

TP MongoDB : interface client

Lancer le serveur mongodb :

```
mongod
```

Se connecter à mongo par le client :

```
mongo
```

Afficher la liste des bases de données :

```
show dbs
```

Se positionner sur la base « blog » :

```
use blog
```

Afficher la liste des bases de données à nouveau.

Lister les collections. Il n'y en a aucune.

Insérer un auteur :

```
db.author.insert({ name : 'Doe', firstname : 'John', age : 30, topics : [ 'dev', 'admin' ], company :  
{ name : 'SQLI', title : 'developer' } })
```

Liste à nouveau les collections. La collection author a été créée.

Requête simple :

```
db.author.find()
```

On remarque la création automatique de l'id.

Affichage mis en forme :

```
db.author.find().pretty()
```

Créer plusieurs auteurs dont certains avec des topics en commun, d'autres non. De même pour l'entreprise.

Requête par critère :

```
db.author.find({ name : 'Doe' })  
db.author.find({ age : { $gte : 20 } })  
db.author.find({ "company.name" : 'SQLI' })  
db.author.find({ firstname : { $in : [ 'John', 'Elvis' ] } })  
db.author.find({ topics : { $elemMatch : { $eq: 'dev' } } })
```

Supprimer un élément :

```
db.author.remove({ _id: ObjectId("591405caa239e43fb893175c") })
```

Ou plusieurs, les critères sont les même que pour le find :

```
db.author.remove({ age : { $gte : 20 } })
```

Supprimer une collection :

```
db.author.drop()
```

Supprimer une base de donnée :

```
db.dropDatabase()
```

Mettre à jour un document

```
db.author.update({ name: "Doe"}, {test: 1})
```

Attention : la syntaxe est trompeuse, on écrase le contenu du document. Pour ajouter ou modifier un attribut du document :

```
db.author.update({ name: "Doe"}, {$set : { toto: 1}})
```

TP MongoDB : PyMongo

A vous de jouer. En vous appuyant sur les commandes vu ensemble. Créer un programme Python qui par plusieurs fonctions permettra :

- Créer des auteurs. Ajoutez la date de création
- Créer un article (collection article) : title, content, category, topics (tableau), date de création
- Publier un commentaire sur un article : pseudo, content, date de création, approved (booléen)

Nous verrons ensemble comment utiliser les capacités de Map/reduce de MongoDB avec l'aggrégation sur les données créer ainsi que sur les jeux de données fournis dans la machine virtuelle pour une volumétrie conséquente.

```
mongoimport companies.json
```

Avec le framework d'aggrégation calculez :

- Nombre moyen d'employé
- Les 5 entreprises avec le plus d'employés, le moins d'employés
- Nombre d'entreprise par état
- Nombre d'acquisition par an
- Montant moyen des levées de fond
- La liste des 10 tags les plus représentés

TP Cassandra : ligne de commande

Lancer le serveur mongod :

```
cassandra
```

Se connecter à mongo par le client :

cqlsh

TP Cassandra : Python

A l'aide de ce que l'on a pu utiliser ensemble. Créer un programme qui va insérer des statistiques en base. Nous allons les simuler :

- Une boucle infini qui s'exécute toutes les secondes :
 - Une chance sur 5 d'insérer des données
 - Insérer un document avec
 - une action parmi une dizaine (view_homepage, add_product, show_basket ...)
 - une date