Prise de note MongoDB

**Lancement serveur et client :**

* ./mongodb 🡪 lancement du serveur
* ./mongod 🡪 lancement du client

**Interface utilisateur**

On utilise Robo3T. La connexion s’établie avec localhost sur le port 27017. On importe une nouvelle base pour effectuer des requêtes

Pour importer les données on utilise la commande suivante :

./mongoimport -d « nom de la base » -c « nom de la collection » --file « fichier de donnée à importer »

**Type de donnée à traiter**

La base de données contient des restaurant où chacun a un nom, un quartier, le type de cuisine, une adresse (document imbriqué, avec coordonnées GPS, rue et code postale), et un ensemble de notes.

**Manipulation des données**

La commande de base pour une requête de filtrage est la suivante :

db.restaurants.find( { "borough" : "Manhattan" } )

Cela les restaurants dans le quartier de Manhattan. Ensuite on ajoute deux clés/valeurs afin de chercher parmi les restaurants de Manhattan ceux qui font de la cuisine Italienne.

db.restaurants.find( { "borough" : " Manhattan ", "cuisine" : "Italian" } )

Ensuite, afin de rechercher un mot en particulier on va rajouter la notation /lemot/pour regrouper tous les mots correspondant tout en étant insensible à la casse :

db.restaurants.find{ "borough" : " Manhattan ",

"cuisine": "Italian",

"address.street" : "21 jump Street",

"name": /burger/i } )

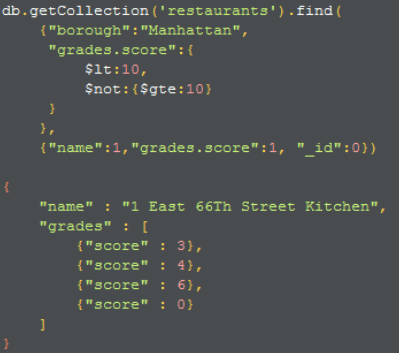
Ce filtre permet de renvoyer les restaurants de Manhattan dans la rue définie donc le nom contient « burger ».

Pour afficher les résultats retournés par le filtrage, on utilise la projection. Pour projeter une valeur spécifique on va compléter le deuxième paramètre de la fonction « find » : {"name":1, "\_id":0 }. Dans cet exemple, on va projeter la valeur selon le « nom » sans projeter « \_id » du document, qui est projeté par défaut.

De plus, pour être plus précis on peut appliquer un filtrage par rapport à des comparateurs sur des valeurs numérique. Par exemple :

* $eq = égale à
* $lt = plus petit que
* $gt = plus grand que

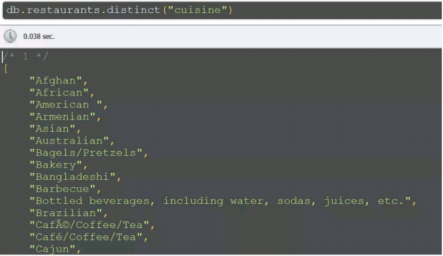
Voici un exemple :



!!!! 🡪 Si on veut combiner plusieurs conditions, on doit utiliser la notation $elemMatch. Cette notation permet que toutes les conditions soient vérifiées élément par élément

Pour afficher toutes les données d’un champ spécifiques il faut utiliser !

db.restaurants.distinct(« cuisine »)



**Les agrégations :**

La fonction « aggregate » est utilisée dans des manipulations plus complexes. Elle va pouvoir prendre en paramètre plusieurs opérateurs. Les principaux opérateurs sont les suivants :

* {$match : {} }
* {$project : {} }
* {$sort : {} }

db.restaurants.find({

"grades.0.grade":"B"

},

{"name":1, "\_id":0} );

La requête équivalente avec une aggrégation est la suivante :

db.restaurants.aggregate( [

{ $match : {

"grades.0.grade":"B" }

},

{ $project : {

"name":1, "\_id":0 }} ]

)