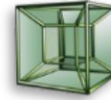


BASES DE DONNÉES NoSQL

Tarek Arif

GoMyCode – 2020



HYPERTABLE^{INC}





Plan

- Qu'est ce qu'une base NoSQL ?
- Utilité et usage des bases NoSQL
- Caractéristiques des bases NoSQL
- Bases NoSQL – 4 approches / 4 types
 - *Type 1 – Bases orientées Clé/valeur*
 - *Type 2 – Bases orientées documents*
 - *Type 3 – Bases orientées colonnes*
 - *Type 4 – Bases orientées graphes*
- Le théorème ACID
 - *ACID / BASE*
- Le théorème CAP
- Avantages des bases NoSQL
 - *... quelques inconvénients*
- Références



PRÉSENTATION DES BASES NOSQL

Définition – Utilité - Caractéristiques



Qu'est-ce qu'une base NoSQL ?



- **NoSQL** correspond à "*not only SQL*" (*pas seulement SQL*) et désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles.
- La raison principale de l'émergence et de l'adoption des SGBD NoSQL serait le développement des centres de données et la nécessité de posséder un paradigme de bases de données adapté à ce modèle d'infrastructure matérielle.
- NoSQL serait une approche de la conception des bases et de leur administration particulièrement utile pour de très grands ensembles de données distribuées et qui remet en cause de nombreux fondements de l'architecture SGBD relationnelle traditionnelle, notamment les propriétés ACID.
- Il ne s'agit pas d'une contrepartie, mais bien un enrichissement et complément utile des bases de données SQL relationnelles traditionnelles. Il existe de nombreuses variantes combinées au sein desquelles les deux solutions peuvent être utilisées et qui restent toutefois englobées sous l'étiquette NoSQL.

...



- Donc les bases de données NoSQL ne forment pas une famille homogène et n'interdisent pas complètement le célèbre langage de requête structurée SQL:
 - Certains systèmes NoSQL sont *intégralement non relationnels*,
 - d'autres se contentent d'éviter certaines fonctions relationnelles, telles que les schémas de tables fixes et les opérations de jointure. Par exemple, plutôt que d'utiliser des tables, une base de données NoSQL pourra organiser les données en objets, en paires clé/valeur ou en uplets (tuples).
 - Et en tant que bases NoSQL, elles abritent des technologies variées et sont souvent citée en association avec d'autres outils Big Data, tels que le traitement massivement parallèle, les bases de données orientées colonnes et la base de données en tant que service, ou DBaaS (Database-as-a-Service).
- Parmi les bases NoSQL les plus connues citons: **Apache Cassandra, SimpleDB, Google BigTable, Apache Hadoop, MapReduce, MemcacheDB et Voldemort.**

Utilité et usage des bases NoSQL

- NoSQL englobe une gamme étendue de technologies et d'architectures, afin de résoudre les problèmes de performances en matière d'évolutivité et de Big Data que les bases de données relationnelles ne sont pas conçues pour affronter. .
- NoSQL est particulièrement utile lorsqu'une entreprise doit accéder, à des fins d'analyse, à de grandes quantités de données non structurées ou semi-structurées ou de données stockées à distance sur plusieurs serveurs virtuels du Cloud. D'ailleurs des entreprises comme Netflix, LinkedIn et Twitter utilisent la technologie NoSQL.
- Les bases No-SQL sont surtout destinées à répondre aux problèmes de performances et aux problèmes de scalabilité. Sur cette base nous pouvons distinguer des problématiques majeures traitées par la technologie NoSQL, que nous allons aborder.

...

a) Les systèmes distribués à grosse volumétrie

- Les solutions No-SQL répondent aux problèmes typiques des grosses applications (gros volume et gros trafic). De fait, tous les acteurs majeurs de l'informatique portent des solutions No-SQL parce qu'ils sont confrontés aux problèmes de disponibilité. Il faut donc penser au No-SQL lorsqu'on souhaite des performances combinées à une forte scalabilité. Les solutions No-SQL garantissent une continuité de service lors des migrations (Il est hors de question que Google réplique ses informations pour effectuer une migration switch).

b) Minimiser les coûts

- Toujours pour les gros consommateurs de data, il faut penser au No-SQL pour minimiser les coûts. Parce qu'elles consomment moins d'énergie les bases de données SQL peuvent amoindrir la facture énergétique.
- Les solutions clé en main No-SQL dans le cloud (SimpleDB, Hypertable) sont 100 fois moins chères que leur alter ego relationnel.

...

c) Les structures évoluant dans le temps

- Le No-SQL est intéressant lorsque la définition de la donnée stockée peut varier dans le temps (développement rapide et prototypage, données polymorphes, systèmes géographiques).
- Le web sémantique est un représentant important de cette classe d'utilisation.

d) Le développement rapide

- Certaines bases No-SQL (CouchDB par exemple) sont intéressantes pour les développements Web. Leurs Apis sont interfaçables avec le protocole JSON (JavaScript Object Notation) et cet atout est précieux pour les développements JavaScript. Par exemple, ce type de base permet de réaliser rapidement un moteur de recherche.
- Concrètement, voici quelques exemples d'utilisation du No-SQL:
 - *Stockage massif de données en grands clusters (Facebook, mail etc.)*
 - *Moteur de recherche et web sémantique*
 - *Base de données géographique.*

Caractéristiques des bases NoSQL

- Bien que très différentes les bases NoSQL ont en commun un certain nombre de caractéristiques :
- Elles privilégient souvent la disponibilité et le partitionnement à la cohérence (théorème CAP)
- Leurs langages de requête sont souvent plus complexes, mais plus flexibles que le SQL (Java, PHP)
- Leurs schémas de stockage sont dynamiques (elles privilégient la flexibilité)
- Les bases NoSQL proposent différentes formes de structuration de données selon la nécessité.
- Adaptées aux données (devenues de nos jours de plus en plus) semi-structurées voire non structurées et rares.
- Elles ont une scalabilité verticale ET horizontale (partitionnement des données et partage des calculs)
- Leur scalabilité est linéaire
- Leur mode d'utilisation : peu d'écritures, beaucoup de lectures
- Leurs données sont distribuées : on a souvent la possibilité d'utiliser des algorithmes MapReduce et/ou du Spark

TYPES DE BASES – 4 APPROCHES / 4 TYPES

On regroupe les bases NoSQL en 4 familles qui diffèrent dans leur manière de stocker et d'accéder aux données en orientées: clé/valeur – documents – colonnes - graphes



amazon
DynamoDB



redis

ORACLE®

BERKELEY DB

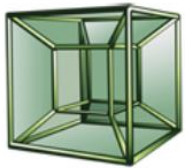
Type 1 – Bases orientées Clé/valeur

- Ces bases, ne permettent de stocker que des couples [clé, valeur]. Cette valeur peut être une simple chaîne de caractères, un entier, ou encore un objet beaucoup plus complexe pouvant contenir une multitude d'informations.
- Ce type de base de données offre de très bonne performances par sa simplicité et peut même être utilisé pour stocker les sessions utilisateur ou le cache de votre site par exemple.
- Dans cette catégorie on peut trouver: Amazon DynamoDB, Oracle BerkeleyDB et Redis.

Type 2 – Bases Orientées documents

- Ces solutions stockent principalement des données semi-structurées. Reposent également sur le paradigme [clé, valeur], et la valeur, dans ce cas, est un document. Ce document a une structure arborescente : il contient une liste de champs, un champs est associée à une valeur qui peut elle même être une liste. Ces documents sont principalement des objets de type JSON ou XML.
- L'avantage est de pouvoir récupérer simplement des informations structurées de manière hiérarchique.
- Les solutions les plus connues sont CouchDB, RavenDB et MongoDB.





HYPERTABLE INC

Type 3 – Bases orientées colonnes

- Ces bases sont très proches des bases de données relationnelles, on y retrouve le principe de « table », mais elles présentent deux grosses différences : les colonnes sont dynamiques (le nombre de colonnes peut varier d'une ligne à l'autre) et l'historisation de la données se fait à la valeur et non à la ligne.
- Les solutions les plus connues sont Apache Hbase, Cassandra ou Hypertable.

Type 4 – Bases orientées graphes

- Ces bases stockent les données en se basant sur la théorie des graphes. Elles s'appuient sur la notion de noeuds, de relations et de propriétés qui leur sont rattachées.
- Cette structure permet de récupérer simplement des relations complexes.
- Comme exemple de base graphe: Neo4J, Titan et Oracle NoSQL.





THÉORÈMES FONDAMENTALES

Théorèmes ACID – ACID et BASE – Théorèmes CAP



Théorème ACID

- Dans le domaine des bases de données, une opération sur les données est appelée une transaction ou transaction informatique. Par exemple, un transfert de fonds d'un compte de banque à un autre, même s'il implique plusieurs actions comme le débit d'un compte et le crédit d'un autre, est une seule transaction. **Les propriétés ACID** sont un ensemble de propriétés qui garantissent qu'une transaction informatique est exécutée de façon fiable.
- Ces propriétés ont été définies par Jim Gray en 1970. En 1983, Andreas Reuter et Theo Härder ont créé l'acronyme ACID pour désigner ces propriétés qui sont : *atomicité, cohérence, isolation* et *durabilité*.
- Les bases de données relationnelles respectent, en général ces propriétés, alors que les NoSQL s'en écartent pour pouvoir répondre à d'autres priorités comme la gestion de données massives et distribuées pour les usages du Big Data par exemple, ces derniers on les appellent **BASE**.

... Propriété ACID

L'Atomicité

Signifie qu'une transaction se fait au complet ou pas du tout. Si une partie d'une transaction ne peut être faite, il faut effacer (Rollback) toute trace et remettre les données à leur état antérieur.

Ainsi, la base ne risque pas d'être corrompue par des opérations à moitié complétées lors d'une panne ou une défaillance.

L'Isolation

Signifie que toute transaction doit s'exécuter comme si elle était la seule sur le système; aucune dépendance possible entre les transactions. Elle garantit que les écritures et lectures des transactions réussies ne seront pas affectées par les écritures et lectures d'autres transactions, qu'elles soient ou non réussies. On qualifie l'isolation de "pessimiste" et "optimiste" .

La Cohérence

Cette propriété assure que chaque transaction amènera le système d'un état valide à un autre état valide. Elle signifie que les transactions doivent respecter les contraintes d'intégrité des données de la base de données.

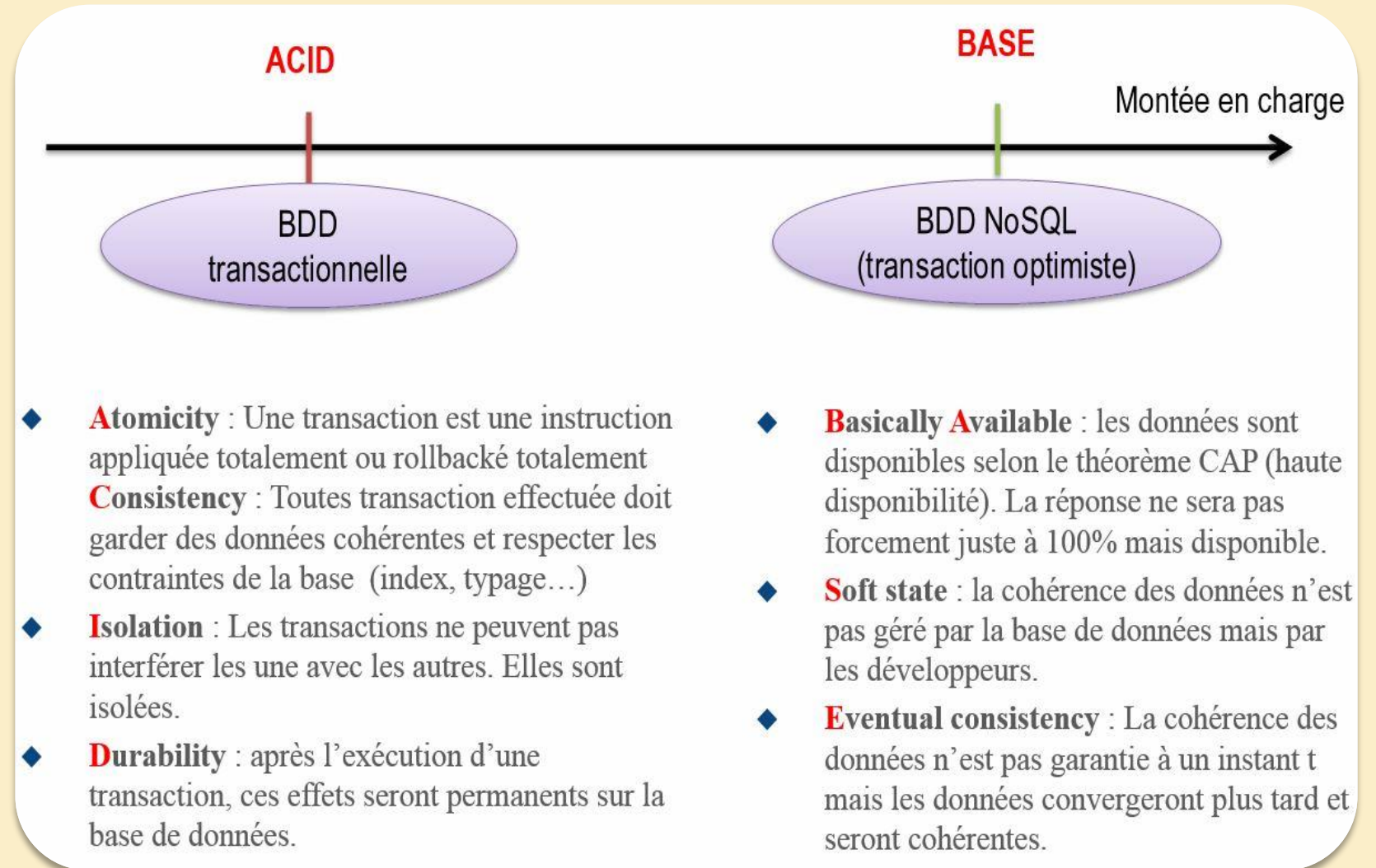
Pour maintenir la cohérence, un DBMS peut abandonner les transactions qui risquent de provoquer une incohérence.

La Durabilité

Cette propriété garantit que les transactions réussies restent enregistrées de façon permanente et ne seront pas affectées par d'éventuelles pannes ou problèmes techniques. Les changements apportés aux données doivent être permanents. Pour y arriver, les DBMS écrivent des logs sur les changements effectués pour les utiliser en cas de panne.

... Les propriétés BASE

La différence entre les bases de données qui respectent la **norme ACID** (les bases de données relationnelles et certaines bases de données NoSQL) et les bases de données qui respectent la **norme BASE** (généralement les bases de données NoSQL).



Théorème CAP

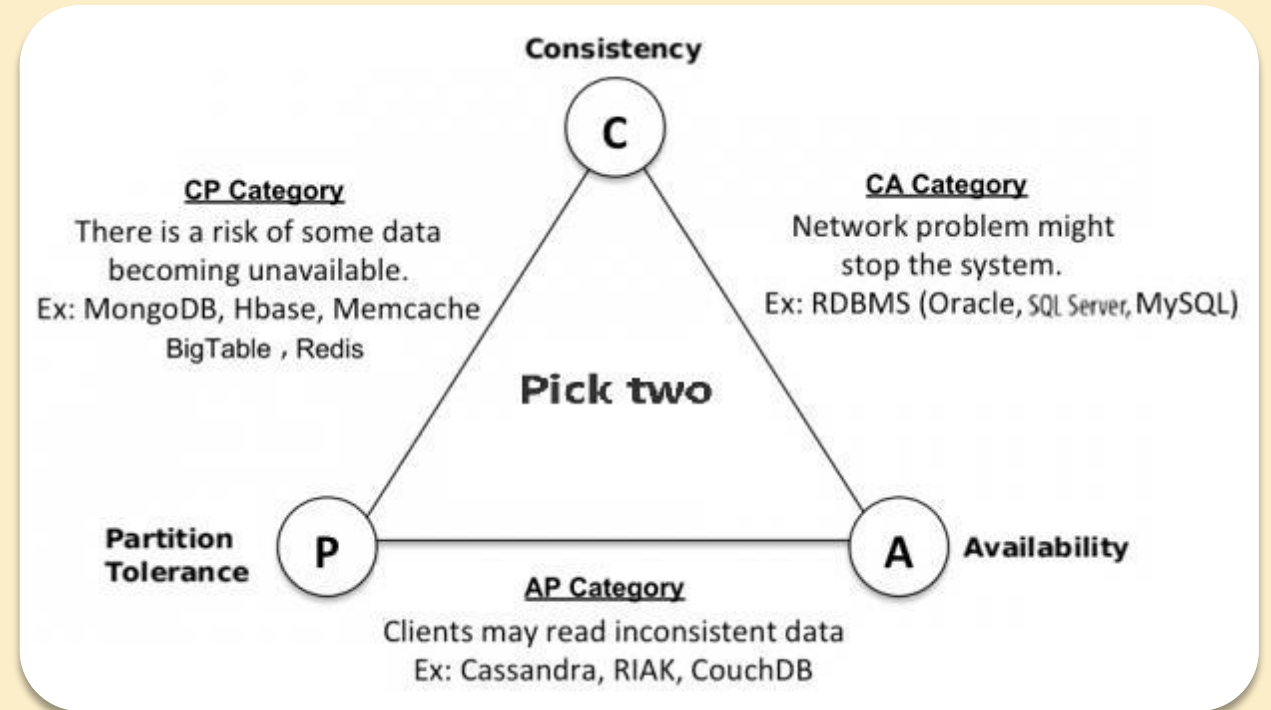
- En 2000, Eric A. Brewer a formalisé un théorème très intéressant reposant sur 3 propriétés fondamentales pour caractériser les bases de données (relationnelles, NoSQL et autres). Le **théorème CAP** ou **CDP**, ou **théorème de Brewer** : dit qu'il est impossible sur un système informatique de calcul distribué de garantir en même temps (c'est-à-dire de manière synchrone) les trois contraintes suivantes:

Cohérence (" <i>Consistency</i> ") :	Tous les nœuds du système (tous les clients) voient exactement les mêmes données au même moment.
Disponibilité (" <i>Availability</i> ") :	Garantir que toutes les requêtes reçoivent une réponse. Les données doivent être disponibles pour tous les clients pour les opérations de lecture et d'écriture.
Tolérance au partitionnement (" <i>Partition Tolerance</i> ") :	En cas de panne d'un nœud individuel ou lorsque des nœuds individuels ne sont plus en mesure de communiquer ensemble, le système peut continuer à fonctionner comme un tout.

- D'après ce théorème, un système de calcul/stockage distribué ne peut garantir à un instant T que deux de ces contraintes mais pas les trois.

... CAP

- Il est très important de comprendre les limites des bases de données NoSQL. NoSQL ne peut pas assurer la cohérence et la haute disponibilité ensemble. Cela est bien exprimé dans le théorème du CAP.



- Les informaticiens s'appuient sur ce théorème lors de la création d'un nouveau système distribué et choisissent un modèle de base axé sur deux des trois propriétés. Ainsi on peut distinguer trois catégories de systèmes :
 - **système CP** (cohérence et tolérance au partitionnement)
 - **système AP** (disponibilité et tolérance au partitionnement)
 - **système CA** (cohérence et disponibilité)



APPORT DES BASES NOSQL

Avantages - inconvénients



Avantages des bases NoSQL

- Les bases de données NoSQL présentent des nombreux avantages par rapport aux bases de données traditionnelles. Leur modèle de données résout des problématiques auxquelles l'architecture relationnelle est incapable de répondre:

Les BDD NoSQL affichent une scalabilité et des performances bien supérieures que les SGBDR.

Volumes importants de données structurées, semi-structurées et non-structurées en constante mutation,

Sprints agiles, itération de schéma rapide et intégrations fréquentes de nouveau code,

Programmation orientée objet d'une grande flexibilité et simplicité d'utilisation,

Architectures à scalabilité horizontale réparties géographiquement sur de nombreux sites, en lieu et place des architectures monolithiques coûteuses,

L'implémentation de bases de données NoSQL de source ouverte est rentable. Puisqu'ils n'ont pas besoin de frais de licence et peuvent fonctionner sur du matériel économique.

... quelques inconvénients

- Les bases de données NoSQL présentent des inconvénients la plupart découlent de leur non respect des propriétés ACID :

La plupart des bases de données NoSQL ne prennent pas en charge les fonctions de fiabilité ACID, qui sont soutenues par les systèmes de bases de données relationnelles.

D'où découle un autre inconvénient, car afin de garantir une certaine fiabilité et cohérence, les développeurs doivent implémenter leur propre code, donc ajouter une complexité supplémentaire au système.

Cela pourrait limiter le nombre d'applications sur lesquelles nous pouvons compter pour effectuer des transactions sécurisées et fiables, telles que des systèmes bancaires.

L'incompatibilité avec les requêtes SQL est l'une des complexités trouvées dans la plupart des bases de données NoSQL. Cela signifie qu'un langage de requête manuelle est nécessaire, ce qui rend les processus beaucoup plus lents et complexes.

Références

- ACID : les 4 propriétés de transactions de bases de données – <https://www.lebigdata.fr/acid-base-de-donnees-definition>
- Base de données NoSQL: quelles sont ses caractéristiques? – <https://pandorafms.com/blog/fr/bases-de-donnees-nosql/>
- Définition bases de données NoSQL | MongoDB – <https://www.mongodb.com/fr/nosql-explained>
- Le NoSQL, qu'est ce que c'est? comment ça marche? et à quoi ça sert?! – <https://www.illustradata.com/nosql-quest-cest-ca-marche-a-quoi-ca-sert>
- Les bases No-SQL – <https://www.technologies-ebusiness.com/langages/les-bases-no-sql>
- nosql - Redis availability and CAP theorem - Stack Overflow – <https://stackoverflow.com/questions/59511275/redis-availability-and-cap-theorem>
- NoSQL — Wikipédia <https://fr.wikipedia.org/wiki/NoSQL>
- Propriétés ACID — Wikipédia – https://fr.wikipedia.org/wiki/Propriétés_ACID

■ ■ ■

- Qu'est-ce qu'une base NoSQL ? Les cas Datastax et MongoDB - Digora.com – <https://www.digora.com/fr/blog/definition-base-nosql-datastax-mongodb>
- Que signifie NoSQL (base de données « Not Only SQL »)? - LeMagIT.fr – <https://www.lemagit.fr/definition/NoSQL-base-de-donnees-Not-Only-SQL>
- Qu'est ce que le NoSQL ? - grafikart.fr – <https://www.grafikart.fr/blog/sql-nosql>
- Qu'est-ce qu'une base de données NoSQL ? - WayToLearnX – <https://waytolearnx.com/2019/10/quest-ce-quune-base-nosql.html>
- Théorème CAP - WayToLearnX – <https://waytolearnx.com/2019/10/theoreme-cap.html>
- Théorème CAP — Wikipédia – https://fr.wikipedia.org/wiki/Théorème_CAP
- Théorème CAP : que dit le théorème de Brewer ? - IONOS – <https://www.ionos.fr/digitalguide/serveur/know-how/le-theoreme-cap-quest-ce-que-cest/>
- TOP 10 - Meilleures bases de données NoSQL 2021 - Ambient-it.net formations – <https://www.ambient-it.net/top-meilleures-db-nosql-2021/>