Projet « Gestion des données à large échelle »

Projet 2 :

BD NOSQL, Modélisation, Création, Interrogation, Mise à jour et MapReduce

Préambule :

Ce second projet est composé de 3 parties : Les opérations de bases pour la création et la mise à jour d'une BD Mongo, la modélisation, l'interrogation et l'affichage des données d'une BD Mongo et enfin mettre en œuvre des traitements Map/Reduce en javascript.

Partie 1 : **Création d'une BD Mongo et opération CRUD**

1. Créer une nouvelle base de données nommée BdProduit et s'assurer qu'elle est sélectionnée.
2. Créer une nouvelle collection nommée produits et y insérer le document suivant:

```
nom: Macbook Pro
fabriquant: Apple
prix: 11435,99
options: Intel Core i5
         Retina Display
         Long life battery
```

3. Rajouter deux autres documents dans produits :

```
nom: Macbook Air
fabriquant: Apple
prix: 125794,73
ultrabook: true
options: Intel Core i7
         SSD
         Long life battery
```

```
nom: Thinkpad X230
fabriquant: Lenovo
prix: 114358,74
ultrabook: true
options: Intel Core i5
         SSD
         Long life battery
```

4. Proposez les requêtes de lecture suivantes:

   a. Récupérer tous les produits.
   b. Récupérer le premier produit
   c. Trouver l'id du Thinkpad et faites la requête pour récupérer ce produit avec son id.
   d. Récupérer les produits dont le prix est supérieur à 13723 DA
   e. Récupérer le premier produit ayant le champ ultrabook à true
   f. Récupérer le premier produit dont le nom contient Macbook
   g. Récupérer les produits dont le nom commence par Macbook
   h. Supprimer les deux produits dont le fabricant est Apple.
   i. Supprimer le Lenovo X230 en utilisant uniquement son id.

Partie 2 : **La modélisation, l'interrogation et l'affichage des données d'une BD Mongo**

Comme présenté en cours, les documents d'une BD NOSQL ne sont pas soumis à un schéma fixe. Cependant, les documents de chaque collection ont une structure similaire. Nous donnons pour ce projet un exemple de chaque collection sous format JSON. La collection ***movies*** contient des informations sur les films, c'est-à-dire leur id, titre et genre.

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 données par chaque utilisateur sont représentées dans un tableau d'objets JSON, chaque objet 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.

Movies

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

Users

```
{
     "_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
          }
```

Modélisation

1) Proposer un modèle conceptuel UML permettant de représenter les données décrites dans les 2 documents JSON. Cette modélisation devrait permettre de mener à ma mise en œuvre proposer par les 2 documents JSON. A partir du modèle conceptuel UML proposé, suggérer une autre implémentation possible de la même base de données NOSQL. Discuter les avantages et les inconvénients de chacune des 2 approches de la BD NOSQL. Discuter les aspects optimisation et performances par rapport aux deux implémentations de la BD NOSQL.

2) Proposer une implémentation en BD relationnelles des 2 collections ***movies et users.*** Illustrer cette implémentations par des tables.

Requêtes simples

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

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 requête dont la réponse affiche uniquement son nom et son occupation.

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

Question 5. () 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 pasinclure pour l'instant le champ *movies*.

Question 9. Choisir un film de la collection movies et mettre à jour l'entrée insérée précédemmenten ajoutant le champ movies respectant le schéma adopté par les autres entrées. Pour le champ timestamp, utiliser l'heure courante (Voir sur Web).

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

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

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

Requêtes sur des tableaux

Lecture

Question 16. Combien d'utilisateurs ont noté le film qui a pour id 1196 (Star Wars: Episode V - TheEmpire Strikes Back (1980)) ?

Question 17. Combien d'utilisateurs ont noté tous les films de la première trilogie Star Wars (id 260,1196, 1210)

Question 18. Combien d'utilisateurs ont notés exactement 48 films ?

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 questionssuivantes.

Question 19. Pour chaque utilisateur, créer un champ num_ratings qui indique le nombre de filmsqu'il a notés.

Question 20. Combien d'utilisateurs ont noté plus de 90 films ?

Question 21. () Combien de notes ont été soumises après le 1er janvier 2001 ?

Question 22. Quels sont les trois derniers films notés par Jayson Brad ?

Question 23. () Obtenez les informations portant uniquement sur Tracy Edward et sanote du film Star Wars: Episode VI - Return of the Jedi, qui a pour id 1210.

Question 24. () Combien d'utilisateurs ont donné au film "Untouchables, The" la note de5.

Ecriture

Question 25. L'utilisateur Barry Erin vient juste de voir le film Nixon, qui a pour id 14 ; il lui attribuela 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 26. L'utilisatrice Marquis Billie n'a en fait pas vu le film "Santa with Muscles", qui a pour id1311. Supprimer la note entrée par mégarde dans la base de données.

Question 27. () Les genres du film "Cinderella" devraient être Animation, Children's etMusical. Modifier en une seule requête le document correspondant pour qu'il contienne ces troisgenres 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 forméepar 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 partoutes les opérations.
Question 28. () Modifier la collection users en y ajoutant un champs movies.movieref quicontient une DBRef vers le film concerné. http://docs.mongodb.org/manual/reference/database-references/

Question 29. () En exploitant le champ nouvellement créé, déterminer combiend'utilisateurs ont noté le film Taxi Driver.

Question 30. () En exploitant le champ nouvellement créé, déterminer combiend'utilisateurs ont attribué au film Taxi Driver une note de 5.

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

Index

Question 31. 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 32. Créer un index sur les champs gender et movies.date.

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

Question 33. Exécuter à nouveau la requête 31. Commenter les différences.

Agrégats

Question 34. 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.

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

Question 35. Quelle est la note moyenne du film Pulp Fiction, qui a pour id 296 ?

Question 36. En une seule requête, retourner pour chaque utilisateur son id, son nom, les notesmaximale, minimale et moyenne qu'il a données, et ordonner le résultat par note moyenne croissante.

Question 37. () Quel est le mois au cours duquel le plus de notes ont été attribuées ?

Question 38. () Créer une nouvelle collection join qui associe à chaque film son _id, sontitre, ses genres, son année et toutes les notes qui lui ont été attribuées.

Partie 3 : **Mise en œuvre des traitement MAP/Reduce (Nicolas travers @CNAM)**

Télécharger le jeu de données JSON "tourPedia" disponible sur le campus :
- Décompresser l'archive
- Importer le fichier dans une base "tourPedia" et une collection "paris"

TourPedia est le Wikipédia du tourisme. Il contient des informations sur les hébergements, les restaurants, les points d'intérêt et les attractions de différentes régions d'Europe. Actuellement, seules huit villes sont couvertes : Amsterdam, Barcelone, Berlin, Dubaï, Londres, Paris, Rome et la Toscane. L'objectif est d'étendre ce service au monde entier.
Les données proviennent de quatre réseaux sociaux : Facebook, Foursquare, GooglePlaces et Booking. Elles ont été élaborées et intégrées afin de constituer un catalogue unique. Dans le jeu de données proposé, la ville de Paris est représenté.

Après vous êtes connecté sur Compass et regarder à quoi ressemble un document JSON sur les lieux de Paris. Dont voici (sur la page suivante) une extraction :
Regarder bien des documents JSON avec les différents champs bien renseignés (« services », « description »,….).

```
{
  "_id" : 455674,
  "name" : "Best Western Premier Trocadero La Tour",
  "category" : "accommodation",
  "location" : {
                "coord" : { "coordinates" : [2.354878, 48.866599], "type" : "Point" },
                "address" : "292 Rue Saint-Martin, Paris, France",
                "city" : "Paris"
               },
  "reviews" : [{
                 "wordsCount" : 30,
                 "rating" : 0,
                 "language" : "en",
                 "source" : "Foursquare",
                 "text" : "Nice beds, rooms andstaff. Perfect central location. Breakfast is very expensive for a contenintal
                           breakfast, however many bakeries and restaurants in the area. Will stay here again my next visit.",
                 "time" : "2010-09-30"
               }],
  "contact" : {
                "website" : "http://www.trocaderolatour.com",
                "GooglePlaces" : "https://plus.google.com/103448496999303589086/about?hl=en-US",
                "phone" : "+33 1 40 27 27 03",
                "foursquare" : "",
                "Booking" : "",
                "Facebook" : ""
               },
"description" : " Situé à 15 minutes à pied de la tour Eiffel, le Best Western Premier Trocadero La Tour bénéficie d'un
                  emplacement idéal pour découvrir Paris",
"services" : [
    "jardin",
    "petit-déjeuner en chambre",
    "bagagerie",
    "blanchisserie",
    "chambres non-fumeurs"
]
}
```


Exécuter la requête suivante :

```
mapFunction = function () {
        emit(this.category, 1);};
reduceFunction = function (key, values) {
        return Array.sum(values);};
queryParam = {"query":{}, "out":{"inline":true}};
db.paris.mapReduce(mapFunction, reduceFunction, queryParam);
```

3.1) Modifier les fonctions MAP et/ou REDUCE afin de répondre aux questions suivantes

a. Pour chaque catégorie, donner le nombre de services en français (utiliser la taille d'une liste en javascript "services.fr.length") ;
b. Pour chaque catégorie, donner le nombre de mots pour les commentaires en français ;
c. Pour chaque "type" de contact (s'il existe), donner le nombre de lieux associés ;
d. Donner le nombre de lieux pour chaque "nombre moyen de mots" par lieu (moyenne de reviews.wordsCount), pour les messages Facebook. Arrondir la moyenne à l'inférieur (Math.floor()) ;
e. Pour chaque source de commentaire, donner le nombre moyen de mots ;
f. Attention, la moyenne n'est pas une fonction associative (contrairement à la somme) - problème de reduce local et global.
g. Même question, mais en produisant la note moyenne (rating), min et max ;
h. Pour chaque langue de services, donner la liste distincte des services proposés ;
i. Nombre de langues de commentaires différents par type de source ;