**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()**

les liste des restaurants de "Brooklyn".

le nombre des restaurants de "Brooklyn"

**db.restaurants.find(**

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

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

**)**

La liste des restaurants de Brooklye qui preparent la cuisine italienne.

**db.restaurants.find(**

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

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

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

**)**

La liste des restaurants de Brooklyn qui preparent la cuisine italienne et qui sont localisés à "5 avenue".

**db.restaurants.find(**

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

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

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

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

**)**

La liste des restaurants de Brooklyn qui preparent la cuisine italienne et qui sont localisés à "5 avenue" et dont le nom est pizza miniscule.

**)**

La liste des restaurants de Brooklyn qui preparent la cuisine italienne , qui sont localisés à "5 avenue" et dont le nom contient le mot 'pizza' case intensive.

Projection

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

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

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

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

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

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

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

Les noms et les ids des restaurants de Brooklyn qui preparent la cuisine italienne , qui sont localisés à "5 avenue" et dont le nom contient le mot 'pizza' case intensive.

regex

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

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

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

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

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

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

**{"name" : 1,**

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

**)**

Les noms et les scores des restaurants de Brooklyn qui preparent la cuisine italienne , qui sont localisés à "5 avenue" et dont le nom contient le mot 'pizza' case intensive.

Les noms , les scores et les ids des restaurants de Brooklyn qui preparent la cuisine italienne , qui sont localisés à "5 avenue" et dont le nom contient le mot 'pizza' case intensive.

**opérateurs**

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

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

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

**},**

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

les noms et les scores des restaurant de manhattan qui ont le score moins de 10

les noms et les scores des restaurant de manhattan qui ont au moins un score moins de 10

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

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

**"grades.score":{**

**$lt:10,**

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

**}**

**},**

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

les nom et les scores des restaurants de manhattan qui ont au moins un score inférieur à 10 et n'est pas supérieur ou égale à 10.

les nom et les scores des restaurants de manhattan qui ont au moins un score inférieur à 10 n'ont aucun score supérieur ou égale à 10

**db.restaurants.find({**

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

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

**},**

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

**);**

La liste des grades et des scores des restaurants qui ont au moins un grade "C" et qui ont au moins un score < 40;

La liste des ids, des scores des restaurants qui ont au moins un grade "C" et qui ont au moins un score < 40;

**db.restaurants.find({**

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

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

**},**

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

**);**

La liste des ids, des grades et des scores des restaurants qui ont au moins un grade "C" et qui ont au moins un score < 40;

**db.restaurants.find({**

**"grades" : {**

**$elemMatch : {**

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

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

**}**

**}**

**},**

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

**);**

La liste des ids, des grades et des scores des restaurants qui ont au moins un grade "C" et qui ont au moins un score < 40;

La liste des ids, des grades et des scores des restaurants qui ont au moins un grade "C" avec un score < 40;

**db.restaurants.find({**

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

**},**

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

**);**

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

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

un tableau des différents noms des quartiers de la base de données. (sans doublons)

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

un tableau des différents noms des grades des restaurants (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 } }**

**])**

la liste des cuisines et des quartiers, triès par quartier ascendant et cuisine ascendante.

la liste différentes cuisines par quartiers, triès par quartier ascendant et cuisine ascendante.

**db.restaurants.aggregate( [**

**{ $match : {**

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

**}},**

**{ $project : {**

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

**}}**

**] )**

les noms et les quartiers des restaurant qui ont le premier grade = "C"

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

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

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

les noms et les quartiers des restaurant qui ont le premier grade = "C"

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

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

les noms et les quartiers des restaurant qui ont le premier grade = "C" triès par nom ascendant

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

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

le groupe de restaurants qui ont le premier grade = "C" et la somme des restaurants.

le nombre des restaurants qui ont le premier grade = "C"

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

le nombre des restaurants qui ont le premier grade = "C" (obsolète - deprecated)

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

le nombre des restaurants qui ont le premier grade = "C"

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

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

le nombre des restaurants qui ont le premier grade = "C"

le nombre des restaurants qui ont le premier grade = "C" avec un id qui contient la valeur "borough"

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

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

**le nombre des restrautrant qui ont le premier grade = "C" par quartier.**

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

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

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

Le regroupement ne peut pas acceder directement aux objets qui se trouvent des tableaux.  
le resultat affiché sera les différents quartiers (id) avec une moyenne null.

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

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

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

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

la moyenne des scores des restaurants par quartier triès par moyenne décroissant.