Bases de données - NoSQL

CM 1

Des SGBD relationnels au NoSQL

Namrata PATEL

Namrata.patel@univ-montp3.fr

Organisation

Date	Session 1 (matin)	Session 2 (après-midi)
18 sept. 2020	CM1 Des SGBD relationnels au NoSQL CM 2 Les caractéristiques du NoSQL	CM3 Les schémas de données NoSQL
25 sept. 2020	CM-TD Les défis du NoSQL	Début TP noté Le partitionnement des données
16 oct. 2020	TP noté suite	Soutenances TP noté 20 min par groupe
20 nov. 2020	Examen sur table	-

Les SGBD

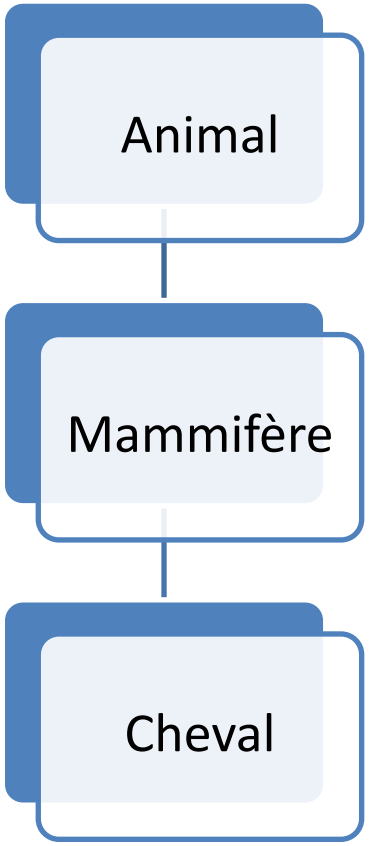
Une brève histoire

Origines

- 1940: Le besoin d'organiser les données
- Les premiers défis des SGBD :
  - Stocker
  - Retrouver
- Une recherche parallèle de différents systèmes

LE MODÈLE HIÉRARCHIQUE

- 1950: la 1ère forme de modélisation de données
- Structure arborescente où chaque enregistrement a un seul parent
- **1 seul arbre** et uniquement des relations descendantes
- Patient -> Docteurs !

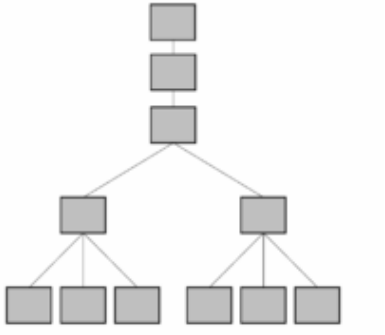


CODASYL et COBOL

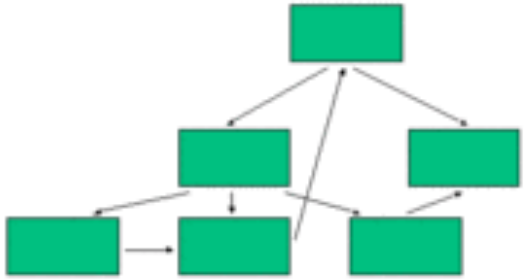
- **1959: CODASYL**
  - **COnference on DAta SYstems Languages**
  - **Spécification d'un langage de programmation standardisé**
- **→ COBOL**
  - **Common Business Oriented Language**
  - **Langage ressemblant le langage parlé (anglais)**
  - **But :**
    - interrogation de données
  - **Approche :**
    - Procédurale (i.e. spécification de commandes en série)
- **Modèle hiérarchique → modèle réseau**

Modèle hiérarchique → modèle réseau

- **Modèle hiérarchique :**
  - **1 seul parent/antécédent**



- **Modèle réseau :**
  - **Possibilité de lier un enregistrement à plusieurs enregistrements**



Passage au modèle relationnel

- **E.F. CODD :**
  - **Britannique travaillant pour IBM**
  - **Définition d'un nouveau modèle**
- **1969 : Présenté à IBM**
- **1970 : Publication scientifique**
  - **« A relational model of data for large shared data banks »**
  - **Les idées fondatrices du modèle relationnel**

De l'idée… aux produits

- **1974 : System R, Laboratoire de San Jose**
  - **Moteur en 2 parties : RSS et RDS**
    - Research Storage System
    - Relational Data System
  - **Langage de manipulation SEQUEL :**
    - Structured English QUEry Language
    - Renommé SQL (SEQUEL déjà pris)
- **1974 : Ingres, Université de Berkeley**
  - **PostgreSQL, Sybase, Informix**
- **1977 : Larry Ellison**
  - **Oracle, Oracle 2 : 1e moteur relationnel commercial**
- **1981 : System R → commercialisé sous SQL/DS**

Les 12 règles de CODD

Les codes de conduite

12 règles de CODD

- **Règle 1**
  - **Unicité :**
    - Toute l'information dans la base de données est **représentée d'une et une seule manière**, à savoir par des valeurs dans des champs de colonnes de tables.
- **Règle 2**
  - **Garantie d'accès :**
    - Toutes les données doivent être **accessibles sans ambiguïté**. Cette règle est essentiellement un ajustement de la condition fondamentale pour des clefs primaires. Elle indique que chaque valeur scalaire individuelle dans la base de données doit être logiquement accessible en indiquant le nom de la table contenante, le nom de la colonne contenante et la valeur principale primaire de la rangée contenante.
- **Règle 3**
  - **Traitement des valeurs nulles (le marqueur NULL) :**
    - Le système de gestion de bases de données doit **permettre à chaque champ de demeurer nul** (ou vide). Spécifiquement, il doit soutenir une représentation "d'information manquante et d'information inapplicable" qui est systématique, distincte de toutes les valeurs régulières (par exemple, "distincte de zéro ou tous autres nombres," dans le cas des valeurs numériques), et **ce indépendamment du type de données**. Cela implique également que de telles représentations doivent être gérées par le système de gestion de bases de données d'une manière systématique.
- **Règle 4**
  - **Catalogue lui-même relationnel :**
    - Le système doit supporter **un catalogue en ligne, intégré, relationnel, accessible aux utilisateurs autorisés au moyen de leur langage d'interrogation régulier.** Les utilisateurs doivent donc pouvoir accéder à la structure de la base de données (catalogue) employant le même langage d'interrogation qu'ils emploient pour accéder aux données de la base de données.

12 règles de CODD

- **Règle 5**
  - **Sous-langage de données :**
    - Le système doit soutenir au moins un langage relationnel qui :
      - a une syntaxe linéaire
      - peut être employé interactivement et dans des programmes d'application,
      - supporte des opérations de définition d'informations supplémentaires (incluant des définitions de vues), de manipulation de données (mise à jour aussi bien que la récupération), de contraintes de sécurité et d'intégrité, et des opérations de gestion de transaction (commencer, valider et annuler une transaction).
- **Règle 6**
  - **Mise à jour des vues :**
    - Toutes les vues pouvant théoriquement être mises à jour doivent pouvoir l'être par le système.
- **Règle 7**
  - **Insertion, mise à jour, et effacement de haut niveau :**
    - Le système doit supporter les opérations par lot d'insertion, de mise à jour et de suppression. Ceci signifie que des données peuvent être extraites d'une base de données relationnelle dans des ensembles constitués par des données issues de plusieurs tuples et/ou de multiples table. Cette règle explique que l'insertion, la mise à jour, et les opérations d'effacement devraient être supportées aussi bien pour des lots de tuples issues de plusieurs tables que juste pour un tuple unique issu d'une table unique.
- **Règle 8**
  - **Indépendance physique :**
    - Les modifications au niveau physique (comment les données sont stockées, si dans les rangées ou les listes liées, etc.) ne nécessitent pas un changement d'une application basée sur les structures.

12 règles de CODD

- **Règle 9**
  - **Indépendance logique :**
    - Les changements au niveau logique (tables, colonnes, rangées, etc) ne doivent pas exiger un changement dans l'application basée sur les structures. L'indépendance de données logiques est plus difficile à atteindre que l'indépendance de donnée physique.
- **Règle 10**
  - **Indépendance d'intégrité :**
    - Des contraintes d'intégrité doivent être indiquées séparément des programmes d'application et être stockées dans le catalogue. Il doit être possible de changer de telles contraintes au fur et à mesure sans affecter inutilement les applications existantes.
- **Règle 11**
  - **Indépendance de distribution :**
    - La distribution des parties de la base de données à diverses localisations doit être invisible aux utilisateurs de la base de données. Les applications existantes doivent continuer à fonctionner avec succès :
      - quand une version distribuée du système de gestion de bases de données est d'abord présentée ; et
      - quand des données existantes sont redistribués dans le système.
- **Règle 12**
  - **Règle de non-subversion :**
    - Si le système fournit une interface de bas niveau, cette interface ne doit pas permettre de contourner le système (par exemple une contrainte relationnelle de sécurité ou d'intégrité).

OLTP et OLAP

Passage aux données complexes

De OLTP à OLAP

- **OLTP (On Line Transaction Processing)**
  - Utilisé dans les PGI/ERP
  - Performant pour une utilisation transactionnelle
  - Envergure limitée
- **OLAP (On Line Analytical Processing)**
  - Utilisé pour les statistiques et les prédictions
  - Notion de cube de données
  - Grand volume de données

L'émergence du Big Data

Le besoin d'un nouveau paradigme

Le Big Data

- En lien avec les évolutions de son époque (l'arrivée d'internet):
  - Bande passante
  - Capacité de stockage
  - Coût de stockage
  - Interconnexion réseau
  - Volume de données disponibles

La solution Google

- Volume extrêmement important de données pour alimenter :
  - Google, Gmail
  - YouTube
  - Google Maps, Google Drive…
- Solution maison :
  - Un système de stockage (GoogleFS, 2003)
  - Un système d'interrogation (MapReduce, 2004)

HADOOP

UNE IMPLÉMENTATION DE MAPREDUCE

- Doug Cutting, développeur de Lucene
  - Nutch
    - moteur d'indexation open source du web
  - Hadoop
    - Implémantation de MapReduce en java
    - HDFS (Hadoop Distributed File System)
  - Projet soutenu et utilisé par
    - Apache, Yahoo!, Microsoft, Facebook, Twitter…

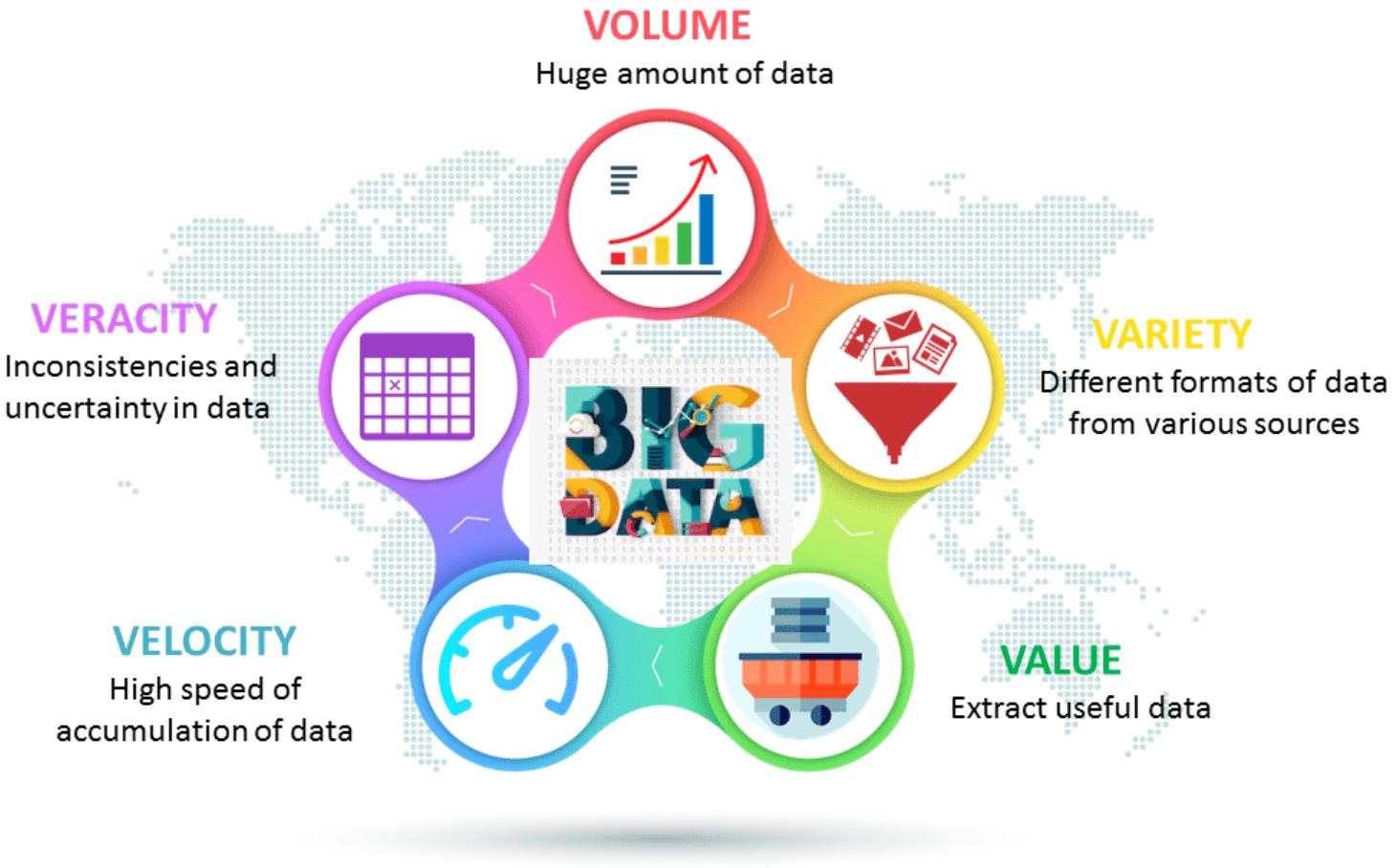
BIGTABLE, HBASE, DYNAMO

- BigTable (2004) - Google
  - Système de gestion de données basé sur GFS
  - Présenté en 2006
- HBase (2006)
  - Une implémentation libre de BigTable
  - Basé sur Hadoop et HDFS
  - Indépendant de la fondation Apache
  - Utilisé par Ebay, Yahoo!, Twitter…
- Dynamo (2007) – Amazon

APACHE CASSANDRA ET DYNAMODB

- Cassandra, 2008 (A. Lakshman)
  - Développeur Dynamo débauché par Facebook
  - Entrepôt destiné à gérer la messagerie
  - Offert à la fondation Apache en 2009
- DynamoDB, 2012
  - Mis à disposition de son moteur de recherche par Amazon
  - Avantage: Simplicité et absence de contraintes matérielles
  - Inconvénient: Données sur serveur Amazon

Les 5V du Big Data



Big Data - Volumétrie

- **1 kilooctet (ko)= 103 octets= 1 000 octets**
- **1 mégaoctet (Mo)= 106 octets= 1 000 ko= 1 000 000 octets**
- **1 gigaoctet (Go)= 109 octets= 1 000 Mo= 1 000 000 000 octets**
- **1 téraoctet (To)= 1012 octets= 1 000 Go= 1 000 000 000 000 octets***
- **1 pétaoctet (Po)= 1015 octets= 1 000 To= 1 000 000 000 000 000 octets***
- **1 exaoctet (Eo)= 1018 octets= 1 000 Po= 1 000 000 000 000 000 000 octets**
- **1 zettaoctet (Zo)= 1021 octets= 1 000 Eo= 1 000 000 000 000 000 000 000 octets**
- **1 yottaoctet (Yo)= 1024 octets= 1 000 Zo= 1 000 000 000 000 000 000 000 000 octets**

Big Data - Volumétrie

- **By 2020, there will be around 40 trillion gigabytes of data (40 zettabytes).**
- **90% of all data has been created in the last two years.**
- **Today it would take a person approximately 181 million years to download all the data from the internet.**
- **In 2018, internet users spent 2.8 million years online.**
- **Social media accounts for 33% of the total time spent online.**
- **Twitter users send nearly half a million tweets every minute.**
- **97.2% of organizations are investing in big data and AI.**
- **Using big data, Netflix saves $1 billion per year on customer retention.**

Conclusion

- Le NoSQL répond à un besoin
  - Le NoSQL est « Web scale »
  - Soutenu par les plus grands acteurs
- Le NoSQL est pluriel
- Les moteurs NoSQL sont multiples
- Beaucoup de bases NoSQL sont libres
  - Gratuit de base avec option payante
    - Oracle, Riak, Mongo DB…
  - Gratuit
    - HBase, Cassandra…