

# TP2 : MongoDB

## Saison III, épisode 1

### Objectifs

- Utilisation de javascript sur une base noSQL
- Apprentissage de mongoDB
- Utilisation du format Json
- Réalisation de petits programmes simples utilisant mongoDB.

### Section

R5\_10: Département Informatique  
IUT CAEN Campus 3

### Auteur

E.Porcq

### Date

08/09/2025 durée 3h00

L'objectif de ce TP est d'apprendre les bases du Big Data et de MongoDB

Rendre un compte-rendu par binôme à la fin de la séance

- nommé "TP2\_<nom1>\_<nom2>.pdf" (sans accent ni espace).
- L'étudiant qui envoie le CR met son binôme en copie.
- Il faut des commandes, des résultats, [des observations] [et des conclusions]
- les captures sont éditablest **obligatoirement** (au moins les commandes)
- le compte-rendu concerne le 1.3 au 2.1

Si plusieurs binômes sont constituables et non constitués, tirage au sort.

Dans le TD, les fichiers XXX sont fournis et les fichiers YYY sont à créer

## 1 Initiation à MongoDB

### 1.1 Présentation

On peut créer des collections qui contiendront des documents.

Un document est 1 enregistrement de type json {} .

Un document JSON ne comprend que deux types d'éléments structurels :

- Des ensembles de paires "nom" (alias "clé") / "valeur";
- Des listes ordonnées de valeurs.
- Ces mêmes éléments représentent trois types de données :
  - Des objets ;
  - Des tableaux ;
  - Des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.



#	title	stuff	moar
1	Bla bla	Mdr	xD
3	TEST	Lmfao	XML
4	Azerty	GUI	Lol

```
{
  { title: "Bla bla", stuff: "Mdr", moar: "xD" }
  { title: "TEST", stuff: "Lmfao", moar: "XML" }
  { title: "Azerty", stuff: "GUI", moar: "Lol" }
}
```

Sur disque, il est enregistré en binaire (Bson).

Dans une même collection, les documents peuvent avoir un format différent.

La façon d'associer plusieurs collections est assez différente aux SGBDR mais pas tant que cela. D'un point de vue UML, on peut parler

- d'association simple (normalisé au sens BD)
- de composition (dénormalisé)



## Normalisé

## Dénormalisé

Si la taille maximale allouée à ce document est dépassée, ce qui est généralement limité à 16Mo à cause de BSON, MongoDB va déplacer le fichier sur le disque-dur. Ceci peut être un facteur de décision pour le choix d'un modèle de données plutôt normalisé ou dénormalisé (**duplicer** l'information, **la répéter**, autant de fois que nécessaire afin de faire « descendre » les attributs au plus près des données : de cette façon on simplifie les relations et on optimise l'accès aux données )

Il est possible d'utiliser une Collection Plafonnée ; une collection plafonnée est un type spécial de collection qui a soit des éléments fixes, soit un nombre d'éléments fixes. Une fois que la collection est pleine, les éléments les plus anciens seront effacés lors de l'ajout de nouveaux éléments.

De plus, si une application exécute de nombreuses opérations de lecture, on peut ajouter des Index.

En cas de " jointure " fréquente, mieux vaut choisir des compositions ; mais en cas de répétitions trop fréquentes, mieux vaut choisir l'association.

On peut aussi organiser la structure en arbre.

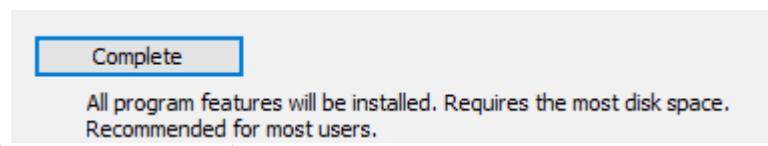
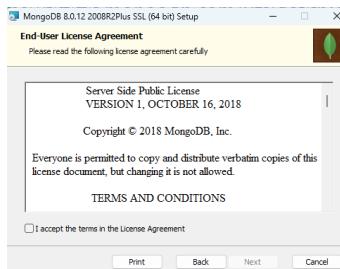
ObjectID représente l'identifiant d'un document, et est toujours contenu dans la propriété \_id. Lorsque l'on insère un nouveau document dans une collection, MongoDB lui définit automatiquement une propriété \_id. On peut définir soit-même un id à un document.

### 1.2 Installation (facultatif)

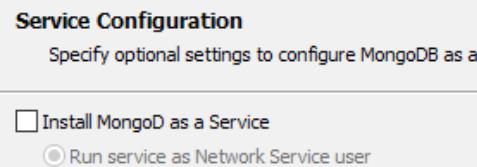
<http://harry-wanki.developpez.com/tutoriels/mongodb/debuter-mongodb-introduction-base-donnees-nosql/>  
<https://www.tutorielsenfolie.com/tutoriels-113-Installation-et-configuration-de-MongoDB-sous-Windows.html>  
[https://www.mongodb.com/try/download/community?tck=docs\\_server](https://www.mongodb.com/try/download/community?tck=docs_server)

En 2236, mongoDB est déjà installé.

- <https://www.mongodb.com/try/download/community>
- <https://www.mongodb.com/download-center/community/releases/archive>
- Version installée 6.0.8 2008R2Plus



- décocher l'installation en tant que service



- Depuis la version 4.4 de mongoDB, certains outils sont à installer à part
  - <https://www.mongodb.com/try/download/database-tools>
  - <https://www.mongodb.com/try/download/shell>
    - copier le contenu des bin dans le bin de mongodb

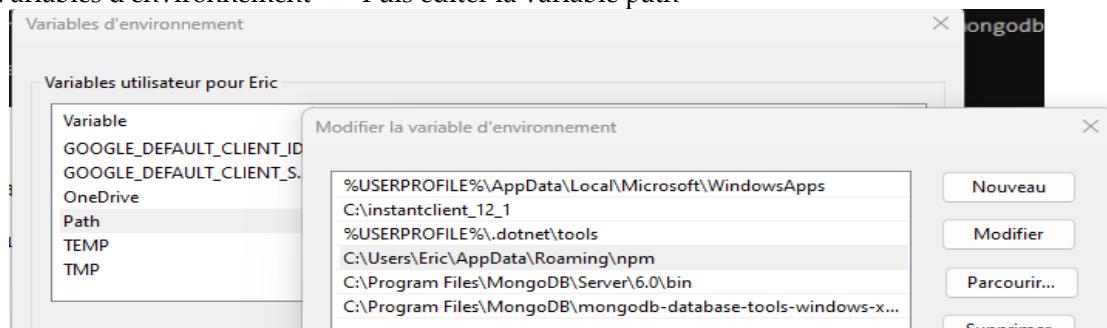
- Vérifiez que vous parvenez à le lancer à partir d'un interpréteur de commande. Ce n'est pas grave si la connexion échoue.

```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.6
Using MongoDB: 8.0.12
Using Mongosh: 2.5.6

```

- si cela ne fonctionne pas, ajoutez-le chemin dans le path (il faudra peut-être redémarrer)
  - paramètres--> Informations systèmes --> paramètres avancés du système
  - Variables d'environnement --> Puis éditer la variable path



- Il peut être nécessaire de redémarrer

### 1.3 tests du serveur

- dans l'interpréteur de commande, créer un lecteur S dans le dossier où seront déposés les scripts de l'exercice et la base (pas dans le dossier de mongoDB)

Exemple :

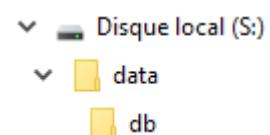
```

Z:\fait_ici\M4101C_2_TP2>subst S: /D
Z:\fait_ici\M4101C_2_TP2>subst S: .
Z:\fait_ici\M4101C_2_TP2>S:
S:\>

```

- Fabriquer le dossier data/db dans son dossier de travail (donc dans [S:\](#)). Ce dossier :
  - contiendra toutes les bases de données.
  - est géré par mongoDB. Ne rien y enregistrer directement.
- Lancer mongod

```
mongod.exe --dbpath S:\data\db --bind_ip 127.0.0.1
```



```

data capture", "attr": {"dataDirectory": "S:/data/db", "diagnostic": {}}
[{"t": {"$date": "2021-01-11T21:37:04.511+01:00"}, "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "attr": {"address": "127.0.0.1"}}
[{"t": {"$date": "2021-01-11T21:37:04.511+01:00"}, "s": "I", "c": "NETWORK", "id": 23016, "ctx": "listener", "msg": "Waiting for connections", "attr": {"port": 27017, "ssl": "off"}}

```

le serveur est lancé

- Ouvrir une autre console pour exécuter le client
- lancer mongosh

```
S:\>mongosh
Current Mongosh Log ID: 68a1fc6dec8e28dea3eec4a8
Connecting to:          mongodb://127.0.0.1:2701
7/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.6
Using MongoDB:          8.0.12
Using Mongosh:          2.5.6
```

le client est lancé

- Tester les commandes suivantes :
  - db
  - show dbs ou show databases
  - use maBDTP2
  - db
  - show dbs
  - show collections
  - db.createCollection("maCollection1")
  - show dbs
  - db.createCollection("maCollection2")
  - db.createCollection("maCollection3")
  - show collections
  - db.maCollection3.insertOne( { "\_id" : "1", "nom" : "nom1" } )
  - db.maCollection3.insertOne( { "\_id" : "35", ville : "Moult" , pays : "France", pop : 2200 , loc : [ 05,05 ] } )
  - db.maCollection3.insertOne ( { "nom" : "nom2" } )
  - db.maCollection3.find()
  - db.maCollection3.find( {}, { \_id:1 } )
  - db.maCollection3.find( { "nom": "nom1" }, { \_id:0, nom:1 } )
  - db.maCollection3.drop()
  - config.set("displayBatchSize", 300)
  - db.dropDatabase()
  - show databases

## 1.4 Creation d'une collection

<https://openclassrooms.com/courses/guide-de-demarrage-pour-utiliser-mongodb>  
<https://docs.mongodb.com/manual/reference/method/>

#### 1.4.1 Utiliser une base de données nommée maBDDTP2

**1.4.2** Créer une collection nommée maCollec1. Vérifier que la création a fonctionné.

**1.4.3** Étudier les fichiers fournis (**p14\_X.json**) et indiquer l'erreur du **p14\_2\_faux.json**.

#### **1.4.4** Y insérer les documents suivants. 3 solutions :

- On peut insérer ligne par ligne avec `db.<nom de la collection>.insert`
  - On peut utiliser l'utilitaire `mongoimport` (ce n'est pas une commande du shell mongo). Le fichier associé contient toutes les données json nécessaires.
  - On peut utiliser la commande `load`. Le fichier associé contient les requêtes « `db.<collection>.insert` ».

#### **1.4.5 Analyser la collection. Quelle est la différence avec les données d'une BDDR ?**

				telephone										
_id	nom	prenom	age	portable	fixe	ville	pays	pop	loc	type	couleur	matiere	fruit	
"Dubois"	"Maurice"		25											
"1"	"Dubois"	"Mauricette"	23											
2	"Dupont"	"Bertrand"	28											
3	"Dulong"	"sylvie"	34											
"4"	"Dubois"	"Sylvian"												
"5"	"Delalune"	"Claire"												
80	"Frere"	"Jacques"	22											
11										"casserolle"	"rouge"	"fonte"		
12										"casserolle"	"verte"	"fer"		
5	"Troj"			"06 85 45 48 10"										
6	"Super"			"06 85 45 48 10"	"02 45 38 45 33"									
7	"Super"	"veronique"		"06 80 23 48 12"	"02 32 41 22 16"									
8	"orange"	"Enpierre"	36											
21													["orange", "banane", "pomme"]	
22													["kiwi", "banane", "pomme"]	
23													["kiwi", "orange", "poire"]	
24													["cerise", "peche", "citron"]	
25													["cerise", "banane", "orange"]	
26													["orange", "pomme", "raisin"]	
"27"													["orange", "prune", "cerise"]	

## 1.5 Requêtes sur maCollection

L'objectif est d'apprendre à effectuer des projections, des restrictions et des mises à jour de base  
Conserver toutes les requêtes (pas de rollback possible)

### 1.5.1 Afficher les documents avec et sans la méthode pretty

### 1.5.2 Afficher pour tous les documents la valeur (attribut) nom

```
{ nom: 'Dubois' }, { nom: 'Dubois' },
{ nom: 'Frere' }, { nom: 'Dupont' },
{},
{ nom: 'Troj' }, { nom: 'Super' },
{ nom: 'orange' }, { nom: 'Dulong' },
{ nom: 'Super' }, {},
{},
{ nom: 'Delalune' }, { nom: 'Dubois' }]
```

### 1.5.3 Afficher l'attribut nom des documents qui en contiennent (utiliser \$exists)

```
{ nom: 'Dubois' },
{ nom: 'Dubois' },
{ nom: 'Frere' },
{ nom: 'Dupont' },
{ nom: 'Troy' },
{ nom: 'Super' },
{ nom: 'orange' },
{ nom: 'Dulong' },
{ nom: 'Super' },
{ nom: 'Delalune' },
{ nom: 'Dubois' }]
```

#### 1.5.4 Afficher l'attribut nom sans doublons

```
[ 'Delalune', 'Dubois', 'Dupont', 'Frere', 'Super', 'Troj' ]
```

**1.5.5 Afficher tous les documents sans afficher l'attribut nom**

```
{
  "_id" : ObjectId("5c8a21aa88037310c73168de"), "prenom" : "Maurice", "age" : 25 },
  {"_id" : "1", "prenom" : "Mauricette", "age" : 23 },
  {"_id" : 2, "prenom" : "Bertrand", "age" : 28 },
  {"_id" : 3, "prenom" : "Sylvie", "age" : 34 },
  {"_id" : "4", "prenom" : "Sylvian" },
  {"_id" : 11, "type" : "casserolle", "couleur" : "rouge", "matière" : "fonte" },
  {"_id" : 12, "type" : "casserolle", "couleur" : "verte", "matière" : "fer" },
  {"_id" : 5, "telephone" : { "portable" : "06 85 45 48 10" } },
  {"_id" : 6, "telephone" : { "portable" : "06 85 45 48 10", "fixe" : "02 45 38 45 3" },
  {"_id" : 7, "prenom" : "veronique", "telephone" : { "portable" : "06 80 23 48 12" },
  {"_id" : 21, "fruit" : [ "orange", "banane", "pomme" ] } }
```

**1.5.6 Afficher les documents sans l'attribut nom, les données qui n'ont pas de nom**

```
{
  "_id" : 11, "type" : "casserolle", "couleur" : "rouge", "matière" : "fonte" },
  {"_id" : 12, "type" : "casserolle", "couleur" : "verte", "matière" : "fer" },
  {"_id" : 21, "fruit" : [ "orange", "banane", "pomme" ] },
  {"_id" : 22, "fruit" : [ "kiwi", "banane", "pomme" ] },
  {"_id" : 23, "fruit" : [ "kiwi", "orange", "poire" ] },
  {"_id" : 24, "fruit" : [ "cerise", "peche", "citron" ] },
  {"_id" : 25, "fruit" : [ "cerise", "banane", "orange" ] },
  {"_id" : 26, "fruit" : [ "orange", "pomme", "raisin" ] },
  {"_id" : 27, "fruit": [ 'orange', 'prune', 'cerise' ] } }
```

**1.5.7 Afficher les documents dont le nom est Dubois****1.5.8 Afficher le nom et prénom des documents dont le nom est Dubois****1.5.9 Afficher les documents dont le nom est Dubois et le prénom Mauricette****1.5.10 Afficher les documents dont le nom contient un " r "****1.5.11 Afficher les documents dont le nom ne commence pas par un " D "**

```
{
  "_id": 80, "nom: 'Frere', prenom: 'Jacques' },
  {"_id: 5, nom: 'Troj' },
  {"_id: 6, nom: 'Super' },
  {"_id: 7, nom: 'Super', prenom: 'veronique' },
  {"_id: 8, nom: 'orange', prenom: 'Enpierre' } }
```

**1.5.12 Afficher les documents d'\_id > 23. Justifier les résultats****1.5.13 Afficher les documents d'\_id < "32". Justifier les résultats****1.5.14 Afficher les documents dont l'age est soit 22, soit 25 ans****1.5.15 Afficher les documents dont les numéros de téléphones sont "06 85 45 48 10" et "02 45 38 45 33"****1.5.16 Afficher les documents dont le numéro de mobile est "06 85 45 48 10"**

```
{ _id: 5, nom: 'Troj', telephone: { portable: '06 85 45 48 10' } },
{ _id: 6, nom: 'Super', telephone: { portable: '06 85 45 48 10', fixe: '02 45 38 45 33' } }
```

**1.5.17 Afficher les documents contenant le fruit " orange ".****1.5.18 Afficher les documents dont le premier fruit est " orange ".****1.5.19 Donner l'age de 34 ans à tous les Dubois.****1.5.20 Donner l'age de 36 ans au premier Dubois.****1.5.21 Donner l'age de 19 ans au second Dubois (merci chatGPT)**

```
const second = db.maCollect1.find().skip(1).limit(1).next()
db.maCollect1.updateOne( { _id: second._id }, { $set: { age: 19 } })
ou
db.maCollect1.updateOne( { _id: { $in: [ db.maCollect1.find().skip(1).limit(1).next()._id ] } }, [ { $set: { age: 19 } } ] )
```

**1.5.22 Donner l'age de 35 ans à Dudouit. S'il n'existe pas, le créer.****1.5.23 Remplacer le document 24 par celui-ci { "\_id" : 24 , fruit:["mandarine","citron","prune"] } ).**

- Quelle est la différence entre update et replace ?

**1.5.24 Supprimer le document 4****1.5.25 Supprimer le premier Dubois**

# TO BE CONTINUED