



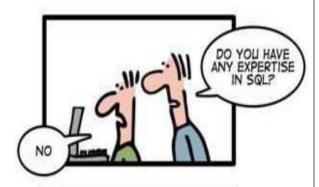
Cours de Bases de données NOSQL (MongoDB)

madaniabdellah@gmail.com

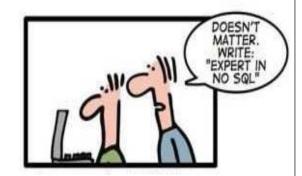
Plan du cours

- Rappels :
 - JSON
 - Bases de Données Relationnelles
- Introduction : Mouvement NoSQL
- MongoDB
 - Caractéristiques de MongoDB
 - Requêtes avec MongoDB
 - MongoDB et Java
 - MongoDB et PHP
 - Réplication et reprise sur panne dans MongoDB
 - Partitionnement dans MongoDB
 - TextSearch

HOW TO WRITE A CV







- use base_donnee : créée une base de données si elle n'existe pas, sinon y accède
- db : affiche la base de données courante
- show dbs : affiche la liste des bases de données
- db.dropDatabase() : supprime la base de données courante

use mabase

db

Show dbs

- db.createCollection(col) : créée la collection « col »
- show collections : liste des collections
- db.col.drop() : supprime la collection « col »

```
use mabase
db.createCollection(«livres »)
show collections
db.livres.drop()
show collections
```

• db.collection.insert(document) : insère un document dans une collection

- db.collection.find(): tous les documents sont renvoyés
- db.collection.find(requête) : requête est un tableau clefs / valeurs spécifiant des opérateurs sur les champs des documents recherchés
- db.collection.find(requête, projection) : projection est un tableau permettant de limiter les champs que l'on souhaite consulter dans les documents recherchés

- db.collection.findOne(): fait la même chose que db.collection.find() mais sans s'embarrasser d'un curseur, lorsqu'on souhaite récupérer un document unique (par son identifiant, par exemple).
- Remarque :

La méthode find() retourne un curseur même si elle retourne une seule valeur

```
db.etudiants.find()
db.etudiants.find({'prenom': 'Camille'})
db.etudiants.find({'nom':/^i/})
db.etudiants.find({'nom':/i$/}) .count()
db.etudiants.findOne()
db.etudiants.find({ville:'casa'},{_id:0}).sort({moyenne:-1})
db.etudiants.find({},{\underline{id:0}}).sort({moyenne:1}).skip(10).limit(5)
```

Opération	Syntax	Example	RDBMS Equivalent
=	{ <key>:<value>}</value></key>	<pre>db.mycol.find({"by":"tuto rials point"}).pretty()</pre>	where by =
<	{ <key>:{\$lt:<value>}}</value></key>	<pre>db.mycol.find({"likes":{\$l t:50}}).pretty()</pre>	where likes < 50
<=	{ <key>:{\$lte:<value>}}</value></key>	db.mycol.find({"likes":{\$l te:50}}).pretty()	where likes <= 50
>	{ <key>:{\$gt:<value>}}</value></key>	<pre>db.mycol.find({"likes":{\$ gt:50}}).pretty()</pre>	where likes > 50
>=	{ <key>:{\$gte:<value>}}</value></key>	<pre>db.mycol.find({"likes":{\$ gte:50}}).pretty()</pre>	where likes >= 50
!=	{ <key>:{\$ne:<value>}}</value></key>	db.mycol.find({"likes":{\$ ne:50}}).pretty()	where likes != 50

- db.mycol.find({key1:value1, key2:value2}).pretty() : la virgule représente le et logique
- db.mycol.find({\$and:[{key1: value1}, {key2:value2}]}).pretty(): \$and représente le et logique
- db.mycol.find({\$or:[{key1: value1}, {key2:value2}]}).pretty(): \$or représente le et logique

```
db.mycol.find({"moyenne": {$gt:10}, $or: [{"prenom":
"Mohamed"}, {"ville": "Casa"}]}).pretty()
```

- Db.mycol.find({key : {\$exists : true | false}) : vérifie l'existence d'un champ
- Db.mycol.find({key : {\$in : [valeur1, valeur2, ...]}}) : vérifie si une valeur existe dans un tableau d'élément

- Db.employes.find({email : {\$exists : true}}) : les employés ayant un email
- 2. Db.employes.find({ville : {\$in : ['Casa', 'Rabat', 'Fes']}}) : les employés de Casa, Rabat ou Fes.

• db.mycol.update(critere, donnée) : est utilisée pour modifier un document.

```
db.mycol.update({'title':'MongoDB Overview'}, {$set: {'title':'New
MongoDBTutorial'}})
db.mycol.update({'title':'MongoDB Overview'}, {$set: {'title':'New
MongoDBTutorial'}}, {multi:true})
db.mycol.update({}, {$inc: {salaire: 400}}, {multi:1})
db.mycol.update({}, {$push: {loisirs: "sport"}}, {multi:1})
db.mycol.update({}, {$rename: {loisirs: "hobbies"}}, {multi:1})
db.mycol.update({}, {$unset: {loisirs: 1}}, {multi:1})
```

• db.mycol.remove(critère) : supprime des documents d'une collections

```
db.mycol.remove({'title':'MongoDB Overview'}) : supprime
plusieurs documents
db.mycol.remove({'title':'MongoDB Overview'}, 1) :
supprime seulement le premier
db.mycol.remove() : supprime tous les documents
```

• Une agrégation permet de transformer et de combiner des documents dans les collections

```
db.mycol.aggregate([{$group:{_id:'$sexe', total: {$sum:1}}}])
db.mycol.aggregate([{$group:{_id:'$sexe', som: {$sum:'$salaire'}}}])
db.mycol.aggregate([{$group:{_id:'$sexe', max: {$max:'$salaire'}}}])
db.mycol.aggregate([{$group:{_id:'$sexe', min: {$min:'$salaire'}}}])
db.mycol.aggregate([{$group:{_id:'$sexe', moy: {$avg:'$salaire'}}}])
```