

TP "MongoDB"

Objectifs

Ce TP a pour objectif de vous familiariser avec la base de données NoSQL MongoDB. MongoDB est une base **orientée documents** car les éléments enregistrés dans la base sont des documents JSON. Vous apprendrez à faire des requêtes à partir de lignes de commandes et d'un programme Python.

Pour communiquer avec la base de données, vous pourrez utiliser les clients suivants :

Le Shell MongoDB depuis le terminal `mongo`
Linux

Depuis le client Web mongo-express

Voici deux commandes mongo utiles pour commencer le TP. Pour ce TP, **nommez votre base en utilisant votre identifiant étudiant**. Si votre login est *bowie65d*, alors vous appellerez votre base *bowie65d*.

```
use <dbname>; // Sélection de la base de données.  
// La base est automatiquement créée si elle n'existe pas  
show collections; // Liste des collections existantes d'une base.  
// Une collection est à ensemble de documents JSON.
```

Documentation

Site officiel	https://docs.mongodb.org/manual/applications/crud/
Tutorials point	http://www.tutorialspoint.com/mongodb/index.htm
PyMongo	https://api.mongodb.org/python/current/

Questions

1. Créer votre base de données depuis le client mongo avec la commande "use". Créer dans cette base une collection appelée "movies".
2. Depuis le client mongo, ajouter le document ci-dessous dans la collection "movies". Quel est le résultat de la requête ? Vérifier que le document a bien été ajouté via mongo-express. Que peut-on observer en regardant le code source du document ?

```
{  
  "titre": "Star Wars, épisode I : La Menace fantôme",  
  "realisateur": "George Lucas"  
}
```

3. Importer le fichier books.json dans la collection "books" en utilisant la commande mongoimport (il est nécessaire de quitter le shell mongo au préalable). Vérifier que les documents ont bien été importés via mongo-express.
4. Écrire la requête qui permet de retrouver le(s) livre(s) dont l'auteur est "J. R. R. Tolkien".
5. Écrire la requête qui permet de retrouver le titre du/des livre(s) dont le paperback est vendu à la Fnac **et** dont le prix est inférieur à 11 euros et les ordonner par leur titre dans l'ordre alphabétique.
6. Écrire la requête qui permet de retrouver le(s) livre(s) n'existant pas au format Kindle **ou** dont le prix de la version Kindle est inférieur à 6 euros.
7. Écrire la requête qui permet de retrouver le nombre de(s) livre(s) dont la description contient le mot "dwar" (vérifier que votre requête est insensible à la case) dans leur titre.
8. Pour le livre dont le titre est "The Hobbit; or, There and Back Again", modifier le prix de sa version Kindle à 7.00€ et ajouter lui un attribut "personnages" associé aux 3 valeurs "Gandalf", "Bilbon", "Elrond" (ne pas oublier d'utiliser l'opérateur \$set). Vérifier que le document a bien été modifié via mongo-express.
9. Ajouter "Cultura" à la liste des magasins qui vendent le livre "Tolkien's World from A to Z: The Complete Guide to Middle-Earth" au format paperback. Vérifier que le document a bien été modifié via mongo-express.
10. Supprimer le(s) livre(s) ayant un éditeur **ou** une liste de personnages renseignée. Vérifier que les données ont bien été supprimées via mongo-express.
11. Créer un index pour la collection "books" indexant le champ "titre". Quel est la taille de cet index ?
12. Supprimer la collection "books" et la réimporter en utilisant la commande utilisée dans la question 2. Sélectionner ensuite la moyenne du prix des version Kindle par auteur (en utilisant une agrégation).

13. En utilisant le patron de conception MapReduce (reproduit ci-dessous), compter le nombre de livres disponibles dans le format "paperback" vendus pour chaque magasin et stocker les résultats dans la collection "books_count_per_shop".

```
db.books.mapReduce(  
  function () {  
    ...  
  },  
  function (key, values) {  
    return ...;  
  },  
  {  
    query: {},  
    out: "<collection_out>"  
  }  
);
```

14. En utilisant le patron de conception MapReduce, sélectionner le(s) livre(s) étant uniquement vendu(s) chez "Amazon" et stocker les résultats dans la collection "books_amazon".
15. Implémenter l'algorithme MapReduce de la question précédente en python, stocker les résultats dans la collection "books_amazon_python" et afficher leur titre. S'appuyer sur la bibliothèque PyMongo déjà installée pour cela.