MOOCEBFR4 : Ingénierie des Données du Big Data : SGBD NoSql et Lacs de Données avec Big Data SQL par la pratique

Gabriel MOPOLO-MOKE
Professeur chargé d'enseignements
Université Côte d'Azur (UCA)

2022 / 2023

Plan Général

- Plan
  - Module M4.1 : Rappel sur les concepts du Big Data et des SGBD NOSQL
  - Module M4.2 : Introduction à Oracle NOSQL
  - Module M4.3 : Oracle NoSql et le Modèle Key/Document
  - Module M4.4 : INTRODUCTION A MONGODB ET LE MONGO SHELL
  - Module M4.5 : INTRODUCTION A MONGODB ET SON API JAVA
  - Module M4.6 : Architectures Big data et construction de lacs de Données avec Big Data SQL par la pratique

Module M4.1 :
Rappel su les Concepts du Big Data et des SGBD NoSql

G. Mopolo-Moké
Professeur chargé d'enseignements
Université Côte d'Azur (UCA)

2022 / 2023

Module M4.1 : Rappel su les Concepts du Big Data et des SGBD NoSql

ESTIA INSTITUTE OF TECHNOLOGY

UNIVERSITÉ CÔTE D'AZUR

Plan

- Module M4.1, section 1 : Rappel sur les Concepts du Big Data
- Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL
- Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL
- Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

UNIVERSITÉ CÔTE D'AZUR

ESTIA INSTITUTE OF TECHNOLOGY

DATUM ACADEMY

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

FUN FRANCE UNIVERSITÉ NUMÉRIQUE

anr

INVESTIR L'AVENIR

MIAGE RÉSEAU DES MIAGE DE FRANCE Nice

MBDS

BIHAR

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

Plan

- **Les cinq V**
  - **V1:** Variété des Données du Big Data
  - **V2:** Volumes des Données du Big Data
  - **V3:** Vélocité des Données du Big Data
  - **V4:** Véracité des Données du Big Data
  - **V5:** Valeur des Données du Big data
- **Quel système pour gérer efficacement les données volumineuses ?**

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V1: Variété des Données du Big Data

▪ Variété des sources

- **Les données classiques des entreprises**
  - ✓ Les données des clients issue des CRM,
  - ✓ Les données transactionnelles des ERP
  - ✓ Les transaction issues du commerce électronique sur le WEB
  - ✓ Les données de comptabilité générale.

- **Les données générées EN TEMPS REEL / les données des capteurs**
  - ✓ Détail des appels
  - ✓ Les logs du web
  - ✓ smart meters (compteurs communicants)
  - ✓ Industrie des capteurs
  - ✓ Les logs des équipements
  - ✓ Les données de la bourse

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V1: Variété des Données du Big Data

- **Variété des sources**
  - **Les données des réseaux sociaux**
    - ✓ Les avis, les commentaires des clients
    - ✓ Sites de micro-blogging tels que Twitter,
    - ✓ Réseaux sociaux tels que Facebook
  - **Les données gouvernementales** : Exemple, Les données du commerce extérieur de la France
  - **Les données du WEB SEMANTIQUE** (RDF …)
  - **Les Données en flots(**Streaming Data)

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V1: Variété des Données du Big Data

- **Les données peuvent être regroupées en TROIS CLASSES**

  - **Les données structurées**
    - ✓ Relationnelles
    - ✓ Objets

    << Schéma séparé des données. Typage fort. Contraintes fortes >>

  - **Les données Semi-structurées**
    - ✓ XML
    - ✓ JSON, …

    << Les données sont structurées mais, chaque donnée à son schéma. Typage faible. Contraintes faibles>>

  - **Les données non structurées**
    - ✓ Textes
    - ✓ Images
    - ✓ Vidéos
    - ✓ …

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V1: Variété des Données du Big Data

- **Les données peuvent être regroupées en TROIS CLASSES**

  - **Les données structurées : Exemple avec Oracle**
    - ✓ CREATE TABLE Etudiant (id number(4), nom varchar2(30));
    - ✓ **INSERT** INTO **ETUDIANT VALUES** (1, 'DUPOND'); -- OK
    - ✓ **INSERT** INTO **ETUDIANT VALUES** ('UN', 10);-- Faux

  - **Les données Semi-structurées : Exemple MONGODB**
    - ✓ db.**etudiant**.insertOne({_id:1, nom:'DUPOND'});
    - ✓ db.**etudiant**.insertOne({_id:'UN', nom:10});

  - **Les données non structurées : Exemple Oracle NOSQL**
    - ✓ **Put kv –key /Key1 –value** "{id:1, nom:'DUPOND'}"

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V2 : Volume des Données du Big Data

Value(Byte)	Métrique	
1000	kB	kilobyte (10^3 bytes)
1000^2	MB	mégabyte (10^3 bytes)
1000^3	GB	gigabyte (10^3 bytes)
1000^4	TB	térabyte (10^{12} bytes)
1000^5	PB	pétabyte (10^{15} bytes)
1000^6	EB	exabyte (10^{18} bytes)
1000^7	ZB	zettabyte (10^{21} bytes)
1000^8	YB	yottabyte (10^{24} bytes)

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V2 : Volume des Données du Big Data

▪ Google

- Plus de **130 milliards de pages** indexées chaque jour
- **20 milliards de sites** parcourus (crawlées) chaque jour
- **80 millions de requêtes chaque seconde**, soit 6,9 milliards par jour
- **15% des requêtes sont nouvelles** (Plus de 500 millions par jour)
- **Plus de 110 millions de Go** sont stockées sur les serveurs de Google
- **Plus de 90%** des recherches **en France** se font sur Google

https://www.blogdumoderateur.com/chiffres-google/#:~:text=130%20000%20milliards%20de%20pages,(500%20millions%20par%20jour)%20!

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

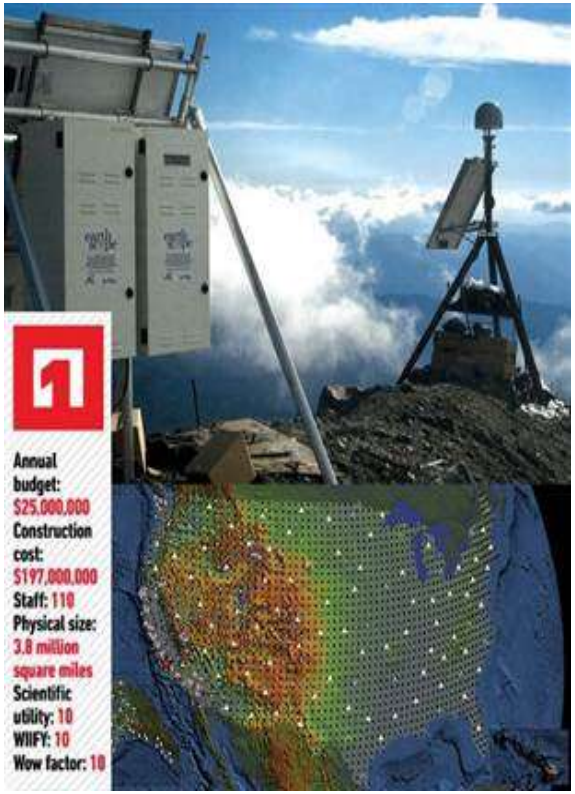
➢V2 : Volume des Données du Big Data

- **Facebook**
  - **Environ 3 Milliard d'utilisateurs** par mois
  - **2.5 Pétabytes de données** utilisateurs **+ 500 Terabytes/jour**
- **Youtube**
  - **2 milliards d'utilisateurs** actifs par mois
  - **43 000 vidéo visionnées chaque seconde** et **1 milliard par heures chaque jour, 24 milliard chaque jour**
  - **500 heures** de vidéos **sont mises en ligne chaque** minutes, **30000 vidéos** par heure, **720 000 heures** de vidéos **par jour**
  - Il **faudrait 82 ans pour regarder la totalité** des vidéos de la plateforme

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V2 : Volume des Données du Big Data

- **Projet Earthscope**
  - **Projet scientifique de recherche** mondial de 2003 à 2018
  - Conçu pour suivre les évolutions géologiques de l'Amérique du nord
  - Cet observatoire enregistre des données sur une superficie de 6 millions de m2
  - Amasse environ **1 Térabytes** de données **tous les trimestres**
  - Permet d'analyse des données sismiques
  - Pour plus d'infos (https://www.earthscope.org/)



Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V2 : Volume des Données du Big Data

- **Aviation**
  - Un seul Avion de ligne peut générer jusqu'à 10 Terabytes de données toutes les 30 minutes.
  - **Avec 25000 Avions par jour**, les données peuvent dépasser le **Pétabyte**

➢V2 : Volume des Données du Big Data

- **Exemple : Voitures Autonomes**

- **Selon Brian Krzanich**, PDG d'Intel, un véhicule autonome va générer et consommer, pour huit heures de conduite :
  - ➢environ 40 téraoctets de données,
  - ➢soit l'équivalent de 40 disques durs d'ordinateur de 1 Téra.

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V2 : Volume des Données du Big Data

- **Exemple : Contrôle facial avec des caméras**

  - 600 Millions de Caméras au moins en chine pour 1,4 milliard d'individus
  - Mise en œuvre d'algorithmes de DEEP LEARNING
  - Près de 500 Millions de chiffres par modèle de visage

  - **Avec des usages divers et parfois controversés**
    - **Surveillances** divers
    - **Système de Crédit Sociale (SCS)**

➢V2 : Volume des Données du Big Data

▪ Impact du temps de réponse

- Une **augmentation de 0.5s** du temps de réponse des services **GOOGLE** entraîne une baisse du trafic pouvant **atteindre 2 chiffres**
- une **augmentation de 0.1s** du temps de réponses chez **AMAZON** pouvait provoquer une **baisse des ventes autour de** 1%
- **Identifier et faire arrêter un criminel dans un lieu public** grâce aux caméras doit se faire dans un temps contraint (**TEMPS REEL**)

Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢V3: Vélocité des données du Big Data

- **Quel usage pour le Big Data ?**
  - **Agrégation et Statistiques**
    - ✓ Data Warehouse et OLAP
  - **Indexation, recherche et Interrogation**
    - ✓ Recherche par mots clés
    - ✓ Pattern Matching (XML/RDF)
  - **Gestion de la connaissance et découverte**
    - ✓ Data Mining
    - ✓ Modèles statistiques

➢V4: Véracité des Données du Big Data

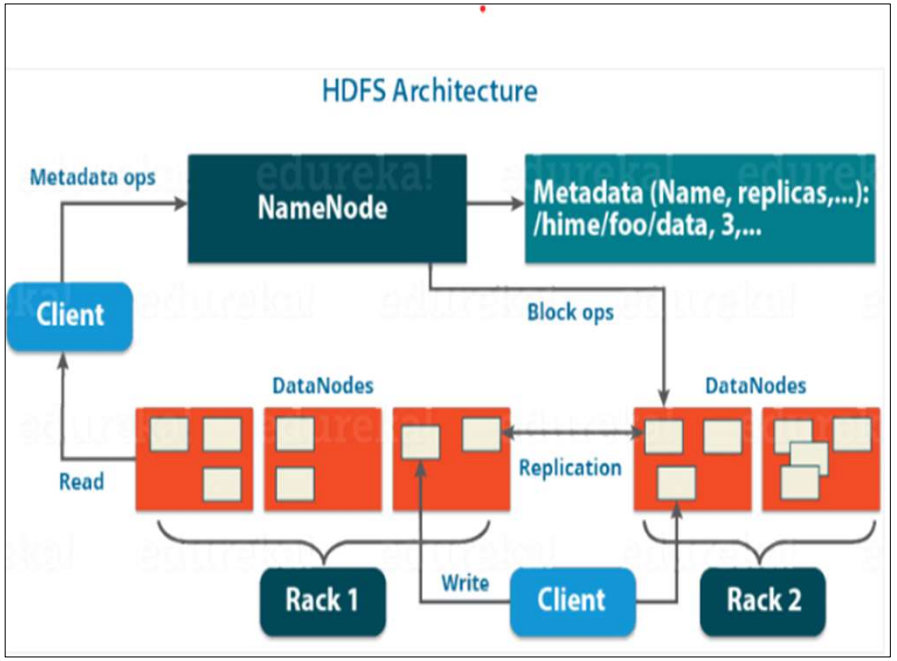
- Les données doivent être FIABLES
- Les données doivent être utilisable (traitées si nécessaire, data cleaning)
- Les données doivent exactes et prêtes à l'emploi

➢V5: Valeur des Données du Big Data

- Les données doivent apportée **une valeur ajoutée** dans les processus d'une entreprise
- Les données doivent **aider à la simplification des processus** de décision
- Les données doivent **être au cœur de la stratégie et de la compréhension** de l'activité
- Les données doivent permettre un **meilleur ciblage de la clientèle et des problèmes**
- Les données doivent aider à améliorer le **ROI** (Retour Of Investisment)

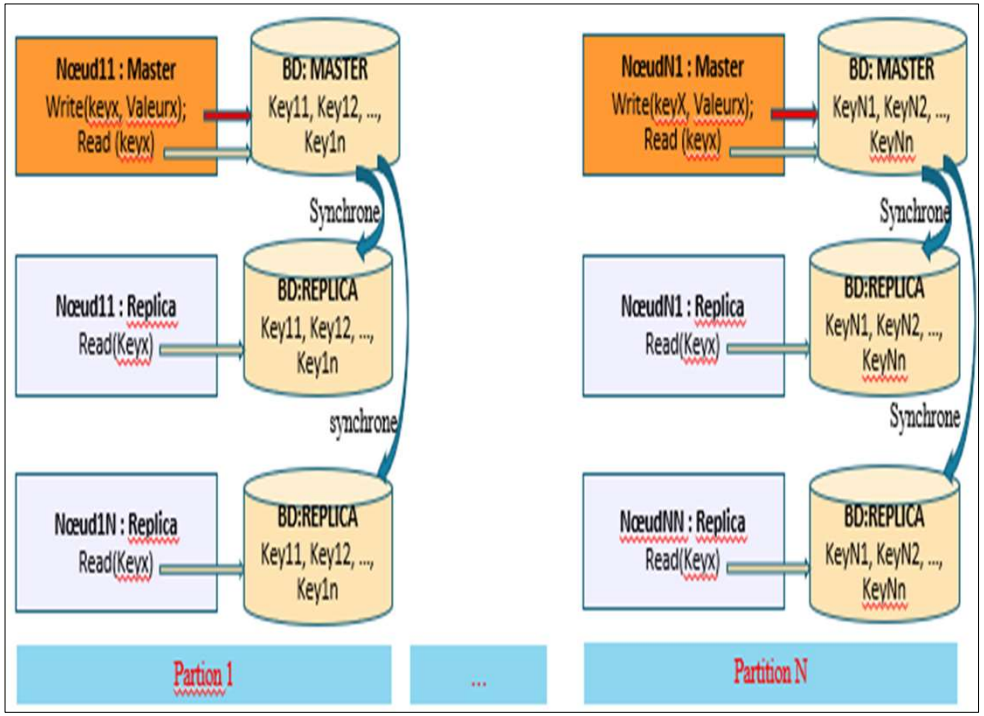
Module M4.1, section 1 : Rappel sur les Concepts du Big Data

- **Quel système pour gérer efficacement les données volumineuses ?**
  - Des Systèmes de Gestion de Fichiers Distribués
  - Exemple : Hadoop HDFS, avec des algorithmes puissants tel que MAP REDUCE
    - Note :
      - Nous revenons sur HDFS dans ce MOOC dans le module consacré aux lacs de données



Module M4.1, section 1 : Rappel sur les Concepts du Big Data

- **Quel système pour gérer efficacement les données volumineuses ?**
  - **Des Systèmes de Gestion des Bases de Données particuliers : NOSQL (Not Only SQL)**
    - Architecture distribuée (partition)
    - Architecture répliquée
    - Une implémentation de la clé plus rapide que les SGBD relationnels
    - **Notes :**
      - Nous détaillons leur concepts dans ce module
      - Nous consacrons dans ce MOOC des modules pour deux SGBDs NOSQL : MongoDB et Oracle NOSQL

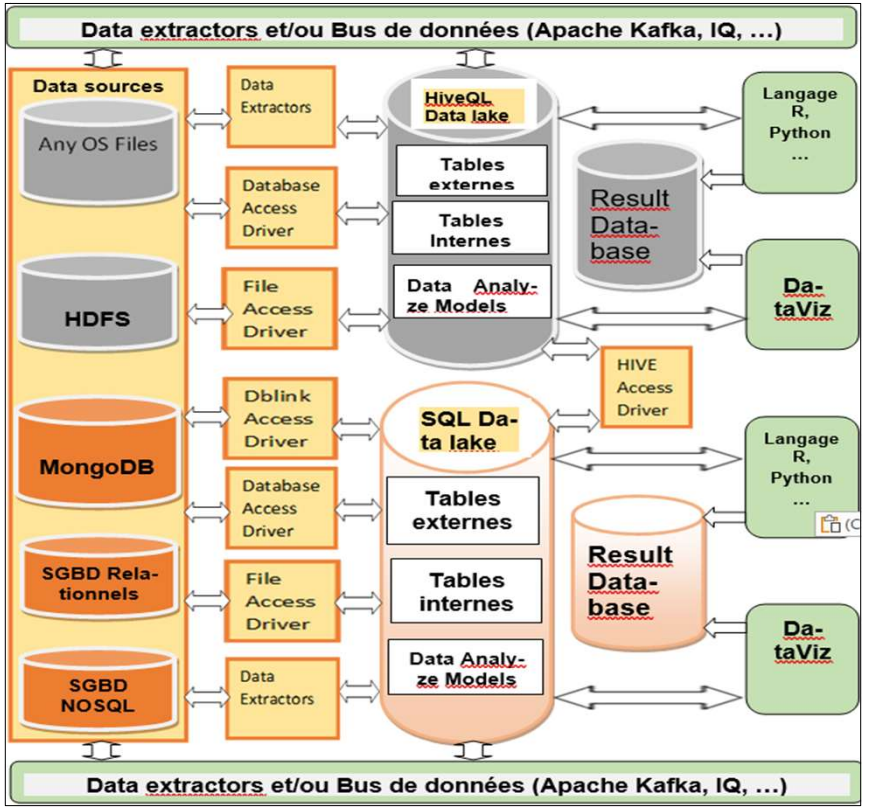


Module M4.1, section 1 : Rappel sur les Concepts du Big Data

➢ **Quel système pour gérer efficacement les données volumineuses ?**

- **Des architectures complexes pour l'analyse de données en temps réels et ou en temps différés**
  - Construction de Data Warehouse (DWH)
  - Construction de data lake

- **Note :**
  - **Nous consacrons dans ce MOOC un module sur le thème : construction de lacs de Données avec Big Data SQL par la pratique**



Module M4.1, section 1 : QUIZ

➢ **Question 1** : Les V présentés dans cette section peuvent être :

- A:La Voyance
- B: La Verdure
- C: La Valeur
- D: La Vélocité
- E: Le Volume

➢ **Question 2** : Les données volumineuses dans cette section sont caractérisées par un certain nombre de V, cochez le bon nombre de V :

- A: 1V
- B: 2V
- C: 3V
- D: 4V
- E: 5V

Module M4.1, section 1 : QUIZ

➢ **Question 3** : Avec les bases de données relationnelles il est possible de gérer les données :

- A: Structurées, semi-structurées et non structurées efficacement
- B: Structurées efficacement
- C: Semi-structurées efficacement
- D: Non structurées efficacement

➢ **Question 4** : Cochez ce qu'est une données NON STRUCTUREE

- A: insert into pilote3 values (1,"Gagarin1","09/03/1934","Klouchino1, Russie", "0071122334455", 10000.75);
- B: '{"plnum":1,"plnom":"Gagarin1","dnaiss":"09/03/1934","adr":"Klouchino1, Russie", "tel":"0071122334455", "sal":10000.75}', ligne insérée dans la table relationnelle PILOTE3
- C: '{"plnum":1,"plnom":"Gagarin1","dnaiss":"09/03/1934","adr":"Klouchino1, Russie", "tel":"0071122334455", "sal":10000.75}' un document JSON
- D: '{"plnum":1,"plnom":"Gagarin1","dnaiss":"09/03/1934","adr":"Klouchino1, Russie", "tel":"0071122334455", "sal":10000.75}' une chaine de caractères

UNIVERSITÉ CÔTE D'AZUR ESTIA INSTITUTE OF TECHNOLOGY

DATUM ACADEMY

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

FUN FRANCE UNIVERSITÉ NUMÉRIQUE

anr

INVESTIR L'AVENIR

MIAGE RÉSEAU DES MIAGE DE FRANCE Nice

MBDS

BIHAR

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

ESTIA INSTITUTE OF TECHNOLOGY

UNIVERSITÉ CÔTE D'AZUR

➢ Plan

- **Définition des SGBD NoSQL (Not only SQL)**
- **Caractéristiques de SGBD NoSQL**
- **Implémentation de la clé dans les SGBD NOSQL**
- **Implémentation de la clé dans les SGBD relationnels versus SGBD NOSQL**
- **Architecture distribuée / répliquée**
- **Les Propriétés BASE**
- **Le Théorème CAP d'Éric BREWER**
- *Les différents types de SGBD non relationnels*
- *Classification de SGBD NOSQL*
- *Popularité des SGBD selon DB Engine*
- *Parts de marché des différents types de SGBD*

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Définition des SGBD NoSQL (Not only SQL tirée de www.nosql-database.org

- SGBD non relationnel
- SGBD distribuée / répliquée
- SGBD open source
- SGBD Sans schéma
- SGBD avec une API Simple (put, get, …)
- SGBD ne supportant pas les propriétés ACID. Mais plutôt les propriétés BASE
- SGBD permettant de gérer de gros volumes de données, Grâce à la réplication et aux Nouvelles techniques d'accès aux données (clé/Valeur)

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Caractéristiques des SGBD NoSQL

- **GESTION DE GROS VOLUMES DE DONNEES**

- **SCALABILITE**
  - **Verticale** (réplication : Favorise les lectures)
  - **Horizontale** (distribution : Favorise les mises à jours)

- **RAPIDITE**
  - En lectures grâce à la réplication
  - En mise à jour grâce à la distribution

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

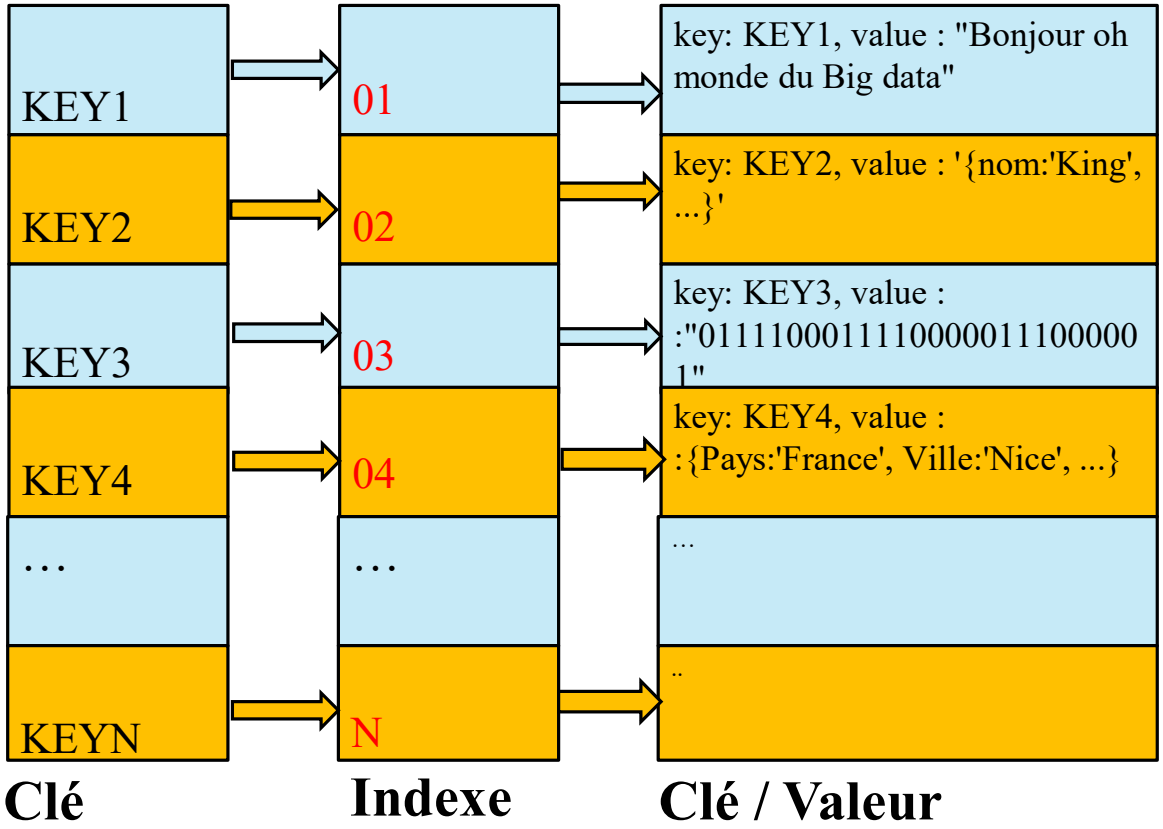
➢Caractéristiques des SGBD NoSQL

- **Mises à jour**
  - ASYNCHRONE (permet la réplication sur un **nombre de nœuds illimités**=> risque de lectures incohérentes)
  - SYNCHRONE (permet la réplication sur un **nombre de nœuds limités**=>lectures cohérentes)
- **Pas de SCHEMA DEFINI STATIQUEMENT** (schéma dynamique: sémistructurés, sans schéma: non structurés)
- **Pas de support des propriétés ACID** mais plutôt des **propriétés BASE** (Eric BREWER) :
  - **B**asically **A**vailable
  - **S**oft state
  - **E**ventually Consistent
- **Théorème CAP** (Consistency, Availability, Partition tolerance)
- Développement essentiellement **Open source**

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Implémentation de la clé dans les SGBD NOSQL

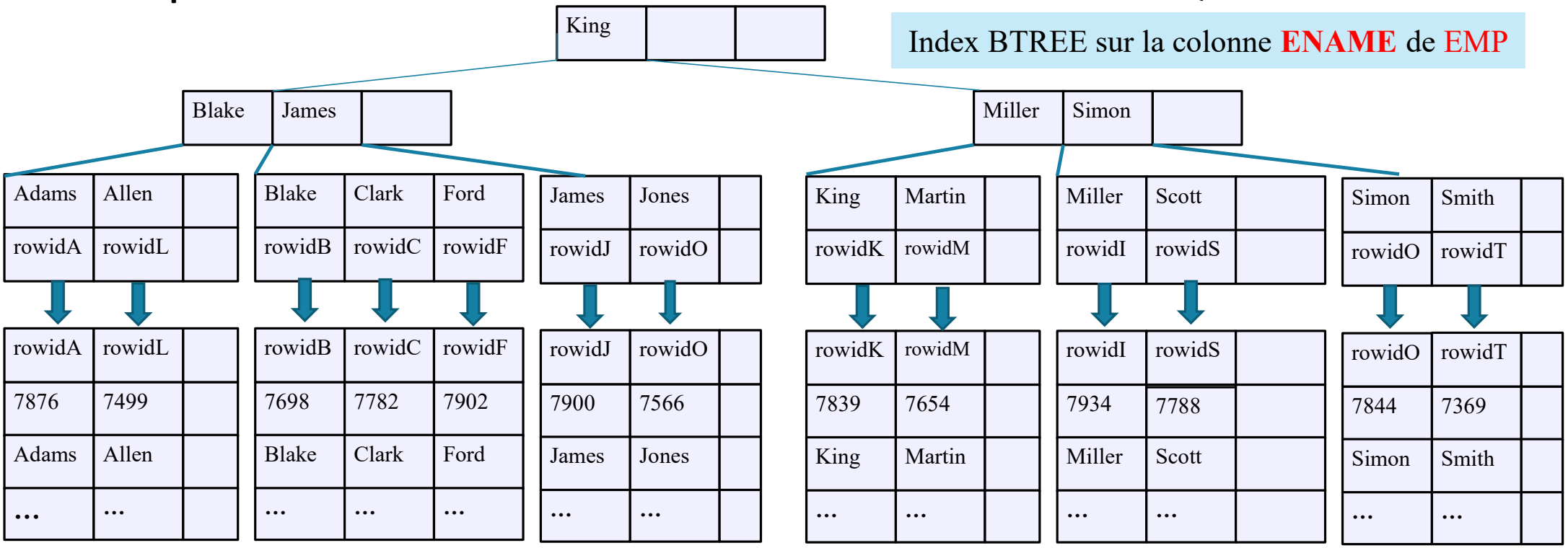
- Modèle **clé / Valeur**
- La **clé** peut être un **entier ou** un **string**
- La **valeur** peut être **structurée**, **semi-structurée** ou **non structuré**
- **Recherche par égalité** ultra rapide
- **Problème** de **collisions** possible



GET (KEY1) => :"Bonjour oh monde du Big data"
Exemple de fonction de hachage
modulo(keyNumerique, NbtotalDindiceDuTableau) => IndiceTableauDeHachage

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ **Implémentation de la clé dans les SGBD relationnels versus SGBD NOSQL**



Index BTREE sur la colonne **ENAME** de EMP

Table EMP : rowid, empno, ename, …

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Implémentation de la clé dans les SGBD relationnel versus SGBD NOSQL

- **Requête de création d'index** : CREATE **UNIQUE INDEX** IDX_EMP_ENAME ON EMP(ENAME)
- **Requête pouvant utiliser l'index** : SELECT * FROM EMP WHERE ENAME='Adams';
- **Principal accélérateur** dans les SGBD relationnels
- **rowid Oracle**=NrSegment.NrFichier.NrBloc.NrLigne
- **Efficace pour tout type de requête MAIS, LENT, en face des tables de hachage !!!! En cas de recherche via l'égalité**

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

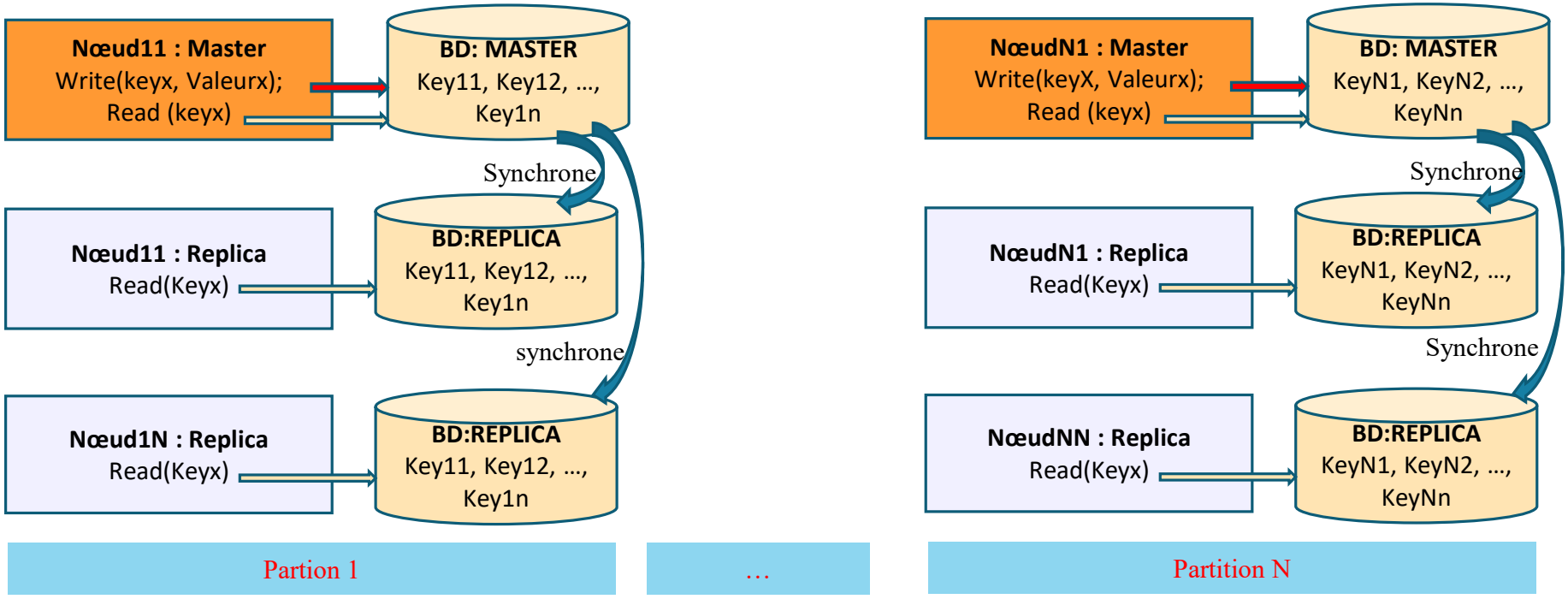
➢ Implémentation de la clé dans les SGBD relationnel versus SGBD NOSQL

▪ B-TREE VERSUS KEY/VALUE

- **La taille d'un index** peu atteindre **5 à 20%** de la taille de table
- **Plus la table est volumineuse plus l'index** est volumineux
- La recherche via l'index implique le **chargement de blocks** d'index
- La recherche **par l'égalité via l'index B-TREE est plus couteuse** que via une table de hachage (Key/Value): Charger 1 block de 4Ko depuis le disque dur vaut environ **10 millisecondes.** Avec le modèle **Key/Value** la vitesse d'accès est en **nanoseconde** car l'application de la fonction de hachage **se fait en mémoire**

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

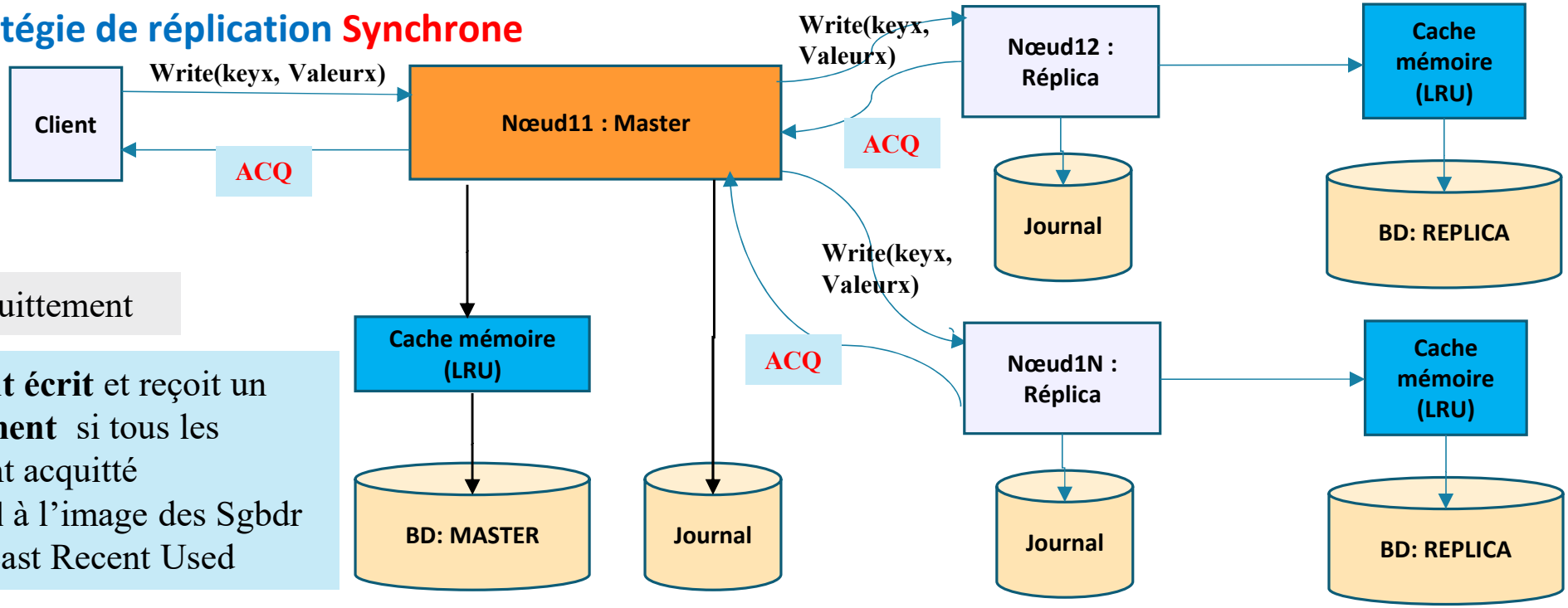
➢Architecture distribuée (partition) / répliquée (Synchrone)



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Architecture distribuée / répliquée (Synchrone)

▪ Stratégie de réplication Synchrone

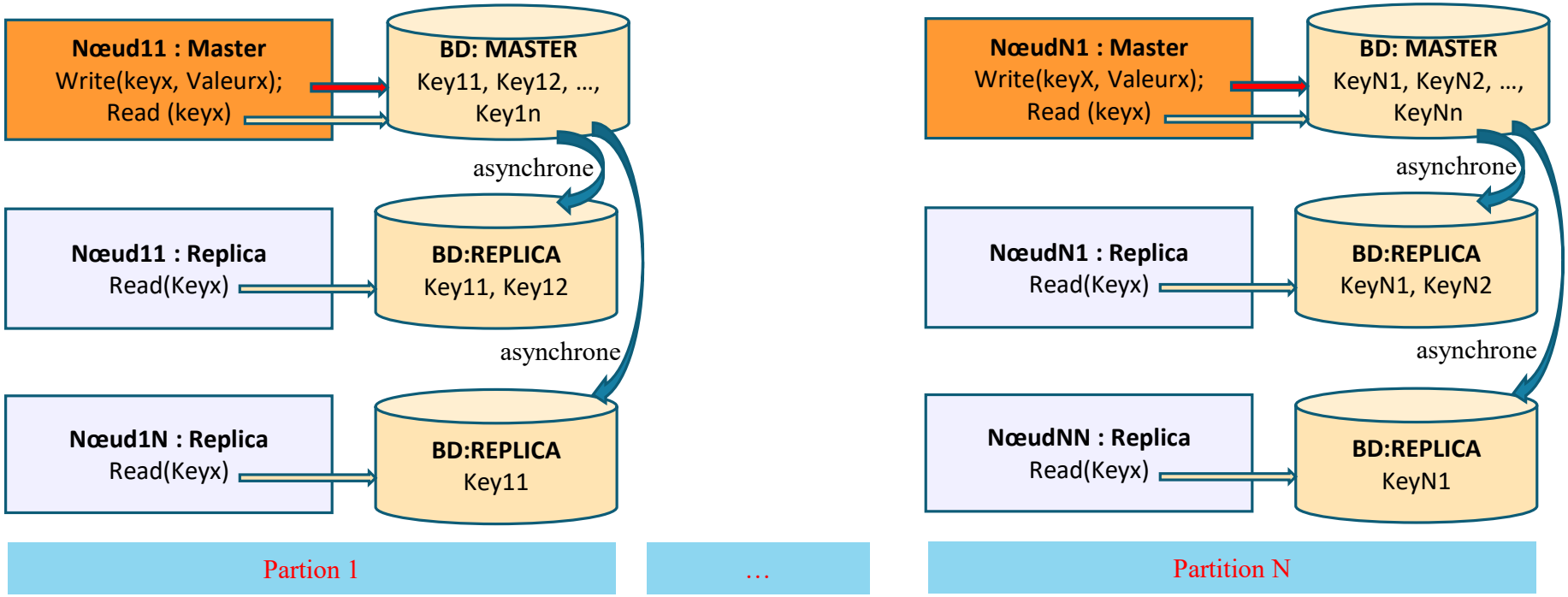


ACQ: Acquittement

1/ Le **client écrit** et reçoit un **acquittement** si tous les réplicas ont acquitté
2/ **Journal** à l'image des Sgbdr
3/ **LRU**: Last Recent Used

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

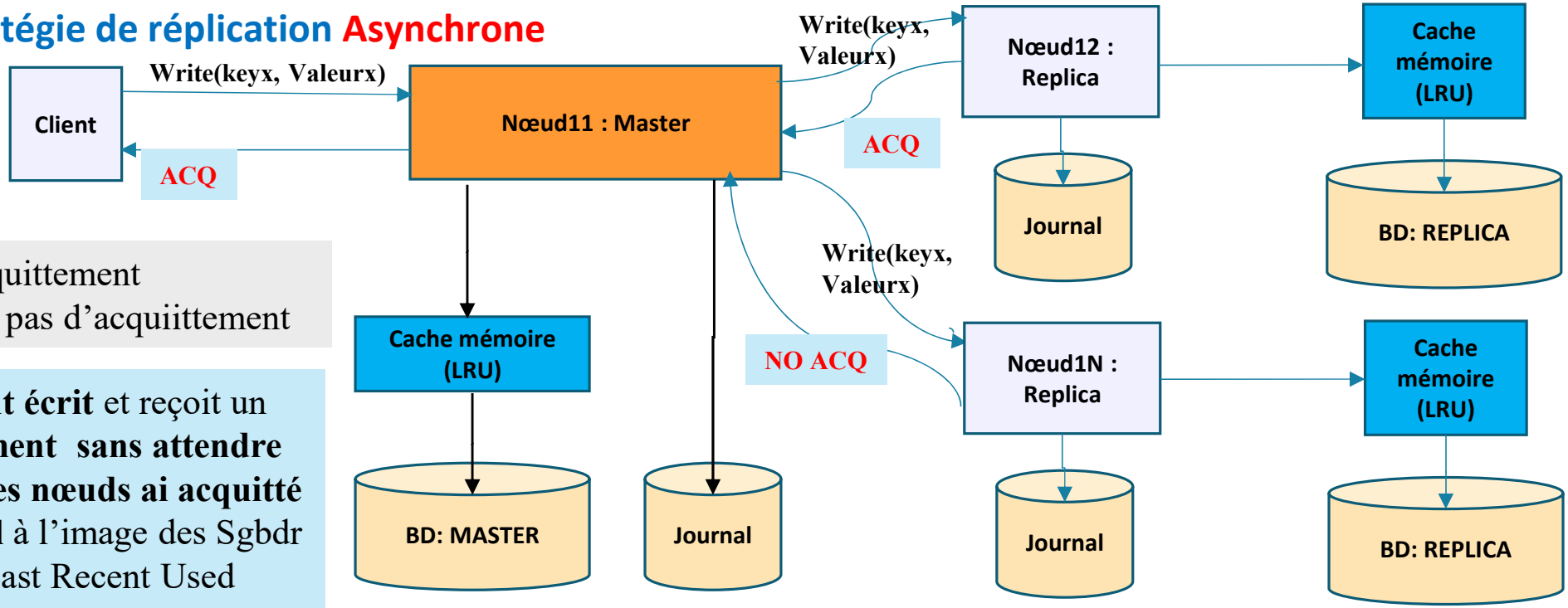
➢Architecture distribuée (partition) / répliquée (Asynchrone)



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Architecture distribuée / répliquée (Asynchrone)

▪ Stratégie de réplication Asynchrone



ACQ: Acquittement
NO ACQ: pas d'acquiittement

1/ Le **client écrit** et reçoit un **acquittement sans attendre** que **tous les nœuds ai acquitté**
2/ **Journal** à l'image des Sgbdr
3/ **LRU**: Last Recent Used

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

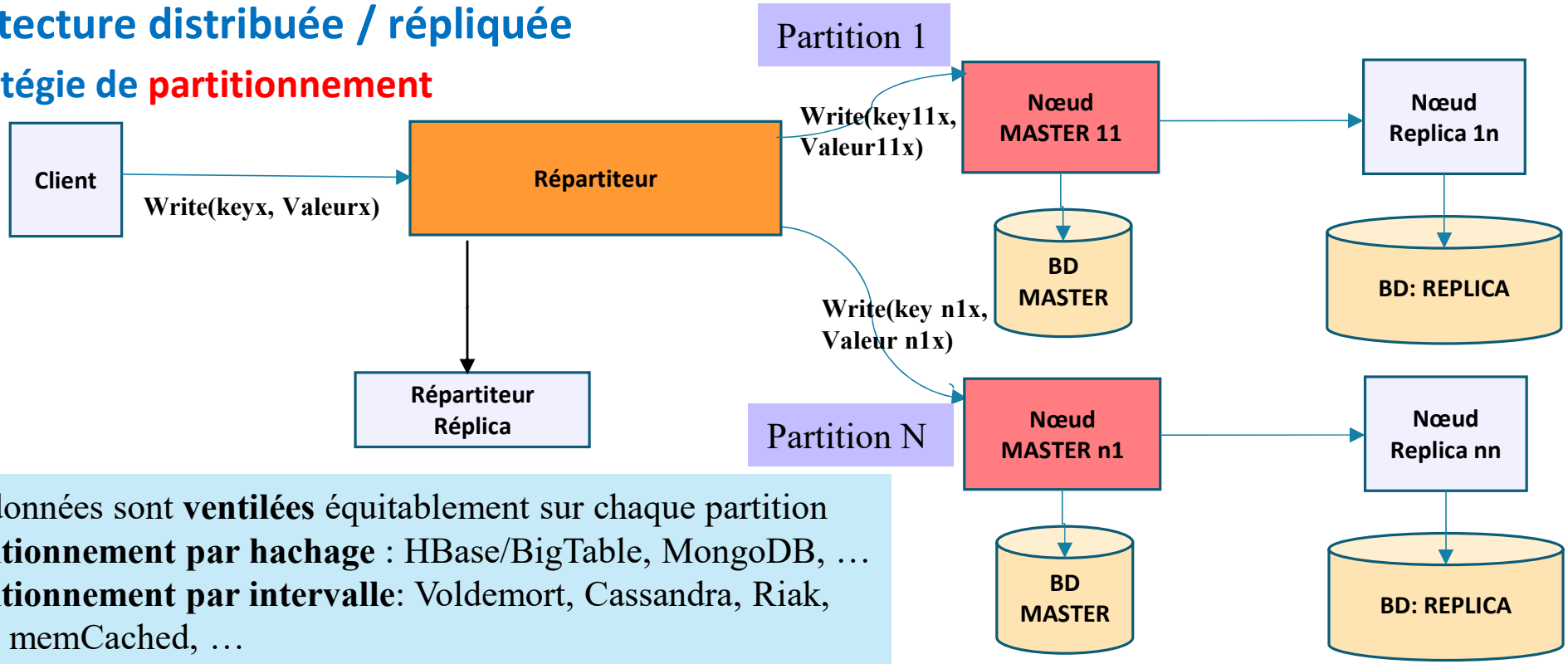
➢ Architecture distribuée / répliquée

- **Stratégie de réplication : Disparition d'un nœud maître**
  - Un nouveau nœud est élu par les esclaves restants dans une partition
- **Stratégie de réplication : Disparition d'un nœud esclave**
  - Les lectures continuent avec les nœuds restants

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Architecture distribuée / répliquée

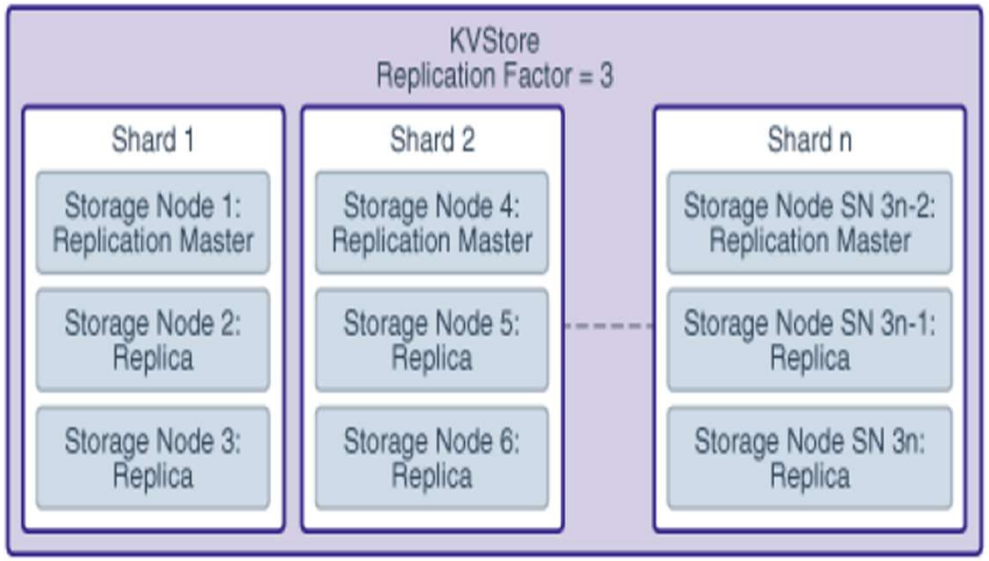
▪ Stratégie de partitionnement



1/ Les données sont **ventilées** équitablement sur chaque partition
2/ Partitionnement par hachage : HBase/BigTable, MongoDB, …
3/ **Partitionnement par intervalle**: Voldemort, Cassandra, Riak, REDIS, memCached, …

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

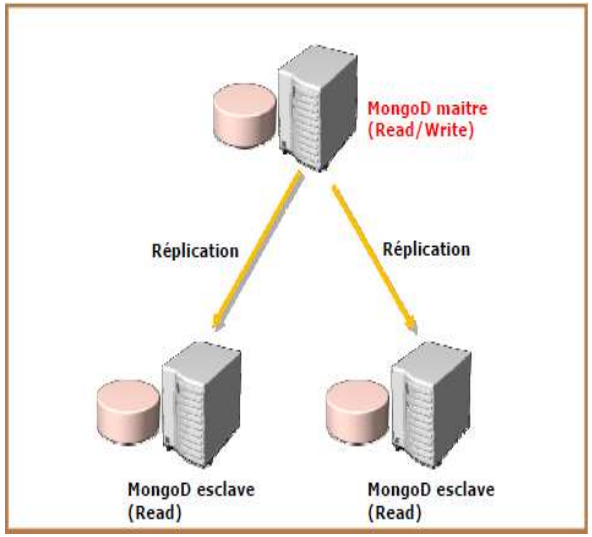
➢Architecture distribuée / répliquée (Asynchrone, Oracle NOSQL)

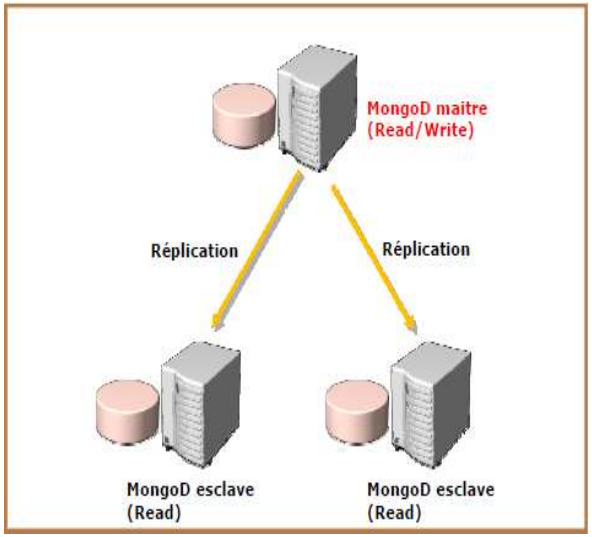


- Un SHARD correspond à un ensemble de nœud dont 1 est maître et les autres Esclaves
- L'écriture s'effectue au niveau du nœud maître
- Chaque nœud d'un SHARD est partitionné

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ **Architecture distribuée / répliquée (SYNCHRONE, MONGODB)**



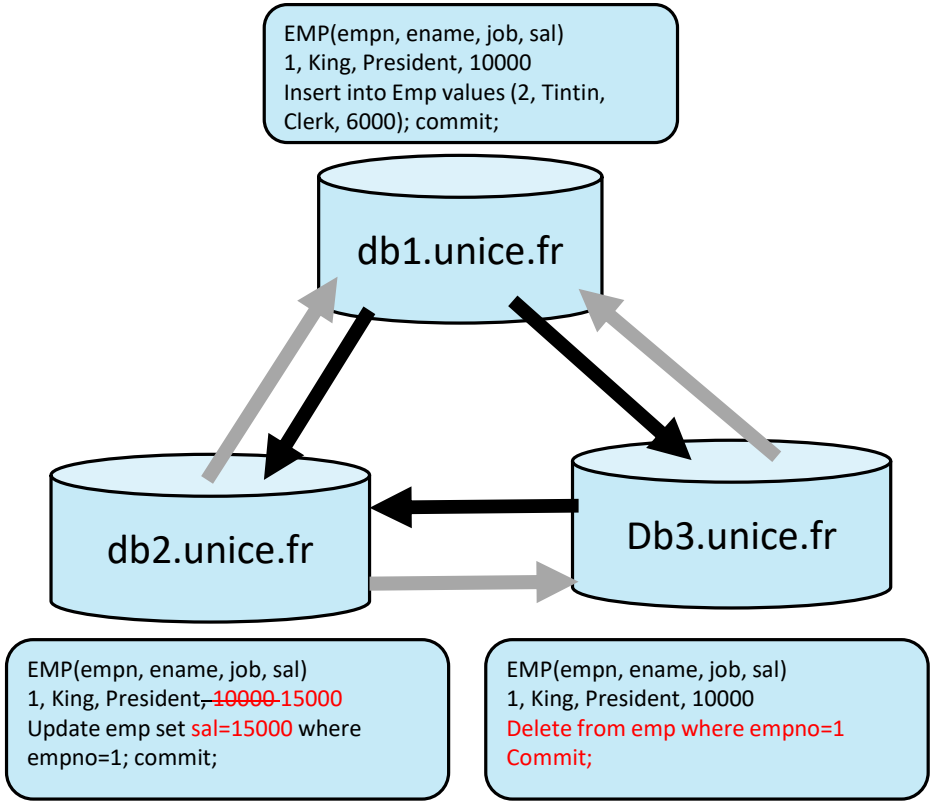


SHARD 1 … SHARD N

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ **Réplication dans un environnement SGBD Relationnels versus NOSQL**

- Chaque serveur est **maître et esclave**
- **Divers problèmes**
  - Problème des clés **primaires**
  - Problème des clés **étrangères**
  - **Problème de de la mise à jour** de la même information sur plusieurs site
  - **Difficulté** pour a**ugmenter le nombre de serveurs** pour la réplication



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Les propriétés BASE

- Les propriétés BASE permettent de **privilégier la disponibilité** à la consistence
- Acronyme **opposé à ACID** : Basically Available, Soft state, Eventually Consistent
- **Caractéristiques des propriétés BASE**
  - **Basically Available** : quelle que soit la charge de la base de données (données/requêtes), le système **garantie un taux de disponibilité** de la donnée
  - **Soft-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**
  - **Eventually consistent** : À **terme, la base atteindra un état cohérent**

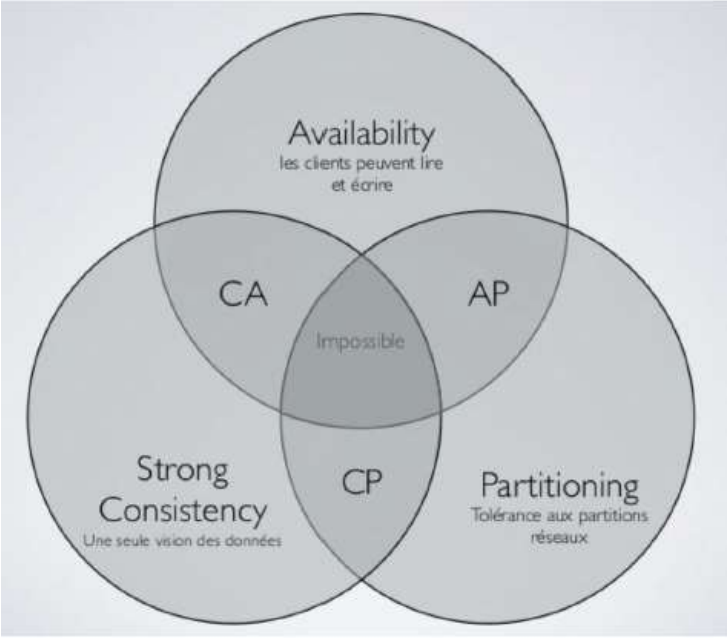
Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Le Théorème CAP d'Éric BREWER

- **Un système distribué/répliqué** (avec un grand nombre de nœuds) ne peut supporter **que deux des TROIS propriétés** du théorème :
  - **Consistency (cohérence)** : tous les nœuds voient les mêmes données au même moment
  - **Availability (disponibilité)** : garantie que toutes requêtes reçoivent une réponse même en cas de panne d'un nœud (grâce à la réplication)
  - **Partition (nœud) tolerance (distribution/réplication)** : seule une panne totale peut empêcher le réseau de fonctionner. S'il y a des sous-réseau chacun doit pouvoir fonctionner de façon autonome

Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

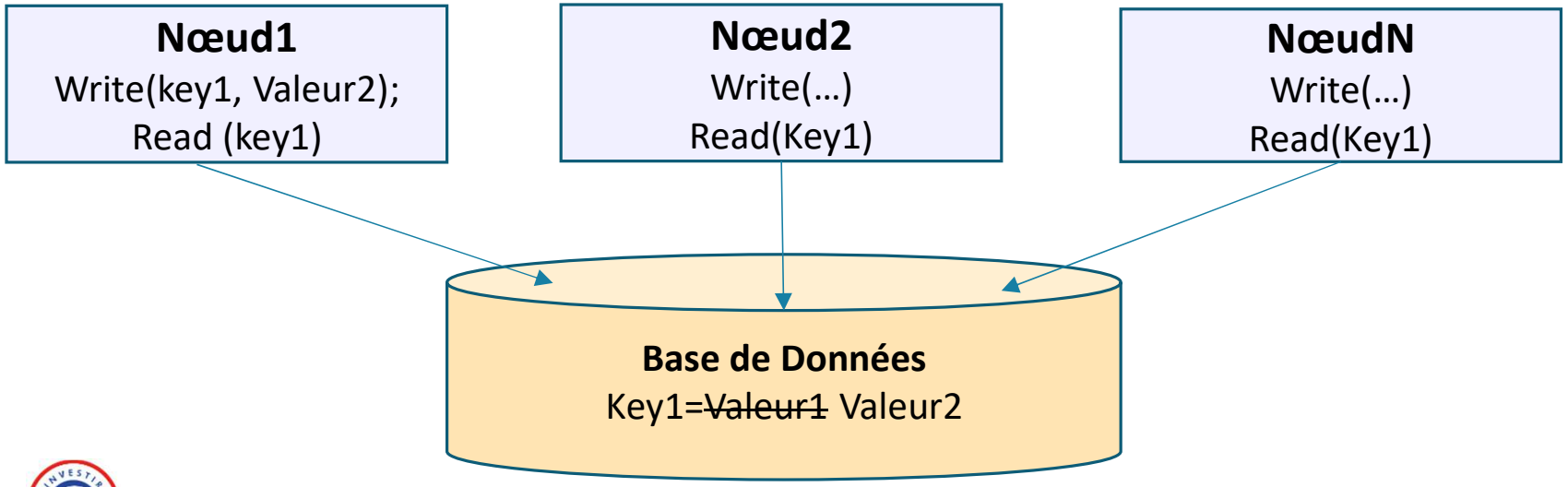
- **Le Théorème CAP d'Éric BREWER**
  - http://fr.wikipedia.org/wiki/Th%C3%A9or%C3%A8me_CAP
  - http://www.royans.net/wp/2010/02/14/brewers-cap-theorem-on-distributed-systems/



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

Le Théorème CAP d'Éric BREWER

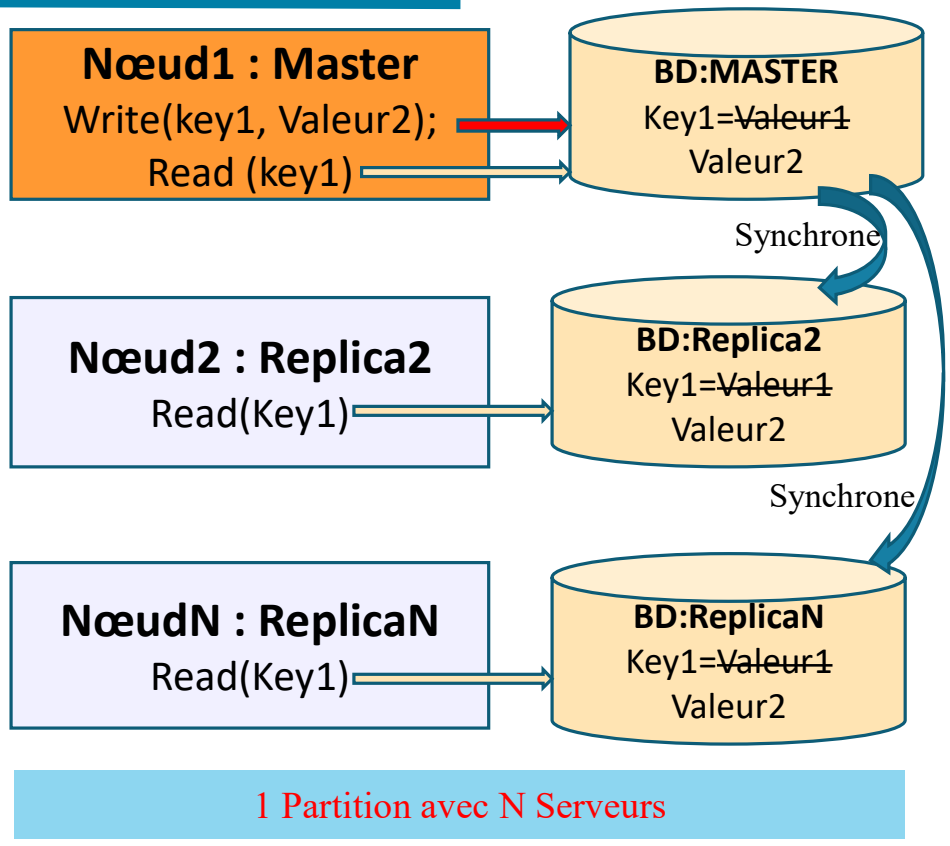
- **CA** : **Consistency and Availablity (Cohérence et Disponibilité):** Nœud1, Nœud2 et NœudN lisent la nouvelle version de la valeur **Valeur2** associée à **Key1** (Système centralisé). SGBD Relationnel (Real Application Cluster). Tous les nœuds peuvent lire et écrire **=>Problème de concurrence.**
  MAXIMUM 100 INSTANCES CHEZ ORACLE !!!



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Le Théorème CAP d'Éric BREWER

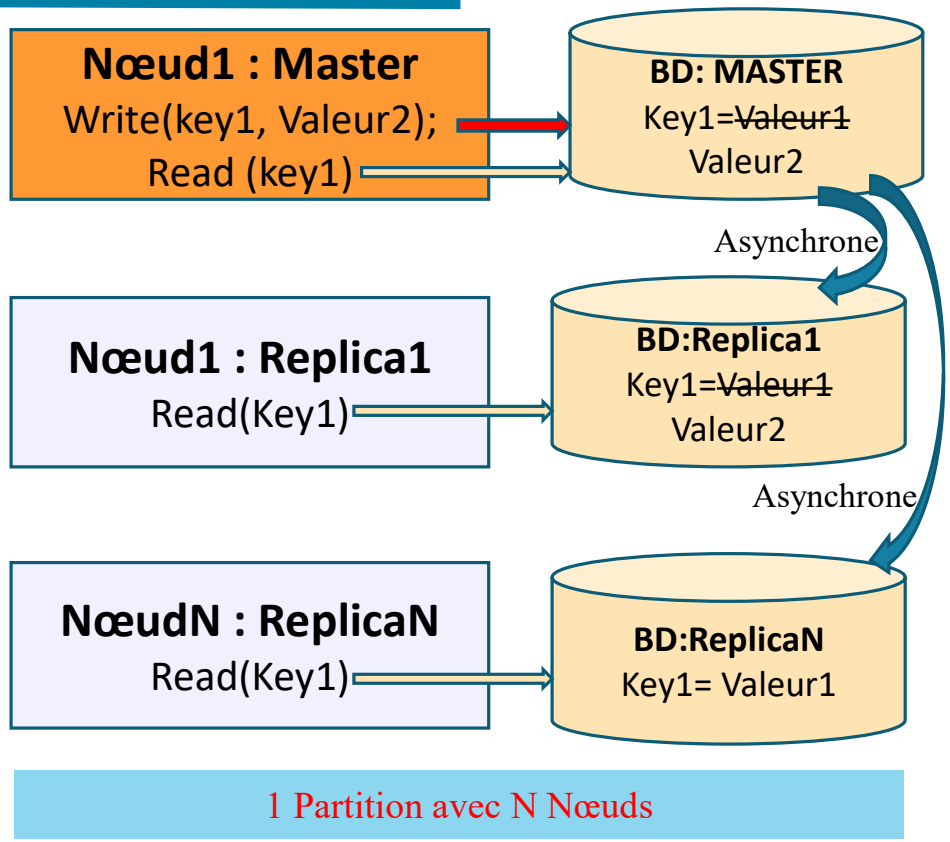
- **CP** : **Consistency and Partition (Cohérence et Distribution)**: Permet de **lire la même chose** sur **l'ensemble des nœuds** d'une partition
  - **Nœud1 MASTER** : READ/**WRITE :réplication synchrone**
  - **Nœud2, …, NœudN Replicas**: READ ONLY
  - Lecture
    - Nœud1, Nœud2 et NœudN lisent la nouvelle version de la valeur **Valeur2** associée à **Key1** (**réplication en mode synchrone**).
  - **Exemple :** MongoDB, HBASE, BIGTABLE
  - **Maximum 50 Nœuds Réplica chez MONGODB !!!**



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Le Théorème CAP d'Éric BREWER

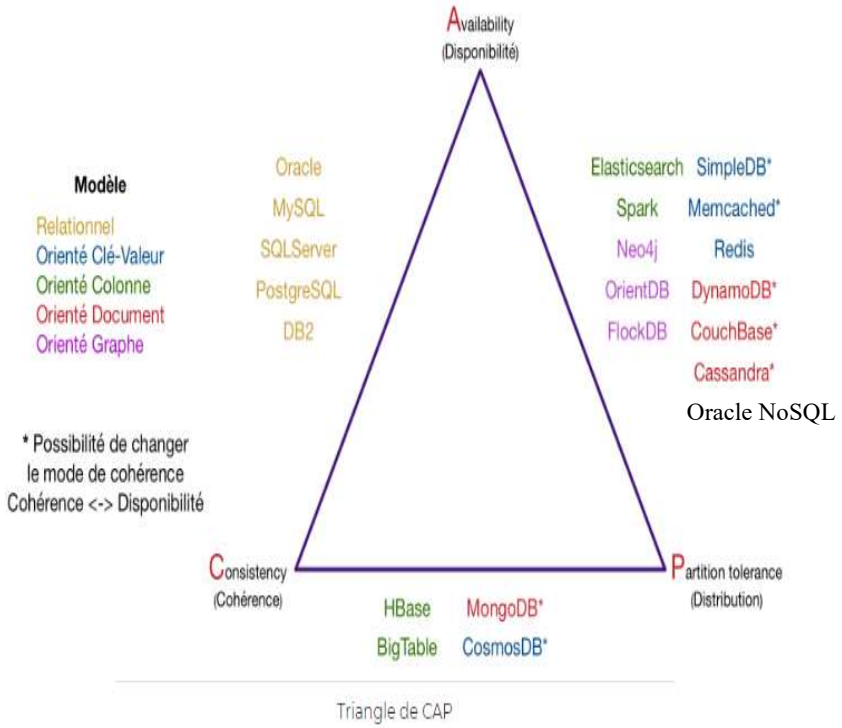
- **AP** : **Availablity and Partition**: fournit un temps de réponse rapide en lecture grâce aux **réplicas illimités** mais, avec le risque de **lire des données anciennes**
  - **Nœud1 MASTER** : READ/**WRITE** avec **réplication Asynchrone**
  - **Nœud2, …, NœudN Replicas**: READ ONLY
  - **Lectures dans le schéma**
    - Nœud1, Nœud2 lisent la nouvelle valeur **Valeur2** associée à **Key1**
    - NœudN lit l'ancienne valeur **Valeur1** associée à **Key1** : **Réplication en mode asynchrone**
  - **Exemple :** CASSANDRA, ORACLE NOSQL, …



Module M4.1, section 2, partie 1 : Rappel sur les Concepts des SGBD NOSQL

➢ Le Théorème CAP d'Eric BREWER

https://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462471-maitrisez-le-theoreme-de-cap



Triangle de CAP

Module M4.1, section 2, partie 1 : QUIZ

- **Question 1** : Pour accéder aux données volumineuses rapidement il vaut mieux d'utiliser (par exemple le profil d'un client)
  - A: la primary key au sens relationnel
  - B: le modèle key value
  - C: les indexes bitmap
  - D: les indexes B-arbres

- **Question 2** : On dit souvent que la primary key au sens relationnel n'est pas efficace dans un contexte big data car
  - A: même pour charger 1 ligne il faut plusieurs blocks d'indexe
  - B: la taille d'un index vaut environ 10% de la table
  - C: Tout va bien c'est la PK qui reste le chemin le plus rapide

Module M4.1, section 2, partie 1 : QUIZ

➢ **Question 3** : Cochez ce qui caractérise le mieux la réplication asynchrone

- A: Elle permet de renforcer la disponibilité dans une partition d'une base de données NOSQL
- B: Elle permet d'augment la puissance de mise à jour dans une partition d'une base de données NOSQL
- C: B: Elle permet d'augment la puissance de lecture dans une partition d'une base de données NOSQL
- D: Une partition à un nœud master et peut avoir un nombre de nœuds réplicas illimité

➢ **Question 4** : Considérons une base de données NOSQL avec 10 partitions. Vous créé une table dans cette base

- A: Vous devez lors de l'insertion des lignes cette dans table indiquer dans quelle partition la ligne va être insérée
- B: Une partition = 1 machine virtuelle ou physique
- C: Le mot le SHARD et le mot partition signifient la même chose
- D: Le SGBD grâce à une clé de réparation, répartie automatiquement les données de la table sur chacune des partitions

Module M4.1, section 2, partie 1 : QUIZ

➢ **Question 5** : Cochez ce qui caractérise le mieux la distribution dans les SGBD NOSQL

- A: Le partitionnement des données dans les différents nœuds maîtres peut être faits par intervalle
- B: Le partitionnement des données dans les différents nœuds maîtres peut être faits par hachage
- C: Le partitionnement des données est géré par un nœud répartiteur
- D: L'ajout d'une partition augmente la disponibilité des données

➢ **Question 6** : Le théorème CAP (Cohérence, Disponibilité, Partition) affirme qu'il n'existe aucun système de gestion de bases de données (répliqué/distribué) qui garantisse les trois propriétés en même temps mais uniquement deux d'entre elles. Cochez les propriétés qui caractérisent les SGBD NOSQL

- A: CP
- B: AP
- C: AC

UNIVERSITÉ CÔTE D'AZUR ESTIA INSTITUTE OF TECHNOLOGY

DATUM ACADEMY

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

FUN FRANCE UNIVERSITÉ NUMÉRIQUE

anr

INVESTIR L'AVENIR

MIAGE RÉSEAU DES MIAGE DE FRANCE Nice

MBDS

BIHAR

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

ESTIA INSTITUTE OF TECHNOLOGY

UNIVERSITÉ CÔTE D'AZUR

➢ Plan

- *Définition des SGBD NoSQL (Not only SQL)*
- *Caractéristiques de SGBD NoSQL*
- *Implémentation de la clé dans les SGBD NOSQL*
- *Propriétés BASE*
- *Le Théorème CAP d'Éric BREWER*
- *Architecture distribuée / répliquée de la plupart des BD Nosql*
- **Les différents types de SGBD non relationnels**
- **Classification de SGBD NOSQL**
- **Popularité des SGBD selon DB Engine**
- **Parts de marché des différents types de SGBD**

anr

INVESTIR L'AVENIR

DATUM ACADEMY

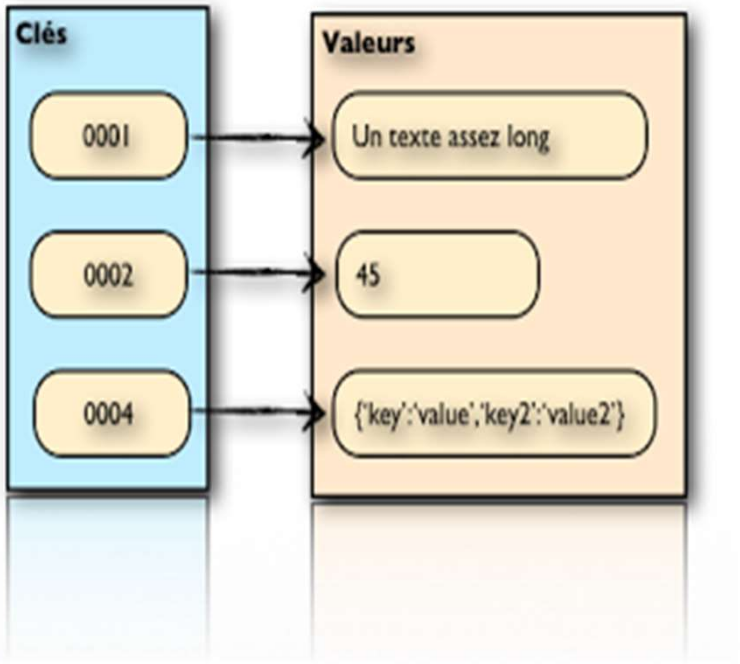
Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢ Les différents types de SGBD NoSQL

- **SGBD Orientés Clé/Valeur** : Table de hachage (valeur non structurée)
- **SGBD Orientés Colonnes** (clé/valeur): chaque block ne contient que les données d'une seule colonne
- **SGBD Orientés Documents** (clé/valeur): les valeurs sont des données complexes de type XML, JSON, …
- **SGBD orientés graphes** (clé/valeur): Permet la modélisation, le stockage et la manipulation de données complexes en graphe

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢ **SGBD NoSQL clé/valeur**



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL clé/valeur

- Les données sont représentés par un **couple Clé/Valeur**
- La base de données est **un grand tableau associatif**. A une clé correspond une entrée du tableau
- La **valeur EST NON STRUCTUREE**
- **Pas de schéma à priori**. L'interprétation de l'objet est laissé au programme client. Attention des versions avec des tables JSON existent (primary = KEY)

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL clé/valeur

- La construction **de la clé n'est plus triviale** : major and Minor Key
- Opération **: CRUD (Create Read Update Delete)**
- Des **API sont disponibles dans plusieurs langages**
- **SGDB connus**: **Amazon Dynamo** (**Riak** en est l'implémentation Open Source**), Redis** (projet sponsorisé par VMWare), **Voldemort** (développé par LinkedIn), **Oracle Nosql**
- **Usage** : Les profiles client, **les preferences**, les logs, **le cache**, etc.

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté clé/Value

- https://db-engines.com/en/ranking/key-value+store

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model
1.	1.	1.	Redis	Key-value, Multi-model
2.	2.	2.	Amazon DynamoDB	Multi-model
3.	3.	3.	Memcached	Key-value
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-model
5.	5.	5.	Hazelcast	Key-value
6.	↑ 7.	↑ 9.	Aerospike	Key-value
7.	↓ 6.	↓ 6.	Ehcache	Key-value
8.	↑ 9.	8.	OrientDB	Multi-model
9.	↓ 8.	↓ 7.	Riak KV	Key-value
10.	10.	↑ 11.	Ignite	Multi-model
11.	11.	↓ 10.	ArangoDB	Multi-model
12.	12.	↑ 14.	InterSystems Caché	Multi-model
13.	13.	↓ 12.	Oracle NoSQL	Key-value, Multi-model
14.	↑ 15.	↑ 16.	LevelDB	Key-value
15.	↓ 14.	↓ 13.	Oracle Berkeley DB	Multi-model
16.	↑ 17.	↑ 19.	RocksDB	Key-value
17.	↑ 18.	↓ 15.	Oracle Coherence	Key-value
18.	↑ 19.	18.	Amazon SimpleDB	Key-value
19.	↓ 16.	↓ 17.	Infinispan	Key-value
20.	20.	20.	GridGain	Multi-model

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL clé/valeur : Exemple Oracle NOSQL

- **Ajouts d'enregistrements sur la ligne de commande**

  kv-> put kv -key /Smith/Bob/-/phonenumber -value "408 555 5555"
  kv-> put kv -key /Smith/Bob/-/adresse -value " place de la république Abidjan"
  kv-> put kv -key /Smith/Bob/-/etatcil -value "bob smith ne a tamatave à Assinie"

- **Lecture des enregistrements avec Oracle NOSQL sur la ligne de commande**

  kv-> get kv -key /Smith/Bob/-/adresse
  " place de la république Abidjan"

- **Suppression d'un enregistrement sur la ligne de commande**

  kv-> delete kv -key /Smith/Bob/-/phonenumber

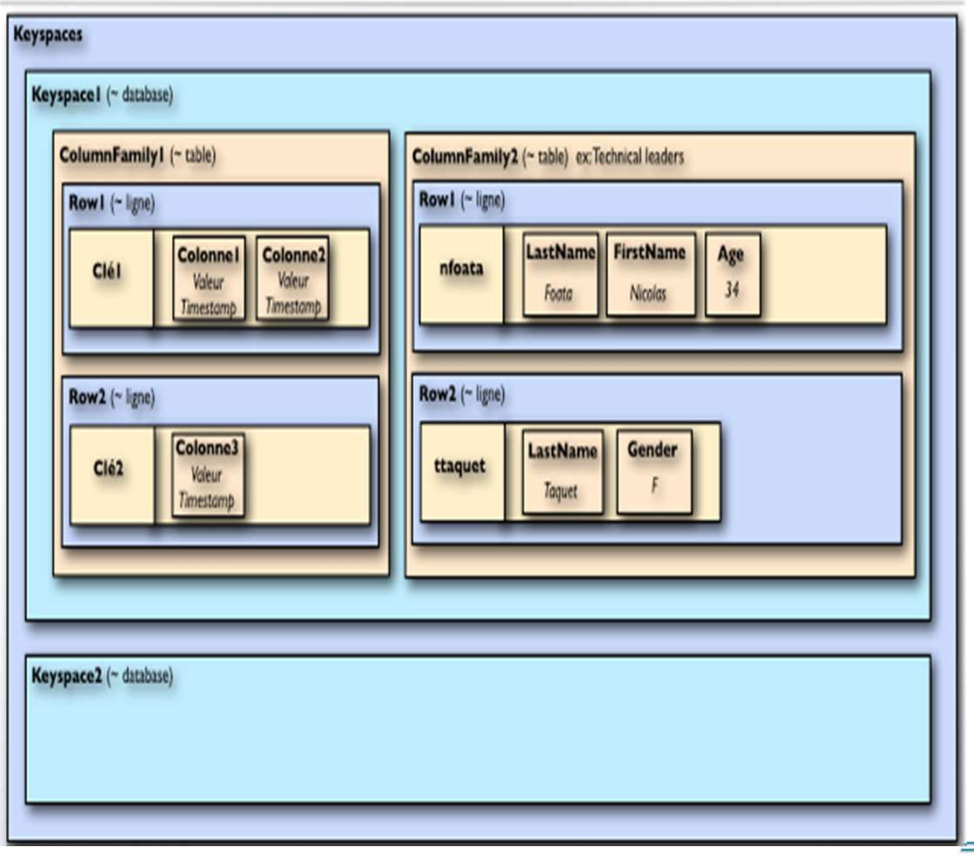
- **Maj d'un enregistrement sur la ligne de commande**

  kv-> put kv -key /Smith/Bob/-/phonenumber -value "225 1111111"

- **NOTE :** Des APIs en Java et d'autres langages existent

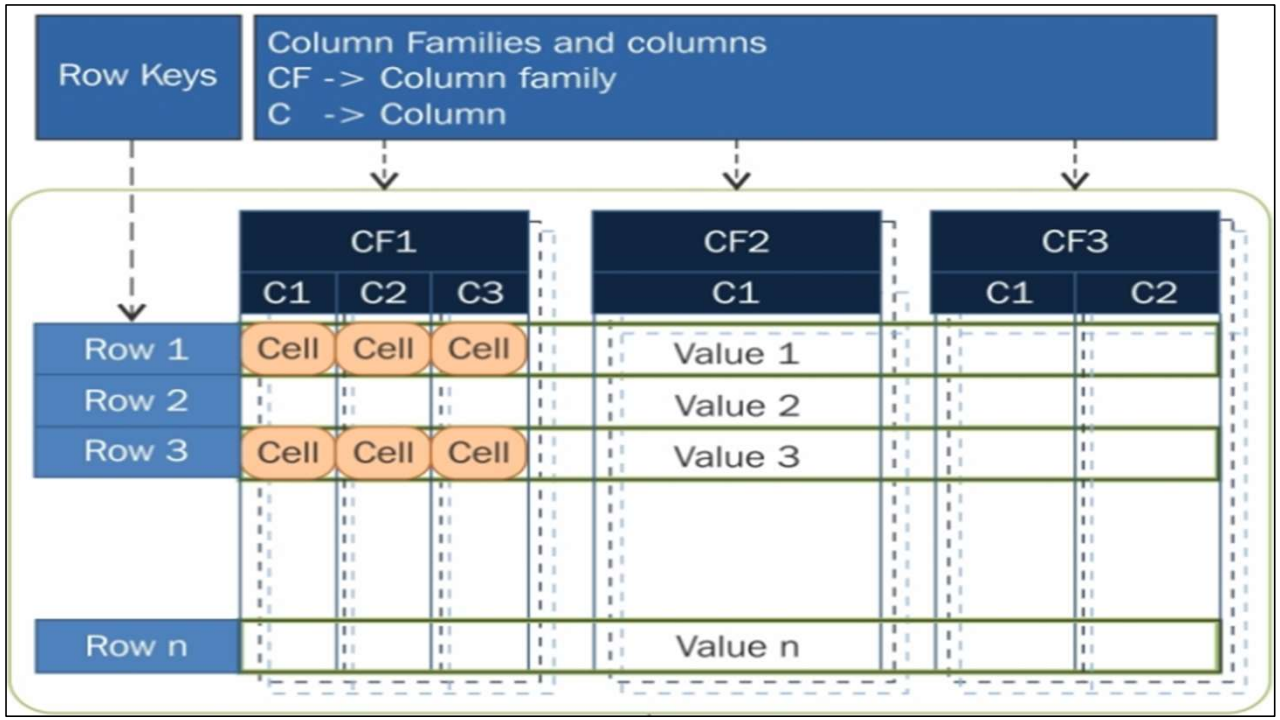
Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

- SGBD NoSQL orienté Colonne



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté Colonne



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté Colonne

- Stockage des données **en colonne et non en ligne**
- Ajout dynamique de nouvelles colonnes
- Les colonnes sont organisées en Familles
- Insertion de lignes couteuses
- Compressions de données identiques pour une même colonne possible
- Chaque enregistrement à ses colonnes

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté Colonne VERSUS stockage Colonnes

Table EMPLOYE

RowId	EmpId	Nom	Prénom	Salaire
001	10	Durant	Jacques	40000
002	12	Dupont	Marie	50000
003	11	Martin	Jeanne	44000
004	22	Dupont	Robert	94000

Stockage **ligne**

```
001:10,Durant,Jacques,40000;
002:12,Dupont,Marie,50000;
003:11,Martin,Jeanne,44000;
004:22,Dupont,Robert,94000;
```

Stockage **colonne**

```
10:001,12:002,11:003,22:004;
Durant:001,Dupont:002,Martin:003,Dupont:004;
Jacques:001,Marie:002,Jeanne:003,Robert:004;
40000:001,50000:002,44000:003,94000:004;
```

Stockage **colonne** avec compression

```
10:001,12:002,11:003,22:004;
Durant:001,Dupont:002,004; Martin:003;
Jacques:001,Marie:002,Jeanne:003,Robert:004;
40000:001,50000:002,44000:003,94000:004;
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté Colonne

- Bon moyen pour éviter les valeurs NULL

- **Implémentations les plus connues** :
  - **HBase** (Open Source de BigTable de Google utilisé pour l'indexation des pages web, Google Earth, Google analytics, ...),
  - **Cassandra** (fondation Apache qui respecte l'architecture distribuée de Dynamo d'Amazon, projet né de chez Facebook),
  - **SimpleDB** de Amazon

- **Quelques utilisateurs de ce type de SGBD**
  - **Orange Portail** : Pour la gestion des profils clients et des données de syndication
  - **Netflix**
  - **Adobe**
  - **Ebay**

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté Colonne

- https://db-engines.com/en/ranking/wide+column+store

Rank Dec 2022	Rank Nov 2022	Rank Dec 2021	DBMS	Database Model	Score Dec 2022	Score Nov 2022	Score Dec 2021
1.	1.	1.	Cassandra	Wide column	114.65	-3.47	-4.55
2.	2.	2.	HBase	Wide column	40.04	-0.38	-5.50
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	37.95	-1.80	-1.77
4.	4.	4.	Datastax Enterprise	Wide column, Multi-model	8.63	-0.05	-0.65
5.	5.	5.	Microsoft Azure Table Storage	Wide column	5.62	-0.40	+0.34
6.	↑ 7.	↑ 7.	Accumulo	Wide column	5.41	+0.43	+1.51
7.	↑ 8.	↓ 6.	ScyllaDB	Wide column, Multi-model	5.22	+0.27	+1.28
8.	↓ 6.	8.	Google Cloud Bigtable	Multi-model	5.01	-0.23	+1.38
9.	9.	↑ 10.	YugabyteDB	Relational, Multi-model	4.47	+0.32	+2.64
10.	10.	↓ 9.	Trino	Relational, Multi-model	4.27	+0.28	+1.72
11.	11.	11.	HPE Ezmeral Data Fabric	Multi-model	1.41	+0.15	+0.52
12.	12.		OceanBase	Relational, Multi-model	1.20	+0.08	
13.	13.	13.	Amazon Keyspaces	Wide column	0.78	+0.01	+0.24
14.	14.	↓ 12.	Elassandra	Wide column, Multi-model	0.51	-0.06	-0.12
15.	15.	↓ 14.	Alibaba Cloud Table Store	Wide column	0.31	-0.02	-0.10
16.	16.	↓ 15.	SWC-DB	Wide column, Multi-model	0.05	-0.02	-0.02

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté colonne : Exemple HBASE

hbase(main):002:0> create 'uhProfile', 'info'
0 row(s) in 1.1190 seconds

-- **Vérification des tables créées**
-- Une table a été créée avec un groupe de colonnes
-- Les colonnes seront précisées lors de l'insertion des lignes

-- Liste des tables
hbase(main):004:0> list TABLE uhProfile
1 row(s) in 0.4400 seconds
["uhProfile"]

-- **Vérification de la structure d'une table**
hbase(main):028:0> describe 'uhProfile'

Table uhProfile is ENABLED
uhProfile
COLUMN FAMILIES DESCRIPTION
{NAME => 'info', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KE
EP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE
', TTL => 'FOREVER', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65
536', REPLICATION_SCOPE => '0'}

1 row(s) in 0.0920 second

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté colonne : Exemple HBASE

--- Insertion de lignes avec la commande put dans la table uhProfile

--- syntaxe

put ’<table name>’,’rowId’,’<colfamily:colname>’,’<value>’

-- insertion de la ligne 1

put 'uhProfile', 1, 'info:id', '1'
put 'uhProfile', 1, 'info:name', 'joe'
put 'uhProfile', 1, 'info:address', '12 rue du Congres à Valbonne'
put 'uhProfile', 1, 'info:dnaiss', '11/11/1960'
put 'uhProfile', 1, 'info:zip_code', '06560'

-- insertion de la ligne 2

put 'uhProfile', 2, 'info:id', '2'
put 'uhProfile', 2, 'info:name', 'jack'
put 'uhProfile', 2, 'info:address', '11 rue du Begonias à Vallauris'
put 'uhProfile', 2, 'info:dnaiss', '13/11/1955'
put 'uhProfile', 2, 'info:zip_code', '06220'

-- insertion de la ligne 3

put 'uhProfile', 3, 'info:id', '3'
put 'uhProfile', 3, 'info:name', 'john'
put 'uhProfile', 3, 'info:address', '13 Avenue de Marseille à Cassis'
put 'uhProfile', 3, 'info:dnaiss', '11/11/1962'
put 'uhProfile', 3, 'info:zip_code', '13260'

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté colonne : Exemple HBASE

-- Lecture des lignes insérées dans la table uhProfile

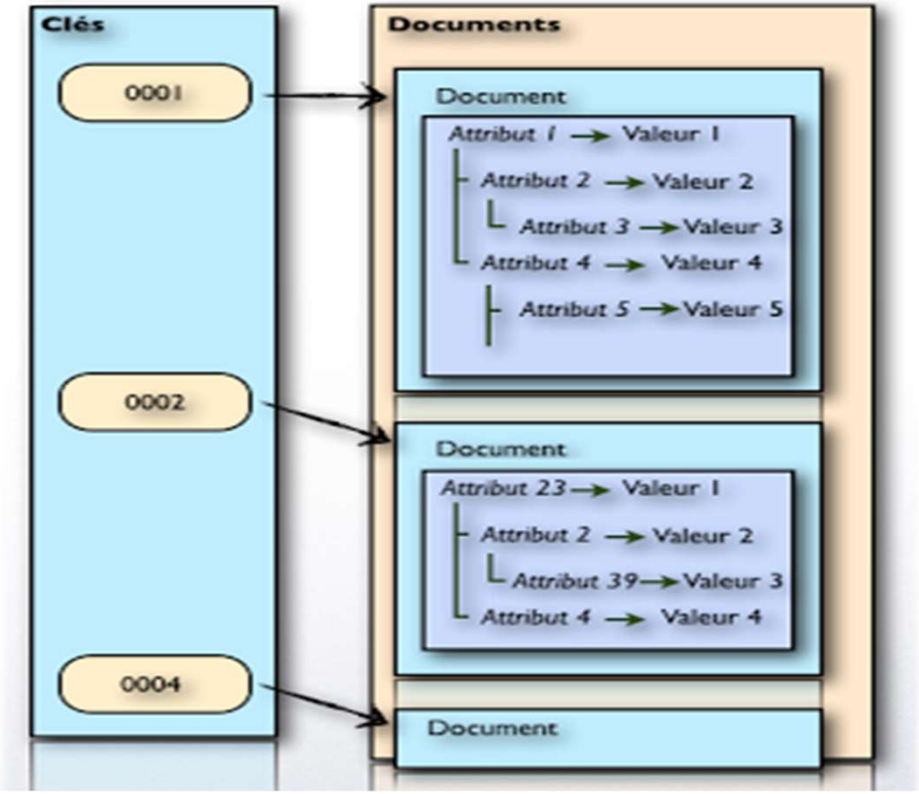
hbase(main):027:0> scan 'uhProfile'

ROW	COLUMN+CELL
1	column=info:address, timestamp=1451131883641, value=12 rue du Congres \xC3\xA0 Valbonne
1	column=info:dnaiss, timestamp=1451131883734, value=11/11/1960
1	column=info:id, timestamp=1451131220578, value=1
1	column=info:name, timestamp=1451131883487, value=joe
1	column=info:zip_code, timestamp=1451131883808, value=06560
2	column=info:address, timestamp=1451131906838, value=11 rue du Begonias \xC3\xA0 Vallauris
2	column=info:dnaiss, timestamp=1451131906942, value=13/11/1955
2	column=info:id, timestamp=1451131906500, value=2
2	column=info:name, timestamp=1451131906741, value=jack
2	column=info:zip_code, timestamp=1451131907081, value=06220
3	column=info:address, timestamp=1451131921427, value=13 Avenue de Marseille \xC3\xA0 Cassis
3	column=info:dnaiss, timestamp=1451131921489, value=11/11/1962
3	column=info:id, timestamp=1451131921237, value=3
3	column=info:name, timestamp=1451131921317, value=john
3	column=info:zip_code, timestamp=1451131921567, value=13260

3 row(s) in 0.1730 seconds

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

- **SGBD NoSQL orienté Documents**



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté documents

- **Les documents sont au format JSON**. Pas de schema mais une structure arborescente
- **Il s'agit aussi d'un modèle KEY/VALUE** dont la valeur est un document
- **Permettent d'effectuer des requêtes sur le contenu des documents/objets** : pas possible avec les BD clés/valeurs simples
- **Elles sont principalement utilisées dans le développement de CMS** (Content management System)
- **Limites** : **inadaptée** pour les données **graphes**
- **SGBD les plus connues** : **CouchDB** (fondation Apache), **MongoDB**, **Oracle Nosql**, …

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté document

- https://db-engines.com/en/ranking/document+store

Rank Dec 2022	Rank Nov 2022	Rank Dec 2021	DBMS	Database Model
1.	1.	1.	MongoDB	Document, Multi-model
2.	2.	2.	Amazon DynamoDB	Multi-model
3.	3.		Databricks	Multi-model
4.	4.	↓ 3.	Microsoft Azure Cosmos DB	Multi-model
5.	5.	↓ 4.	Couchbase	Document, Multi-model
6.	6.	↓ 5.	Firebase Realtime Database	Document
7.	7.	↓ 6.	CouchDB	Document, Multi-model
8.	8.	↑ 9.	Google Cloud Firestore	Document
9.	9.	↓ 8.	MarkLogic	Multi-model
10.	10.	↓ 7.	Realm	Document
11.	↑ 12.	↓ 10.	Aerospike	Multi-model
12.	↓ 11.	↑ 13.	Google Cloud Datastore	Document
13.	13.	↓ 11.	Virtuoso	Multi-model
14.	14.	↓ 12.	ArangoDB	Multi-model
15.	15.	↓ 14.	OrientDB	Multi-model
16.	↑ 18.	16.	IBM Cloudant	Document
17.	17.	↓ 15.	Oracle NoSQL	Multi-model
18.	↓ 16.	↓ 17.	RavenDB	Document, Multi-model
19.	↑ 20.	19.	PouchDB	Document

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

SGBD NoSQL orienté document : MONGODB

- Insertion des clients du client 1

```
c:\>mongo
> use airbase
> db.createCollection("clients");
> db.createCollection("vols");
> document={
   _id: 1,
   nom: "ERZULIE",
   prenoms:["Maria", "Freda"],
   telephone:"00509232485",
   DateNaiss:"01/01/1881",
   adresse: {
            numero: 11,
                        rue:"Rue des miracles",
                        codePostal: "HT8110",
            ville: "PORT-AU-PRINCE",
            pays: "HAITI"
           }
}
>db.clients.insert(document);
WriteResult({ "nInserted" : 1 })
```

La collection clients est aussi créés par la même occasion.

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté document : MONGODB

- **Insertion des clients du client 2**

```
c:\>mongo
> use airbase
> db.clients.insertOne(
{
  _id: 2,
  nom: "BARON",
  prenoms:["Samedi"],
  telephone:"00509232488",
  DateNaiss:"01/01/1870",
  adresse: {
          numero: 1,
                        rue:"Rue des lwa",
                        codePostal: "HT8111",
          ville: "PORT-AU-PRINCE",
          pays: "HAITI"
         }
});
{ "acknowledged" : true, "insertedId" : 2 }
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté document : MONGODB

- **Insertion d'un VOL**

```
c:\>mongo
> use airbase
> document ={ _id: "100",
   villeDepart: "Nice",
   villeArrivee: "Paris",
   heureDepart: "10:10",
   heureArrivee: "11:30",
   dateVol: "12/12/2018",
   appreciations:[ {idClient:1,
      notes:[
      {apid: 11, critereANoter: "SiteWeb", note: "BIEN"},
      {apid: 12, critereANoter: "Prix", note: "BIEN"},
      {apid: 13, critereANoter: "Nourriture à bord", note: "BIEN"},
      {apid: 14, critereANoter: "Qualité siège", note: "MOYEN"},
      {apid: 15, critereANoter: "Accueil guichet", note: "TRES_BIEN"},
      {apid: 16, critereANoter: "Accueil à bord", note: "EXCELLENT"}
      ]
     },
```

… Voir la suite page suivante

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté document : MONGODB

- **Insertion d'un vol suite**

```
{idClient:2,
      notes:[
      {apid: 21, critereANoter: "SiteWeb", note: "TRES_BIEN"},
      {apid: 22, critereANoter: "Prix", note: "MEDIOCRE"},
      {apid: 23, critereANoter: "Nourriture à bord", note: "BIEN"},
      {apid: 24, critereANoter: "Qualité siège", note: "MOYEN"},
      {apid: 25, critereANoter: "Accueil guichet", note: "TRES_BIEN"},
      {apid: 26, critereANoter: "Accueil à bord", note: "BIEN" }
      ]
     }
      ]
}

>db.vols.insertOne(document);
{ "acknowledged" : true, "insertedId" : 100 }
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

- **Recherche de tous les documents d'une collection : MONGODB**

-- **Pour un meilleurs affichage**

>Db.clients.find().pretty();

```
> db.clients.find().pretty();
{
        "_id" : 1,
        "nom" : "ERZULIE",
        "prenoms" : [
                "Maria",
                "Frida"
        ],
        "telephone" : "00509232472",
        "DateNaiss" : "01/01/1880",
        "adresse" : {
                "numero" : 11,
                "rue" : "Rue des miracles",
                "codePostal" : "HT8110",
                "ville" : "PORT-AU-PRINCE",
                "pays" : "HAITI"
        }
}
{
        "_id" : 2,
        "nom" : "BARON",
        "prenoms" : [
                "Samedi"
        ],
        "telephone" : "00509232488",
        "DateNaiss" : "01/01/1870",
        "adresse" : {
                "numero" : 1,
                "rue" : "Rue des lwa",
                "codePostal" : "HT8111",
                "ville" : "PORT-AU-PRINCE",
                "pays" : "HAITI"
        }
}
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

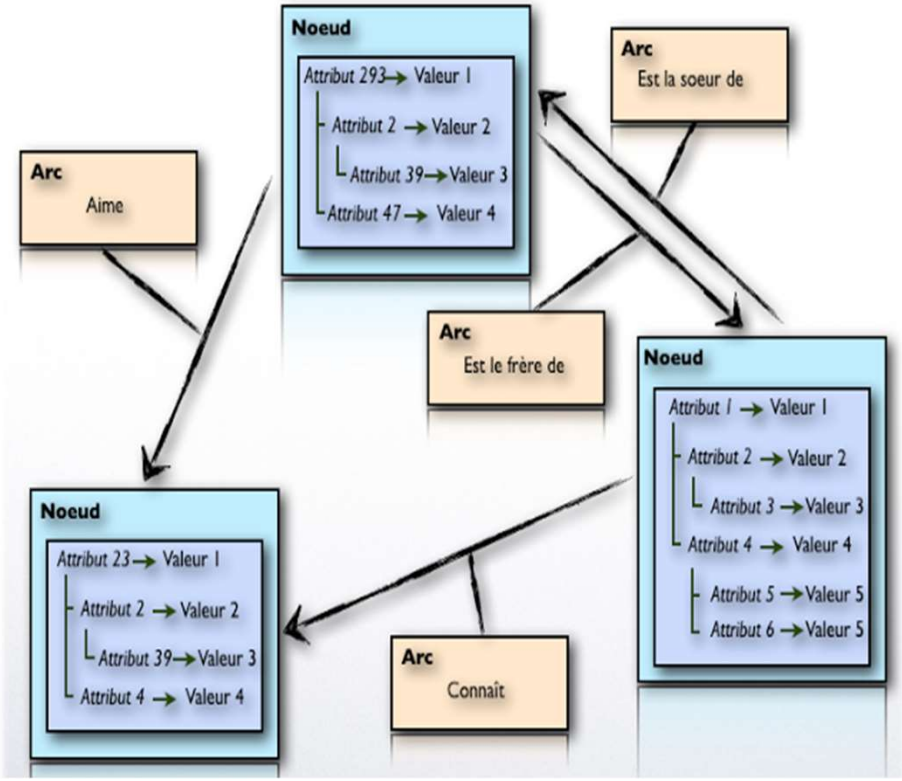
- **Recherche de documents connaissant la valeur d'une propriété : MONGODB**

```
-- Recherche de clients de nom ERZULIE
> db.clients.find({nom:"ERZULIE"}).pretty();
```

```
> db.clients.find({nom:"ERZULIE"}).pretty();
{
        "_id" : 1,
        "nom" : "ERZULIE",
        "prenoms" : [
                "Maria",
                "Frida"
        ],
        "telephone" : "00509232472",
        "DateNaiss" : "01/01/1880",
        "adresse" : {
                "numero" : 11,
                "rue" : "Rue des miracles",
                "codePostal" : "HT8110",
                "ville" : "PORT-AU-PRINCE",
                "pays" : "HAITI"
        }
}
>
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

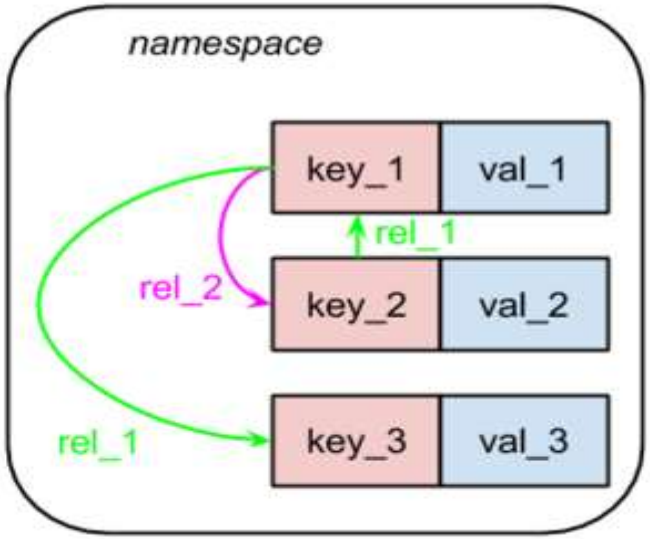
- SGBD NoSQL orienté graphe



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté graphe

- Quelque soit le modèle, la base est toujours Clé/valeur



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté graphe

- **Modèle de représentation** des données basé sur la **théorie des graphes**. Modèle puissant
- S'appuie sur les notions de **nœuds**, de **relations** et de **propriétés** qui leur sont rattachées
- Possibilité de **joindre des graphes**
- Possibilité de **fusionner des graphes**
- chaque ligne a une structure «**nœud-lien-noeud**» (sujet-prédicatobjet)
- **Domaines d'applications** : données de **cartographie**, de **relations** entre personnes, **moteur** de recommandation, **généalogie**, web **sémantique**, catalogue **produit**, **sciences** de la vie, **géolocalisation**,
- **SGDB les plus connues** : **Neo4J**, **OrientDB** (fondation Apache)

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

- **SGBD NoSQL orienté Graphe**
  - https://db-engines.com/en/ranking/graph+dbms

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model
1.	1.	1.	Neo4j	Graph
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model
3.	3.	3.	OrientDB	Multi-model
4.	4.	4.	ArangoDB	Multi-model
5.	5.	5.	Virtuoso	Multi-model
6.	6.	↑ 13.	JanusGraph	Graph
7.	↑ 8.	↓ 6.	Giraph	Graph
8.	↓ 7.	↓ 7.	Amazon Neptune	Multi-model
9.	↑ 10.	↑ 11.	GraphDB	Multi-model
10.	↓ 9.	↓ 8.	AllegroGraph	Multi-model
11.	↑ 13.	↑ 22.	TigerGraph	Graph
12.	12.	↓ 9.	Stardog	Multi-model
13.	↓ 11.	↑ 18.	Dgraph	Graph
14.	14.	↓ 12.	Graph Engine	Multi-model
15.	15.	15.	Blazegraph	Multi-model
16.	↑ 18.	↑ 21.	FaunaDB	Multi-model
17.	↓ 16.	↓ 10.	Sqrrl	Multi-model
18.	↓ 17.	↑ 19.	InfiniteGraph	Graph
19.	↑ 20.	↑ 20.	InfoGrid	Graph
20.	↓ 19.	↓ 16.	FlockDB	Graph

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté graphe : NEO4J

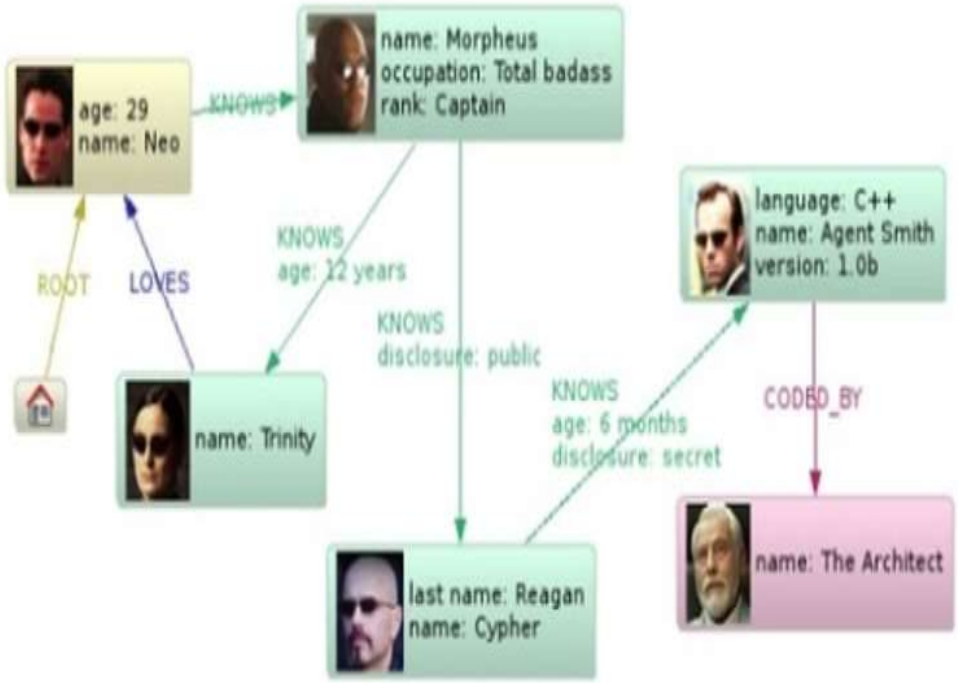
- **Un graphe attribué est modélisée grâce à trois blocs de base:**
  - Le **nœud** ou sommet (node, vertex)
  - La **relation** ou arête (relationship, edge), avec une orientation et un type (orienté et marqué)
  - La **propriété** ou attribut (property, attribute), portée par un nœud ou une relation

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté graphe : NEO4J

- Exemple du film MATRIX (https://www.infoq.com/fr/articles/graph-nosql-neo4j/

  Par Peter NEUBAR, InfoQ est un site communautaire



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢SGBD NoSQL orienté graphe : NEO4J

```
-- Le code complet pour créer le graphe Matrix
graphdb = new EmbeddedGraphDatabase("target/neo4j");
index = new LuceneIndexService(graphdb);
Transaction tx = graphdb.beginTx();
try {Node root = graphdb.getReferenceNode();
   // we connect Neo with the root node, to gain an entry point to the graph
   // not neccessary but practical.
   neo = createAndConnectNode("Neo", root, MATRIX);
   Node morpheus = createAndConnectNode("Morpheus", neo, KNOWS);
   Node cypher = createAndConnectNode("Cypher", morpheus, KNOWS);
   Node trinity = createAndConnectNode("Trinity", morpheus, KNOWS);
   Node agentSmith = createAndConnectNode("Agent Smith", cypher, KNOWS);
   architect = createAndConnectNode("The Architect", agentSmith, HAS_CODED);
   trinity.createRelationshipTo(neo, LOVES); // Trinity loves Neo. But he doesn't know.
   tx.success();
} catch (Exception e) { tx.failure();
} finally {tx.finish();
}
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢ **SGBD NoSQL orienté graphe : NEO4J**

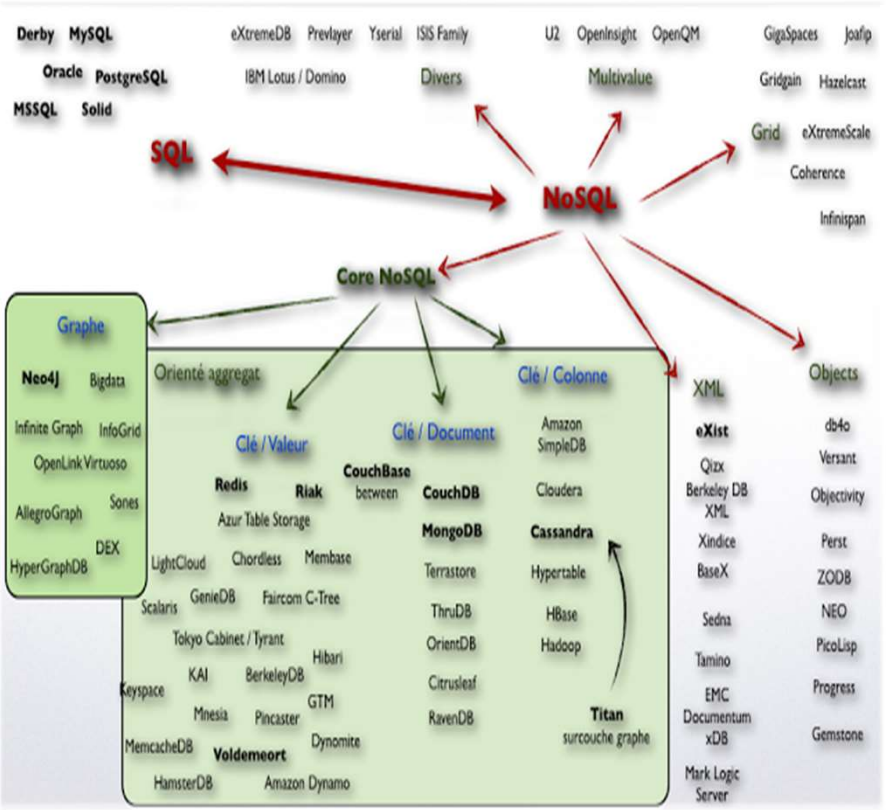
```
-- Le code complet pour créer le graphe Matrix
-- Avec la fonction interne...pour créer les noeuds et relations.
private Node createAndConnectNode(String name, Node otherNode,
   RelationshipType relationshipType) {
   Node node = graphdb.createNode();
   node.setProperty("name", name);
   node.createRelationshipTo(otherNode, relationshipType);
   index.index(node, "name", name);
   return node;
}

-- Requête à travers le graphe Matrix : Qui sont les amis de Néo?
-- L'API Neo4j a un certain nombre de méthodes orientées Collections Java pour répondre à des requêtes simples. Ici, un regard aux relations du noeud "Néo" est suffisant
-- pour trouver ses amis:
for (Relationship rel : neo.getRelationships(KNOWS)) {
   Node friend = rel.getOtherNode(neo);
   System.out.println(friend.getProperty("name"));
}
retourne "Morpheus" comme seul ami.
```

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢ Classification de SGBD NoQL

- http://administration-systeme.blogspot.fr/2013/10/bases-de-donnees-big-data-et-nosql.html



Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

➢ Popularité des SGBD selon DB Engine

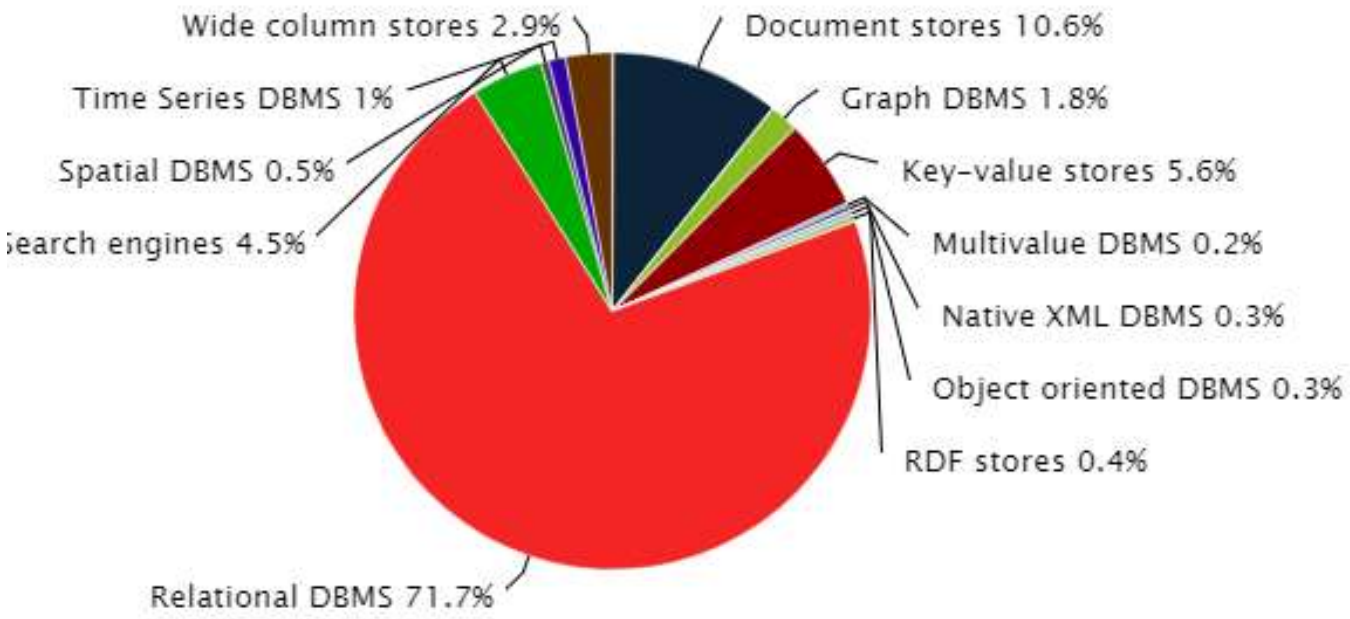
- **Critères :** nombre de mention du SGBD, Intérêt général au système, fréquence de discussions techniques sur le système, Nombre de jobs proposés, nombre profiles mentionnant le SGBD

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document
6.	6.	6.	IBM Db2	Relational, Multi-model
7.	↑ 9.	7.	Microsoft Access	Relational
8.	↓ 7.	8.	Redis	Key-value, Multi-model
9.	↓ 8.	9.	Elasticsearch	Search engine, Multi-model
10.	10.	↑ 11.	SQLite	Relational
11.	11.	↓ 10.	Cassandra	Wide column
12.	12.	↑ 15.	MariaDB	Relational, Multi-model
13.	13.	13.	Splunk	Search engine
14.	14.	↓ 12.	Teradata	Relational
15.	15.	↑ 18.	Hive	Relational
16.	16.	↓ 14.	Solr	Search engine
17.	17.	17.	HBase	Wide column
18.	18.	↑ 19.	FileMaker	Relational
19.	↑ 20.	↓ 16.	SAP Adaptive Server	Relational
20.	↓ 19.	20.	SAP HANA	Relational, Multi-model
21.	21.	21.	Amazon DynamoDB	Multi-model
22.	22.	22.	Neo4j	Graph
23.	23.	23.	Couchbase	Document

https://db-engines.com/en/ranking

Module M4.1, section 2, partie 2 : Rappel sur les Concepts des SGBD NOSQL

- Parts de marché des différents types de SGBD, Décembre 2022
  - https://db-engines.com/en/ranking_categories



Module M4.1, section 2, partie 2 : QUIZ

➢ **Question 1 :** Notez ce qui caractérise Le modèle Key value de base (valeur non structurée)

- A: l'accès à une valeur est indépendante du nombre et de la taille de la clé
- B: la taille d'un index de hachage vaut environ 10% des données
- C: L'application d'une fonction de hachage à la clé rend l'accès à 1 valeur rapide
- D: Toute recherche dans 1 base nosql quel qu'elle soient se fait toujours et uniquement via la clé

➢ **Question 2** : Cochez ce qui est un SGBD NOSQL

- A: Voldemort
- B: Mysql
- C: Cassandra
- D: Neo4j
- E: Sqlserver

Module M4.1, section 2, partie 2 : QUIZ

- **Question 3 :** Cochez ce qui est un SGBD NOSQL orienté Clé/valeur de base
  - A: Cassandra
  - B: Oracle NOSQL
  - C: Hbase
  - D: Neo4j
  - E: Voldemort
  - F: MongoDB

- **Question 4** : Cochez ce qui est unn SGBD NOSQL orienté colonne
  - A: Cassandra
  - B: Oracle NOSQL
  - C: Hbase
  - D: Neo4j
  - E: Voldemort
  - F: MongoDB

Module M4.1, section 2, partie 2 : QUIZ

- **Question 5 :** Cochez les SGBD NOSQL supportant le CP du théorème CAP
  - A: Cassandra
  - B: Oracle NOSQL
  - C: Hbase
  - D: Neo4j
  - E: Voldemort
  - F: MongoDB
- **Question 6** : Cochez les SGBD NOSQL supportant le AC du théorème CAP
  - A: Cassandra
  - B: Oracle NOSQL
  - C: Hbase
  - D: Neo4j
  - E: Voldemort
  - F: MongoDB
  - G: aucun

Module M4.1, section 2, partie 2 : QUIZ

- **Question 7 :** Cochez les SGBD NOSQL supportant le AP du théorème CAP
  - A: Cassandra
  - B: Oracle NOSQL
  - C: Hbase
  - D: Neo4j
  - E: Voldemort
  - F: MongoDB
  - G: Aucun
- **Question 8** : Cochez les SGBD supportant le AC du théorème CAP
  - A: Cassandra
  - B: Oracle 21c
  - C: Habase
  - D: Sqlserver
  - E: Voldemort
  - F: MongoDB
  - G: Aucun

UNIVERSITÉ CÔTE D'AZUR ESTIA INSTITUTE OF TECHNOLOGY

DATUM ACADEMY

Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

FUN FRANCE UNIVERSITÉ NUMÉRIQUE

anr

INVESTIR L'AVENIR

MIAGE RÉSEAU DES MIAGE DE FRANCE Nice

MBDS

BIHAR

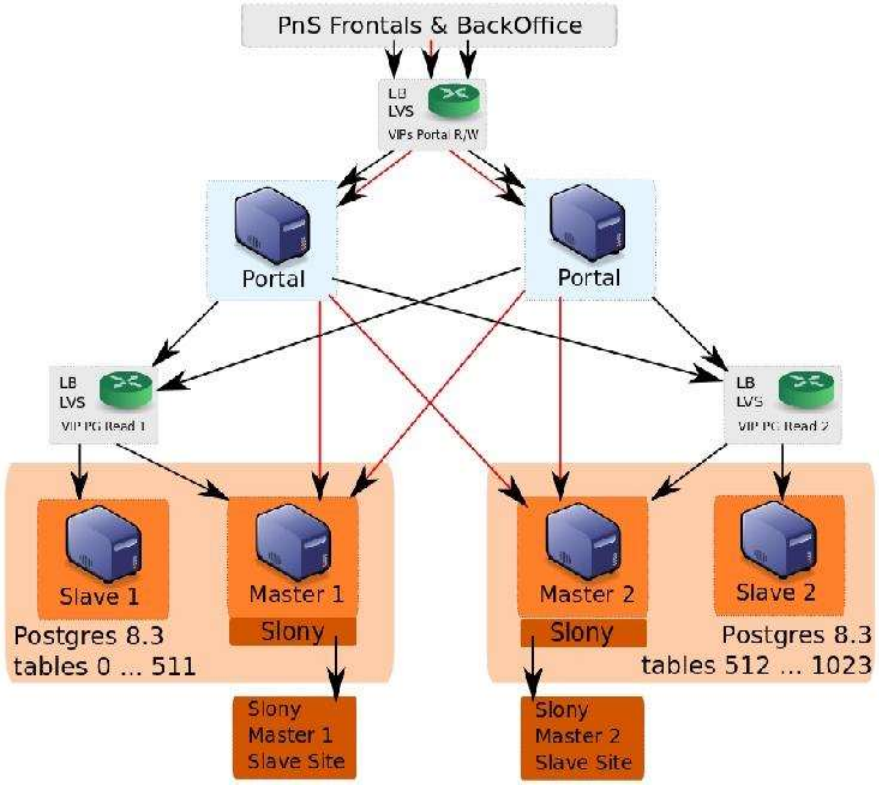
Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

Plan

- Bases de profiles gérée dans un cluster POSTGRES
- Base de Syndication : actuelle gérée dans un cluster MYSQL
- Limites de la solution actuelle
- Critères d'évaluation de systèmes NOSQL
- Environnement de TEST
- Résultat des évaluations

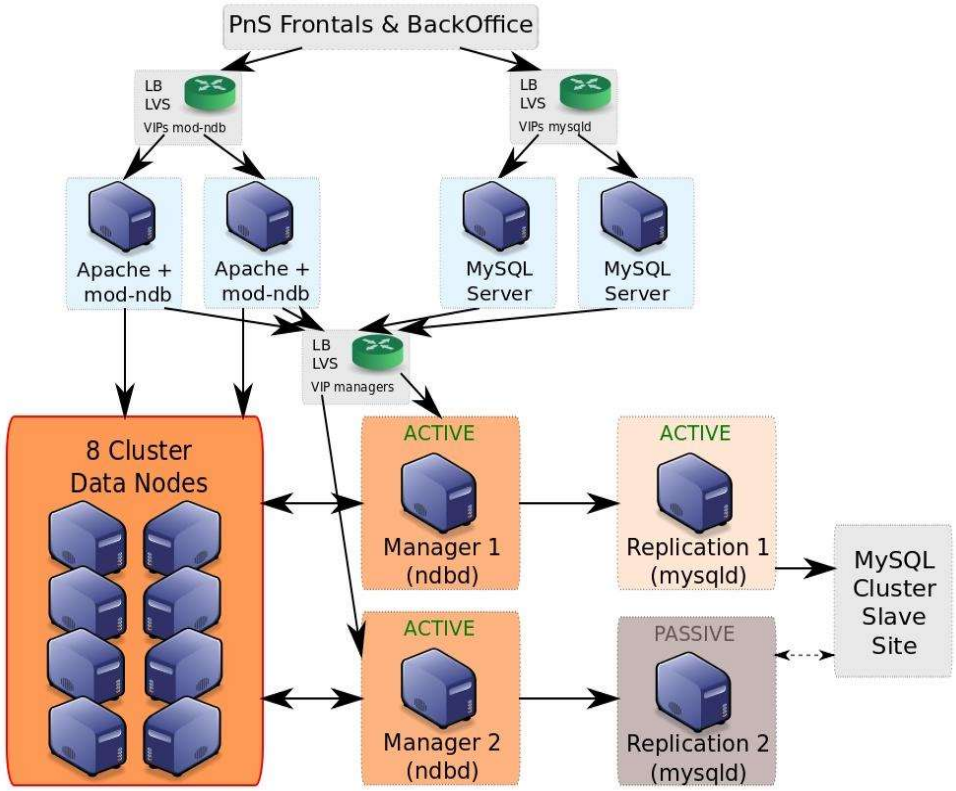
Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

➢ **Bases de profiles gérée dans un cluster POSTGRES**



Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

- **Base de Syndication : actuelle gérée dans un cluster MYSQL**
  - Base en mémoire contenant les infos sur le nombre de mails lus, non lus, …)



Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

➢ Limites de la solution actuelle

- **Volumétrie croissante de la base de syndication** (50Go) et de la base des profiles (100Go) avec des LOGs pour l'analyse d'erreurs importants (1h de log = 12Go, soient 105 Téraoctets de Log par an)
- **De nouveaux types de données Semi-structurées voir Non structurées** à supporter. Données au format XML, JSON
- **Contraintes en terme de performances menacées** (6000 à 7000 requêtes/secondes)
- **Haute disponibilité exigées** : 99.9%
- **Pas de facilité pour ajouter dynamiquement nouvelles colonnes**

Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

➢ Critères d'évaluation de systèmes NOSQL

- Haute disponibilité
- Performances
- Scalabilité
- Exploitation et administration
- Hardware
- Support et pérennité du système

Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

Environnement de TEST

- **Environnement Ubuntu** 10.0.4 / Ubuntu 11.10 (Riak)
- Serveurs **EC2 Amazon**
- **Outils** utilisés :
  - jMeter
  - jtlHistogram
  - VMware
- **SGBD NoSQL** : MongoDB, Riak, Citrus Leaf, Cassandra, Redis, Hbase

Module M4.1, section 3 : Comment choisir un système NoSQL : cas Orange Portail

- **Résultat des évaluations**

		MongoDB		Cassandra		CitrusLeaf		Redis		Hbase		Riak	
	Poids	Note	Total	Note	Total	Note	Total	Note	Total	Note	Total	Note	Total
Haute disponibilité	8	15	120	19	152	17	136	15	120	15	120	17	136
Performances	6	15	90	17	102	18	108	16	96	12	72	14	84
Scalabilité	4	15	60	18	72	17	68	10	40	19	76	16	64
Exploitation/ administration	4	10	40	15	60	15	60	16	64	13	52	15	60
Hardware	3	12	36	15	45	16	48	8	24	18	54	14	42
Support/ pérennité	3	18	54	17	51	12	36	11	33	18	54	17	51
Total			400/560		482/560		456/560		377/560		428/560		437/560

Module M4.1, section 3 : QUIZ

➢ **Question 1 : Avant de passer au NOSQL, cochez ce qui caractérisaient l'environnement applicatif d'Orange Portail**

- A: Certaines applications tournaient sur le SGBD Oracle
- B : Certaines applications tournaient sur le SGBD Sqlserver
- C: Certaines applications tournaient sur le SGBD Mysql
- D: Certaines applications tournaient sur le SGBD Postgres
- E: Certaines applications tournaient sur le SGBD Informix

- **Question 2** : Cochez ce qui a motivé Orange portail à passer au NOSQL
- A: 'impossibilité d'ajouter de nouveaux nœuds dans les deux clusters qu'ils avaient
- B: Les problèmes de temps de réponse
- C: La colère de leurs utilisateurs
- D: L'absente de simplicité lors de l'ajout de nouveaux champs
- E: Le besoin de gérer des données semi-structurées

Module M4.1, section 3 : QUIZ

➢ **Question 3 : Cochez les principaux critères de choix retenus pour l'étude**

- A: L'inviolabilité, la sécurité, les performances, exploitation, le hardware, la réplication asynchrone
- B : La consistance, l'atomicité, l'Isolation, la durabilité
- C: Haute disponibilité, Performances, Scalabilité, Exploitation et administration, Hardware, Support et pérennité du système
- D: La monté en charge, la facilité d'ajout de nœud, la disponibilité continue, l'intolérance aux pas pannes, la haute disponibilité à 99.9999999

- **Question 4** : Cochez les SGBD retenus pour l'étude
- A: Cassandra, Oracle nosql, couchDB, Microsoft Azur Cosmos
- B: MongoDB, Riak, Citrus Leaf, Cassandra, Redis, Hbase
- C: Google cloud FireStore, MarkLogic, ArangoDB, OrientDB, IBM Cloudant
- D: MongoDB, Cassandra, PouchDB, Amazon DynamoDB, Microsoft Azur Cosmos

Module M4.1 : Rappel su les Concepts du Big Data et des SGBD NoSql

- Bilan

- Exercices

Module M4.1 : Références Web

- [R1] “Exploring CouchDB: A document-oriented database for Web applications”, Joe Lennon, Software developer, Core International. http://www.ibm.com/developerworks/opensource/library/os-couchdb/index.html
- [R2] “Graph Databases, NOSQL and Neo4j” Posted by Peter Neubauer on May 12, 2010 at: http://www.infoq.com/articles/graph-nosql-neo4j
- [R3] “Cassandra vs MongoDB vs CouchDB vs Redis vs Riak vs HBase comparison”, Kristóf Kovács. http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis
- [R4] “Distinguishing Two Major Types of Column-Stores” Posted by Daniel Abadi onMarch 29, 2010 http://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of_29.html
- [R5] Bases de données : Big Data et NoSQL http://administration-systeme.blogspot.fr/2013/10/bases-de-donnees-big-data-et-nosql.html

Module M4.1 : Références Web

- [R6] “MapReduce: Simplified Data Processing on Large Clusters”, Jeffrey Dean and Sanjay Ghemawat, December 2004. https://static.googleusercontent.com/media/research.google.com/fr//archive/mapreduce-osdi04.pdf
- [R7] “Scalable SQL”, ACM Queue, Michael Rys, April 19, 2011 http://queue.acm.org/detail.cfm?id=1971597
- [R8] “a practical guide to noSQL”, Posted by Denise Miura on March 17, 2011 at http://blogs.marklogic.com/2011/03/17/a-practical-guide-to-nosql/
- [R9] Visual Guide to NoSQL Systems http://blog.nahurst.com/visual-guide-to-nosql-systems
- [R10] Infos sur Map Reduce. http://en.wikipedia.org/wiki/MapReduce
- [R11] WEBRANKINFO : site d’informations et de statistiques sur le WEB
  http://www.webrankinfo.com/dossiers/googles
  http://www.webrankinfo.com/dossiers/facebook
- [R12] Echo Système HADOOP

http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F

Module M4.1 : Références Web

- [R13] Hadoop Vive Tutorial
  https://www.tutorialspoint.com/hive/hive_create_table.htm

- [R14] Site web Hadoop Hive
  https://hive.apache.org/

- [R15] Wiki Apache Hive user documentation
  https://cwiki.apache.org/confluence/display/Hive/Home#Home-UserDocumentation

- [R16] HPL/SQL documentation (procédure stockées)
  https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=59690156

- [R17] Structured vs Unstructured Data: 5 Key Differences, by Mark Smallcombe
  https://www.integrate.io/blog/structured-vs-unstructured-data-key-differences/

- [R18] Semistructured Data, Peter Buneman, Department of Computer and Information Science, University of Pennsylvania
  https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.5869&rep=rep1&type=pdf

Module M4.1 : **Bibliographie**

- [B1] HADOOP in practice, Alex HOLMES, Editions Manning Publications, 2012

- [B2] Principles of Big Data
  Preparing, Sharing, and Analyzing Complex Information,
  Par Jules J. Berman, Edition Morgan Kaufmann, 2013

- [B3] "LES BASES DE DNNÉES NOSQL: COMPRENDRE ET METTRE EN ŒUVRE"
  Edition Eyrolles 2013

- [B4] Building the Data Warehouse; W.H. INMON, Éditeur : John Wiley & Sons (19 septembre 2005), ISBN-10 : 6610279713, ISBN-13 : 978-6610279715