

Systemes de gestion de donnees

TP NoSQL avec MongoDB

Connexion aux PC de l'Ensimag

MongoDB est installé sur le serveur `mongodb.ensimag.fr`. Pour s'y connecter :

A partir d'un PC de l'Ensimag (ou d'une connexion VPN Grenoble INP si vous avez installé le client Mongo 3.6 localement).

```
$ mongo -host mongodb.ensimag.fr -u <login> <login> -p
```

Cette commande se connecte à votre base de données de l'instance de Mongo sur le serveur (remplacez `<login>` par votre login). Si cette base de données n'existe pas, elle sera créée à la première insertion de données.

Chargement des données dans la base

Entrez les commandes suivantes dans le shell Mongo :

```
$ use TP
$ var resto = db.restaurants.find()
$ use <login>
$ resto.forEach(function(r) { db.restaurants.insert(r); })
```

L'import de ces données a pour effet de créer une collection de documents. La plupart des documents de cette collection ont le schéma suivant :

```
{
  "address" : {
    "building" : "----",
    "coord" : [ -- , -- ],
    "street" : "----",
    "zipcode" : "----"
  },
  "telephoneNumber" : "----",
  "cuisine" : "----",
  "grades" : [
    {
      "date" : ISODate("----"),
      "grade" : "----",
      "score" : --
    }
  ],
  ...
},
"name" : " ----",
"restaurant_id" : "----"
}
```

Interrogation de la base

À vous de jouer maintenant. Toutes les données sont chargées. Vous allez maintenant interroger la base de données que vous avez créée.

Question 1 : Donnez la liste de tous les restaurants de la collection, triée par ordre de noms croissant.

Question 2 : Donnez la liste de tous les restaurants proposant une cuisine de type `"Italian"` et affichez, pour chacun d'entre eux, le nom, le code postal et les coordonnées géographiques. De plus, assurez-vous que la réponse soit ordonnée selon la clef de tri (code postal croissant, nom décroissant).

Question 3 : Donnez la liste de tous les restaurants italiens ayant pour code postal `"10075"` et pour lesquels le numéro de téléphone est fourni dans la base de données. Affichez nom, code postal et numéro de téléphone.

Question 4 : Trouvez tous les restaurants ayant au moins un score supérieur ou égal à 50.

Question 5 : Donnez la liste de tous les restaurants qui sont soit italiens, soit ayant le code postal `"10075"`.

Question 6 : Donnez la liste de tous les restaurants dont le code postal est `"10075"` ou `"10098"`, dont la cuisine est soit italienne, soit américaine, et ayant au moins un score supérieur ou égal à 50.

Question 7 : Donnez la liste de tous les restaurants ayant au moins un score concernant le grade *customer service* (*C*), *price* (*P*) ou *quality* (*Q*). Affichez simplement les noms, la cuisine et le code postal.

Mises à jour

Question 8 : Modifiez le type de cuisine du restaurant `"Juni"` pour le mettre à `"American (new)"`. Enregistrez également dans un champ `"lastModified"` la date et l'heure du système au moment où la modification est effectuée. S'il existe plusieurs restaurants portant le même nom, seul le premier doit être modifié.

Question 9 : Changez l'adresse du restaurant dont l'identifiant (`restaurant_id`) est `"41156888"` en `"East 31st Street"`.

Question 10 : Changez le type de cuisine de tous les restaurants dont le code postal est `"10016"` et le type de cuisine `"Other"` en `"Cuisine to be determined"`. Enregistrez également dans un champ `"lastModified"` la date et l'heure du système au moment où la modification est effectuée.

Question 11: Remplacez toute l'information concernant le restaurant dont l'identifiant est `"41154403"` par l'information suivante :

```
"name" : "Vella 2",
"address" : {
  "city" : "1480",
  "street" : "2 Avenue",
  "zipcode" : "10075"
}
```

Interrogation complexe (agrégation)

Question 12 : Dressez la liste des types de cuisine représentés dans la base. Pour chaque type, affichez le nombre de restaurants associés. Ordonnez le résultat par nombre de restaurants décroissants.

Question 13 : Affichez, pour chaque code postal, le nombre de restaurants italiens ayant ce code postal. Ordonnez le résultat par nombre de restaurants décroissants.

Question 14 : Considérez les restaurants italiens dont l'identifiant (`restaurant_id`) est supérieur ou égal à "41205309", et possédant un attribut `averagePrice`. Calculez la moyenne de ces prix moyens. Puis refaites la même opération en calculant la moyenne par code postal.

Références entre collections et jointures

Tout cela est bien intéressant, mais notre collection de restaurants est bien seule pour le moment. Nous allons maintenant travailler avec plusieurs collections, liées entre elles par des références (à la manière des contraintes d'intégrité référentielles, ou clefs étrangères, dans les bases de données relationnelles).

Question 15 : Créer une nouvelle collection appelée `comments`, dans la même base de données.

Question 16 : Insérez trois documents dans la collection précédemment créée, en utilisant le schéma suivant :

```
{
  "_id" : "----",
  "restaurant_id" : "----",
  "client_id" : "----",
  "comment" : "----",
  "date" : ISODate("----"),
  "type" : "----"
}
```

Quelques précisions utiles :

- les identifiants de restaurants doivent correspondre à des restaurants existants dans la collection `restaurants` ;
- vous devez fournir des commentaires de différents clients, et pour des restaurants différents ;
- l'attribut `type` peut prendre uniquement les valeurs `positive` ou `negative`.

Question 17 : Donnez la liste de tous les commentaires de votre base de données. Chaque commentaire doit contenir également toutes les informations concernant le restaurant auquel il se rapporte.

Question 18 : Insérez 7 autres documents dans la collection `comments`, en suivant le schéma précédemment décrit, et en respectant les règles suivantes :

- les identifiants de restaurants doivent correspondre à des restaurants existants dans la collection `restaurants` ;
- l'un des restaurants au moins doit avoir plusieurs commentaires ;
- l'un des clients au moins doit avoir commenté plusieurs fois ;
- l'un des clients au moins doit avoir commenté plusieurs fois le même restaurant ;
- l'attribut `type` peut prendre uniquement les valeurs `positive` ou `negative`.

Question 19 : Trouvez la liste des restaurants ayant des commentaires, et affichez pour chaque restaurant uniquement le nom et la liste des commentaires. Plusieurs stratégies sont possibles.

Index

Occupons-nous maintenant des aspects efficacité de notre base de données. Nous allons créer des index, afin d'optimiser l'accès à notre collection de restaurants.

Question 20 : Créez un index ascendant sur l'attribut `cuisine` de la collection `restaurants`.

Question 21 : Créez un autre index pour la collection `restaurants`, composé de l'attribut `cuisine` (ascendant) et de l'attribut `zipcode` (descendant).

Question 22 : Affichez la liste des index créés sur la collection `restaurants`.

Question 23 : Utilisez la méthode `explain` pour afficher le plan d'exécution d'une requête renvoyant tous les restaurants italiens. Quelles informations sont fournies par le système ?

Question 24 : Même question mais en ajoutant l'argument `"executionStats"` en paramètre de la méthode `explain`.

Question 25 : Supprimez les deux index que vous avez précédemment créés, puis affichez à nouveau les statistiques sur le plan d'exécution de la requête renvoyant tous les restaurants italiens. Que constatez-vous?