Mini Projet : APACHE SPARK

1 - Creation de session

In [ ]:
```
import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession
```

In [ ]:

2 - Ingestion du CSV Donors

In [ ]:

3 - Affichage

Afficher les 20 premières lignes de dataset Donors (Utilisez la fonction show())

In [ ]:

Conversion en dataframe pandas (utilisez la fonction "toPandas()")

In [ ]:

Trouver le nombre nul dans chaque colonne

In [ ]:

Imprimer le schéma de dataset (pour imprimer le schéma, on utilise la fonction "printSchema")

In [ ]:

4 - Filtrage

Laissez que les enregistrement dont Donor City commence par A

Vous pouvez utiliser la fonction "filter"

Exemple : "My_data.filter(My_data.name_colonne.like("A%"))"

Like("A%") : le caractère "%" est un caractère joker qui remplace tous les autres caractères. Ainsi, ce modèle permet de rechercher toutes les chaines de caractère qui commence par un "A".

In [ ]:

Affichez les résultats

In [ ]:

5 - Transformation

Construisez une nouvelle colonne `Address` en faisant une concaténation `Donor_City, Donor_State, Donor_Zip`

In [1]:
```
from pyspark.sql.functions import concat_ws
from pyspark.sql.types import *
```

Vous pouvez utiliser la fonction "withColumn" et "concat_ws"

In [ ]:

Afficher les résultats

In [ ]:

6 - Moteur SQL

Persister le dataset de départ comme une Temporary View

Vous pouvez utliser la fcontion createOrReplaceTempView

In [ ]:

Comptez le nombre de professeurs ayant participé à la donation

Vous pouvez utiliser la fonction count() et le langage SQL

In [ ]:

utiliser juste 10% du dataset c'est très grand complet pour des jointures pour votre petite machine... avec la method `sample`

In [ ]:
```
df_e = df_depart.sample(fraction=0.1, seed=3)
```

Afficher que les id des donateurs qui habite à California

Vous pouvez utiliser le langage SQL qu'on vu dans le TP 5 suivant select col_x from donors where col_y ="California

In [ ]:

Ingestion des données et publication en temporary view du fichier Donations.CSV

In [ ]:

In [ ]:

Afficher le DF

In [1]:
```
# df_donations.count()
# df_donations.show()
```

Calculer le montant minimum, le montant maximum, le montant moyen en arrondissant à l'unité après la virgule de la colonne Donation_Amount

pour l'ensemble Donations

Utiliser les alias "maxMontant", "minMontant", "avgMontant". et la colonne "Donation_Amount"

Pour rappel en SQL, un alias ressemble à ça : "as maxMontant".

In [ ]:

utiliser juste 10% du dataset c'est très grand complet pour des jointures... avec la method `sample`

In [ ]:
```
df_donations_e = df_donations.sample(fraction=0.1, seed=3)
```

Faites une jointure Entre le data set des donneurs Donors, et le dataset des Donations Donations

Indication : utilisez "inner join" de langage spark.sql

In [ ]:

Calculez la somme de l'argent donnée par Les Professeurs (Donor Is Teacher=Yes) et les non professeurs utilisant seulement SQL

Indication : ('select sum(dt.col4) as amountProf from donations dt inner join donors dr on dt.col2 = dr.col0 and dr.col3 = "Yes" ')

In [ ]: