

Entête commun :

```
import pandas as pd
import seaborn as sns

#Charge les different offshore leaks, la parametre low_memory=False permet de preciser que les donnees sont grandes
#et donc de desactiver les warnings
df_entite = pd.read_csv("../assets/full-olddb-20220110/nodes-addresses.csv", low_memory=False)
df_officers = pd.read_csv("../assets/full-olddb-20220110/nodes-officers.csv", low_memory=False)
```

Voici l'entête commun à tous les codes présentés.

Exercice avec Pandas (Exercice 1) :

```
#Ici je defini une fonction affichant a l'ecrant la reponse au questions du premier set
def Exo1() :
    print("\n\nExo 1 : \n")
    #la ligne suivante affiche le nombre d'entite en mesurant le retour de la dataframe avec un filtre sur les
    'offshore leaks'
    print("Nombre d'entites : " + str(len(df_entite[df_entite["sourceID"].str.lower() == "offshore leaks"]
    ["countries"].value_counts()))
    #Celle-ci mesure le retour du filtre sur 'ireland' dans la dataframe
    print("Ireland apparait : " + str(len(df_entite[df_entite["countries"].str.lower() == "ireland"])) + " fois.")
    #pour fini cette exercice, dans la ligne suivante j'affiche le nombre d'occurence des nom egal a 'el portdor'
    print("El Portador appait : " + str(len(df_officers[df_officers["name"].str.lower() == "el portador"])) + "
    fois.")
```

Pour ce TP j'ai décidé de réaliser chaque exercice dans une fonction et la fin d'exécuter le tout dans une fonction main.

Exercice avec Seaborn (Exercice 2) :

```
#Ici je defini une fonction affichant a l'ecrant la reponse au questions du second set
def Exo2() :
    print("\n\nExo 2 : \n")
    #cette ligne va permettre d'afficher un graphiique correlant les occurence de nom d'un pays et sont nom
    sns.countplot(x="countries", data=df_entite[df_entite["sourceID"].str.lower() == "offshore leaks"])
    #ici j'affiche les 20 premiere pays avec le plus d'occurence
    df_entite[df_entite["sourceID"].str.lower() == "offshore
leaks"].groupby(["countries"]).size().sort_values(ascending=False)[:20]
```

Reconstruction de la base de données

```
#je n'ai malheureusement pas compris la questions
def Exo3() :
    pass
```

Conclusion,

voici le code général :

```
import pandas as pd
import seaborn as sns

#Charge les different offshore leaks, la parametre low_memory=False permet de preciser que les donnees sont grandes
#et donc de desactiver les warnings
df_entite = pd.read_csv("../assets/full-olddb-20220110/nodes-addresses.csv", low_memory=False)
df_officers = pd.read_csv("../assets/full-olddb-20220110/nodes-officers.csv", low_memory=False)

#Ici je defini une fonction affichant a l'ecrant la reponse au questions du premier set
def Exo1() :
    print("\n\nExo 1 : \n")
    #la ligne suivante affiche le nombre d'entite en mesurant le retour de la dataframe avec un filtre sur les
    'offshore leaks'
    print("Nombre d'entites : " + str(len(df_entite[df_entite["sourceID"].str.lower() == "offshore leaks"]
    ["countries"].value_counts()))
    #Celle-ci mesure le retour du filtre sur 'ireland' dans la dataframe
    print("Irland apparait : " + str(len(df_entite[df_entite["countries"].str.lower() == "ireland"])) + " fois.")
    #pour fini cette exercice, dans la ligne suivante j'affiche le nombre d'occurence des nom egal a 'el portador'
    print("El Portador appait : " + str(len(df_officers[df_officers["name"].str.lower() == "el portador"])) + "
    fois.")

#Ici je defini une fonction affichant a l'ecrant la reponse au questions du second set
def Exo2() :
    print("\n\nExo 2 : \n")
    #cette ligne va permettre d'afficher un graphique correlant les occurence de nom d'un pays et sont nom
    sns.countplot(x="countries", data=df_entite[df_entite["sourceID"].str.lower() == "offshore leaks"])
    #ici j'affiche les 20 premiere pays avec le plus d'occurence
    df_entite[df_entite["sourceID"].str.lower() == "offshore
    leaks"].groupby(["countries"]).size().sort_values(ascending=False)[:20]

#je n'ai malheureusement pas compris la questions
def Exo3() :
    pass

def main():
    Exo1() #Execute la fonction contenant l'exercice 1
    Exo1() #meme chose mais pour l'exercice 2
    #Exo3()

if __name__ == "__main__":
    main()
```