



MongoDB, administration Pour NoSQL

Dr. Ali SKAF



Les outils et commandes MongoDB





Les outils et commandes MongoDB

- Inventaire
 - mongod (le moteur de base)
 - mongo (le shell javascript)
 - mongos (le contrôleur de Sharding)
 - Les outils d'import/export
 - mongoimport, mongoexport, mongodump, mongorestore, bsondump
 - mongofiles (l'utilitaire GridFS)
 - mongostat (visualisation des stats d'une instance mongoDB)
 - mongosniff (le tcpdump de mongo)
 - mongotop, mongoperf



Bases de données MongoDB





Les bases de données

- Création / Suppression
 - Création de bases de données
 - Automatique
 - # mongo <db_name>
 - > use <db_name>
 - Suppression de bases de données
 - # mongo
 - > use <db_name;>
 - > db.runCommand({dropDatabase: 1});
 - > db.dropDatabase();



Les bases de données

- Stockage des données physiques d'une base
 - Limite en environnement 32 bits
 - Ne permet pas de stocker plus de 2.5Gb de données
 - Déconseillé pour de la production
 - Le répertoire de stockage utilise la variable --dbPath
 - En production
 - Utiliser la version 64 bits
 - Le répertoire de stockage utilise la variable --dbPath



Les bases de données

- Stockage des données physiques d'une base
 - Caractéristiques
 - Pour chaque base
 - initialisation de deux fichiers
 - `<db_name>.0` et `<db_name>.1`
 - La taille de chaque fichier est le double de la taille du précédent (64, 128, 256, etc)
 - Taille maximale d'un fichier = 2Gb
 - Le fichier `<nom_fichier>.ns` stocke les namespaces de la base (16 Mb par défaut)
 - modifiable via l'option `-nssize`, avec pour taille maximum 2Gb



Les espaces de noms





Les espaces de noms

- Caractéristiques
 - Chaque collection et chaque indexe compte comme un espace de nom
 - Concrètement: concaténation du nom de la base et celui de la collection
 - Limites
 - Par défaut: 24000 espaces de noms
 - Possibilité d'augmenter cette taille avec le paramètre -nssize
 - Possibilité de redimensionner une base existante
 1. On redémarre mongodb avec l'option -nssize
 2. On exécute la commande `db.repairDatabase()`



Travailler avec les documents





Travailler avec les documents

Les collections



Les collections

- Une Collection c'est quoi ?
 - Equivalent de la table en relationnel
 - Deux modes de création
 - Automatiquement lors d'une insertion de document (tuple)
 - En exécutant la commande `db.createCollection('<nom_collection;>')`



Les collections

- Les opérations sur les Collections

- Créer

- > db.createCollection('gens;')

- Lister

- > show collections;
 - > db.getCollections();

- Insérer un document

- > db.<collection_name>.insert({ var1: "valeur", var2: "valeur", var3: "valeur", });

- Supprimer

- > db.<nom_collection>.drop();



Les documents





Travailles avec les documents

Les documents



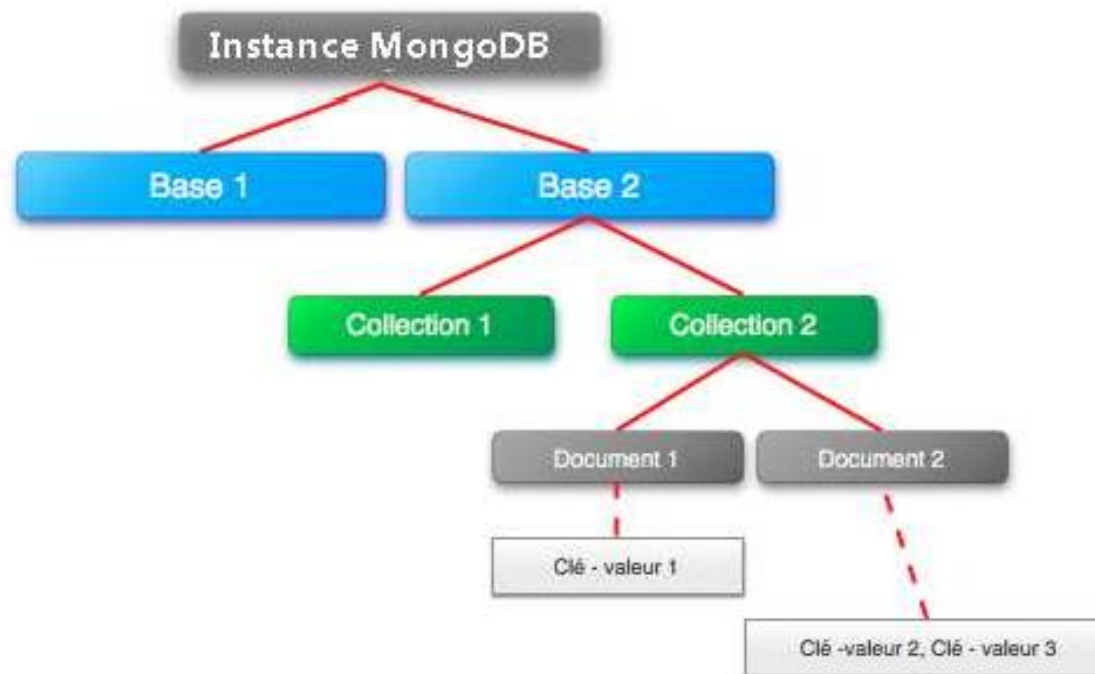
Les documents

- Un document c'est quoi ?
 - Equivalent d'un enregistrement (tuple) en relationnel
 - Deux modes d'insertion
 - En utilisant la commande `db.<nom_collection>.insert()` du schell
 - En utilisant la commande `db.<nom_collection>.save ()` du schell
 - En utilisant un driver: PHP, Ruby, Java, Python
 - Chaque document crée contient le champ « `_id` »
 - Généré automatiquement
 - Type primary key
 - Indéxé



Les documents

- Rappel de la hiérarchie des objets





Les documents

- Le format JSON/BSON
 - MongoDB stocke les documents au format BSON (Binary JSON)
 - BSON est la représentation binaire des objets JSON (JavaScript Object Notation)



Les documents

- Le format JSON c'est quoi ?
 - JSON (JavaScript Object Notation)
 - Permet de représenter de l'information structurée
 - Format de données textuelle
 - Utilise une notation JavaScript
 - Décrit par la RFC 4627
 - Un document JSON, ne comprend que deux éléments structurels :
 - des ensembles de paires nom / valeur ;
 - des listes ordonnées de valeurs.



Les documents

- Le format JSON

- Exemple

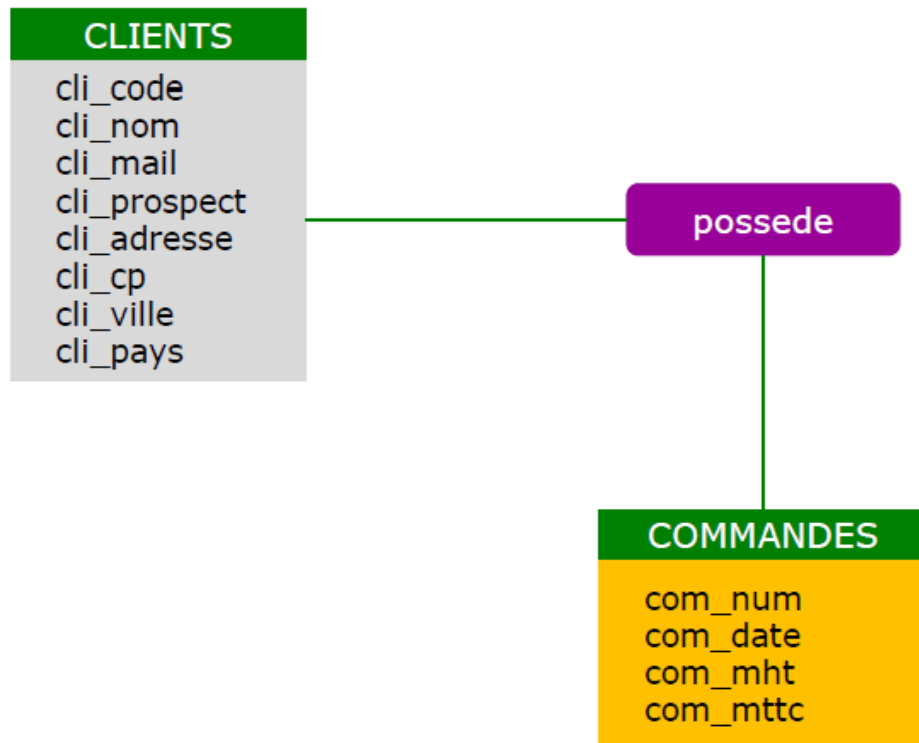
```
{  
  "pospect" : "ALFKI",  
  "nom" : "ORSYS",  
  "adresse"  
    : {  
      {  
        "numero" : "77",  
        "codepostal" : "31000",  
        "ville" : "Toulouse"  
      }  
    }  
}
```



Les documents

■ Exemple pratique d'utilisation

Schema relationnel



Document NoSQL

```
{  
  "cli_code": "ALFKI",  
  "cli_nom": "ORSYS",  
  "cli_mail": "contact@orsysfr",  
  "cli_prospect": false,  
  "commandes": {  
    "com_num": 1077,  
    "com_date": "10/12/2007",  
    "com_mht": 190.00  
  }  
}
```



Les documents

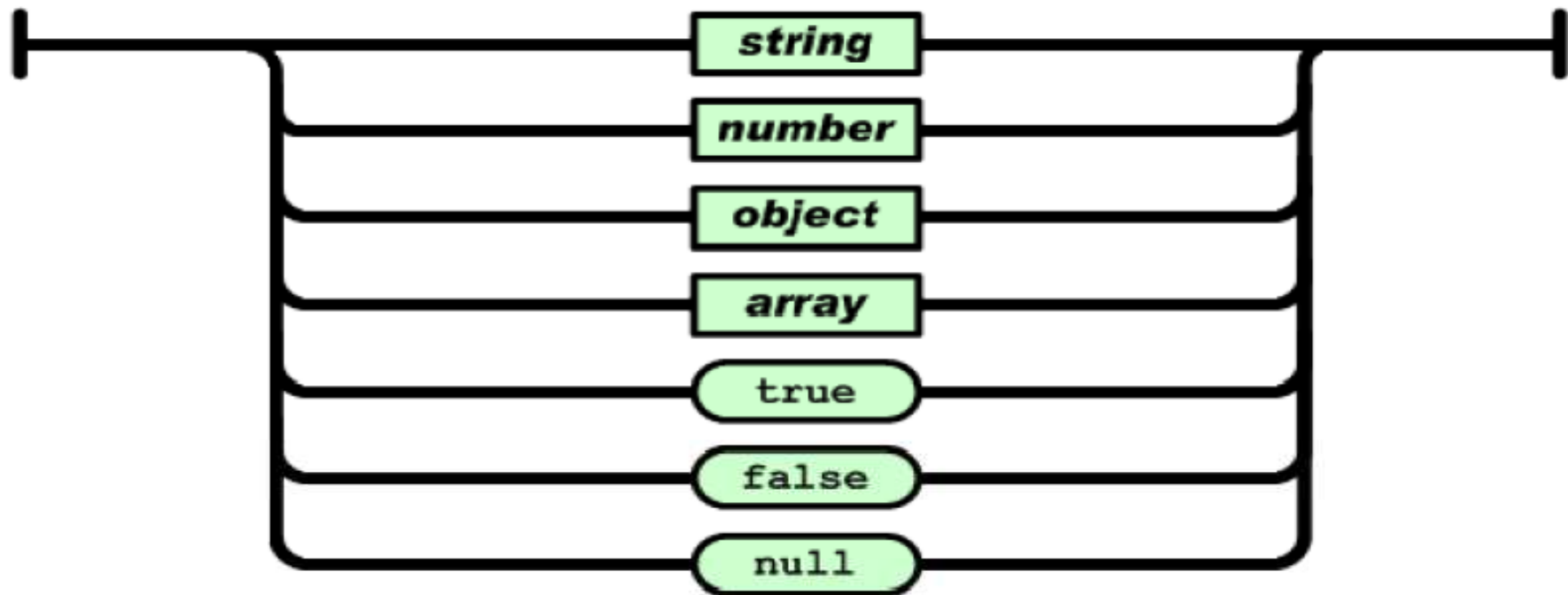
- Avantage du format JSON
 - Format abstrait pour une représentation simplifiée dans les différents langages
 - Indépendant du langage de programmation
 - Simple et complet pour la représentation des objets
 - Utilise des types de données connus et simples à décrire
 - Une bonne lisibilité de la syntaxe
 - Temps de traitement très proche de celui des fichiers XML
 - Moins volumineux en terme de stockage
 - Pour JavaScript un document JSON représente un objet



Les documents

- Types disponibles en JSON
 - issus de json.org

value





Les documents

- Le format BSON c'est quoi ?
 - BSON est la représentation binaire des objets JSON
 - Ajoute des améliorations et des capacités supplémentaires
 - expressions régulières
 - Bytes Arrays
 - Date et TimeStamps
 - stockage de blocs de code ou de fonctions JavaScript



Les documents

- Les types de données
 - Une base MongoDB est constituée de Collection(s)
 - Une collection est constituée de document(s) regroupant des paires clef/valeur dont la clef est une string et la valeur de type :
 - des objets ;
 - des tableaux ;
 - des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.



Performance et indexation





Performance et indexation

- Les indexes sous MongoDB
 - Pourquoi faire ?
 - Améliorer les temps d'exécution des requêtes
 - Les différents types d'indexes
 - Index sur un seul champ
 - Index unique
 - Index composé, Sparse et Hashed
 - Indexation par défaut
 - Le champ `_id` est indexé par défaut (impossible de le supprimer)



Performance et indexation

- Index sur un seul champ
 - Deux modes
 - Ordre ascendant (1 pour ASC)
 - Ordre descendant (-1 pour DESC)
 - Poser un index (ascendant et descendant)
 - `db.<collection>.ensureIndex({<nom_champ>:1});`
 - `db.<collection>.ensureIndex({<nom_champ>:-1});`



Performance et indexation

- Index composés
 - Deux modes
 - Ordre ascendant (1 pour ASC)
 - Ordre descendant (-1 pour DESC)
 - Poser un index (ascendant et descendant)
 - `db.<collection>.ensureIndex({<nom_champ1>:1, <nom_champ2>:1});`
 - `db.<collection>.ensureIndex({<nom_champ1>:-1, <nom_champ2>:-1});`



Performance et indexation

- Les index unique
 - Deux modes
 - Ordre ascendant (1 pour ASC)
 - Ordre descendant (-1 pour DESC)
 - Poser un index unique (ascendant et descendant)
 - `db.<collection>.ensureIndex({<nom_champ1>:1, {unique: true}});`
 - `db.<collection>.ensureIndex({<nom_champ1>:-1, {unique: true}});`

Performance et indexation

- Création des index en arrière plan
 - Par défaut construction de l'index en premier plan
 - Possibilité de construire un index en arrière plan
 - `db.collection.ensureIndex({ a: 1 }, { background: true })`

Performance et indexation

- Visualisation des index
 - Index d'une collection
 - `db.<collection>.getIndexes()`
 - Index d'une base de données
 - `db.system.indexes.find()`

Performance et indexation

- Reconstruction d'index
 - Utilisation de la méthode `db.collection.reIndex()`
- Suppression d'index
 - > `db.<collection>.dropIndex('<nom_index>')`

Performance et indexation

- Monitoring des opérations sur les index (création/reconstruction)
 - Progression: `db.currentOp()`
 - Interruption: `db.killOp()`



Administration de MongoDB



Sauvegarde et restauration

- Sauvegarde à froid (le plus simple)
 - Les étapes
 - Arrêter le service mongod
 - Copier tous les fichiers de --dbpath
 - <nom_base>.<numero_fichier>
 - <nom_base>.ns
 - Démarrer le service mongod

Sauvegarde et restauration

- Export MongoDB
 - mongodump
 - Les principales options
 - --db
 - --collection
 - --out (précise le répertoire d'export)
 - --dbpath (précise la racine à exporter)

Sauvegarde et restauration

- Import MongoDB
 - Commande mongoimport
 - Permet d'importer
 - Une base
 - Une collection
 - Des champs
 - Deux formats d'imports possible
 - JSON
 - CSV

Sauvegarde et restauration

- Import MongoDB
 - Commande mongorestore
 - Permet de restaurer
 - Pré-requis
 - Droper la base cible
 - Les principales options
 - --db
 - --collection
 - --dbpath



Les principaux paramètres de configuration d'une instance





Les principaux paramètres

- Renseignés dans le fichier mongod.conf
 - dbpath emplacement des données d'une instance
 - logpath emplacement des logs
 - logappend continu a écrire des anciens logs ou créer un nouveau fichier
 - nojournal = true désactive ou pas la journalisation
 - noprealloc = true désactive la pré-allocation des fichiers data
 - nssize = <size> taille du fichier d'espace de nom
 - port = 27017 port utilisé par une instance
 - auth = true active l'authentification sous MongoDB
 - fork = true démarrage de mongod en tâche de fond
 - objcheck = true inspection des documents avant insertion (utf-8 , type ect ..)
 - nohttpinterface = true active ou désactive l'interface web de monitoring (Defaults to localhost:27018).
 - ...