Réponses aux questions

✔ pour fini

# Définir le BigData

1. Pourquoi le BigDate ... Définir les 5V du Big Data ... ✔
2. Définir le NoSQL :

- données structurées / semi / non

- Clé/Valeur, colonnes, documents, graphes

- théorème de CAP de Brewer

1. Définir le DataLake

# Présentation des outils

1. UN EXEMPLE d'architecture BigData à étudier :

- Hadoop HDFS : intégration et stockage des fichiers et données brutes (non transformées) dans HDFS

- Yarn : l’exécution simultanée de multiples applications + mixer des workloads en batch, interactif et en temps réel

- MapReduce / Spark : pour réaliser les traitements d’intégration et de calculs (entre HBase et Hive)

- HBase (BD NoSQL colonnes) / Phoenix (permet de faire du OLTP/ACID) : stockage data

- Hive : reporting / analyse de données / SQL (permet de faire du SQL sur HBase)

Pour compléter, voici une vidéo que j'avais trouvé très instructive quand j'avais étudié le sujet :

https://www.youtube.com/watch?v=CNUlNk0KtKs

1. Présenter les distributeurs Hadoop sur la place (Cloudera, MAPR, Hortonworks,...)

https://www.lebigdata.fr/hadoop-top-des-distributeurs

1. Présenter le mode de fonctionnement de Hadoop HDFS (NameNode/Job Tracker, DataNode/Task Tracker, ...)

# Démonstration

En fonction de l'architecture technique que vous allez présenter, on vous invite à nous présenter concrètement :

1. Comment s'installe l'outil (DFS, DRILL, ...)
2. À quoi ressemble l'interface de développement
3. À quoi ressemble le développement

DRIll : moteur SQL facile à installer avec du "real SQL" s'exécutant sur toutes les plateformes

# Qu’est-ce que le BIGDATA ?

Le big data (où données massives) défini un ensemble de données trop volumineux pour être stocker ou analyser avec les outils classiques de base de données.

De nos jours, la quantité de données partagé dans le monde ne cesse de croitre. Ces données et leurs analyses sont recherché par certaines organisations dans de multiples buts. Elles permettraient notamment de pouvoir faire de meilleure analyse tendancielle dans les domaines de la sociologie (prévision des risques de conflits, lutte contre l’insécurité …), de la science (compréhension des épidémies, analyse des changements climatiques) et de l’économie (meilleure prévision du marché, meilleure analyse des bénéfices possibles).

## Définition des 5V (3V étendu au 5V)

*Source :* [*http://www.lsis.org/espinasseb/Supports/BD/Article-BigData-TI-2016.pdf*](http://www.lsis.org/espinasseb/Supports/BD/Article-BigData-TI-2016.pdf)[*https://fr.blog.businessdecision.com/bigdata/2015/01/3v-opportunites-big-data/*](https://fr.blog.businessdecision.com/bigdata/2015/01/3v-opportunites-big-data/)

Pour pouvoir stocker des données massives, le domaine du BIG DATA fut caractérisé. Les trois premières caractéristiques sont le Volume, la Variété et la Vélocité. Ces caractéristiques ont ensuite été rejoint par deux nouvelles caractéristiques, la Valeur et la Véracité.

### Le Volume

Le volume fait référence à la quantité de données à traiter. Elles sont parfois trop volumineuses pour les outils d’acquisition standard. On interprète la caractéristique « Volume » comme étant le traitement à faire sur un grand ensemble de données.

### La Variété

La variété fait référence aux différents types de formats présents dans un ensemble massifs de données. Cette variété de format peut être dû à l’origines de ces informations (différents capteurs, différentes sources), au format d’échange de ces informations (http, Json, XML) et aux différents formats de pré-stockage des données (Json, XML).

### La Vélocité

On parle de vélocité lorsque le temps entre l’acquisition de la données et son traitement sont à minimiser. En effet, l’exploitation d’une masse d’information doit être fait rapidement pour que des prises de décision en conséquence puissent être rapidement prise. De plus, un très grand volume de données ne peut être facilement stocker. Il est donc nécessaire de rapidement les analyser en flux.

### La Valeur

La valeur définit le caractère économique de l’informations une fois analyser. Chaque ensemble de données, une fois analyser, pourrait permettre à un organisme de gagner de l’argent (ou au mieux, en économiser).

### La Véracité

La véracité d’une donnée fait référence à la validité de la donnée. En d’autre thermes, on cherchera à savoir si une donnée est aberrante, manquante ou la confiance envers la donnée. Le Volume important de données peut compliquer la vérification de la véracité de nos données.

Ces caractéristiques, une fois respectés, permettent de garantir une analyse d’un grand ensemble de données de qualité.

# Qu’est-ce que le NO SQL ?

Pour mettre en place une base de données répondant aux demandes du BIGDATA, L’utilisation des bases de données classique était impossible. Il a donc été développé un nouveau style de base de données dit NoSQL pour « Not only SQL » répondant à plusieurs caractéristiques :

- Le coût raisonnable et la facilité de mise en œuvre.

- Le partitionnement et la copie des fichiers de données sur plusieurs machines.

- La structure dynamique n’ayant pas de schéma de données fixe.

- L’évolutivité en rajoutant des colonnes, ce qui permet de traiter les données plus rapidement, surtout les plus volumineuses.

- La rapidité du transfert des données comparé aux bases de données classiques.

- L’évolutivité en rajoutant des nœuds supplémentaires dans le Cluster sans avoir besoin de faire une répartition.

Ces caractéristiques font que les bases de données NoSQL ne respectent pas les règles classiques des bases de données relationnels. En effet, les base de données NoSQL ne sont pas conformes au propriétés ACID et sont limités par le théorème CAP. Les réseaux sociaux appliquent fortement l’utilisation des bases de données NoSQL, vu leurs besoins compatibles avec CAP et contrairement aux banques nécessitant plus de rigidité.

## ACID vs BASE

*Source :* [*https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap*](https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap)

Les propriété ACID pour les transactions d’une base de données relationnelle sont :

* **A**tomicité : Une transaction s’effectue entièrement ou pas du tout
* **C**ohérence : Le contenu d’une base doit être cohérent au début et à la fin d’une transaction
* **I**solation : Les modifications d’une transaction ne sont visibles/modifiables que quand celle-ci a été validée
* **D**urabilité : Une fois la transaction validée, l’état de la base est permanent (non affecté par les pannes ou autre)

Comme on l’a vu précédemment, Une base de données NoSQL ne respecte pas ces propriétés ACID. En effet, une base de données NoSQL peut être déployé sur plusieurs serveur ce qui nécessite une synchronisation de l’ensemble des serveurs lors d’une transaction. La transaction sera donc lente.

On peut néanmoins remarquer qu’une base de données NoSQL est caractérisé par des propriétés, on nomme cet ensemble de propriété « BASE » :

* **B**asically **A**vailable : quelle que soit la charge de la base de données (données/requêtes), le système garantie un taux de disponibilité de la donnée
* **S**oft-state : La base peut changer lors des mises à jour ou lors d'ajout/suppression de serveurs. La base NoSQL n'a pas à être cohérente à tout instant
* **E**ventually consistent : À terme, la base atteindra un état cohérent

Avec NoSQL, on remarque que la disparition de contrainte ACID permet d’optimiser le temps de réponse entre les transactions.

## Le théorème de Brewer

*Source :* [*https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap*](https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap)

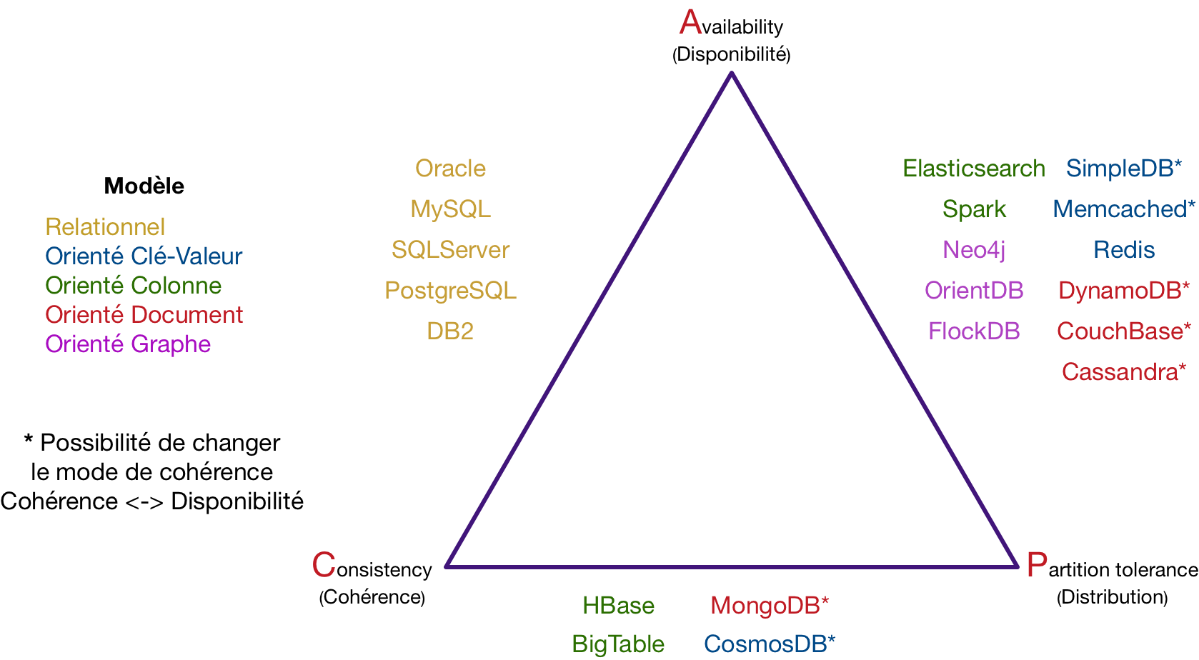
Le théorème de Brewer (dit théorème CAP) **caractérise une base de données en 3 points** :

* **C**onsistency (Cohérence). Les données sont uniformisées, il n’y a pas deux manières de rentrer une même information.
* **A**vailability (Disponibilité). La donnée doit être disponible.
* **P**artition Tolerance (Distribution). Le résultat retourné doit être correcte peut importe le nombre de serveurs utilisés pour stocker les données.

Ce théorème nous dit aussi **qu’une base de données ne peut respecter au mieux que 2 points**. Une base de données peut donc être caractérisé par trois couples :

**CA** : Cohérence et Disponibilité, ceci n’est possible que dans le cadre des Base de données relationnel (donc les SQBDR)

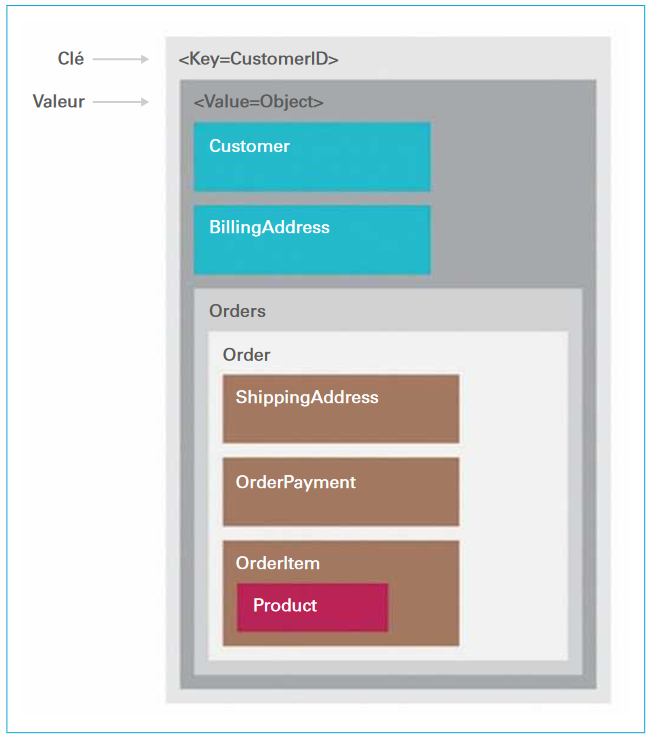
**CP** : Cohérence et Distribution, on réparti notre information sur plusieurs serveur. La communication avec les différents serveurs prend du temps, on ne peut pas garantir la disponibilité des données. On retrouve cela dans la base NoSQL MongoDB.

**AP** : Disponibilité et Distribution, on privilégie le temp de réponse à la cohérence de l’information. La base de données NO SQL Cassandra a ces caractérisations.

Certaines Base de données permettent de changer de caractéristique. On peut notamment passer du **AP** au **CP** avec MongoDB.

## Les types de base de données NO SQL //à analyser

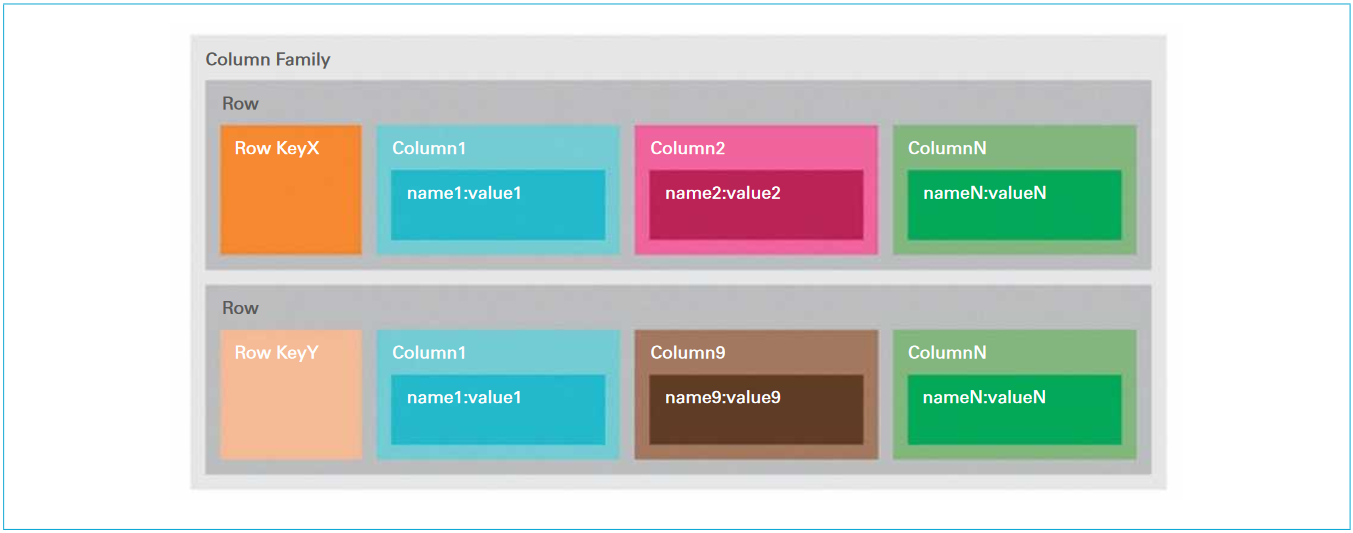
*Source :* [*http://www.lsis.org/espinasseb/Supports/BD/Article-BigData-TI-2016.pdf*](http://www.lsis.org/espinasseb/Supports/BD/Article-BigData-TI-2016.pdf)



### Modèle orienté « clé-valeur »

Les données sont simplement représentées par un couple clé-valeur. La valeur peut être une simple chaîne de caractères, un objet sérialisé... Cette absence de structure et de typage a un impact important sur le requêtage. Ainsi la complexité d’une requête qui sera dans un système relationnel prise en charge par le langage SQL, sera dans ce modèle NoSQL prise en charge par l’applicatif interrogeant la base de données, la communication avec la base se limitant généralement à des ordres PUT, GET et DELETE. Les systèmes NoSQL orientés clé-valeur les plus connus sont Memcached, Amazon’s Dynamo, Redis, Riak et Voldemort créé par LinkedIn.

### Modèle orienté « colonnes »

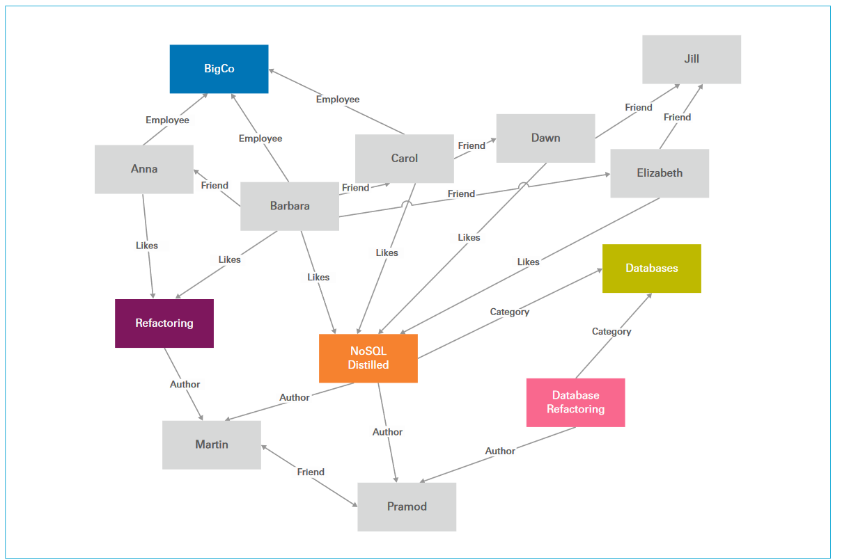
Ce modèle (figure 3) ressemble à première vue à une table du modèle relationnel, du fait que les attributs sont regroupés en famille de colonnes. Ainsi, deux attributs qui sont fréquemment interrogés ensemble seront stockés au sein d’une même famille de colonnes. Cependant la différence est que, dans cette base NoSQL orientée colonnes, le nombre de colonnes est dynamique, alors que dans une table relationnelle, le nombre de colonnes est fixé dès la création du schéma de la table. De plus, dans ce modèle, contrairement au modèle relationnel, le nombre de colonnes peut varier d’un enregistrement à un autre, ce qui évite de retrouver des colonnes ayant des valeurs inconnues (Null Value). Les systèmes NoSQL orientés colonnes les plus connus sont principalement HBase, implémentation Open Source du modèle BigTable développé par Google, et Cassandra, projet Apache qui respecte l’architecture distribuée de Dynamo d’Amazon, et le modèle BigTable de Google.

### Modèle orienté « documents »



Ce modèle (figure 2) se base sur le paradigme clé-valeur précédent ; cependant dans ce nouveau modèle la valeur est un document de type JSON ou XML. Ainsi, dans ce modèle, les données sont stockées à l’intérieur de documents. Un document peut être vu comme un n-uplet d’une table dans le monde relationnel, à la différence toutefois que les documents peuvent avoir une structure complètement différente les uns des autres. L’avantage est de pouvoir récupérer, via une seule clé, un ensemble d’informations structurées de manière hiérarchique. La même opération dans le monde relationnel impliquerait plusieurs jointures. Les systèmes NoSQL orientés documents les plus connus sont CouchDB d’Apache, RavenDB (destiné aux plateformes .NET/Windows avec la possibilité d’interrogation via LINQ) et MongoDB.

### Modèle orienté « graphe »

Ce modèle (figure 4) qui repose sur la théorie des graphes, permet de représenter les données sous la forme de graphes. Le modèle s’appuie sur la notion de nœuds, de relations et de propriétés qui leur sont rattachées. Les entités sont alors les nœuds du graphe et les relations que partagent les entités sont alors des arcs que ces entités. Ce modèle est notamment adapté au traitement des données des réseaux sociaux. Notons que les systèmes NoSQL orientés graphe trouvent un certain intérêt pour des applications dans le domaine du Web Sémantique, dans la gestion de bases de données de triplets RDF (triple-stores), permettant de stocker des connaissances ou ontologies, un triplet étant une arête d’un graphe. Les systèmes NoSQL orientés graphe les plus connus sont Neo4J, Infinite Graph, OrientDB.

# Qu’es ce que le DataLake ?

Un Data Lake est un dépôt de données brutes accessible en lecture seule qui doit être la source de données de référence utilisée par les autres applications de traitement de données. Le Data Lake contiendra des jeux de données qui serviront de point de référence : c'est ce qu'on appelle le master dataset.

On peut déduire de ces observations nos premiers besoins concernant le master dataset :

1. Les données ne seront écrites une seule fois (Write once)
2. Le master dataset ne subira que des ajouts (Append-only)
3. Le master dataset sera lu de nombreuses fois (Read many times)
4. Les opérations de lecture seront séquentielles (Sequential reads)

Le master dataset est voué à croître strictement avec le temps, il faut que la solution de stockage soit peu coûteuse. Par ailleurs, il faut que la solution retenue passe à l'échelle : il faut donc que le stockage se fasse de manière distribuée sur plusieurs machines.