

# Base de données NoSQL – TD 1

## MongoDB

Master 2 BI&BD - 2020–2021 - 11/10/2021

### 1 Rappel sur MongoDB

MongoDB est une base purement "documentaire" qui représente les données au format JSON. MongoDB est un des SGBD NoSQL les plus populaires du moment. MongoDB est particulièrement appréciée pour sa capacité à passer en mode distribué pour répartir le stockage et les traitements.

#### 1.1 Principes de base

En bases de données orientées documents, une table est appelée *collection* et une entrée dans une table mongo est appelée *document*. Les documents sont stockés dans une collection. Les collections sont stockées dans une base de données.

Un document peut être vu comme un ensemble de couples {clé:valeur}. Exemples de documents très simples :

```
{ 'prenom': 'James', 'nom': 'Bond' }
{ 'prenom': '007' }
{
  'code': 'NS001',
  'intitule': 'NoSQL',
}
```

Les documents d'une collection peuvent avoir des champs différents. Les champs de même nom peuvent contenir des valeurs de types différents. Une collection n'est qu'un ensemble de documents. Les valeurs peuvent être complexes (listes, dictionnaires,...).

#### Identifiants

Un identifiant unique est supposé être affecté à chaque document. C'est identifiant est la valeur de la clé `_id`.

#### Les opérations

Toutes opérations effectuées avec une commande prennent en paramètres des tableaux de clé / valeur.

- Les documents sont naturellement fournis comme des tableaux {clé:valeur}
- Les requêtes sont fournies sous la forme de tableaux {clé:valeur}
- Les options des commandes sont spécifiées sous la forme de tableaux {clé:valeur}

## 2 Installation de MongoDB

MongoDB fonctionne en mode classique client(mongo)/serveur(mongod). MongoDB est multiplateforme, les instructions d'installation sont sur :

<https://docs.mongodb.com/manual/installation/>

### 2.0.1 Sous Windows

- Avant de commencer : taper la commande `C :> wmic os get osarchitecture` sous DOS pour vérifier que votre Windows est 64 bits.
- Télécharger la dernière version de MongoDB en cliquant ici. Assurez-vous d'obtenir la dernière version.
- MongoDB nécessite un dossier de données pour stocker ses fichiers. L'emplacement par défaut pour le répertoire de données MongoDB est `c:\data\db`. Donc, vous devez créer ce dossier en utilisant l'invite de commande. Exécuter la séquence de commande suivante :  
`C :>md data` puis `C :>md data\db`
- Après l'installation de MongoDB, ouvrez un premier invite de commande dans le dossier *bin* pour lancer *mongod*.
- Si le démarrage est réussi, vous devez voir que le processus se lance sur le port 27017 qui est le port par défaut utilisé par MongoDB.
- Un message comme celui-ci vous l'indique : *Waiting for connections on port 27017*.
- Puis ouvrez un second invite de commande pour lancer l'interpréteur de commandes : *mongo*
- Vous êtes maintenant connecté à votre moteur de base de données.

### 2.0.2 Sous Linux (Ubuntu)

- Exécutez la commande suivante pour importer le GPG publique MongoDB Key :  
`sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927`
- Créez un fichier `/etc/apt/sources.list.d/mongodb.list` en utilisant la commande suivante :  
`\echo "deb http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list`
- Maintenant, exécutez la commande suivante pour mettre à jour le référentiel :  
`sudo apt-get update`
- Maintenant, installez le MongoDB en utilisant la commande suivante :  
`sudo apt-get install -y mongodb-org`
- Démarrer MongoDB : `sudo service mongodb start`
- Arrêter MongoDB : `sudo service mongodb stop`
- Redémarrer MongoDB : `sudo service mongodb restart`
- Pour utiliser MongoDB exécutez la commande suivante : *mongo*

## 3 Activité 1 : Les commandes de base

Les commandes de base que nous allons voir durant ce TD sont à utiliser avec le client *mongo* de MongoDB.

- Créer une base de données

- > `use NOM_DE_VOTRE_BASE_DE_DONNEES`
- Si la base de données n'existe pas, elle sera créée. Créer une base données nommée `td1`.*
- Pour vérifier le nom de la base de données en cours d'utilisation
- > `db`
- Lister les bases données existantes
- > `show dbs`
- Si votre base de données est vide, elle ne sera pas listée.*
- Supprimer une base de données
- > `db.dropDatabase()`
- Si vous n'avez pas sélectionné une base de données, la commande supprimera la base de données `test` de `mongoDB`. Si vous voulez supprimer une base de données en particulier, lancez d'abord la commande `use NOM_DE_LA_BASE_DE_DONNEES`, suivi de `db.dropDatabase()`*
- Une base de données est constituée d'un ensemble de collections, l'équivalent d'une table en SGBD relationnel. Pour créer une collection :
- > `db.createCollection("NOM_DE_LA_COLLECTION", options)`
- options correspond au document et est utilisé pour spécifier les configurations de la collection (taille de la mémoire, indexation,...).*
- Source : <https://docs.mongodb.com/manual/reference/method/db.createCollection/>
- Lister les collections de votre base de données
- > `show collections`
- Insérer un document JSON dans une collection
- > `db.NOM_DE_LA_COLLECTION.insert({"ATTRIBUT_CLÉ": "VALEUR"})`
- Si la valeur de l'attribut `_id` n'est pas spécifiée, une valeur générée automatiquement lui sera affectée*
- Afficher le contenu d'une collection
- > `db.NOM_DE_LA_COLLECTION.find()`
- L'aide de MongoDB
- > `db.help()`
- Pour obtenir la liste des commandes de `MongoDB`.*
- MongoDB Statistiques
- > `db.stats()`
- Pour obtenir des statistiques sur le serveur `MongoDB`. Cela affichera le nom de la base, le nombre de collections et de documents dans la base de données.*

### 3.1 Activité 2 : Collection, documents

1. Créer une collection nommée livres

```
> db.createCollection("livres");
```

2. Insérer deux documents dans votre collection livres

```
> db.livres.insert({titre: "L'Histoire de Pi", annee: 2001,
auteur: {prenom: "Yann", nom: "Martel"}});
> db.livres.insert( {titre: "Le Trône de fer", annee: 1996,
auteur: {prenom: "George R. R.", nom: "Martin"}})
```

Dans livres, auteur est ici un document incorporé (ou imbriqué) "Embedded Documents".

3. Pour afficher/rechercher/sélectionner les documents de votre collection

```
### db.collection.find()
### db.collection.find(requête)
### db.collection.find(requête, projection)

### 1- sans paramètres, tous les documents sont renvoyés
### 2- requête est un tableau clé / valeurs spécifiant des opérateurs sur les
###    champs des documents recherchés
### 3- projection est un tableau permettant de limiter les champs que l'on
###    souhaite consulter dans les documents recherchés (cette option sera
###    traitée ultérieurement)
```

```
> db.livres.find();
> db.livres.find({titre:"L'Histoire de Pi"});
```

**N.B.** mongo a rajouté automatiquement le champ (field) `_id` qui est unique, créé par défaut et indexé.

4. Pour modifier le prénom de "George R. R." en "George Raymond Richard" :

```
### db.collection.findOne(requête) permet de rechercher un document précis
```

```
> db.livres.findOne({titre:'Le Trône de fer'})
```

```
### Le résultat sera le document correspondant, copiez la valeur du champ _id
### pour faire la modification
### db.collection.update(requête, modifications)
```

```
> db.livres.update({'_id':ObjectId("OBJECT_ID")},
{'$set':{auteur:{prenom:"George Raymond Richard", nom:"Martin"}}});
```

**N.B. :**

- `$set` permet de spécifier la modification à faire. `$set` est ici utilisé comme clé et en valeur il prend le `field` à modifier et sa nouvelle valeur ;
- auteur étant un `Embedded Documents` il faut aussi y **faire référence** et dans le **bon ordre** pour faire la modification. Ceci est valable pour tous les `Embedded Documents`.

5. Pour supprimer un document

```
### db.collection.remove()
### db.collection.remove(requête)
```

```
### 1- sans paramètres, tous les documents sont supprimés (attention, donc)
### 2- requête est de la même forme que pour find, elle désigne les documents
### qui seront supprimés
```

**À faire :** supprimez le livre "L'Histoire de Pi".

6. Importer un fichier dans votre collection en utilisant `mongoimport`. Télécharger le fichier `livres.json` depuis Moodle.

(a) Quitter le client mongo

```
> exit;
```

(b) Copier le fichier `livres.json` téléchargé précédemment dans le :

```
C:\> data (sous Windows)
```

```
$ /home/ (sous Linux).
```

(c) Lancer la commande `mongoimport` avec (l'installation de de MongoDB Tools est nécessaire, voici le lien) :

Sous Windows :

```
C:\>mongodb\bin\mongoimport --host localhost:27017 --db td1
--collection livres < C:\Data\livres.json
```

Sous Linux :

```
$ mongoimport --host localhost:27017 --db td1
--collection livres --file /home/livres.json
```

Les options, de la commande `mongoimport`, utilisées sont :

- `--host localhost:27017` : le serveur où se trouve le mongod
- `--collection livres` : le nom de la collection dans laquelle on va importer le contenu du fichier
- `--file /home/livres.json` : le fichier à importer

(d) Vérifier sur votre client mongo que le document a bien été importé dans la collection livres

```
> use td1
```

```
> db.livres.find();
```

**N.B.** Le `find()` affichera 20 enregistrements par défaut. Suivez les instructions du client mongo si vous souhaitez afficher plus.

### 3.2 Activité 3 : Sélection avancée et Indexation

1. Mongo lors des sélections avec `find()` différencie les majuscules et les minuscules

```
> db.livres.find({titre: "harry potter à l'école des sorciers"})
```

```
### Le résultat sera Null, Harry Potter a été écrit avec des majuscules
```

```
> db.livres.find({titre: /harry potter à l'école des sorciers/i})
```

```
### Mettre la valeur à rechercher entre "/.../" permet de faire des expressions
### régulières. Le "i" c'est pour lui dire de ne pas différencier majuscule
### et minuscule
```

- ```

> db.livres.find({titre: /^harry potter à l'école des sorciers$/i})
### "^" : commence par ...
### "$" : se termine par ...
> db.livres.find({titre: /potter/i})
### tous les titres contenant "potter"

```
2. Les opérateurs de comparaison :
- ```

$gt : > ; $gte : >= ; $lt : < ; $lte : <= ; $ne : !=
> db.livres.find({annee:{$gte:1997}})
### Affichera tous les livres dont l'année d'édition est supérieure ou égale à 1997
### La condition de restriction est mise entre {} sous forme de clé/valeur

> db.livres.find({annee:{$gte:1997}, titre: /sorcier/i});
### Livre sorti après 1996 ET dont titre contient le mot sorcier

```
3. Opérateurs logique :
- ```

$or : OR ; $and : AND ; $nor : NOR (Negation OR) ; $not : NOT
> db.livres.find({$and : [
    {annee:{$gte:1936}},
    {annee:{$lte:1997}}
  ]
})
### Entre 1936 et 1997
### Un tableau de clé/valeur

```
4. Les options du find()
- ```

> db.livres.find({}, {titre:1})
> db.livres.find({}, {_id:0, titre:1})
### _id est toujours affiché par défaut

```
5. Compter les enregistrements :
- ```

> db.livres.count()
> db.livres.count({titre: /potter/i})

```
- À faire :** commentez le résultat de la dernière requête.

6. Recherche par mot clé :

```
> db.livres.find({$text:{$search:"potter"}})
### cette commande renvoie une erreur, car il faut créer un ou plusieurs
### index avant de l'exécuter
> db.livres.createIndex({titre:'text'})
> db.livres.find({$text:{$search:"potter"}})
```

**N.B. :**

- `$text` est un opérateur qui permet de spécifier le type de donnée à chercher ;
- `$search` est une fonction disponible dans mongodb, elle est utilisée en tant que clé pour rechercher la valeur associée dans une collection.

7. Lister les indexes

```
> db.livres.getIndexes()
```

8. Supprimer un index

```
> db.livres.dropIndex("INDEX_NAME")
```

9. Afficher le plan d'exécution

```
> db.livres.find({annee:1937}).explain("executionStats")
```

**Travail à faire :**

- (a) Rajoutez un index sur le champ année  

```
> db.livres.createIndex({annee:1})
```
- (b) Affichez à nouveau le plan d'exécution  

```
> db.livres.find({annee:1937}).explain("executionStats")
```
- (c) Comparez les deux plans d'exécutions avec et sans index. Commentaires ?

### 3.3 Activité 4 : À vous de jouer

MongoDb permet d'importer des fichiers de type .csv. L'option `--type` avec la valeur `csv` permet de spécifier le format du fichier à importer (`--type csv`).

L'option `--headerline` utilisera l'entête de votre csv comme champ (`field`) dans vos documents mongo.

Le site open data du gouvernement permet de récupérer et de réutiliser certaines données ouvertes. La liste et la localisation des Musées de France (au 31/12/15) a été **téléchargée**, **nettoyée** et **exportée** en .csv. Vous pouvez le télécharger sur Moodle.

#### 3.3.1 Travail à faire :

1. Téléchargez et importez dans mongodb le fichier `musees.csv` (utilisez la commande `mongoimport` vue précédemment et rajoutez les options manquantes).
2. En utilisant une "requête" mongo, quel est le nombre total de musées ?
3. Combien de musées sont situés à Lyon ?
4. Combien de musées en AUVERGNE - RHÔNE-ALPES ? En sachant que dans le fichier .csv fourni les deux régions sont séparées.
5. Listez les musées "Municipal". Combien sont-ils ?
6. Listez les musées qui n'ont pas de site Web. Combien sont-ils ?
7. Lister le nom et la ville des musées qui ont le code postal 21000.
8. Combien de musées ne sont pas à Paris.
9. Affichez les musées du DOUBS. Combien sont-ils ?
10. Lister les musées qui ont le mot "Musée" dans leur nom. Combien sont-ils ?