



BASES DE DONNÉES AVANCÉES PROJET 02 **NoSQL sous MongoDB**

Realisé Par :

SEMMAR Hichem
AMEZIANE Abdelghani

Responsable du module :

Azzouz

Assistante TP :

Challal

Année 2021/2022

Partie I : préparation d'environnement de Travail :

1-Services (Linux users Only) :

Il est strictement important de lancer le service de mongod avant de commencer le travail sinon python ne peut pas accéder à votre base de données

Linux :

```
sudo systemctl start mongod
```

2-Insertion de la base de données :

Pour insérer la base de données dans mongodb, insérer cette requête dans votre terminal :

```
mongoimport --db BDD --collection world --file world-mongodb.json
```

3-Installation des requirements : (Pymongo - PyQt5)

```
pip install -r requirements.txt
```

(au cas où vous avez déjà ces deux bibliothèques-là, exécuter ces deux commandes)

```
pip install sip --upgrade
```

```
pip install PyQt5-sip --upgrade
```

4-Lancement de l'application:

Python main.py

interface.py (Le fichier de l'interface)

manual.py (Le fichier de l'interface du manuel)

world.py (Le fichier qui contient les fonctions manipulatrice de la base de données)

main.py (le fichier qui organise l'interaction entre ces derniers)

Partie II : Requêtes, Explications et résultats :

1-Déterminer le nombre exact de pays :

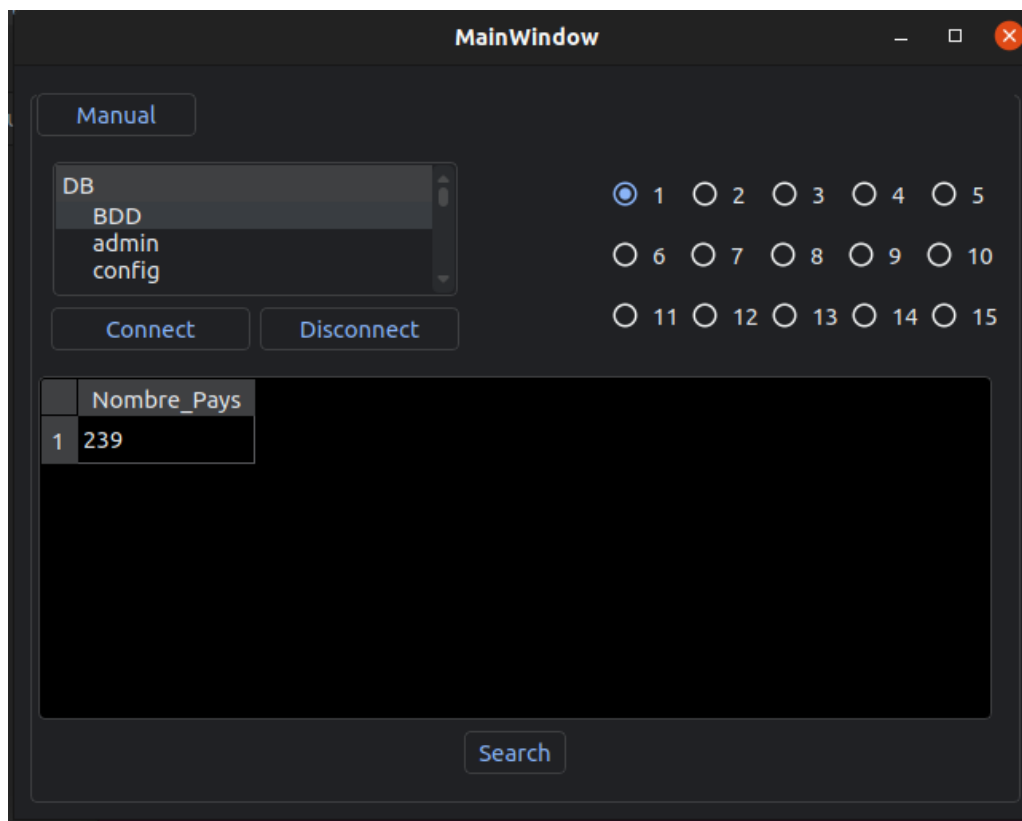
Requête MongoDB + Python :

```
Result = world.find()  
len(list(Result))
```

Explications :

Affecter tout les elements existant dans la variable Result de type liste de Dictionnary, ensuite compter le nombre de ses elements.

Resultat :



2-Lister les différents continents :

Requête MongoDB + Python

```

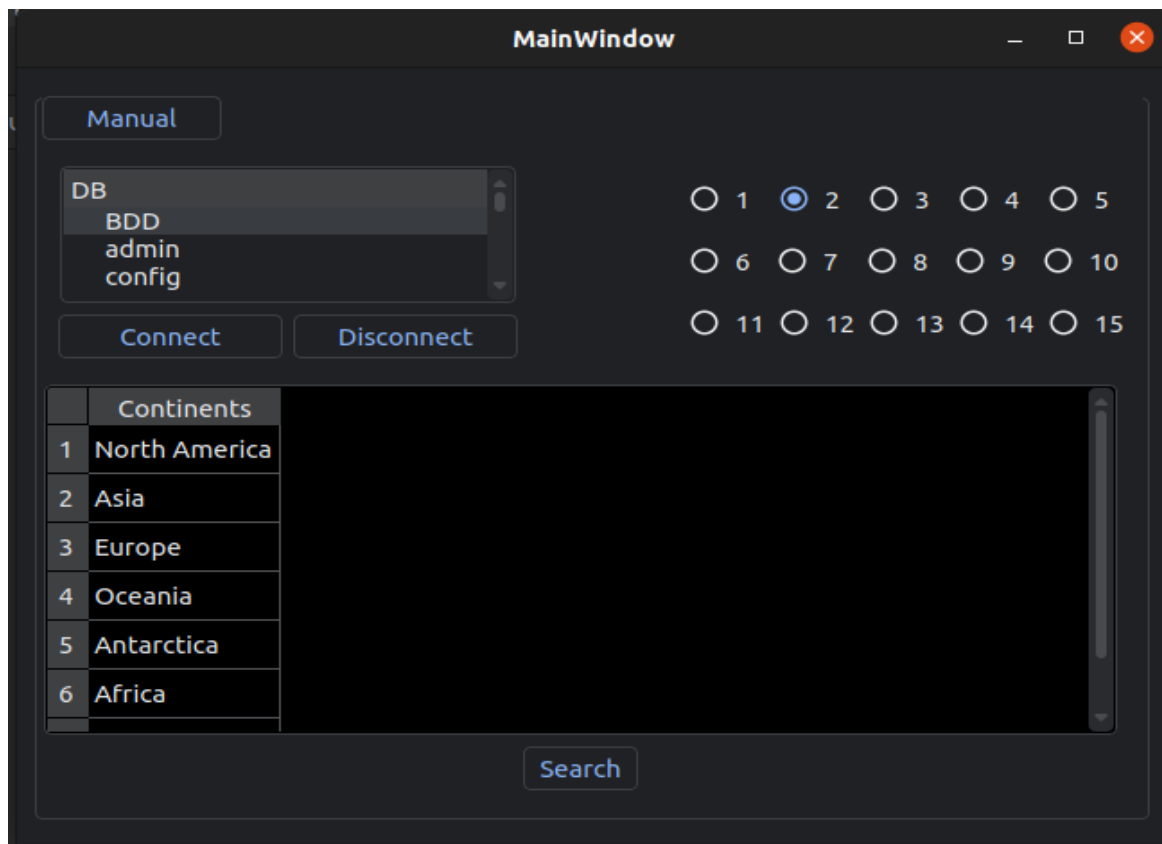
Result = world.find()
Continents = []
for x in Result:
    if(x["Continent"] not in Continents):
        Continents.append(x["Continent"])

```

Explications :

Affecter tout les elements existant dans la variable Result de type liste de Dictionnary ensuite, affecte tout les continents vers une liste Continents sans repetition.

Resultat :



3-Lister les informations de l'Algérie :

Requête MongoDB + Python

```

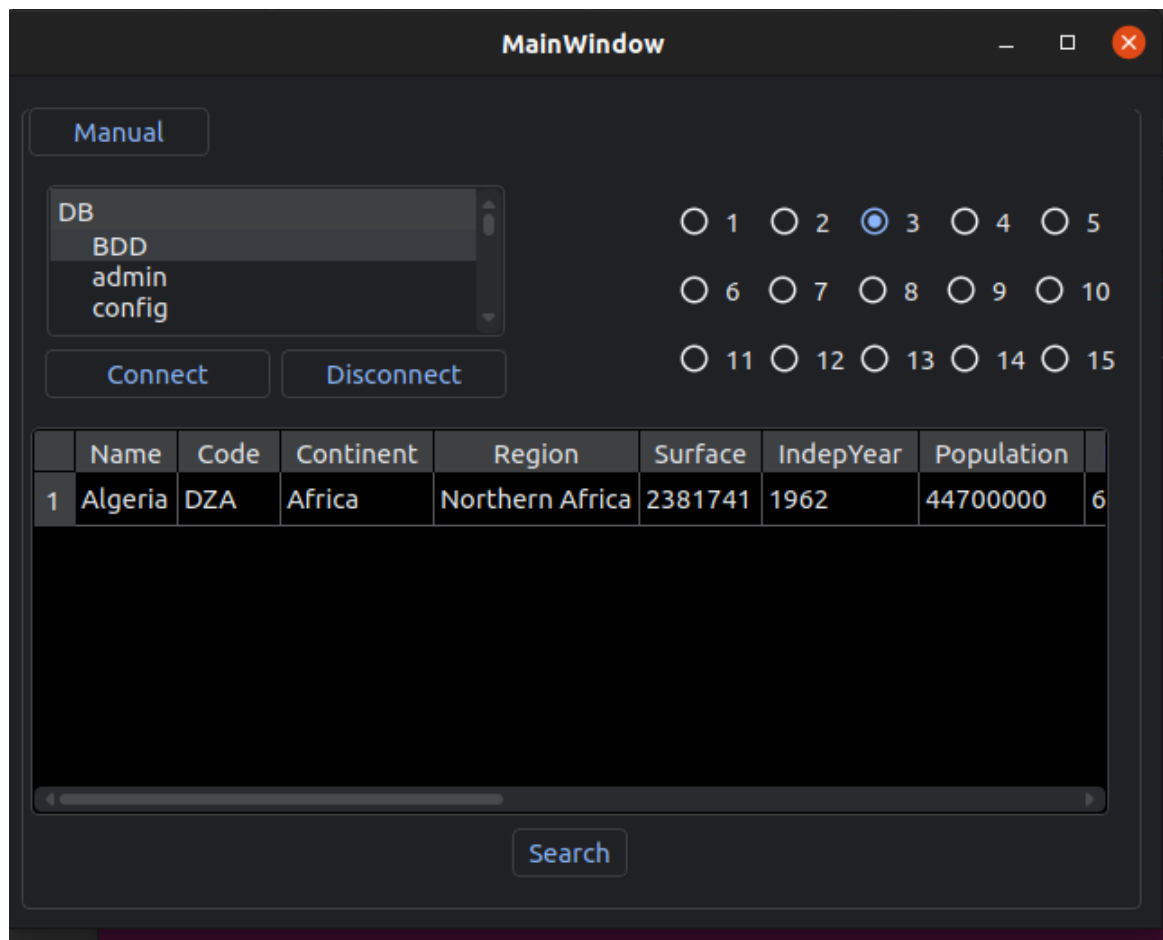
Query = {"Name" : "Algeria"}
Result = world.find(Query)

```

Explications :

La variable Result reçoit l'element du pays avec le nom « Algeria »

Resultat :



4- Lister les pays du continent Africain, ayant une population inférieure à 100000 habitants :

Requête MongoDB + Python :

Query = {"Continent" : "Africa", "Population": {"\$lt" : 100000} }

Result = world.find(Query)

Explications :

Result reçoit tout les elements du Continent d’afrique avec une population inférieure à 10000

Resultat :

	Pays
1	British Indian Ocean Territory
2	Saint Helena
3	Seychelles

5-Lister les pays indépendant du continent océanique ? :

Requête MongoDB + Python :

Query = {"IndepYear" : { "\$ne" : "NA"}, "Continent" : "Oceania"}

Projection = {"Name" : 1, "_id" : 0}

Explications :

La variable result reçoit les noms de tout les pays ayant une valeur existante de l'année d'indépendance du continent océanique .

Resultat :

	Pays Independants
9	Papua New Guinea
10	Samoa
11	Solomon Islands
12	Tonga
13	Tuvalu
14	Vanuatu

6-Quel est le plus gros continent en termes de surface ? :

Requête MongoDB + Python :

Query = [{"group": {"_id": "\$Continent", "SurfaceArea": {"sum": "\$SurfaceArea"}}, {"sort": {"SurfaceArea": -1}}, {"limit": 1}]

Result = world.aggregate(Query)

Explications :

La requête regroupe les éléments par leurs Continent, ensuite, elle somme la surface de ces groupes en les triant par décroissance et en limitant à 1 seul élément.

Resultat :

	Continent	Surface
1	Asia	31881005

7-Donner par continents le nombre de pays, la population totale et en bonus le nombre de pays indépendant. :

Requête MongoDB + Python :

```
Query = [  
{"$group" : {  
  "_id" : "$Continent",  
  "Population_Total" : {"$sum" : "$Population"},  
  "Pays" : {"$sum" : 1},  
  "Pays_indep": { "$sum": {"$cond": [{"$eq": ["$IndepYear", "NA"]}, 0, 1] } }  
}  
}]
```

Explications :

La requete regroupe les elements par leur continents ensuite, somme la population et compte le nombre de pays (à travers \$sum : 1) et calcule le nombre des pays independant en sommant le nombre des pays n'ayant pas la valeur « NA » dans le champ « IndepYear »

Resultat :

	Continent	Population	Pays	IndepPays
1	Asia	3705025700	51	47
2	Europe	730074600	46	43
3	North America	482993000	37	23
4	Oceania	30401150	28	14
5	Antarctica	0	5	0
6	Africa	797704000	58	53

8-Donner la population totale des villes d'Algérie :

Requête MongoDB + Python :

```
Query = {"Name" : "Algeria"}
```

Explications :

Cette requete retourne le pays ayant le nom « Algerie », ensuite une boucle « FOR » en python somme la population des villes.

Resultat :

	Sans capital	Avec capital
1	3024179	5192179

9-Donner la capitale (uniquement nom de la ville et population) d'Algérie :

Requête MongoDB + Python :

```
Query = {"Name" : "Algeria"}
```

Explications :

Cette requete retourne le pays ayant le nom « Algerie » ensuite, une simple operation en python retourne la capital et la population de l'element. (Key & values)

Resultat :

	Capital	Population
1	Alger	2168000

10-Quelles sont les langues parlées dans plus de 15 pays :

Requête MongoDB + Python :

```
For x in world.find():
    for y in (dict_sum(x)):
        if(y['Language'] not in languages.keys()):
            languages[y['Language']] = 1
        else:
            languages[y['Language']] += 1
```

```
for lang in languages.keys():
    if(languages[lang] > 15):
        print(lang)
```

Explications :

La premiere boucle sert a compter le nombre de l'utilisation de chaque langue dans chaque pays dans une Variable Languages de type Dictionnary, la 2eme boucle sert a filtrer les langues parlée dans plus de 15 pays de la variable Languages.

Hint : Dict_sum sert a regrouper les langues officielles avec Non officielles.

Resultat :

	Langues
1	English
2	Spanish
3	French
4	Arabic
5	German
6	Russian

11-Calculer pour chaque pays le nombre de villes (pour les pays ayant au moins 100 villes), en les triant par ordre décroissant du nombre de villes :

Requête MongoDB + Python :

```
cities = {}
for x in world.find():
    if("Cities" in x):
        if(len((x['Cities'])) > 99):
            cities[x["Name"]] = len(x['Cities'])
```

Explications :

Boucle python sert a sauvegarder les pays ayant plus de 100 villes dans une variable Cities

Resultat :

	Pays
3	United States
4	Brazil
5	Japan
6	Russian Federation
7	Mexico
8	Philippines

12-Lister les 10 villes les plus habitées, ainsi que leur pays, dans l'ordre décroissant de la population :

Requête MongoDB + Python :

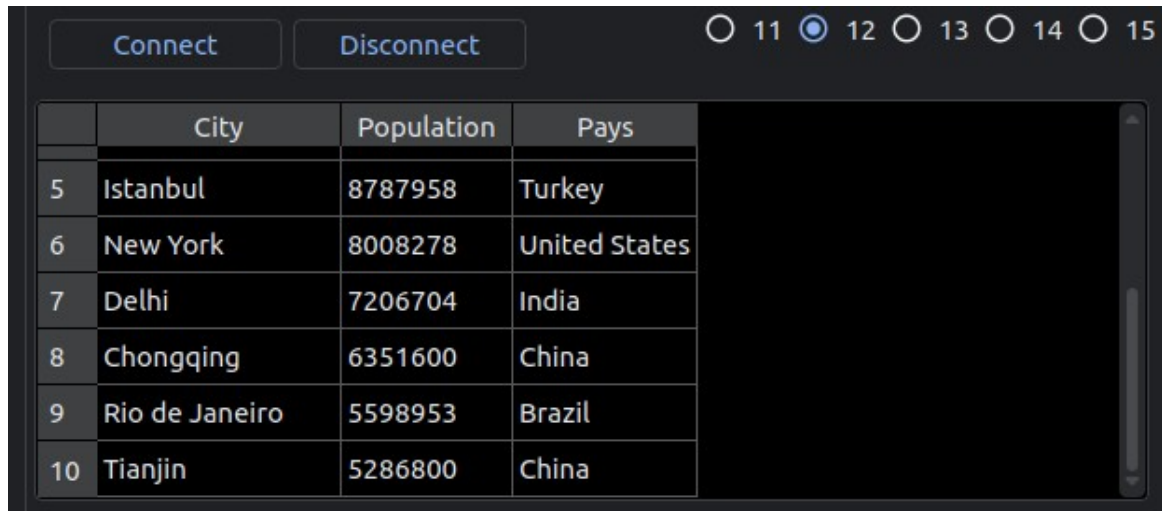
```
Query = [
    {"$unwind" : "$Cities"},
    {"$project" :
        {"Name": 1, "Cities.Population" : 1, "_id":0, "Cities.Name":1}
    },
    {"$sort" :
        {"Cities.Population" : -1}
    },
    {"$limit" : 10}
]
```

Explications :

Cette requête retourne chaque élément de la liste « Cities » (Élément par élément et non

pas une liste) ensuite, tri ces éléments par leurs populations en mode décroissant en limitant les 10 éléments premiers seulement.

Resultat :



The screenshot shows a MongoDB interface with a dark theme. At the top, there are 'Connect' and 'Disconnect' buttons. To the right, there are radio buttons for selecting a database, with '12' selected. Below this is a table with four columns: an index, 'City', 'Population', and 'Pays'. The table displays the top 10 cities by population. To the right of the table is a large empty area, likely for a map or additional data visualization.

	City	Population	Pays
5	Istanbul	8787958	Turkey
6	New York	8008278	United States
7	Delhi	7206704	India
8	Chongqing	6351600	China
9	Rio de Janeiro	5598953	Brazil
10	Tianjin	5286800	China

13-Lister les pays pour lesquels l'Arabe est une langue officielle :

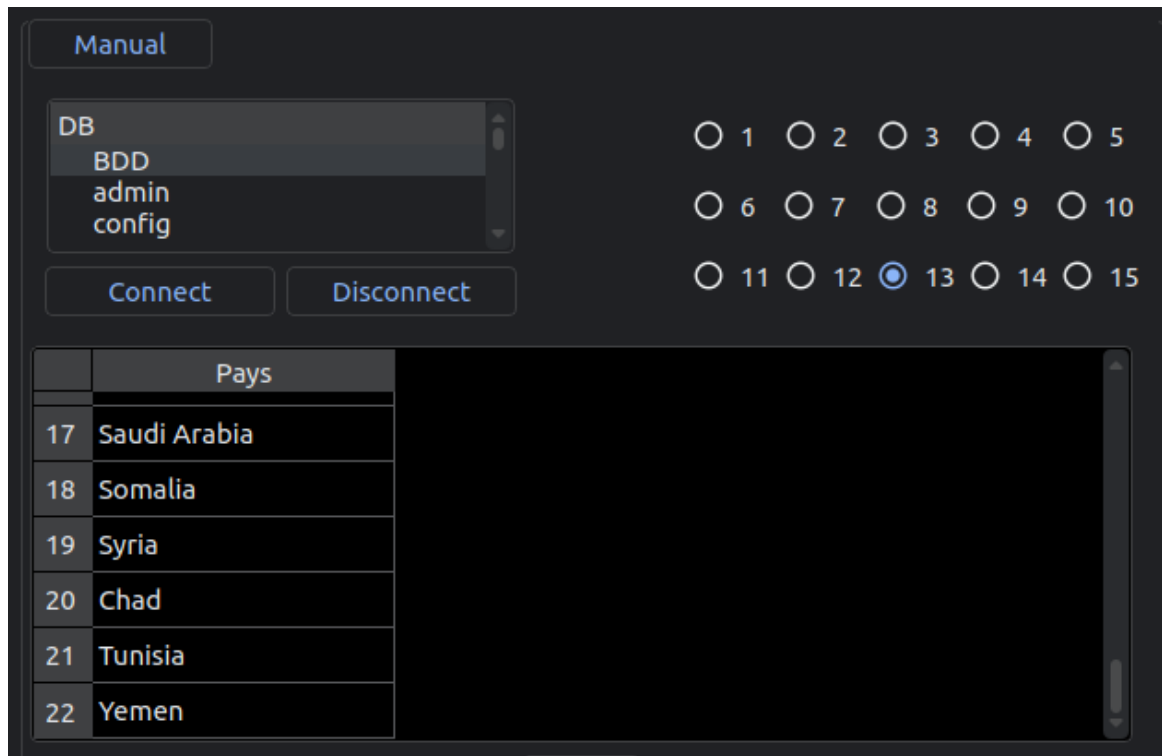
Requête MongoDB + Python :

```
Query = [  
    {"$unwind" : "$OffLang"},  
    {"$project" :  
        {"Name" : 1, "OffLang.Language" : 1, "_id":0}  
    }  
]
```

Explications :

Le resultat de cette requete retourne le nom du pays avec ses langues officielles

Resultat :



14-Lister les 5 pays avec le plus de langues parlées :

Requête MongoDB + Python :

```

Languages_spoken = {}
for x in world.find() :
    Languages_spoken[x["Name"]] = len(dict_sum(x))
i=1
for key in sorted(Languages_spoken, key=Languages_spoken.get, reverse=True):
    print("Le pays: ",key)
    print("Le nombre de langues parlées :", Languages_spoken[key])
    print('-----')
    if(i != 5):
        i += 1
    else:
        break

```

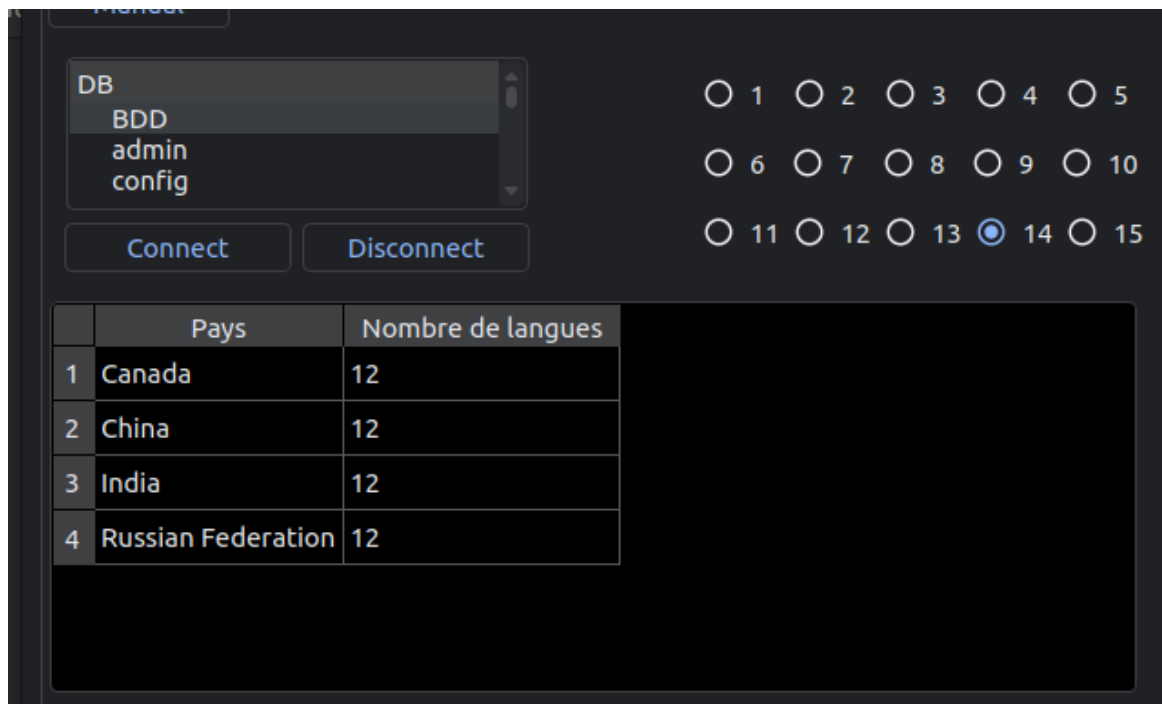
Explications :

La première boucle sert à calculer le nombre de langues parlé officielles et non officielles dans chaque pays et les sauvegarde dans une variable de type dictionary

Languages_spoken.

La 2eme boucle Tri les pays de la variable Languages_spoken par les valeurs stocké en mode décroissant en affichant seulement les 5 premiers éléments.

Resultat :



15-Lister les pays pour lesquels la somme des populations des villes est supérieure à la population du pays :

Requête MongoDB + Python :

```
for x in world.find():
    sum = 0
    if("Cities" in x):
        for y in x['Cities']:
            sum += y["Population"]
    if("Capital" in x):
        sum += x["Capital"]["Population"]
    if(sum > x["Population"]):
        ...
```

Explications :

La 2eme boucle somme la population de tout les villes dans un pays ensuite apres quitter la boucle, on ajouter la population de la capitale et on compare avec la population du pays

Resultat :

Connect		Disconnect		○ 11 ○ 12 ○ 13 ○ 14 ● 15	
	Pays	Population_total	Population des villes		
1	Cocos (Keeling) Islands	600	670		
2	Gibraltar	25000	27025		
3	Singapore	3567000	4017733		

Partie III : Pymongo & MongoDB (world.py) :

Le fichier world.py contient la solution code de ce project, pour acceder a la reponse de chaque question en detail il faut lire le contenu de chaque fonction Question#() numéroté de 1 jusqu'à 15.

Question1()
 Question2()
 Question3()
 Question4()
 Question5()
 Question6()...

```
def Question1() :
    Result = world.find()
    print("Le nombre total des pays est :",len(list(Result)))

def Question2():
    Result = world.find()
    Continents = []
    for x in Result:
        if(x["Continent"] not in Continents):
            Continents.append(x["Continent"])
    print("La liste des Continents :")
    for x in Continents:
        print(x)
```