du temps

−

- Duplication des données
- Cohérence pas forcément assurée

Exemples d'implémentation

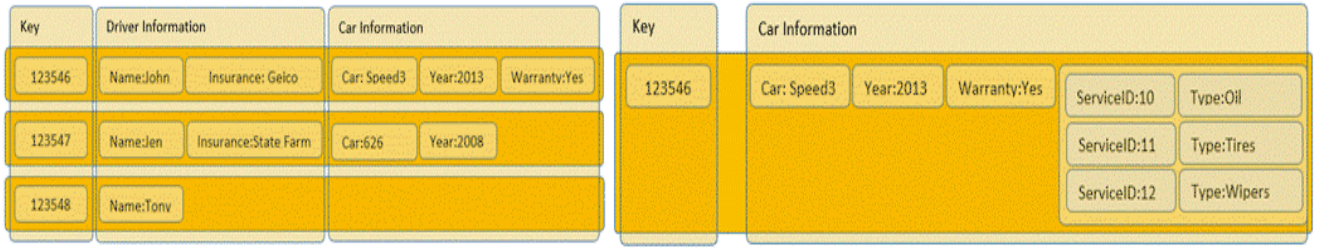
CouchDB relax

mongoDB

terrastore

Type 3 : Bases orientées colonnes

Ces bases de données se rapprochent des bases de données relationnelles, à ceci près qu'elles permettent de remplir un nombre de colonnes variable.



Key	Name	Insurance	Car	Year	Warranty
123456	John	Geico	Speed3	2013	Yes
123457	Jen	State Farm	626	2008	NULL
123458	Tony	NULL	NULL	NULL	NULL

ServiceID	Type	Key
10	Oil	123456
11	Tires	123456
12	Wipers	12345

+

Capacité de stockage accrue

Accès rapide aux données

−

Efficace surtout pour des données de même type et similaires

Requêtage limité

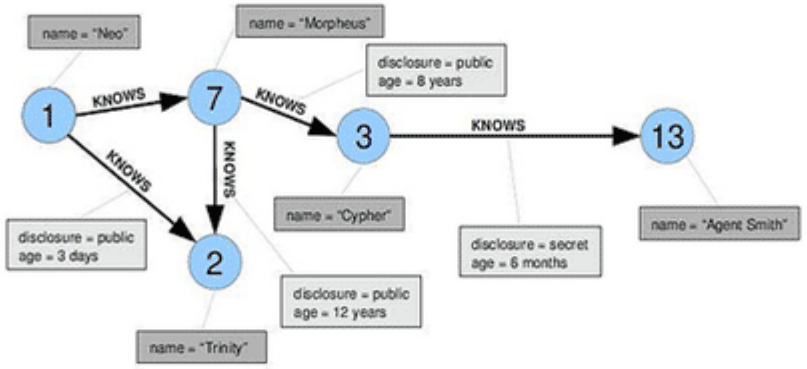
Exemples d'implémentation

APACHE HBASE

cassandra

Type 4 : Bases de données orientées graphes

Ces bases de données, basées sur la théorie des graphes, sont gérées par noeuds, relations et propriétés. Elles gèrent des données spatiales, sociales ou financières (dépôts/retraits).



Adapté à la gestion de données relationnelles

Architecture modelable

Architecture limitée à certains cas

Exemple d'implémentation

TITAN

Neo4j

Quel est le théorème ACID ?

ACID

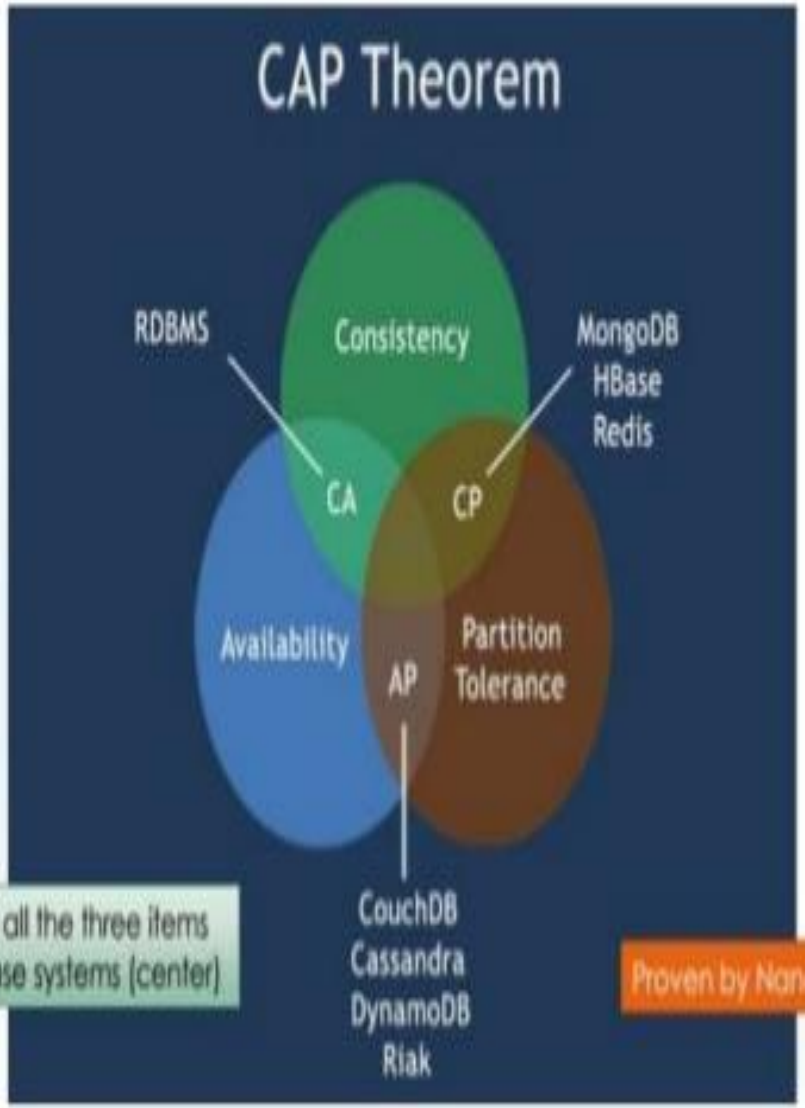
- **A**tomicity (Les tâches d'une transaction est exécutée ou aucune d'entre elles. C'est le principe du tout ou rien. Si un élément d'une transaction échoue, toute la transaction échoue)
- **C**onsistency (La BDD doit rester dans un état cohérent au début et à la fin d'une transaction. il n'y a jamais de transactions à moitié terminées).
- **I**solation (Aucune transaction n'a accès à une autre transaction dont l'état n'est pas terminé. Ainsi, chaque transaction est indépendante en soi. Cela est nécessaire pour la cohérence des transactions dans une base de données).
- **D**urability (Une fois la transaction terminée, elle persistera et ne pourra plus être annulée, Elle survivra aux pannes du système, aux coupures de courant et à d'autres types de pannes du système).

Quel est le théorème CAP ?

Théorème de CAP

Un système distribué ne peut prendre en charge que deux des caractéristiques suivantes:

- **Consistency** (toutes les nœuds dans un system distribué renvoie la même valeur)
- **Availability** (Chaque nœud non défaillant renvoie une réponse pour toutes les demandes de lecture et d'écriture dans un délai raisonnable)
- **Partition Tolerance** (Le système continue de fonctionner et maintient sa cohérence malgré les partitions de réseau)



We can not achieve all the three items
In distributed database systems (center)

Proven by Nancy Lynch et al. MIT labs.

Avantages NoSQL

Avantages

- Pas des données dupliquer,
- Très rapide,
- La simplicité de clé/ valeur les rend très rapides et léger,
- hautement évolutive (Highly Scalable),
- hautement disponible (Highly availability),
- Peut stocker n'importe quel type de valeur,
- Des operations basic : Insert(key, value),Update(key), Delete(key).