


Le NoSQL et MongoDB

Le Big Data – défis du siècle

Le Big Data – Défis du siècle : plan

- Introduction
- Le modèle des 3V
- Le mode de stockage
- Cas d'utilisations



Le Big Data – Introduction

- Pourquoi parle-t-on de Big Data ?

Aujourd'hui, la quantité de données, la vitesse de génération ainsi que la variété dont elles font preuve, a marqué la limite d'utilisation des systèmes de gestion traditionnels.

« Big » car les données générées ces deux dernières années représentent près de 90% de toutes les données créées par l'homme depuis la nuit des temps*.

*Données 2015



- Pourquoi les systèmes traditionnels ont-ils atteint leurs limites ?

Les systèmes utilisés sont devenus peu performants pour le traitement de ces données. En effet, celles-ci, en plus d'être nombreuses, sont semi-structurées alors que les systèmes traditionnels ont besoin de données structurées pour être performants.

- Quelles technologies ont résolu une partie de ces défis ?


Le NoSQL est une des techniques qui a été développée par de grands leaders du web (Google, Facebook, Amazon, ...) afin de répondre à leurs besoins internes.



Le Big Data – Le modèle des « 3V »

Le Big Data a trois caractéristiques : la vitesse, le volume et la variété.

- La vitesse est la vitesse de création et de modification des données.
- Le volume est la quantité de données à traiter, stocker et retrouver.
- La variété des données provient de leur structure qui n'est pas toujours semblable.



Le Big Data – Le mode de stockage

Les données du Big Data peuvent être stockées dans différentes structures comme le Cloud, les data centers ou dans un système de fichiers distribués.

- Un système de fichiers distribués applique le principe de « Data Locality ». Ce principe signifie que les données sont stockées là où elles seront traitées.
Ce système privilégie également le stockage des données sur un très grand nombre de petites machines au lieu d'une grosse.



Le Big Data – Utilisation

Le Big Data est une source d'information sans précédent dans l'histoire de l'humanité.

Il est actuellement utilisé dans :

- le domaine scientifique afin de traiter les données des grosses expérimentations.
- le domaine commercial afin d'aider les décisions d'entreprise en fonction des profils clients analysés, de leurs habitudes de consommation, etc.
- le domaine politique pour influencer l'électorat, orienter un débat, ...

Le NoSQL – L'avenir des bases de données ?


Le NoSQL : plan

- Introduction / Défis
- Théorème CAP
- Utilisation
- Système de gestion de données
- Schéma et Contenu
- Type de moteur NoSQL



Le NoSQL – Introduction

- La technologie NoSQL a été créée dans le but de répondre au besoin du Big Data.
- Le NoSQL est un système distribué de gestion de base de données.
- Le NoSQL privilégie une montée en charge horizontale permettant de répondre plus facilement à l'augmentation des quantités de données. Pour parvenir à suivre la vitesse de génération des données, les moteurs NoSQL ont généralement abandonné le modèle relationnel et son schéma structuré ce qui leur permet une plus grande flexibilité.



NoSQL - Les défis

Les bases de données NoSQL ayant été créés pour répondre au besoin du Big Data, il est évident que ces défis sont ceux du Big Data :

- Capture des données
Gestion des données semi-structurée fournies par plusieurs types de support.
- Stockage des données
- Recherche rapide
- Partage
- Transfert
- Analyse
- Présentation



Le NoSQL – Le théorème CAP

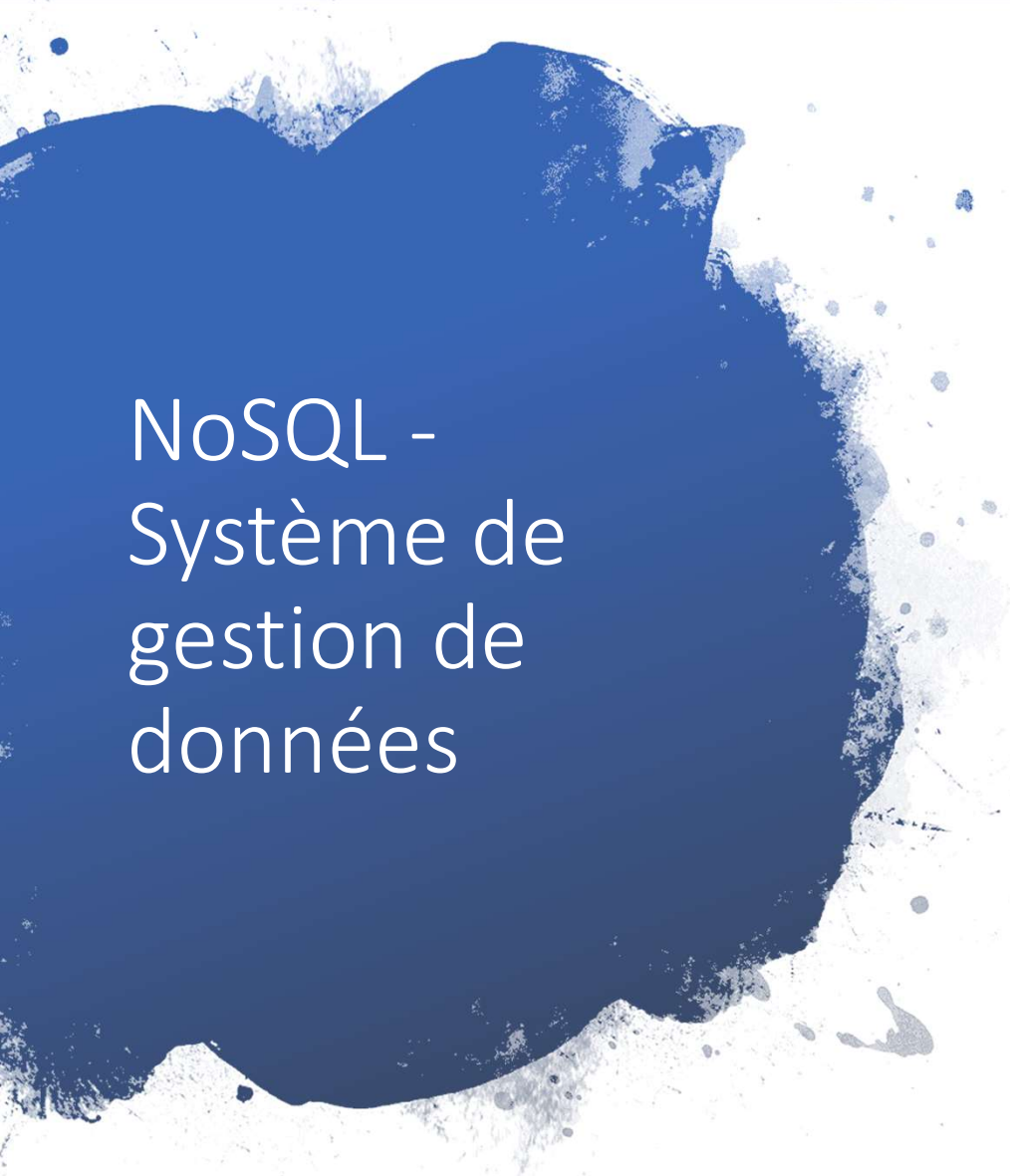
- CAP (Consistency, Availability and Partition tolerance) pour Consistance, Disponibilité et tolérance à la Partition.
- Selon ce théorème, les bases de données distribuées ne sont capables de garantir que deux de ces trois points. Ainsi, en privilégiant la disponibilité et la tolérance à la partition, les bases de données NoSQL perdent la capacité de garantir une certaine consistance.
- Pour palier ce problème, les bases de données NoSQL garantissent une cohérence finale des données à travers tous les nœuds.



NoSQL- Utilisation

Question: Que peut-on faire avec ces données ?

- Le NoSQL a été créé dans le but d'analyser les données. Ainsi, nous pouvons :
 - analyser le comportement d'achat des clients ;
 - analyser les produits les plus vendus ;
 - prendre des décisions en fonction de ces analyses ;
 - produire des factures, etc.
- La différence, entre le NoSQL et les bases de données traditionnelles, est que la difficulté de traitement est laissée au code clients. Ce qui permet une plus grande flexibilité, vitesse de traitement et accessibilité.




NoSQL - Système de gestion de données

La gestion d'une base de données est laissée à un logiciel spécialisé appelé: *système de gestion de base de données (SGBD)*

- Les SGBD s'occupent de :
 - L'organisation des données.
Organise les données en entités stockées sur le disque de manière à garantir un accès rapide à celles-ci.
 - La gestion des données.
Garantir la cohérence finale des données stockées.
 - Assurer l'accès aux données.
Accès aux données stockées par plusieurs clients et logiciels de traitement simultané.



- La protection contre les incidents
Les moteurs NoSQL garantissent cette protection par la réplication des données.
- Le partitionnement des données
Les bases de données NoSQL permettent un partitionnement des données automatiques permettant un gain en vitesse de traitement.
- La réplication des données
La majorité des moteurs NoSQL permettent une réplication automatique des données sur différents serveurs.



NoSQL - Schéma et contenu

- La plupart des moteurs NoSQL sont dits « schema-free » ou « schema-less ». Cela signifie que l'existence d'une structure de données prédéfinie n'est pas requise pour l'insertion de données.
- La non-existence d'un schéma permet au NoSQL d'être plus adapté aux méthodes de développement actuel.
- Elle permet également d'optimiser l'espace en enregistrant que ce qui est nécessaire.



NoSQL - La redondance interne

- Dans les bases de données NoSQL, la redondance d'information est utilisée afin de permettre de meilleures performances dans les opérations de lecture de données en évitant de nombreuses requêtes parfois complexes.
- Le prix de l'espace de stockage par rapport au prix d'opération processeur en est la raison.



NoSQL - Les structures physiques

Les données étant stockées sur le disque dur, y accéder peut prendre du temps. Comment le diminuer ?

- En NoSQL, les données sont généralement réparties sur plusieurs serveurs qui exécutent une partie de la requête. Cela permet de diminuer le temps d'accès aux données.
=> Mécanisme de répartition.
- Le NoSQL intègre également un mécanisme emprunté au monde relationnel : l'indexation
L'indexation est un « catalogue » permettant de retrouver plus rapidement une information.



Le NoSQL – Les types de moteurs

Il existe différents types de moteurs NoSQL :

- orientée colonne
Ces moteurs sont généralement assez proches conceptuellement des bases de données traditionnelles mais permettent de gérer de plus grosses quantités de données.
- orienté document
Ces moteurs sont destinés à des besoins plus « modestes » et s'adaptent plus au paradigme de programmation orienté Objet.
- orientée clé-valeur
Ces moteurs sont des dépôts de données.
- orienté graph
Ces moteurs sont destinés à être utilisés dans le cadre des relations complexes entre entités.

MongoDB – Un moteur NoSQL

MongoDB – Un moteur NoSQL : Plan

- Introduction
- Les données
- Les concepts
- La réplication
- La répartition
- Le DDL
- Le DML

MongoDB - Introduction



MongoDB – Introduction

- MongoDB est une base de données NoSQL orienté document. Ce moteur a été développé initialement pour des besoins plus modestes. Cependant, il est capable de réaliser une montée en charge horizontale.
- Ce moteur garantit la disponibilité et la tolérance au partitionnement ainsi qu'une cohérence finale des données distribuées.

MongoDB – Les données



MongoDB – Les données

- Plan
 - Première approche
 - Les collections
 - Les documents
 - Utilisation
 - Première conclusion
 - Exercices

MongoDB - Première approche des données

Exemple de données à traiter

Commande N° : 30188		Date : 2/1/2009		
Numéro client	B512			
Nom	GILLET			
Adresse	14, r. de l'Eté			
Localité	Toulouse			
N° PRODUIT	LIBELLE PRODUIT	PRIX	QUANTITE	SOUS-TOTAL
CS464	CHEV. SAPIN 400x6x4	220	180	39600
PA45	POINTE ACIER 45 (20K)	105	22	2310
PA60	POINTE ACIER 60 (10K)	95	70	6650
PH222	PL. HETRE 200x20x2	230	92	21160
TOTAL COMMANDE				69720

MongoDB - Première approche des données

- À partir de cet exemple, il nous est possible de construire une base de données orientée document.
- En regardant ce bon de commande, on remarque que certaines données ont un lien. Il suffit de regrouper les données reliées au sein d'une même collection.
- Ainsi, cet exemple nous demandera d'avoir plusieurs collections:
 - une collection « clients »
 - une collection « produits »

MongoDB – Première approche des données

- Les collections:
 - une collection est l'équivalent des tables dans les bases de données traditionnelles dites relationnelles, elles sont donc composées de données regroupées en « document ».
 - Ces collections n'obligent pas les documents à posséder une structure commune.
 - Il est cependant conseillé d'adopter une certaine rigueur dans la structure.

MongoDB – Première approche des données

- Les documents
 - Un document est l'équivalent d'un enregistrement dans les bases de données traditionnelles dites relationnelles, ils composent donc l'unité de stockage.

```
{
  "_id": ObjectId('56d3fc4a9290688a3b724486'),
  "IDENTIFIANT": {
    "NOM": "Exemple",
    "PRENOM": "Document",
    "DATE_NAISSANCE": ISODate('2000-12-31T23:00:00.000Z')
  },
  "ADRESSE": {
    "RUE": "exempleRue",
    "LOCALITE": "exempleLocalite",
    "CODE_POSTALE": 1000,
    "NUM": 10
  },
  "DROIT": [
    "modo",
    "user",
    "redacteur"
  ]
}
```

MongoDB – Première approche des données

	_id	NOM	ADRESSE	LOCALITE	CAT	COMPTE	COMMANDES
1	B332	MONTI	23,a. Dumont	Geneve	B2	0	
2	B112	HANSENNEA	23,a. Dumont	Poitiers	C1	1250	
3	F010	TOUSSAINT	23,a. Dumont	Poitiers	C1	0	
4	S127	VANDERKA	23,a. Dumont	Namur	C1	-4580	[1 element]
5	B062	GOFFIN	23,a. Dumont	Namur	B2	-3200	
6	B512	GILLET	23,a. Dumont	Toulouse	B1	-8700	[1 element]
7	K111	VANBIST	23,a. Dumont	Lille	B1	720	[1 element]
8	C123	MERCIER	23,a. Dumont	Namur	C1	-2300	
9	C003	AVRON	23,a. Dumont	Toulouse	B1	-1750	
10	K729	NEUMAN	23,a. Dumont	Toulouse		0	
11	C400	FERARD	23,a. Dumont	Poitiers	B2	350	[3 elements]
12	F011	PONCELET	23,a. Dumont	Toulouse	B2	0	[1 element]
13	L422	FRANCK	23,a. Dumont	Namur	C1	0	
14	D063	MERCIER	23,a. Dumont	Toulouse		-2250	
15	F400	JACOB	23,a. Dumont	Bruxelles	C2	0	
16	S712	GUILLAUME	23,a. Dumont	Paris	B1	0	
17	K112	EXO	4/105, r. Ch...	Namur		125	
18	F401	1234	23,a. Dumont	Bruxelles	C2	2523	
19	B632	AROUNDI	23,a. Dumont	Geneve	B2	0.95	

MongoDB – Première approche des données

	id	CHAMP	NOM	ADRESSE	LOCALITE	CAT	COMPTE	COMMANDES
1	B332		MONTI	23,a. Dumont	Geneve	B2	0	DOCUMENT
2	B112		HANSENNEA	23,a. Dumont	Poitiers	C1	1250	Champ inexistant
3	F010		TOUSSAINT	23,a. Dumont	Poitiers	C1	0	
4	S127		VANDERKA	23,a. Dumont	Namur	C1	-4580	
5	B062		GOFFIN	23,a. Dumont	Namur	B2	-3200	[1 element]
6	B512		GILLET	23,a. Dumont	Toulouse	B1	-8700	[1 element]
7	K111		VANBIST	23,a. Dumont	Lille	B1	720	[1 element]
8	C123		MERCIER	23,a. Dumont	Namur	C1	-2300	
9	C003		AVRON	23,a. Dumont	Toulouse	B1	-1750	
10	K729		NEUMAN	23,a. Dumont	Toulouse		0	
11	C400		FERARD	23,a. Dumont	Poitiers	B2	350	[3 elements]
12	F011		PONCELET	23,a. Dumont	Toulouse	B2	0	[1 element]
13	L422		FRANCK	23,a. Dumont	Namur	C1	0	
14	D063		MERCIER	23,a. Dumont	Toulouse		-2250	COLLECTION
15	F400		JACOB	23,a. Dumont	Bruxelles	C2	0	
16	S712		GUILLAUME	23,a. Dumont	Paris	B1	0	
17	K112		EXO	4/105, r. Ch...	Namur		125	
18	F401		1234	23,a. Dumont	Bruxelles	C2	2523	
19	B632		AROUNDI	23,a. Dumont	Geneve	B2	0,95	

MongoDB - Première approche des données

- Exercices

- Dans le cadre de la collection « clients », quels en serait les composants ?
- Dans le cadre de la collection « produits », quels en serait les composants ?

MongoDB - Utilisation

- Le rôle d'une base de données est la mise à disposition et le stockage des données nécessaires au fonctionnement d'une société.
- On peut donc les interroger de manière plus ou moins aisée
 - Quel est le numéro, le nom et le montant du compte de tous les clients habitant à Toulouse ?
 - `db.clients.find({LOCALITE: 'Toulouse'}, {_id: true, NOM: true, COMPTE: true})`
 - Quelles sont les commandes réalisées par les clients de Toulouse
 - `db.clients.find({LOCALITE: 'Toulouse'}, {_id: false, COMMANDES: true})`

MongoDB - Première conclusion

- Une base de données MongoDB est constituée d'un ensemble de collections
- Chaque collection est composée de documents
- Chaque document constitue un ensemble de données liées entre elles
- Chaque document est unique et identifié par un identifiant

MongoDB - Première conclusion

- Les données calculables ne sont pas stockées en base de données
- On conserve les données relatives au sein d'un même document.
 - Redondance d'information, mais rapidité de traitement.
- Les bases de données MongoDB s'adaptent aux besoins actuels d'une société. Elles s'adaptent donc plus facilement au modèle de développement « Agile » de par sa flexibilité.
- Ces bases de données demandent une analyse moins rigoureuse puisqu'il est possible d'ajouter ou de retirer des attributs à un ou plusieurs documents.

MongoDB – Les concepts



MongoDB – Les concepts

- Plan
 - Illustration
 - Répartition des données
 - Réplication des données
 - Une base de données distribuée
 - Valeur NULL
 - Identifiant
 - Curseur

MongoDB – Illustration

Visualisation sous
forme de table

	_id	NOM	ADRESSE	LOCALITE	CAT	COMPTE	COMMANDES
1	B332	MONTI	23,a. Dumont	Geneve	B2	0	
2	B112	HANSENNEA	23,a. Dumont	Poitiers	C1	1250	
3	F010	TOUSSAINT	23,a. Dumont	Poitiers	C1	0	
4	S127	VANDERKA	23,a. Dumont	Namur	C1	-4580	[1 element]
5	B062	GOFFIN	23,a. Dumont	Namur	B2	-3200	
6	B512	GILLET	23,a. Dumont	Toulouse	B1	-8700	[1 element]
7	K111	VANBIST	23,a. Dumont	Lille	B1	720	[1 element]
8	C123	MERCIER	23,a. Dumont	Namur	C1	-2300	
9	C003	AVRON	23,a. Dumont	Toulouse	B1	-1750	
10	K729	NEUMAN	23,a. Dumont	Toulouse		0	
11	C400	FERARD	23,a. Dumont	Poitiers	B2	350	[3 elements]
12	F011	PONCELET	23,a. Dumont	Toulouse	B2	0	[1 element]
13	L422	FRANCK	23,a. Dumont	Namur	C1	0	
14	D063	MERCIER	23,a. Dumont	Toulouse		-2250	
15	F400	JACOB	23,a. Dumont	Bruxelles	C2	0	
16	S712	GUILLAUME	23,a. Dumont	Paris	B1	0	
17	K112	EXO	4/105, r. Ch...	Namur		125	
18	F401	1234	23,a. Dumont	Bruxelles	C2	2523	
19	B632	AROUNDI	23,a. Dumont	Geneve	B2	0.95	

MongoDB – Illustration

Visualisation sous
forme de document

```
{
  "_id": ObjectId('52873b364038253faa4bbc0e'),
  "student_id": "STU002",
  "sem": "sem1",
  "english": "A",
  "maths": "A+",
  "science": "A",
  "dateTime": ISODate('1992-07-18T22:00:00.000Z')
}
```

```
{
  "_id": ObjectId('52873b5d4038253faa4bbc0f'),
  "student_id": "STU001",
  "sem": "sem1",
  "english": "A+",
  "maths": "A+",
  "science": "A",
  "dateTime": ISODate('1991-07-18T22:00:00.000Z')
}
```

MongoDB – Illustration

Visualisation sous forme de liste

```
{ "_id": "ObjectId(52873b364038253faa4bbc0e)", "student_id": "STU002", "sem": "sem1", "english": "A", "maths": "A+", "science": "A", "dateTime": "ISODate(1992-07-18T22:00:00.000Z)" }
{ "_id": "ObjectId(52873b5d4038253faa4bbc0f)", "student_id": "STU001", "sem": "sem1", "english": "A+", "maths": "A+", "science": "A", "dateTime": "ISODate(1991-07-18T22:00:00.000Z)" }
{ "_id": "ObjectId(52873b7e4038253faa4bbc10)", "student_id": "STU003", "sem": "sem1", "english": "A+", "maths": "A", "science": "A+", "dateTime": "ISODate(1991-07-18T22:00:00.000Z)" }
{ "_id": "ObjectId(56b320cc67f06e94da74caf0)", "student_id": "STUD004", "sem": "sem1", "english": "B-", "maths": "A+", "science": "C" }
{ "_id": "STU56bca724a02e2340fd50493e", "name": "Test", "prenom": "id" }
{ "_id": "STU5" }
```

MongoDB – Répartition de données



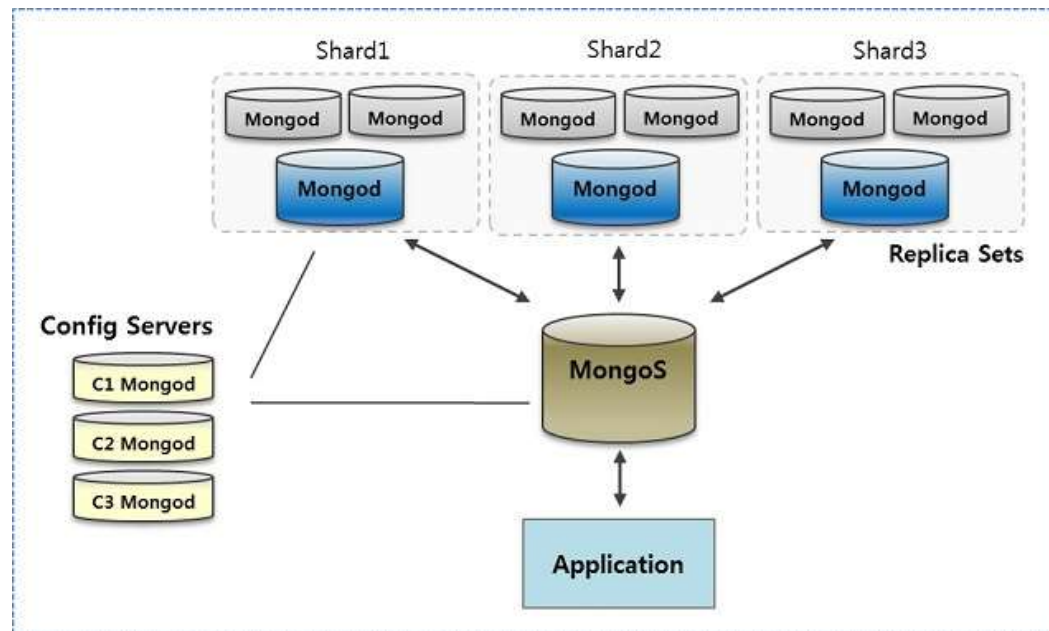
- La répartition est un principe fondamental. Il permet de répartir une collection sur plusieurs serveurs en les répartissant selon un ensemble de clés-valeurs défini par le gestionnaire de base de données.
- Cette répartition possède une architecture maître-esclave. Cela lui permet de répartir la charge de travail sur les esclaves qui n'exécutent que des requêtes affectant leurs contenus.

MongoDB – Réplication de données

- La réplication est un principe fondamental. Il permet d'effectuer une copie de la base de données contenue dans un serveur. Cela permet une haute disponibilité.
- Les serveurs effectuant une réplication sont regroupés au sein d'un « ensemble de réplication » (replica set) qui possède une architecture maître-esclave avec basculement automatique du maître en cas de défaillance de ce dernier.

MongoDB – Base de données distribuée

- Organisation d'une base de données distribuée selon l'architecture maître-esclave.



MongoDB – La valeur « NULL »

Dans les moteurs NoSQL, les données ne doivent pas respecter un schéma préétabli. Dès lors, l'absence d'un attribut, lors d'opérations de lecture, est comprise comme l'association de cet attribut par une valeur « NULL », mais qui n'est pas stocké en mémoire.

MongoDB –La valeur « NULL »

- Exemple d'utilisation de la valeur « NULL »
 - `db.<collection>.find(<champ>: null)`

Renvoie les documents contenant le champ <champ> et dont la valeur est null, et les documents dont le champ <champ> n'existe pas.
 - `db.<collection>.find(<champ>: {$exists: false})`

Renvoie que les documents ne contenant pas le champ <champ>.
 - `db.<collection>.find($and: [{<champ>: {$exist: true}}, {<champ>: null}])`

Renvoie les documents possédant le champ <champ> ayant une valeur « null »

MongoDB - Identifiant

- Le rôle de l'identifiant dans une base de données est d'identifier de manière unique une entité.
- Ainsi l'identifiant de MongoDB est un nombre de 12 octets constitué d' :
 - un numéro attribué au serveur (3 octets)
Ce numéro est utilisé lors de la répartition des données. Permet de connaître le serveur sur lequel est stocké le document.
 - une date (permettant la gestion de versions des données) (4 octets)
Une date en seconde depuis le 1er janvier 1970 (Création de Linux)
 - un numéro unique généré pour la collection (3 octets)
Compteur commençant à une valeur aléatoire.
 - le « process id » du démon Mongod (2 octets)

MongoDB - Identifiant

- L'identifiant définit une contrainte d'unicité du document
- Il est généré automatiquement par le moteur ou défini par le code client
- Un identifiant peut être constitué de plusieurs attributs
- L'identifiant d'un autre document peut être inséré dans un document. Ceci permettra de réaliser une requête au serveur sur base de la référence récupérée
 - Cette « jointure » est réalisée par le code client

MongoDB – Le curseur

- Dans MongoDB, les requêtes renvoient un objet appelé « cursor ». Cet objet contient quelques méthodes permettant de le traiter. Ainsi, en récupérant cet objet dans une variable, nous pouvons l'utiliser pour effectuer des requêtes sur base d'autres requêtes.
- Une variable est un nom symbolique permettant de stocker un objet et ainsi de le réutiliser ultérieurement.
 - Pour déclarer une variable, « var <nom_symbolique> »
- Exemple
 - var produits = <requête>

MongoDB – Le DDL



MongoDB – Le DDL

- Plan
 - Introduction
 - Création de collections
 - Suppression de collections
 - Insertion de documents
 - Suppression de documents
 - Création d'index
 - Suppression d'index
 - Exercices

Le DDL - Introduction

- Le DDL (Data Description Language) est un langage permettant de définir les structures d'une collection, d'un document.
- MongoDB a été développé dans le paradigme « Objet »
 - Il y a un objet parent global: « db ». Celui-ci possède des méthodes et des objets accessibles par l'opérateur « . ». Ainsi, « db.collection » fait appel à l'objet collection de l'objet db. (Important pour la suite)
- Dans MongoDB, le DDL est composé de plusieurs méthodes:
 - la création de collection
 - l'insertion de document
 - la suppression de document
 - la suppression de collection
 - la création et suppression des index

db.createCollection()

Le DDL – Création de collections

- Dans MongoDB, la création d'une base de données se fait par son utilisation et l'insertion de données dans celle-ci.
 - use <nom_base_de_données>
- Pour créer une collection dans ce moteur NoSQL, il y a deux possibilités:
 - en utilisant la méthode « createCollection() » de l'objet « db » qui permet de définir différentes options de contrôle sur les documents
 - MongoDB crée automatiquement la collection utilisée lors d'une insertion si elle n'existe pas au sein de la base de données. Les options sont celles par défaut.
- Les collections n'ayant pas de schéma préétabli, seuls leurs noms sont nécessaires à leurs créations.

Le DDL – Suppression de collections

- La suppression d'une collection se fait facilement au vu de l'inexistence de contrainte de schéma.
- Elle se fait par la méthode « drop() » de l'objet collection dont le nom est la collection à supprimer.

Le DDL – L'insertion de documents



- L'insertion de document se fait grâce à la méthode « insert() » d'une collection.
- L'insertion d'un document se fait toujours entre « { } »
- L'insertion de plusieurs documents peut se faire de deux manières différentes:
 - en utilisant la méthode « insert([{}, {}, {}, ...]) »
 - en réalisant l'importation de données d'un fichier JSON

Le DDL – L'insertion de documents

- L'insertion de plusieurs documents par la méthode « insert() »

```
db.collection.insert([
    {<document1>},
    {<document2>},
    ...
])
```
- L'insertion de plusieurs documents par Console
 - Dans le fichier binaire du dossier d'installation, il existe un utilitaire : mongoimport.
 - Utilisation:

```
/ « chemin_accès »/mongoimport --dbpath  
« chemin_accès_db » -c « collection » <  
« chemin_fichier json »
```

Le DDL – L'insertion de documents

Import documents

local/exercice

Source

☒ Import documents from file
☐ Import documents from textbox

Name of destination-collection*

NOM de la collection

Import mode

☐ Replace documents in collection by import (disabled when unknown collection)
☒ Append documents in collection with import

Ignore duplicate keys

☒

GLISSER ou CHARGER votre fichier JSON

Drop your file here or use the 'choose file' button

Choose File

No file chosen

The import process may take some time to complete, with a larger number of records or a memory size greater than 100 MB of the collection we recommend mongoexport and mongoimport through the console.

Cancel

Import

Le DDL – L'insertion de documents

- Les types de données pouvant être insérées sont de type JavaScript.
- Exemple de type:
 - entier, double pour les nombres
 - chaîne de caractère (string)
 - Date (\$date: {'aaaa-mm-jj'})
 - Fichier binaire
 - ...

Le DDL – La suppression de documents

- Pour réaliser la suppression d'un document dans une collection, il faut exécuter la méthode « remove() »
- Le comportement par défaut de cette méthode est la suppression de tous les documents répondant au critère de recherche.
- Exemple:
 - nous souhaitons supprimer le document contenant un client dont le nom est FERARD.

```
db.clients.remove(  
    {NOM: 'FERARD'},  
    {justOne: true}  
)
```

Le DDL – La création d'index

- Dans le moteur NoSQL MongoDB, il est possible de créer plusieurs index différents au sein d'un document. Cela permet une amélioration des performances lors de lectures de données.
- Pour créer un index en MongoDB, il faut utiliser la méthode « `createIndex()` » de la collection.
 - Le nombre situé après le champ à indexer est l'ordre d'indexation
 - > 0 : ordre croissant
 - < 0 : ordre décroissant
- Exemple : `db.clients.createIndex({NOM: 1})`

Le DDL – La suppression d'index

- La suppression d'un index se fait grâce à la méthode « `dropIndex()` » de la collection
- La structure de cette méthode est la même que celle permettant sa création

Le DDL - Exercice

Réaliser la base de données permettant le stockage et le traitement d'un client et ses commandes de produits.

- Le nom de la base de données est libre.
- Cette base de données doit contenir deux collections:

- clients

Un client est composé d'un NOM, d'une ADRESSE, d'une LOCALITE d'une CAT, d'un COMPTE et d'un ensemble de COMMANDES.

Une COMMANDES est généralement composée d'un NCOM, d'une DATECOM et d'un DETAIL.

Un DETAIL est constitué d'un PRODUIT, et d'une QCOM.

- produits

Un produit est défini par un NPROD, un LIBELLE, un PRIX, une DATELASTMODIF et une QSTOCK