TP4 – MongoDB
Yelp

Pour ce TP, il est conseillé d'utiliser le terminal intégré à MongoDB Compass. Cela vous permettra d'utiliser les outils mis à disposition par Compass (comme l'analyse par exemple) tout en apprenant la syntaxe du langage MongoDB.

Yelp est un site répertoriant des entreprises, des commerces et des artisans et recueillant des avis sur les adresses répertoriées. Dans ce TP, nous travaillerons sur un extrait de leur base de données nord-américaine se concentrant sur 8 régions aux États-Unis et au Canada.

Découverte des données.

Q_1). Récupérer le fichier `yelp.json` sur Moodle.

Q_2). Importer les données en tapant la commande suivante dans le terminal (le terminal système et non pas celui intégré à Compass) :

```
mongoimport --host londres.uca.local --authenticationDatabase "admin"
                -u "<login>" --db="<dblogin>" --collection="yelp" yelp.json
```

Vous devriez obtenir un message similaire à celui-ci :

```
150346 document(s) imported successfully. 0 document(s) failed to import.
```

Q_3). Lancer `mongodb-compass` et vérifier que les données sont bien présentes.

Si elles ne sont pas visibles et que vous n'avez pas eu de message d'erreur lors de l'import, cliquer sur l'icône de rafraîchissement en haut à gauche de la fenêtre.

Q_4). Combien y a t'il de documents ?

Q_5). Quelle est la taille de la base de données ?

Q_6). Afficher un unique document contenu dans cette collection. Que représente ce document ? De quelles données dispose-t'on ?

Q_7). Utiliser l'interface de Compass pour parcourir quelques documents. Que remarquez-vous ? Ont-ils tous la même structure ?

Q_8). Analyser la collection grâce à Compass. Cliquer plusieurs fois sur le bouton 

(en n'ayant rien renseigné dans la barre Filter). Que remarquez-vous ? Pourquoi ?

Requêtes.

Pour pouvoir répondre aux questions suivantes, vous allez devoir étudier les documents constituants cette base de données ainsi que leur structure afin de comprendre comment sont organisées les données

et d'écrire une requête permettant d'obtenir les informations recherchées.

Q_9). Lister les noms des commerces répertoriés dans la ville de Tampa.

Q_{10}). Lister les 10 premières adresses qui sont soit en Floride (FL) ou soit dans le Nevada (NV). Proposez deux façons d'écrire cette requête (en utilisant deux opérateurs différents).

Q_{11}). Lister les adresses de la ville d'Edmonton ayant une note d'au moins 4 étoiles. Proposez deux façons d'écrire cette requête (l'une avec un opérateur explicite, l'autre avec la version implicite).

Q_{12}). Lister les adresses ayant une note d'au plus une étoile avec plus de 400 commentaires (`review_count`).

Q_{13}). Lister les noms des entreprises et villes des plombiers (*Plumbing*) d'Arizona (AZ).

Q_{14}). Lister les noms et adresses des restaurants de Philadelphie (*Philadelphia*) qui possèdent une terrasse (`OutdoorSeating`).

Q_{15}). Lister tous les restaurants japonais ayant obtenu une note supérieure à quatre étoiles.

Q_{16}). Lister tous les restaurants japonais ou mexicains du Missouri (MO).

Q_{17}). Lister toutes les adresses ayant un nombre d'avis inférieur ou égal à leur nombre d'étoiles.

Q_{18}). Lister tous les restaurants qui proposent à la fois la livraison et à emporter ou ni l'un ni l'autre.

Proposer une requête qui compare les valeurs de deux champs afin de répondre à cette question.

Q_{19}). Lister les pharmacies (*drugstores*) de la Nouvelle-Orléans dont on connaît les horaires d'ouverture.

Q_{20}). Lister les parcs des villes de Bristol, Hulmeville, Langhorne, Newtown et Pendell en Pensylvannie (PA).

Q_{21}). Lister les églises de Floride qui ne sont pas dans les plus grandes villes (Miami, Orlando, Tampa et Jacksonville) par ordre alphabétique.

Q_{22}). Lister les noms des adresses référencées sur Virginia Street ("Virginia St") à Reno.

Q_{23}). Lister les noms et adresses magasins du Tennessee (TN) dont le nom contient le mot `"Tree"` ou le mot `"Earth"`, les adresses ayant eu le plus de commentaires en premier.

Q_{24}). Le restaurant Twin Peaks d'Indianapolis change de propriétaire. Il faut donc mettre à jour certaines informations. Enlevez la catégorie `"Bars"`.

Q_{25}). Supprimez également les catégories `"Sports Bars"`, `"American (New)"` et `"American (Traditionnal)"`.

Q_{26}). Le restaurant propose désormais de la cuisine française. Ajoutez la catégorie `"French"`.

Q_{27}). Ajoutez également les catégories `"Creperies"` et `"Seafood"`. Vous devrez n'utiliser qu'une seule instruction.

Q_{28}). Vérifiez que vos modifications ont bien été prises en compte.

Pour aller plus loin : Les index

Comme pour PostgreSQL, il est possible de créer des index dans une base de données MongoDB afin d'améliorer les performances des requêtes.

Par défaut, un index est créé sur le champ `_id`. Pour créer soi-même un index, il faut utiliser la commande `createIndex` comme dans l'exemple ci-dessous :

```
db.ma_collection.createIndex({"mon_champ" : 1}, {"name": "mon_index"})
```

Dans cet exemple, un index est créé sur le champ `mon_champ` de la collection `ma_collection`. Le `1` correspond au type d'index, ici les valeurs de `mon_champ` sont triés en ordre croissant. Le nom de cet index est `mon_index`.

Q_{29}). Lister les bars de la ville de Santa Barbara ayant obtenu plus de 3 étoiles.

Q_{30}). Utiliser l'onglet **Explain Plan** pour afficher le plan d'exécution de cette requête. Combien de documents sont retournés ? Quel est son temps d'exécution ? Un index est-il utilisé pour calculer le résultat ?

Q_{31}). Créer un index sur le champ représentant la ville.

Q_{32}). Relancer l'analyse du plan d'exécution et comparer les résultats.

Q_{33}). Essayer de créer d'autres index pour améliorer les performances de cette requête.

Q_{34}). Consulter l'onglet **Indexes** pour la taille nécessaire au stockage des index que vous avez créé. (Utiliser le bouton de rafraîchissement en haut à gauche de la fenêtre si vos index n'apparaissent pas.) Supprimer les index que vous venez de créer.

Pour aller plus loin : Les données géospatiales

L'un des avantages de MongoDB est son intégration d'outils permettant de manipuler facilement des données géospatiales, autrement dit des informations de localisation. Pour représenter ces données, MongoDB utilise le format GeoJSON, un format standardisé dont vous pouvez voir les possibilités dans la documentation associée (`https://www.mongodb.com/docs/manual/reference/geojson/`).

Dans la base de données Yelp, nous disposons de données géospatiales. En effet, chaque adresse dispose d'un champ `location` qui est un point GeoJSON comme par exemple :

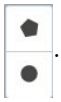
```
{
  "_id": {
    "$oid": "63972dd0158ef602a68ac261"
  },
  "name": "Abby Rappoport, LAC, CMQ",
  "address": "1616 Chapala St, Ste 2",
  "city": "Santa Barbara",
...
  "location": {
    "coordinates": [-119.7111968, 34.4266787],
    "type": "Point"
  }
}
```

Les coordonnées sont données sous forme de tableau, la première valeur étant la longitude, la deuxième étant la latitude.

MongoDB Compass permet de visualiser directement les données sous forme de carte.

Q_{35}). Aller dans l'onglet **Schema** et cliquer sur Analyze Schema.

Au niveau des informations sur le champ `location`, vous devez voir apparaître une carte. Si ce n'est pas le cas, aller dans le menu Help » Privacy Settings et cocher "Enable geographic visualizations", puis raffraîchir l'analyse. (La connexion au plugin servant à afficher la carte peut prendre quelques minutes.)

Q_{36}). Utiliser les outils à droite de la carte pour dessiner une zone autour de Philadelphie :  .

Cela modifie automatiquement la requête de la zone "Filter".

Q_{37}). Cliquer sur le bouton ANALYZE pour voir le résultat. Qu'obtient-on ?

Q_{38}). Écrire une requête permettant d'obtenir la latitude et la longitude de la salle de sport "Live Oak Yoga" de la Nouvelle-Orléans.

La plupart des opérateurs sur les données géospatiales requièrent un index spécial de type `2dsphere`.

Q_{39}). Créer un index sur le champ `location` de type `2dsphere` en remplaçant le `1` dans la commande montrée en exemple précédemment par `2dsphere`.

L'opérateur `$nearSphere` permet de trouver données géospatiales à une certaine distance d'un point donné. La syntaxe est la suivante :

```
{
  "location":{
    $nearSphere: {
      $geometry: {
        "type": "Point",
        "coordinates": [latitude, longitude]
      },
      $minDistance: valeur_en_metres,
      $maxDistance: valeur_en_metres
    }
  }
}
```

Les champs `$minDistance` et `$maxDistance` sont optionnels.

Q_{40}). Lister les noms des adresses à moins de 25 mètres de "Live Oak Yoga".

Q_{41}). Y a t'il des restaurants à moins de 200m de "Live Oak Yoga" ? Quel type de cuisine proposent-ils ?

Q_{42}). Lister les noms et adresses des salles de sports (*Fitness & Instruction*) de la Nouvelle-Orléans à au moins 1km de "Live Oak Yoga", les mieux notées en premier.