Modélisation des Données

Dr I GAYE

Universite Alioune DIOP (UAD) de Bambey

27 novembre 2024

Outline

Les bases de données nouvelles générations (NoSQL)

Implémentation MongoDB

Forces des bases de données relationnelles

SGBDR se resument comme suit :

- ☞ Faire des jointures entre les tables de la base de données ;
- ☞ Faire des requêtes complexes avec un langage de haut niveau sans se préoccuper des couches basses ;
- ☞ Gérer l'intégrité des données de manière infaillible.

Forces des bases de données relationnelles

Force des SGBDR : propriétés ACID pour les transactions (séquences d'opérations/requêtes) :

- ☞ Atomicité : Une transaction s'effectue entièrement ou pas du tout ;
- ☞ Cohérence : Le contenu d'une base doit être cohérent au début et à la fin d'une transaction ;
- ☞ Isolation : Les modifications d'une transaction ne sont visibles/modifiables que quand celle-ci a été validée ;
- ☞ Durabilité : Une fois la transaction validée, l'état de la base est permanent (non affecté par les pannes ou autre).

Faiblesses des bases données relationnelles

- ☞ Incapable de gérer de très grands volumes de données (de l'ordre du péta-octet)
- ☞ Impossible de gérer des débits extrêmes (plus que quelques milliers de requêtes par seconde)
- ☞ Le modèle relationnel est parfois peu adapté au stockage et à l'interrogation de certains types de données (données hiérarchiques, faiblement structurées, semi-structurées)
- ☞ Les propriétés ACID entraînent de sérieux surcoûts en latence, accès disques, temps CPU (verrous, journalisation, etc.)
- ☞ Performances limitées par les accès disque.

Bases de données nouvelles générations ou NoSQL

Q'est-ce qu'une base de données NoSQL

NoSQL, également appelé "pas seulement SQL" ("Not only SQL" en anglais) ou « non-SQL », est une approche de la conception de bases de données qui permet le stockage et l'interrogation de données en dehors des structures traditionnelles que l'on trouve dans les bases de données relationnelles.

Propriétes du NoSQL

Propriétés BASE ont été proposées pour caractériser les bases NoSQL :

- Basically Available : 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.

Propriétes du NoSQL

Stokage

- But : Répondre efficacement aux réquêtes des utilisateurs ;
- Plusieurs indexes permettent d'accéder directement aux informations :
  - Les arbres : plus utilisé est BTree qui adopte une structure arborescente pour chercher toute valeur indexée
  - Le hachage : table de hachage est de placer les données par paquets via une fonction de hachage
  - La distribution : utiliser depuis les bases de données distribuées ; les données sont réparties dans plusieurs serveurs
  - ...

Propriétes du NoSQL

Stokage

- But : Répondre efficacement aux réquêtes des utilisateurs ;
- Plusieurs indexes permettent d'accéder directement aux informations :
  - ...
  - L'élasticité : capacité du système à s'adapter automatiquement en fonction du nombre de serveurs
  - Le sharding : une technique permettant de distribuer des chunks (morceaux de fichiers) sur un ensemble de serveurs

Propriétes du NoSQL

Stokage

- Trois familles de sharding pour NoSQL :
  - Hadoop Distributed File System (HDFS) (basé sur la distribution),
  - le clustered index (basé sur le BTree) et
  - le consistent hashing (basé sur les tables de hachage)

Propriétes du NoSQL

Stokage

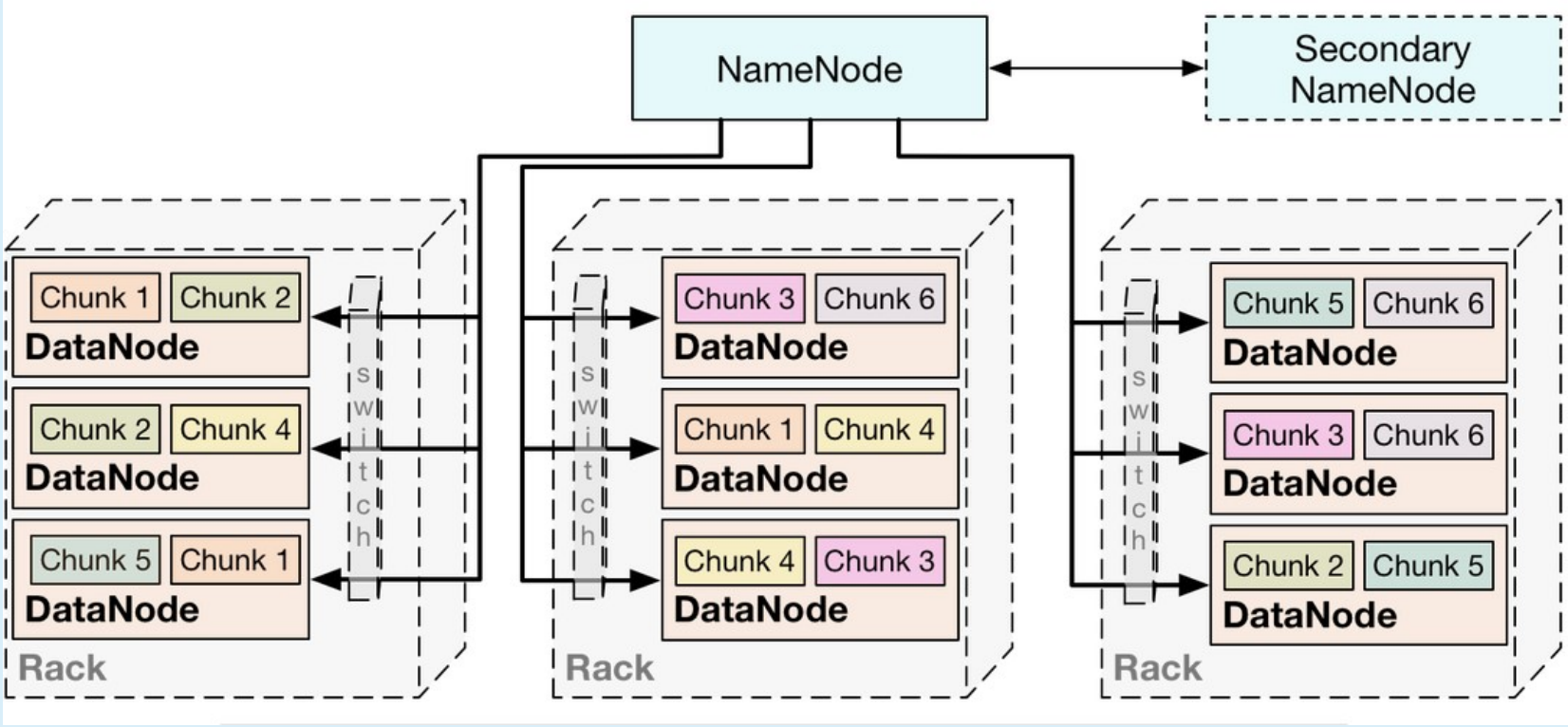


FIGURE: Architecture Hadoop Distributed File System (HDFS)

Source : https ://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462476-passez-a-lechelle

Propriétes du NoSQL

Stokage

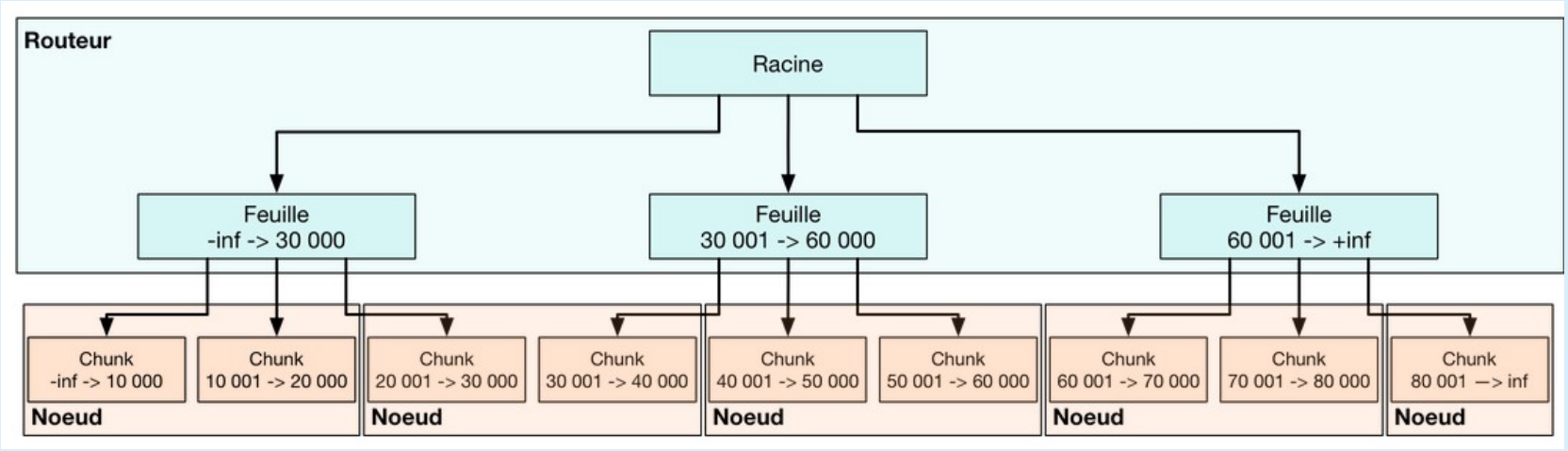


FIGURE: Architecture Btree

Source : https ://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462476-passez-a-lechelle

Propriétes du NoSQL

Stokage

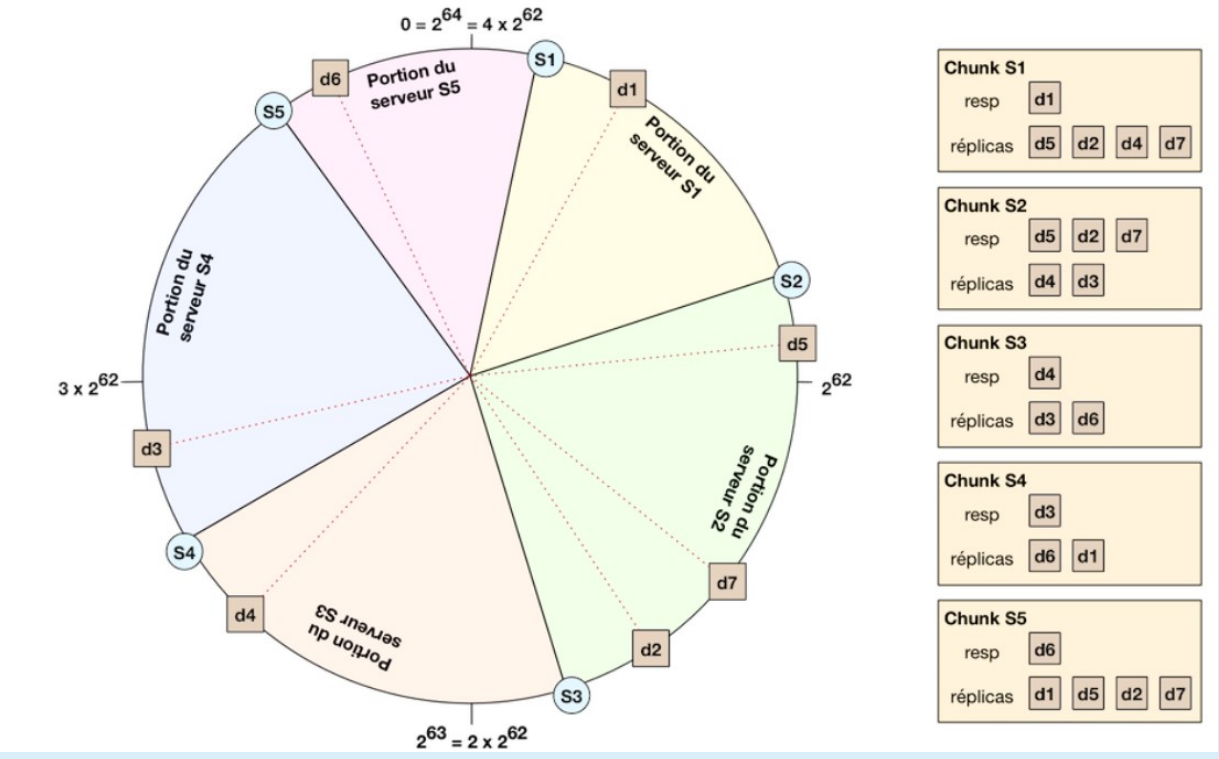


FIGURE: Architecture basée sur table de hachage

Source : https ://openclassrooms.com/fr/courses/4462426-maitrisez-les-bases-de-donnees-nosql/4462476-passez-a-lechelle

Catégories de BD NoSQL

Les quatre principaux types de bases de données NoSQL sont les suivants :

- Clé-valeur ;
- Document ;
- Colonnes ;
- Graphe.

Catégories de BD NoSQL

Orienté Clé-valeur

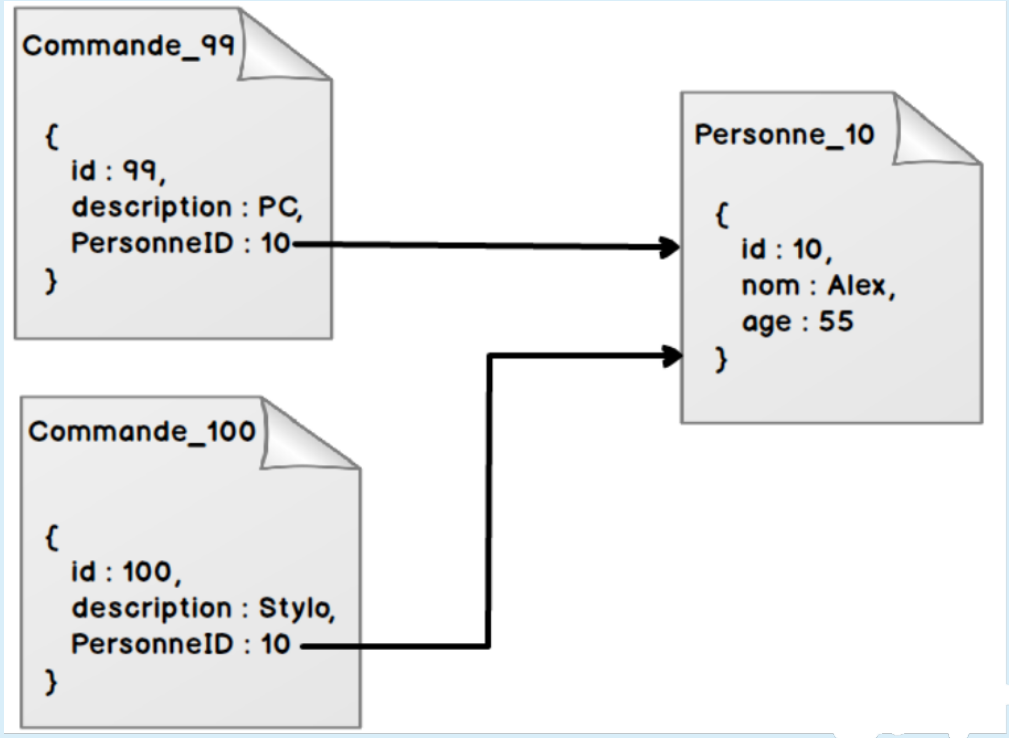
Clé-valeur stocke des paires de clés et de valeurs à l'aide d'une table de hachage. Les types clé-valeur sont particulièrement adaptés lorsqu'une clé est connue et que la valeur associée à la clé est inconnue.

Clé	Valeur
C1	XXXX, YYYYYY, ZZZZZ
C2	123, 456, 789
C3	AA, BB, CC
C4	123, AAA, DDDD
C5	X
C6	KKKK, LLLLL, MMMM

Catégories de BD NoSQL

Orienté Documents

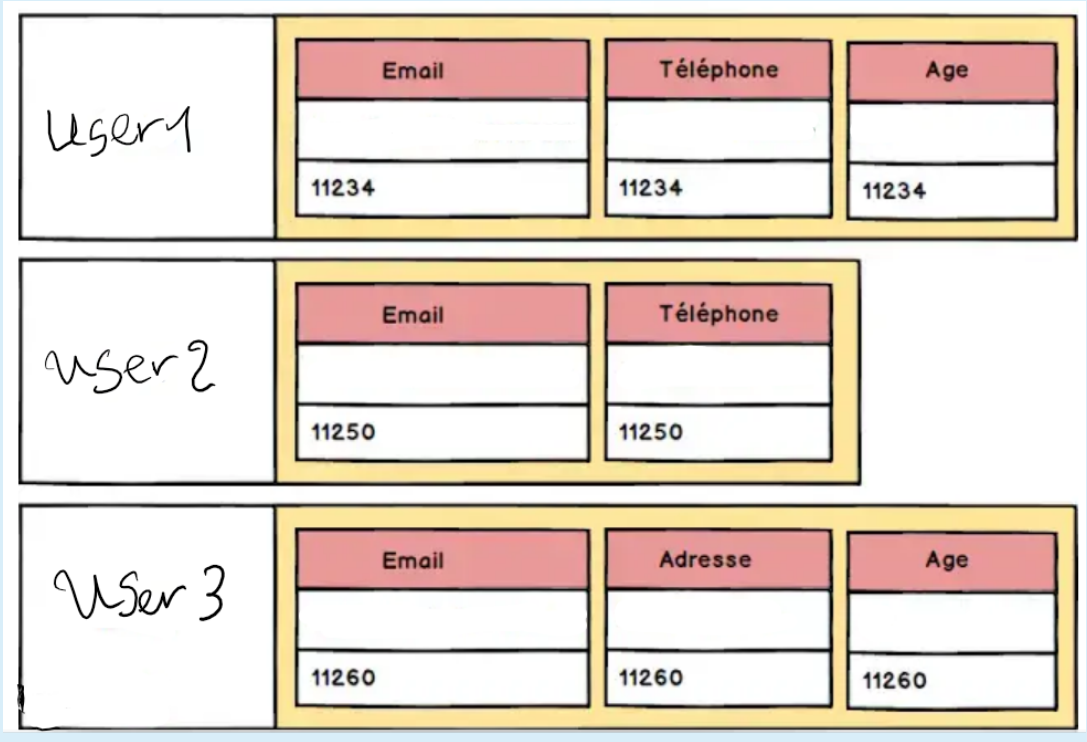
Les BD orentés documents étendent le concept de base de données clé-valeur en organisant des documents entiers dans des groupes appelés collections. Elles prennent en charge les paires clé-valeur imbriquées et autorisent les requêtes sur tous les attributs d'un document.



Catégories de BD NoSQL

Orienté colonnes

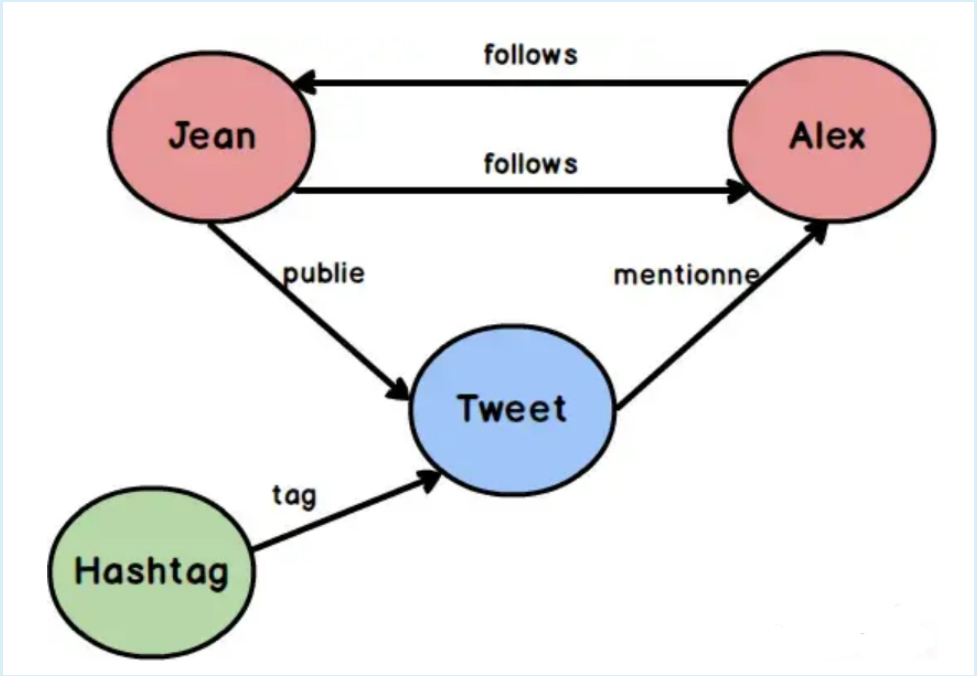
Les BD orientées colonnes stockent efficacement les données et interrogent les lignes de données éparses, et offrent la possibilité d'interroger les colonnes spécifiques d'une base de données.



Catégories de BD NoSQL

Orienté Graphe

Les BD otientées graphe utilisent un modèle basé sur les noeuds et les arêtes pour représenter les données interconnectées (relations entre membres d'un réseau social, par exemple), et offrent un stockage et une navigation facilités en présence de relations complexes.



MongoDB

- MongoDB est développé depuis 2007 par MongoDB qui est une entreprise qui évolue dans le domaine de Cloud Computing ;
- MongoDB est une base de données NoSQL orientée documents, Open-source et Cross-plateforme ;
- MongoDB utilise le format BJSON (JSON Binaire) et pas obligatoire d'avoir un schémas de BD.

MongoDB Installation

MongoDB a besoin de trois choses :

- Téléchargement du serveur : https ://www.mongodb.com/download-center ?jmp=navcommunity :
  - Windows : pas difficile (laisser les paramètres par défaut)
  - MacOSx : décompressez l'archive et déplacez le répertoire dans vos applications "/Applications" ;
  - Debian/Ubuntu : utilisez la commande "sudo apt install mongodb".

MongoDB Installation

- ☞ création d'un répertoire pour stocker les données :
  - ☛ Windows : créer le répertoire "data" puis dans ce répertoire, créer le répertoire "db".
  - ☛ Linux ou MacOSx : créer le répertoire data à la racine avec "sudo mkdir /data" Ensuite donnez les droits (ex. utilisateur "toto") : "sudo chown toto /data" enfin, créer le répertoire db : "mkdir /data/db".
- ☞ Le lancement du serveur, avec l'exécutable mongod (disponible sur $MONGO/bin)

MongoDB Interface utilisateur

- Administration avec une interface graphique.
  - Robo3T
    (https ://fastdl.mongodb.org/windows/mongodb-windows-x86_64-8.0.3.zip).
    - Windows : application est installée par défaut dans le répertoire C :FilesT3T
    - Mac : suffit de placer Robo3T dans le répertoire "Applications"
    - Linux : le fichier d'installation doit être décompressé (tar xzvf robo3t-1.1.1-linux-x86_64-c93c6b0.tar.gz). Puis, vous pouvez exécuter le binaire "bin/robo3t"

Création d'une collection

- MongoDB permet de manipuler des objets structurés au format BSON (JSON binaire), sans schéma prédéterminé.
- Les données prennent la forme de documents enregistrés eux-mêmes dans des collections
- Exemple :

ID	NOM	PRENOM	AGE
13	NIANG	Ko Mbodj	30
1234	GAYE	Aminata Racine	8
10	NGONE	ALy	34
2345	DIOP	Babacar	33

Base de Données orientée Documents

Modélisation de documents

- ☞ Documents JSON :

  Tout est clé/valeur : "clé" : "valeur"
  Un document est encapsulé dans des accolades {...}, pouvant contenir des listes de clés/valeurs
  Une valeur peut être un type scalaire (entier, nombre, texte, booléen, null), des listes de valeurs [...], ou des documents imbriqués

Modélisation de documents

- MongoDB permet de manipuler des objets structurés au format SON (JavaScript Object Notation)), sans schéma prédéterminé.
- Les données prennent la forme de documents enregistrés eux-mêmes dans des collections
- Exemple :

ID	NOM	PRENOM	AGE
13	NIANG	Ko Mbodj	30
1234	GAYE	Aminata Racine	8
10	NGONE	ALy	34
2345	DIOP	Babacar	33

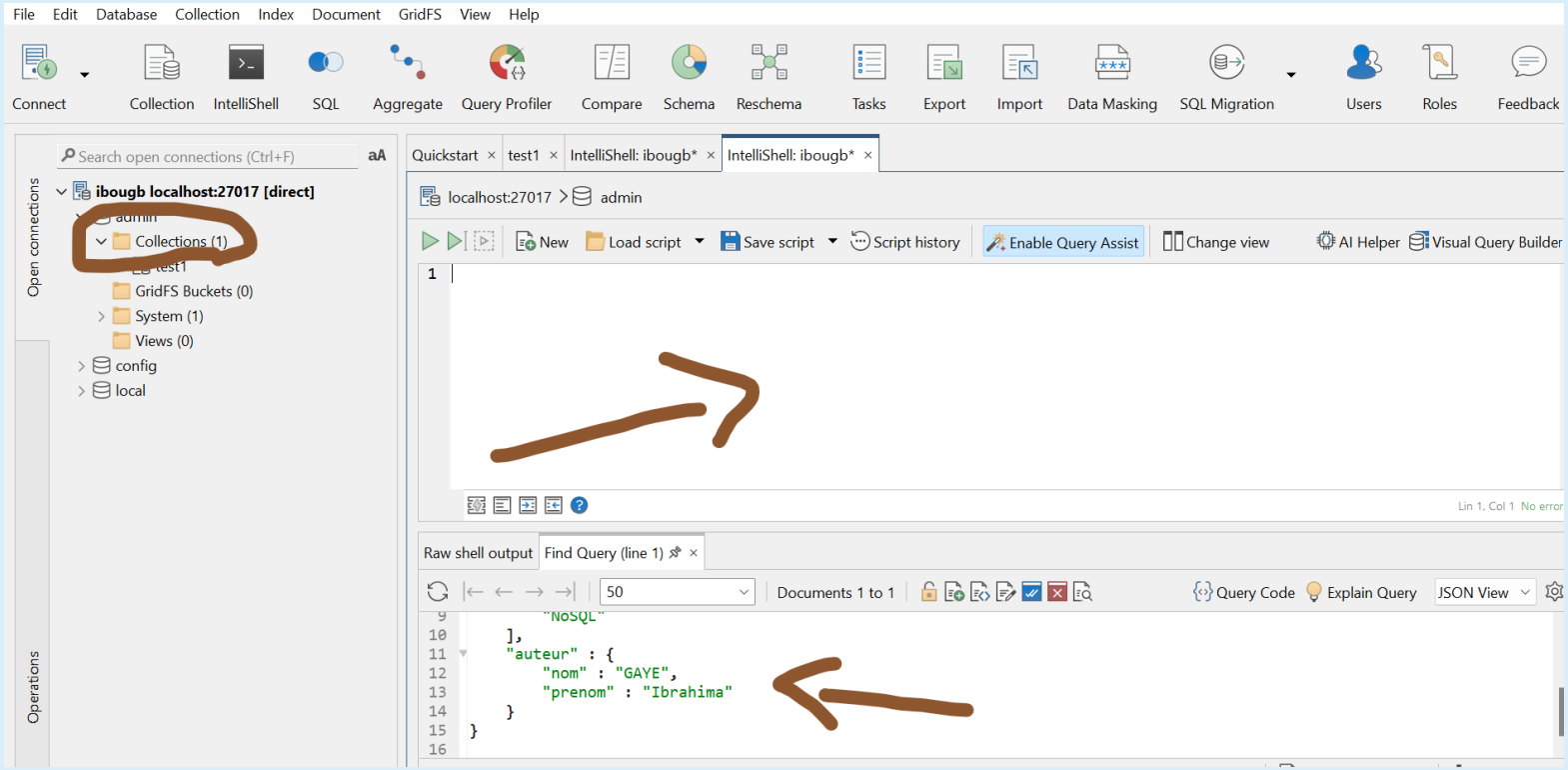
Modélisation de documents

☞ Le document JSON correspondant :

{ "id" : 13, "NOM" : "NIANG", "PRENOM" : "Ko Niang", "AGE" : 30 },
{ "id" : 1234, "NOM" : "GAYE", "PRENOM" : "Aminata Racine", "AGE" : 8 },
{ "id" : 10, "NOM" : "NGONE", "PRENOM" : "Aly", "AGE" : 34 },
{ "id" : 2345, "NOM" : "DIOP", "PRENOM" : "Babacar", "AGE" : 33 },

Création d'une collection

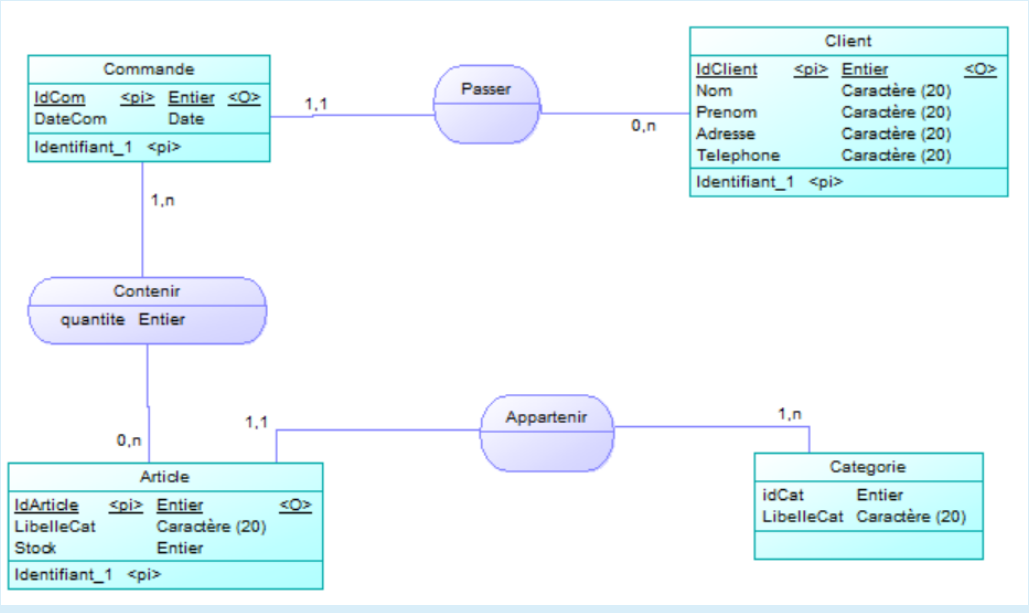
☞ Dans Robo3T, il faut créer une connexion (localhost et 27017 le port par défaut)



Relationnel vers JSON (La Dénormalisation)

- ☞ Les données utilisées de manière classiques sont en relationnel.
- ☞ On doit faire face à un gros problème dans les bases NoSQL : Il faut proscrire les jointures.
- ☞ Dans un SGBDR avec deux tables, faire de requêtes de jointure serait très fastidieux.
- ☞ Une solution est de fusionner les tables : C'est ce qu'on appelle la dénormalisation.

Exemple de MCD



Exemple de MCD

- ☞ Une commande est passée pas un client et un client peut passer plusieurs commandes ;
- ☞ Une commande peut contenir plusieurs articles et un article peut être dans plusieurs commandes ;
- ☞ Un article appartient à une seule catégorie et une catégorie peut avoir plusieurs articles ;

Règles de la dénormalisation

Quelques règles qui vont vous permettre de produire des documents JSON qui répondront à votre demande, tout en minimisant les problèmes de jointures et d'incohérences

- ☞ Des données fréquemment interrogées conjointement
  - ☛ Regarder les entités les plus sollicitées (Exemple : Commande)
- ☞ Toutes les données d'une entité sont indépendantes
  - ☛ On peut avoir une imbrication d'objets mais sous forme de liste

Règles de la dénormalisation

Quelques règles qui vont vous permettre de produire des documents JSON qui répondront à votre demande, tout en minimisant les problèmes de jointures et d'incohérences

- Une association avec des relations 1-n des deux côtés
  - Cette question est plus problématique ;
  - Il faut transférer les informations de l'association dans un document et les proprietés dans l'autre entité ;
  - Exemple : La quantité sera envoyée dans commande plus l'proprietés du produit

Exemple de JSON correspondant au MCD