MongoDB

1. Premiers pas

1.1. Mac OS

On se base sur :
https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/

Si pas déjà fait, installer Homebrew :
```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

1.1.1. Installation

Dans un terminal :
```
$ brew update
$ brew install mongodb
```

Créer le répertoire de stockage:
```
$ sudo mkdir -p /data/db
$ sudo chmod 777 /data/db
```

1.1.2. Démarrage

Dans un terminal, démarrer le serveur :
```
$ mongod
```

Dans un second terminal (laisser le premier actif), démarrer le shell Mongo :
```
$ mongo
```
C'est dans ce terminal qu'on écrira les requêtes. Une alternative possible est l'IDE Robo3T.

1.2. Debian / Ubuntu

https://docs.mongodb.com/manual/administration/install-on-linux/

1.3. Documentation

https://docs.mongodb.com/

Comparaison avec la syntaxe SQL :
https://docs.mongodb.com/manual/reference/sql-comparison/

Guides de démarrage :
https://docs.mongodb.com/manual/tutorial/getting-started/

http://blog.xebia.fr/2010/12/15/mongodb-en-pratique/
https://scotch.io/tutorials/an-introduction-to-mongodb
https://www.codeproject.com/Articles/1037052/Introduction-to-MongoDB

1.4. Création d'une collection à partir d'un fichier JSON

Par exemple, création de la **collection *emails*** dans la **base de données *test*** (dans un 3ème terminal)

```
$ mongoimport --db test --collection emails --file enron.json
```

2. TP

2.1. Emails

Décompresser le fichier enron.json.bz2 (peut être fait en double-cliquant sur le fichier dans l'explorateur):

```
$ bunzip2 enron.json.bz2
```

1. Importer enron.json dans une collection nommée "mails"
2. Nombre total d'emails ?
3. Nombre total d'emails envoyés avec des adresses du domaine enron.com
4. Compter les emails transférés ("*subject"* commence par "FW:")
5. Combien de temps a pris la dernière requête ?
6. Créer un index avec le bon champ pour faire en sorte que la requête précédente soit plus rapide
7. Combien de temps a pris cette nouvelle requête ?
8. Trouver la date, l'expéditeur (*sender*) et l'objet (*subject*) de tous les messages envoyés à rosalee.fleming@enron.com
9. Supprimer lizard_ar@yahoo.com dans tous les emails dont il est le destinataire. Ne pas oublier les requêtes nécessaires à la vérification de la suppression.
10. Ajouter rob.bradley@enron.com comme destinataire de tous les emails envoyés par rosalee.fleming@enron.com. Ne pas oublier les requêtes nécessaires à la vérification de l'ajout.

2.2. Codes postaux

Décompresser le fichier zips.json.bz2 et importer zips.json dans une collection "zips"

1. Lister les 10 zones (une zone correspond à un code postal = 1 "zip code") les plus peuplées en Californie. Même chose pour la Louisiane
2. Lister les 10 zones suivantes (classées entre 11 et 20)
3. Ajouter un champ "country" avec la valeur "USA" à tous les documents
4. Lister toutes les zones avec plus de 100 000 habitants situées à l'ouest du méridien -74
5. Quelle est la zone la plus proche des coordonnées -73.996705, 40.74838 ?
   Il faut créer un index :
   ```
   > db.zips.createIndex( { loc : "2d" } ) ;
   ```
6. Lister les villes qui sont distantes de moins de 5 km de -73.996705, 40.74838
7. Lister les villes qui ont plus de 500 000 habitants
8. La requête suivante ne retourne aucun résultat :
   ```
   > db.zips.find({city:{regex:"^N"}})
   ```
   Quel est le problème ?
   Pourquoi MongoDB ne retourne pas d'erreurs lors de son exécution ?
   Quelle est la requête correcte ?