**NOTES BIG DATA**

* **CREATION DE BASE DE DONNEES PAR INVITE DE CMD**

C:\Users\blanc>mongosh

test> use PERSONNEL permet de créer la base de données

PERSONNEL> db.createCollection("Enseignant") Créer une collection (table ) nommé Enseignant

PERSONNEL> db.createCollection("Etudiant") Créer une collection (table ) nommé Etudiant

PERSONNEL> db.Enseignant.insert({"Nom":"BALOGOUN", "Prénom":"Inès Houéfa", "Fonction": "Statisticienne", "Pays": "Bénin" }) Insérer une donnée dans la collection Enseignant

PERSONNEL> db.Enseignant.find()

[ { \_id: ObjectId('65e8d6f9e47a85e34b416ea9'),

Nom: 'BALOGOUN',

'Prénom': 'Inès Houéfa',

Fonction: 'Statisticienne',

Pays: 'Bénin' } ]

PERSONNEL> show collections Pour afficher toutes les collections dans la base de données Personnel

PERSONNEL> show dbs Pour afficher toutes les bases de données

PERSONNEL> exit : Pour revenir à la base(C:\Users\blanc>)

**NB : Après chaque ligne de code on appui « Entrez »**

* **COMMANDE MONGODB**

1-PERSONNEL> db.Enseignant.find().pretty() : Afficher les documents de la collection Enseignant

2-PERSONNEL> db.Enseignant.find({"Pays": "France"}) : Afficher les documents ayant pour Pays : France de la collection Enseignant

3-PERSONNEL> db.Enseignant.find({"Pays": "France"}).pretty() : Rendre l’affichage plus lisible

4-PERSONNEL> db.Enseignant.find().limit(3) : Afficher 3 documents de la collection Enseignant

5-PERSONNEL> db.Enseignant.find({"Fonction": /Chercheur/}) : Afficher tous les documents contenant chercheur dans leur fonction dans la collection Enseignant

6-PERSONNEL> db.Enseignant.find({"Fonction": /Chercheur/i}) : Avec le i à la fin ça ne fait pas de différence entre majuscule et minuscule

7-PERSONNEL> db.Enseignant.find({Fonction: /Chercheur/i}).count() Afficher le nombre de documents contenant chercheur dans leur fonction dans la collection Enseignant

8-PERSONNEL> db.Enseignant.find().count() Afficher le nombre de documents dans la collection Enseignant

9-mongoimport --db Essai --collection Cols --type json --jsonArray --file "chemin du fichier où se trouve les documents " : Pour importer directement le fichier contenant les documents télécharger sur le PC ; Mais avant s’assurer que le package mongoimport est installé

9-db.Livre.distinct("publisher") Pour afficher les modalités des variables

10-Bibliotek> db.Livre.distinct("publisher").length : Pour trouver le nombre retourné par une requête on mais count() à la fin mais en présence de distinct on met length

Comprendre les agrégations : <https://welovedevs.com/fr/articles/aggregation-mongo/>

**TP1\_Mongodb**

**Enoncé**

1-creer la Base de Donnée Bibliotek: use Bibliotek

2-creer la collection Livre: db.createCollection("Livre")

3-Télécharger le fichier contenant les données : DBLP.json.zip (http://b3d.bdpedia.fr/files/dblp.json.zip)

4- Décompresser le fichier dblp.json.zip Dans le même répertoire,

5-lancer l’importation du fichier via MongoShell: mongoimport --db Bibliotek --collection Livre --type json --jsonArray --file "C:\Users\blanc\Desktop\FORMATION CERTIFICAT DATA STEWARDS\BIG DATA\BASE JSON\dblp.json" (Faire d'abord exit pour sortir de mongosh)

6- éalisez les requêtes suivantes :

Travail à faire au niveau des requêtes :

a. Liste de tous les livres (type « Book ») : db.Livre.find({"type": "Book"}) (11074)

b. Liste des publications depuis 2011 : db.Livre.find({"year": {$gte :2011}}) (29499)

c. Liste des livres depuis 2014 : db.Livre.find({"type":"Book", "year": {$gte :2014}}) (288)

d. Liste des publications de l’auteur « Toru Ishida » : db.Livre.find({"authors": "Toru Ishida"}) (22)

e. Liste de tous les éditeurs (type « publisher »), distincts : db.Livre.distinct("publisher") (539)

f. Liste de tous les auteurs distincts : db.Livre.distinct("editor")

g. Trier les publications de « Toru Ishida » par titre de livre : db.Livre.aggregate([{ $unwind: "$authors" }, { $match: { "authors": "Toru Ishida" }},{$sort:{"booktitle":-1}}])

h. Compter le nombre de ses publications : db.Livre.find({"authors": "Toru Ishida"}).sort("booktitle").count() (22)

i. Compter le nombre de publications depuis 2011 et par type : db.Livre.aggregate([{$match:{"year":{$gte: 2011}}},{$group:{\_id:"$type", count:{$sum:1}}}])

j. Compter le nombre de publications par auteur et trier le résultat par ordre croissant : db.Livre.aggregate([{ $unwind: "$authors"}, {$group:{\_id:{ $toLower: "$authors" }, count: { $sum: 1 } } },{$sort:{count: 1}}])