

## MongoDB

### Accès à Mongo

**Base de Données : Accès, Crédation, Suppression**

**Base de Données : Insertion, Sélection, Récupération, Suppression**

### ACCES à Mongo

- Pour savoir quelle version nous allons utiliser, il suffit d'exécuter la commande suivante dans le terminal de commande.

#### **Mongo -version**

- Pour lancer le shell de Mongo, il suffit d'exécuter la commande suivante dans le terminal de commande.

#### **Mongo**

=> Vous devriez voir apparaître des avertissements divers. Il ne faut pas en tenir compte.

- Pour accéder à toutes les commandes, il suffit d'exécuter la commande suivante dans le terminal de commande.

#### **db.help()**

- Pour accéder à toutes les statistiques, il suffit d'exécuter la commande suivante dans le terminal de commande.

#### **db.stats()**

## Base de Données : Accès, Crédit, Suppression

- Une fois connecté au shell, il est possible de connaître l'ensemble des bases de données présentes sur le serveur. Pour ceci, vous devez utiliser la commande **show dbs** comme ci-dessous.

**show dbs**

=> Vous devriez voir apparaître la liste des bases de données

- Pour créer une base de données, commande **use « nomdelabase »**
- Il faut ensuite créer une collection, commande **db.createCollection("Name")**

=> Vous devriez voir apparaître { "ok" : 1 }

- Pour accéder à une BD existante, commande **use « nomdelabase »**

=> Vous devriez voir apparaître switched to db « nomdelabase »

- Une base de données est constituée d'une ou plusieurs bases de données. Chacune de celles-ci contient un ensemble de collections. Pour lister celles-ci, on utilise la commande **show dbs**

=> Vous devriez voir apparaître la liste des bases de données

- Pour savoir dans quelle base de données vous êtes. on utilise la commande **db**

=> Vous devriez voir le nom de la base de données

- Pour supprimer une base de données **db.dropDatabase()**

=> Vous devriez voir { "dropped" : "nomdelabase", "ok" : 1 }

- Pour créer une collection, commande **db.createCollection(Nomcollection,Options)**
- Pour supprimer une collection, commande **db.nom.drop()**

=> Vous devriez avoir true si la collection a été supprimée, false dans le cas contraire

## Base de Données : Insertion, Sélection, Récupération, Suppression

- Le formalisme de base est le suivant : **db.collection.fonction()**
  - db représente la base de données choisie grâce à la commande use (ce mot clé est non modifiable)
  - collection représente la collection dans laquelle nous allons effectuer l'opération, et doit donc correspondre à une des collections présentes dans la base
  - fonction() détermine l'opération à effectuer sur la collection.

### INSERER

- Insérer **un seul** enregistrement de données  
**db.nomdelacollection.insertOne( {.....});**
- Insérer **plusieurs** enregistrements de données  
**db.nomdelacollection.insertMany ( [... { .....},... { .....},... ] );**
- Une base de données est constituée d'une ou plusieurs collections. Chacune de celles-ci contient un ensemble de documents. Pour lister celles-ci, on utilise la commande **show collections**  
**show collections**

=> Vous devriez voir apparaître la liste des documents

### RECHERCHE d'INFORMATIONS (Selection)

- Fonction find()

=> Vous devriez voir apparaître la liste des documents, c'est l'équivalent du SELECT  
SELECT \*  
FROM table  
WHERE condition;  
**db.nomdelacollection.find()**

La fonction find() peut aussi prendre deux paramètres :

- **les critères de sélection des documents**
- **les choix d'items des documents à afficher**

Ces deux paramètres doivent être écrits sous la forme d'objets JSON.

En remplaçant la paire : **db.nomdelacollection.find({ "X": "Y" }{ : })**

- Fonction pretty()

**db.collections.find({ "X": "Y" }).pretty()** pour avoir une présentation plus propre.

Cela correspond à SELECT \* FROM table WHERE X= "Y";

Autre exemple :

```
db.collections.find ( { "X" : "Y", "Z" : "T", "A" : "B" })
db.collections.find ( { "X" : "Y"}, {"Z" : 1, "A" : 1 })
db.collections.find({ "year": 2000 }).pretty()
```

- Fonction findOne()

La fonction findOne() permet de n'obtenir qu'un seul résultat  
**db.nomDeMaCollection.findOne()**. (sans pretty)

- Fonction update ()

**db.collectionName.update({KeyToUpdate},{Set Command})**

Exemple

```
db.nomdelacollection.update({"Key" : "valeur"},{$set : {"Qty": 10}} )
```

- Fonction distinct ()

**db.collectionName.distinct("champsderecherche")**

Exemple : **db.maCollection.distinct('product',{"Qty":20})**

- Fonction renameCollection ()

**db.source.renameCollection('destination')**

=> Vous devriez voir apparaître : { "ok" : 1 }

- Fonction count()

=> Vous devriez savoir combien il y a des résultats

**db.collections.count()**

- Fonction limit ()

Pour limiter le nombre de réponses à X.

### **db.nomdelacollection.find().limit(X)**

- Fonction skip ()

Pour gérer un argument à partir du nombre X

### **db.nomdelacollection.find().skip(X)**

Exemple : db.nomdelacollection.find().skip(X).limit(Y)

=> Vous devriez savoir combien il y a de résultats avec à partir du Xième+1 document et d'afficher Y documents

- Fonction sort ()

Pour trier par ordre croissant ou décroissant

### **db.nomcollection.find().sort({"X":1})**

- Operateurs de Comparaison

les opérateurs logiques habituels **>, >=, <, <=, !=** (différent), “**est dans la liste**”, “**n'est pas dans la liste**” correspondent dans MongoDB à des opérateurs dédiés **\$gt, \$gte, \$lt, \$lte, \$ne, \$in, \$nin**

**remarque :** **\$in** pour l'un ou l'autre valeur d'un champ est égal à n'importe quelle valeur du tableau spécifié.

**\$all** pour les deux, l'opérateur **\$all** sélectionne les documents où la valeur d'un champ est un tableau contenant tous les éléments spécifiés.