**Gestion Des données - MongoDB**

Exercice 1

Créez une base de données qui porte le nom de votre groupe « exemple (DEV203)

Importez le fichier **restaurants.json** dans une collection qui porte le même nom :

Executez les requetes suivantes en utilisant le mongosh en ligne de commande ou sur mongodbcompass puis notez dans les lignes vides l’explication des résultats obtenus

**Exemple :**

**db.restaurants.find( { "borough" : "Brooklyn" } )**

Cette requête permet d’afficher la liste des restaurants qui se trouvent dans le quartier « Brooklyn »……………………………………………………

**NB : ne pas faire copier/coller, retappez la requête pour mémoriser la syntaxe et faire/corriger les erreurs.**

**db.restaurants.find( { "borough" : "Brooklyn" } ).count()**

**affiche le nombre des restaurants qui se trouvent a brooklyne**

**db.restaurants.find(**

**{ "borough" : "Brooklyn",**

**"cuisine" : "Italian" }**

**)**

la liste des restaurant qui preparent la cuisine italienne et qui se trouvent a brooklyne

**db.restaurants.find(**

**{ "borough" : "Brooklyn",**

**"cuisine" : "Italian",**

**"address.street" : "5 Avenue" }**

**)**

la liste des restaurant qui preparent la cuisine italienne et qui se trouvent a brooklyne dans l'avenue 5

**db.restaurants.find(**

**{ "borough" : "Brooklyn",**

**"cuisine" : "Italian",**

**"address.street" : "5 Avenue",**

**"name" : /pizza/i }**

**)**

**)**

la liste des restaurant qui preparent la cuisine italienne et qui se trouvent a brooklyne dans l'avenue 5 dont le nom contient le mot pizza insensible à la casse (Maj-min)

Projection

-------------------

**db.getCollection('restaurants').find(**

**{"borough":"Brooklyn",**

**"cuisine":"Italian",**

**"name":/pizza/i,**

**"address.street" : "5 Avenue"},**

**{"name":1})**

**)**

la liste des noms et des \_ids des restaurant qui preparent la cuisine italienne et qui se trouvent a brooklyne dans l'avenue 5 dont le nom contient le mot pizza insensible à la casse (Maj-min)

regex

--------------------

**db.getCollection('restaurants').find(**

**{"borough":"Brooklyn",**

**"cuisine":"Italian",**

**"name":/pizza/i,**

**"address.street" : "5 Avenue"},**

**{"name" : 1,**

**"grades.score" : 1}**

**)**

la liste des noms , des scores et des \_ids des restaurant qui preparent la cuisine italienne et qui se trouvent a brooklyne dans l'avenue 5 dont le nom contient le mot pizza insensible à la casse (Maj-min)

**opérateurs**

**db.getCollection('restaurants').find(**

**{"borough":"Manhattan",**

**"grades.score":{$lt : 10}**

**},**

**{"name":1,"grades.score":1, "\_id":0})**

affiche les nom et les scores des restaurants de manhattan qui ont au moins un score inférieur à 10

**db.getCollection('restaurants').find(**

**{"borough":"Manhattan",**

**"grades.score":{**

**$lt:10,**

**$not:{$gte:10}**

**}**

**},**

**{"name":1,"grades.score":1, "\_id":0})**

affiche les nom et les scores des restaurants de manhattan qui ont tous les scors inférieur à 10

**db.restaurants.find({**

**"grades.grade" : "C",**

**"grades.score" : {$lt : 40}**

**},**

**{"Grades.grade":1, "grades.score":1} -- problèmes Grades G maj**

**);**

affiche les scores et les \_ids des restaurant qui ont au moins un grade C et au moins un score inférieur à 40

**db.restaurants.find({**

**"grades.grade" : "C",**

**"grades.score" : {$lt : 40}**

**},**

**{"grades.grade":1, "grades.score":1}**

**);**

affiche les grades, les scores et les \_ids des restaurant qui ont au moins un grade C et au moins un score inférieur à 40

**db.restaurants.find({**

**"grades" : {**

**$elemMatch : {**

**"grade" : "C",**

**"score" : {$lt :40}**

**}**

**}**

**},**

**{"grades.grade" : 1,"grades.score" : 1}**

**);**

affiche les grades, les scores et les \_ids des restaurant qui ont au moins un grade C dont le score est inférieur à 40 (dans le meme document)

**db.restaurants.find({**

**"grades.0.grade":"C"**

**},**

**{"name":1, "borough":1, "\_id":0}**

**);**

affiche les nom et les quartiers des restaurants qui ont le premier grade "C"

**db.restaurants.distinct("borough")**

un tableau qui affiche la liste des différents quartiers (sans doublons)

**db.restaurants.distinct("grades.grade");**

un tableau qui affiche la liste des différents grades (sans doublons)

aggregation

----------------

**db.restaurants.aggregate([**

**{ $group: {**

**\_id: { cuisine: "$cuisine",**

**borough: "$borough" } } },**

**{ "$project": { \_id: 0,**

**cuisine: "$\_id.cuisine",**

**borough: "$\_id.borough" } },**

**{ $sort: { borough:1,**

**cuisine:1 } }**

**])**

affiche les différentes cuisines par quartier triès par quartier puis par cuisine

**db.restaurants.aggregate( [**

**{ $match : {**

**"grades.0.grade":"C"**

**}},**

**{ $project : {**

**"name":1, "borough":1, "\_id":0**

**}}**

**] )**

affiche les noms et les quartiers qui ont le premier grade = "C" (peut être remplacée par find)

**varMatch = { $match : { "grades.0.grade":"C"} };**

**varProject = { $project : {"name":1, "borough":1, "\_id":0}};**

**db.restaurants.aggregate( [ varMatch, varProject ] );**

affiche les noms et les quartiers qui ont le premier grade = "C" on a stockées les etapes de la fonction aggrégate dans des variables pour simplifier la présentation et reutiliser ces etapes des les futures requetes

**varSort = { $sort : {"name":1} };**

**db.restaurants.aggregate( [ varMatch, varProject, varSort ] );**

on a reutiliser les etapes précédentes et on a jouté le tri par nom croissant

**varGroup = { $group : {"\_id" : null, "total" : {$sum : 0} } };**

**db.restaurants.aggregate( [ varMatch, varGroup ] );**

requete sans utilité (affiche un document avec \_id=null et total=0)

**db.restaurants.count({"grades.0.grade":"C"})**

affiche le nombre des restaurants qui ont le premier grade égale à "C" (deprecated ou obsolète)

**db.restaurants.find({"grades.0.grade":"C"}).count()**

affiche le nombre des restaurants qui ont le premier grade égale à "C"

**varGroup2 = { $group : {"\_id" : "borough", "total" : {$sum : 1} } };**

**db.restaurants.aggregate( [ varMatch, varGroup2 ] );**

affiche le nombre des restaurant dont le premier grade égale à C avec le id "borough"

**varGroup2 = { $group : {"\_id" : "$borough", "total" : {$sum : 1} } };**

**db.restaurants.aggregate( [ varMatch, varGroup2 ] );**

**affiche le nombre des restaurants dont le premier grande égale à C par quartier**

**varGroup4 = { $group : {"\_id" : "$borough", "moyenne" : {$avg : "$grades.score"} } };**

**varSort2 = { $sort : { "moyenne" : -1 } }**

**db.restaurants.aggregate( [ varGroup4, varSort2 ] );**

les sores se trouvent dans un tableau impossible de calculer la moyenne par restaurant il faut utiliser unwind

**varUnwind = {$unwind : "$grades"}**

**varGroup4 = { $group : {"\_id" : "$borough", "moyenne" : {$avg : "$grades.score"} } };**

**varSort2 = { $sort : { "moyenne" : -1 } }**

**db.restaurants.aggregate( [ varUnwind, varGroup4, varSort2 ] );**

affiche la moyenne des scores par restaurant