TP MongoDB

Ce TP a pour objectif de vous familiariser avec MongoDB et son shell. Vous allez dans la suite charger un jeu de données et écrire les requêtes permettant chacune de répondre à un besoin en information précis.

Introduction

MongoDB est une base de données NoSQL orientée documents. Une base de données MongoDB consiste en un ensemble de collections, elles-mêmes formées de documents, construits selon un schéma dynamique et non prédéfini.

Un document est un ensemble ordonné de paires clé-valeurs, où les clés sont des chaines de caractères et les valeurs peuvent une instance de n'importe quel type de donnée parmi ceux prédéfinis (*null*, *boolean*, *number*, *string*, *date*, *regular expression*, *array*, *object id*, *binary data* et *code*), ou bien un autre document.

Préparation

1. Télécharger, installer et lancer un serveur MongoDB.
    - https://www.mongodb.com/download-center#community
    - https://docs.mongodb.com/manual/installation/#mongodb-community-edition
2. Télécharger les fichiers suivants :
    - http://camillepradel.fr/teaching/nosql/movielens_export/movielens_movies.json
    - http://camillepradel.fr/teaching/nosql/movielens_export/movielens_users.json
3. Importer ces deux fichiers dans MongoDB :

```
# $MONGO_HOME/bin/mongoimport --db MovieLens --collection movies --file
movielens_movies.json

connected to: localhost

imported 3883 objects

# $MONGO_HOME/bin/mongoimport --db MovieLens --collection users --file
movielens_users.json

connected to: localhost

[...]

imported 6040 objects
```

Ces commandes permettent de créer les collections movies et users dans la base de données MovieLens et de les peupler avec les données issues des fichiers json.

4. Connectez-vous à la base de données MovieLens :

```
# $MONGO_HOME/bin/mongo MovieLens

MongoDB shell version: 3.2.10

connecting to: MovieLens

>
```

Shell MongoDB

Cette dernière commande lance le shell MongoDB qui est en fait un interpreteur javascript complet. Vous pouvez donc y exécuter n'importe quel code javascript. Le shell propose en plus un certain

nombre de commandes spécifiques courantes dans les interfaces avec une base de données, par exemple :

- `use <db name>` permet de connecter la session à la base de donnée *db name*.
- `show collections` affiche les collections de la base de données courante.
- `help` donne un aperçu des commandes les plus importantes et de leur usage.

Schéma de la base de données

Comme dit plus haut, les documents d'une collection ne sont pas soumis à un schéma fixe. Cependant, les documents de chaque collection ont une structure similaire. Nous donnons dans cette partie un exemple de chaque collection.

La collection movies contient des informations sur les films, c'est-à-dire leur id, titre et genre.

```
> db.movies.findOne()
{
        "_id" : 1,
        "title" : "Toy Story (1995)",
        "genres" : "Animation|Children's|Comedy"
}
```

La collection users contient des informations portant sur les utilisateurs et les notes qu'ils ont données aux films. Parmi les informations sur les utilisateurs, on trouve leur id, nom, âge, occupation et sexe. Les notes donées par chaque utilisateur sont représentées dans un tableau de document, chaque document contenant l'id d'un film (faisant référence aux id de la collection movies), la note attribuée et la date à laquelle l'utilisateur a laissé la note.

```
> db.users.findOne({},{movies : {$slice : 2}});
{
        "_id" : 6038,
        "name" : "Yaeko Hassan",
        "gender" : "F",
        "age" : 95,
        "occupation" : "academic/educator",
        "movies" : [
                {
                        "movieid" : 1419,
                        "rating" : 4,
                        "timestamp" : 956714815
                },
                {
                        "movieid" : 920,
                        "rating" : 3,
                        "timestamp" : 956706827
                }
        ]
}
```

Requêtes simples

Question 1. Combien y a-t-il d'utilisateurs dans la base de données ?

- https://docs.mongodb.com/manual/reference/command/count/

Question 2. Combien y a-t-il de films dans la base de données ?

Question 3. Quelle est l'occupation de Clifford Johnathan ? Ecrivez une requêtes dont la réponse affiche uniquement son nom et son occupation.

- http://docs.mongodb.org/manual/reference/method/db.collection.find/

Question 4. Combien d'utilisateurs ont entre 18 et 30 ans (inclus) ?

Question 5. (optionnelle) Combien d'utilisateurs sont artistes (*artist*) ou scientifiques (*scientist*) ?

Question 6. Quelles sont les dix femmes auteurs (*writer*) les plus âgées ?

Question 7. Quelles sont toutes les occupations présentes dans la base de données ?

Insertions, mises-à-jour et suppressions

Question 8. Insérer un nouvel utilisateur dans la base de données (vous, par exemple). Ne pas inclure pour l'instant le champ *movies*.

Question 9. Choisir un film de la collection movies et mettre à jour l'entrée insérée précédemment en ajoutant le champ movies respectant le schéma adopté par les autres entrées. Pour le champ timestamp, utiliser l'heure courante : `Math.round(new Date().getTime() / 1000)`

Question 10. Supprimer l'entrée de la base de données.

Question 11. Pour tous les utilisateurs qui ont pour occupation "programmer", changer cette occupation en "developer".

Expressions régulières

- http://docs.mongodb.org/manual/reference/operator/query/regex/

Question 12. Combien de films sont sortis dans les années quatre-vingt ? (l'année de sortie est indiquée entre parenthèses à la fin du titre de chaque film)

Question 13. (optionnelle) Combien de films sont sortis entre 1984 et 1992 ?

Question 14. Combien y a-t-il de films d'horreur?

Question 15. (optionnelle) Combien de films ont pour type à la fois "Musical" et "Romance"?

ForEach

Question 16. Comme vous avez pu le constater, stocker l'année de sortie du film dans son titre n'est pas très pratique. Modifier la collection movies en ajoutant à chaque film un champ year contenant l'année et en supprimant cette information du titre. Ne nombreuses méthodes peuvent répondre à ce besoin ; privilégier au maximum les approches exploitant les fonctionnalités de MongoDB (il est par exemple déconseillé, pour des raisons évidentes de performances, de demander l'intégralité des films à la base de données, de les stocker dans une liste javascript, puis d'itérer sur cette liste pour calculer les nouvelles valeurs de champs et mettre à jour les éléments, toujours en javascript).

- http://docs.mongodb.org/manual/reference/method/cursor.forEach/
- http://docs.mongodb.org/manual/reference/method/cursor.snapshot/

Question 17. Modifier la collection movies en replaçant pour chaque film la valeur du champ genres par un tableau de chaines de caractères.

Question 18. Modifier la collection users en remplaçant pour chaque utilisateur le champ timestamp par un nouveau champ date, de type Date. Le champ timestamp est exprimé en secondes depuis l'*epoch Unix*, c'est-à-dire le 1er janvier 1970. En javascript, les instances de Date sont crées en utilisant le nombre de millisecondes depuis l'*epoch Unix*.

Requêtes sur des tableaux

Lecture

Question 19. Combien d'utilisateurs ont noté le film qui a pour id 1196 (Star Wars: Episode V - The Empire Strikes Back (1980)) ?

Question 20. Combien d'utilisateurs ont noté tous les films de la première trilogie Star Wars (id 260, 1196, 1210) ?

- http://docs.mongodb.org/manual/reference/operator/query/all/

Question 21. Combien d'utilisateurs ont notés exactement 48 films ?

- http://docs.mongodb.org/manual/reference/operator/query/size/

Notez que $size ne peut être apparié qu'à des nombres exacts. La sélection des utilisateurs qui ont vu plus d'un certain nombre de films doit être effectuée en deux étapes ; c'est le sujet des questions suivantes.

Question 22. Pour chaque utilisateur, créer un champ num_ratings qui indique le nombre de films qu'il a notés.

Question 23. Combien d'utilisateurs ont noté plus de 90 films ?

Question 24. (optionnelle) Combien de notes ont été soumises après le 1er janvier 2001 ?

Question 25. Quels sont les trois derniers films notés par Jayson Brad ?

Question 26. (optionnelle) Obtenez les informations portant uniquement sur Tracy Edward et sa note du film Star Wars: Episode VI - Return of the Jedi, qui a pour id 1210.

Question 27. (optionnelle) Combien d'utilisateurs ont donné au film "Untouchables, The" la note de 5.

Ecriture

Question 28. L'utilisateur Barry Erin vient juste de voir le film Nixon, qui a pour id 14 ; il lui attribue la note de 4. Mettre à jour la base de données pour prendre en compte cette note. N'oubliez pas que le champ num_rattings doit représenter le nombre de films notés par un utilisateur.

Question 29. L'utilisatrice Marquis Billie n'a en fait pas vu le film "Santa with Muscles", qui a pour id 1311. Supprimer la note entrée par mégarde dans la base de données.

Question 30. (optionnelle) Les genres du film "Cinderella" devraient être Animation, Children's et Musical. Modifier en une seule requête le document correspondant pour qu'il contienne ces trois genres sans doublon.

Références

MongoDB n'intègre pas de support des jointures. Le plus souvent, les références sont dénormalisées (dupliquées) et stockées sous forme de documents internes. Mais il est parfois plus judicieux (ou inévitable) de stocker des informations liées dans des documents différents appartenant à des collections différentes et de les relier par des références. Il y a deux façons de faire ça :

- Références manuelles : le champs _id d'un document est stocké dans un autre document. C'est le cas dans notre jeu de données, où les champs _id des films sont stockés dans les tableaux de votes.
- DBRef : DBRef est une convention qui permet de référencer un document. Elle est formée par ses information de collection, son id et éventuellement la base de données où il est enregistré. Il n'est pas conseillé d'utiliser cette méthode car elle n'est pas supportée par toutes les opérations.

Question 31. (optionnelle) Modifier la collection users en y ajoutant un champs movies.movieref qui contient une DBRef vers le film concerné.

- http://docs.mongodb.org/manual/reference/database-references/

Question 32. (optionnelle) En exploitant le champ nouvellement créé, déterminer combien d'utilisateurs ont noté le film Taxi Driver.

Question 33. (optionnelle) En exploitant le champ nouvellement créé, déterminer combien d'utilisateurs ont attribué au film Taxi Driver une note de 5.

Vous pouvez désormais supprimer le champ movieref qui ne sera plus utilisé dans la suite :

```
> db.users.find().forEach(function(u) {
    for (i = 0; i < u.movies.length; i++) {
        delete u.movies[i].movieref;
    }
    db.users.save(u);
});
```

Notez qu'il n'est pas possible de mettre à jour avec un seul update() tous les éléments d'un tableau.

Index

Question 34. Chercher le nom des dix femmes qui ont noté un film le plus récemment. Notez que si l'on ajoute la fonction explain() à la fin de la requête, on obtient des informations sur son exécution.

Question 35. Créer un index sur les champs gender et movies.date.

- http://docs.mongodb.org/manual/reference/method/db.collection.ensureIndex/

Question 36. Exécuter à nouveau la requête 34. Commenter les différences.

Agrégats

Question 37. Montrer combien de films ont été produits durant chaque année des années 90 ; ordonner les résultats de l'année la plus à la moins fructueuse.

- http://docs.mongodb.org/manual/core/aggregation-pipeline
- http://docs.mongodb.org/manual/reference/operator/aggregation-pipeline

Remarque : cette requête est bien plus simple si l'on exploite les informations crées dans la requête 16.

Question 38. Quelle est la note moyenne du film Pulp Fiction, qui a pour id 296 ?

Question 39. En une seule requête, retourner pour chaque utilisateur son id, son nom, les notes maximale, minimale et moyenne qu'il a données, et ordonner le résultat par note moyenne croissante.

Question 40. (optionnelle) Quel est le mois au cours duquel le plus de notes ont été attribuées ?

Question 41. (optionnelle) Créer une nouvelle collection join qui associe à chaque film son _id, son titre, ses genres, son année et toutes les notes qui lui ont été attribuées.

- Indice : utiliser aggregate + insert + forEach.

Map-Reduce

Comme alternative au pipeline d'agrégation, il est possible d'utiliser Map-Reduce pour effectuer des agrégations. Cette deuxième approche est souvent moins performante, mais offre beaucoup plus de souplesse et permet d'écrire du code librement.

- http://b3d.bdpedia.fr/mapreduce.html#s2-frameworks-mapreduce-mongodb
- http://docs.mongodb.org/manual/core/map-reduce

Question 42. (optionnelle) Quel est le genre le plus populaire en termes de nombre de notes ?

- Indice : utiliser la collection join créée dans la question 41.

Question 43. (optionnelle) Quel est le genre le mieux noté (celui dont la moyenne de toutes les notes est la plus élevée) ?

- Indice : utiliser finalize.

Question 44. (optionnelle) Déterminer, pour chaque année de production, le film qui a reçu le plus grand nombre de notes.

Question 45. (optionnelle) Déterminer, pour chaque année de production, le film qui a reçu la meilleure note moyenne.

Question 46. (optionnelle) Déterminer, pour chaque année de production, le film qui a reçu la meilleure note moyenne parmi les films qui ont reçu au moins 1000 notes.

- Indice : utiliser l'option query de Map-Reduce pour filtrer les documents avants d'exécuter les autres fonctions.