Mini Projet : APACHE SPARK

1 - Creation de session

```
import findspark
findspark.init()
import pyspark
from pyspark.sql import SparkSession

#https://spark.apache.org/docs/latest/sql-getting-started.html
spark = SparkSession \
        .builder \
        .appName("firstSpark") \
        .getOrCreate()
```

2 - Ingestion du CSV Donors

```
# df = spark.read.format('csv').options(header='true').load(".")
def load_dataframe(filename):
    df =
spark.read.format('csv').options(header='true').load(filename)
    return df

#creating a dataframe
df_donors  = load_dataframe('Donors.csv')
```

3 - Affichage

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

```
df_donors.show()
```

```
+--------------------+------------+--------------+----------------
+---------+
|            Donor_ID|  Donor_City|   Donor_State|Donor_Is_Teacher|
Donor_Zip|
+--------------------+------------+--------------+----------------
+---------+
|00000ce845c00cbf0...|    Evanston|      Illinois|              No|
602|
|00002783bc5d10851...|  Appomattox|         other|              No|
245|
|00002d44003ed46b0...|      Winton|    California|             Yes|
953|
|00002eb25d60a09c3...|Indianapolis|       Indiana|              No|
462|
|0000300773fe015f8...|    Paterson|    New Jersey|              No|
075|
|00004c31ce07c2214...|        null|         other|              No|
null|
|00004e32a448b4832...|    Stamford|   Connecticut|              No|
```

```
069|
|00004fa20a986e60a...|   Green Bay|     Wisconsin|              No|
543|
|00005454366b6b914...|      Argyle|      New York|              No|
128|
|0000584b8cdaeaa6b...|  Valparaiso|       Indiana|              No|
463|
|00005f52c98eeaf92...|   Villanova|  Pennsylvania|              No|
190|
|00006084c3d92d904...|       Brick|    New Jersey|             Yes|
087|
|00006c6b8c3225a54...|  Wilmington|      Delaware|              No|
198|
|0000812bd5629117f...|    Pasadena|         Texas|              No|
775|
|0000889adf4cc958a...|Mohegan Lake|      New York|              No|
105|
|00008eec5aab22286...|    Old Fort|North Carolina|              No|
287|
|0000954e7c49ebfbc...|      Quincy|      Illinois|              No|
623|
|0000a1288b8ccdeaa...|        null|         other|              No|
null|
|0000a2175753bc165...|Grand Rapids|      Michigan|              No|
495|
|0000a3fd8b8a3d1a9...|   Lancaster|  Pennsylvania|              No|
176|
+--------------------+------------+--------------+----------------
+---------+
only showing top 20 rows
```

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

```
df_donors.limit(20).toPandas()
```

```
                           Donor_ID    Donor_City     Donor_State  \
0   00000ce845c00cbf0686c992fc369df4      Evanston        Illinois
1   00002783bc5d108510f3f9666c8b1edd    Appomattox           other
2   00002d44003ed46b066607c5455a999a        Winton      California
3   00002eb25d60a09c318efbd0797bffb5  Indianapolis         Indiana
4   0000300773fe015f870914b42528541b      Paterson      New Jersey
5   00004c31ce07c22148ee37acd0f814b9          None           other
6   00004e32a448b4832e1b993500bf0731      Stamford     Connecticut
7   00004fa20a986e60a40262ba53d7edf1     Green Bay       Wisconsin
8   00005454366b6b914f9a8290f18f4aed        Argyle        New York
9   0000584b8cdaeaa6b3de82be509db839    Valparaiso         Indiana
10  00005f52c98eeaf92b2414a352b023a4     Villanova    Pennsylvania
11  00006084c3d92d904a22e0a70f5c119a         Brick      New Jersey
12  00006c6b8c3225a54438f878d59e650a    Wilmington        Delaware
13  0000812bd5629117f8909f73acbe8b7d      Pasadena           Texas
```

```
14  0000889adf4cc958a35daee1f2529b48  Mohegan Lake        New York
15  00008eec5aab2228652e22457881f2d0      Old Fort  North Carolina
16  0000954e7c49ebfbcd91ed9052070bee        Quincy        Illinois
17  0000a1288b8ccdeaaf716a2480d7b06a          None           other
18  0000a2175753bc165e53c408589a3bd6  Grand Rapids        Michigan
19  0000a3fd8b8a3d1a90fbb1e0cd44c62b     Lancaster    Pennsylvania
```

```
   Donor_Is_Teacher Donor_Zip
0                No       602
1                No       245
2               Yes       953
3                No       462
4                No       075
5                No      None
6                No       069
7                No       543
8                No       128
9                No       463
10               No       190
11              Yes       087
12               No       198
13               No       775
14               No       105
15               No       287
16               No       623
17               No      None
18               No       495
19               No       176
```

Trouver le nombre nul dans chaque colonne

```
from pyspark.sql.types import *
from pyspark.sql.functions import *

df_donors.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c)
for c in df_donors.columns]).show()
```

```
+--------+----------+-----------+----------------+---------+
|Donor_ID|Donor_City|Donor_State|Donor_Is_Teacher|Donor_Zip|
+--------+----------+-----------+----------------+---------+
|       0|    105086|          0|               0|    88694|
+--------+----------+-----------+----------------+---------+
```

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

```
df_donors.printSchema()
```

```
root
 |-- Donor_ID: string (nullable = true)
```

```
 |-- Donor_City: string (nullable = true)
 |-- Donor_State: string (nullable = true)
 |-- Donor_Is_Teacher: string (nullable = true)
 |-- Donor_Zip: string (nullable = true)
```

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".

```
df_donors.filter(df_donors.Donor_City.like("A%"))
```

```
DataFrame[Donor_ID: string, Donor_City: string, Donor_State: string,
Donor_Is_Teacher: string, Donor_Zip: string]
```

Affichez les résultats

```
df_donors.filter(df_donors.Donor_City.like("A%")).show()
```

```
+--------------------+-------------+-------------+---------------
+---------+
|            Donor_ID|   Donor_City|  Donor_State|Donor_Is_Teacher|
Donor_Zip|
+--------------------+-------------+-------------+---------------
+---------+
|00002783bc5d10851...|   Appomattox|        other|              No|
245|
|00005454366b6b914...|       Argyle|     New York|              No|
128|
|0001ef9f64a7e1038...|         Ames|         Iowa|              No|
500|
|00024e86676fc2c3b...|    Ann Arbor|     Michigan|              No|
481|
|0002a45d0b45a78e9...|    Ann Arbor|     Michigan|              No|
481|
|0002cb56c84b1cba7...|         Avon|  Connecticut|              No|
060|
|00050297e37eb7632...|       Austin|        Texas|              No|
787|
|0005327bfe18229b9...|        Acton|   California|              No|
935|
|00054f4b278af0c8d...|American Fork|         Utah|              No|
840|
|00055ed4f4745e71d...|  Albuquerque|   New Mexico|              No|
```

```
871|
|000581cf61255745a...|       Ankeny|         Iowa|              No|
500|
|000583fdc4983283e...|       Agawam|Massachusetts|              No|
010|
|00072a3616151a38a...|      Atlanta|        Texas|              No|
303|
|00086e4e19e80f3f1...|       Albany|     New York|              No|
122|
|0008ebb089883290a...|         Avon|      Indiana|              No|
461|
|0008f2f55a3af2432...|   Alexandria|     Virginia|             Yes|
223|
|0008ff59ba832f21a...|   Atascadero|   California|              No|
934|
|000b625c3ba612734...|          Ava|     Illinois|              No|
629|
|000bb57e7995b68d0...|      Anaheim|   California|              No|
928|
|000bba7748e9b7011...|      Allegan|     Michigan|              No|
490|
+--------------------+-------------+-------------+----------------
+---------+
only showing top 20 rows
```

5 - Transformation

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

```
from pyspark.sql.functions import concat_ws
```

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

```
df_donors.withColumn("Address",concat_ws(",",col("Donor_City"),col("Do
nor_State"),col("Donor_Zip")))
```

```
DataFrame[Donor_ID: string, Donor_City: string, Donor_State: string,
Donor_Is_Teacher: string, Donor_Zip: string, Address: string]
```

Afficher les résultats

```
df_donors.withColumn("Address",concat_ws(",",col("Donor_City"),col("Do
nor_State"),col("Donor_Zip"))).show()
```

```
+--------------------+------------+--------------+---------------
+---------+--------------------+
|            Donor_ID|  Donor_City|   Donor_State|Donor_Is_Teacher|
Donor_Zip|             Address|
```

```
+------------------+------------+-------------+---------------
+--------+------------------+
|00000ce845c00cbf0...|    Evanston|      Illinois|             No|
602|Evanston,Illinois...|
|00002783bc5d10851...|  Appomattox|         other|             No|
245|Appomattox,other,245|
|00002d44003ed46b0...|      Winton|    California|            Yes|
953|Winton,California...|
|00002eb25d60a09c3...|Indianapolis|       Indiana|             No|
462|Indianapolis,Indi...|
|0000300773fe015f8...|    Paterson|    New Jersey|             No|
075|Paterson,New Jers...|
|00004c31ce07c2214...|        null|         other|             No|
null|               other|
|00004e32a448b4832...|    Stamford|   Connecticut|             No|
069|Stamford,Connecti...|
|00004fa20a986e60a...|   Green Bay|     Wisconsin|             No|
543|Green Bay,Wiscons...|
|00005454366b6b914...|      Argyle|      New York|             No|
128| Argyle,New York,128|
|0000584b8cdaeaa6b...|  Valparaiso|       Indiana|             No|
463|Valparaiso,Indian...|
|00005f52c98eeaf92...|   Villanova|  Pennsylvania|             No|
190|Villanova,Pennsyl...|
|00006084c3d92d904...|       Brick|    New Jersey|            Yes|
087|Brick,New Jersey,087|
|00006c6b8c3225a54...|  Wilmington|      Delaware|             No|
198|Wilmington,Delawa...|
|0000812bd5629117f...|    Pasadena|         Texas|             No|
775|  Pasadena,Texas,775|
|0000889adf4cc958a...|Mohegan Lake|      New York|             No|
105|Mohegan Lake,New ...|
|00008eec5aab22286...|    Old Fort|North Carolina|             No|
287|Old Fort,North Ca...|
|0000954e7c49ebfbc...|      Quincy|      Illinois|             No|
623| Quincy,Illinois,623|
|0000a1288b8ccdeaa...|        null|         other|             No|
null|               other|
|0000a2175753bc165...|Grand Rapids|      Michigan|             No|
495|Grand Rapids,Mich...|
|0000a3fd8b8a3d1a9...|   Lancaster|  Pennsylvania|             No|
176|Lancaster,Pennsyl...|
+------------------+------------+-------------+---------------
+--------+------------------+
only showing top 20 rows
```

6 - Moteur SQL

Persister le dataset de départ comme une Temporary View

Vous pouvez utliser la fcontion createOrReplaceTempView

```
df_donors.createOrReplaceTempView("df_donors")
```

Comptez le nombre de professeurs ayant participé à la donation

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

```
filtereddf_donors = spark.sql(""" select * from df_donors where
Donor_Is_Teacher = 'Yes' """)
filtereddf_donors.count()
```

```
104650
```

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

```
df_donors_e = df_donors.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

```
Carlifornia_residents = spark.sql(""" select Donor_ID from df_donors
where  Donor_State = 'California' """)
Carlifornia_residents.show()
```

```
+--------------------+
|            Donor_ID|
+--------------------+
|00002d44003ed46b0...|
|00010615b56ff057f...|
|000177bef7ed7b7d1...|
|00017bad30d4b5d99...|
|000181f354f1cba54...|
|0001abd0c3f256bcd...|
|0001bde8e87c867f3...|
|0002128c613edd04b...|
|00031a7e84cb620d9...|
|00032349e9b32f61f...|
|0003748b6011978a1...|
|00038313e9d8b0d93...|
```

```
|00040a049ad9f348f...|
|00041fc8b829a8135...|
|0004265c44e425d71...|
|0004298ea9ff1bf0d...|
|00047ac546738c937...|
|00049338d2a420cd9...|
|0004be01ccfd90c20...|
|0004ceb1d06fd98f0...|
+--------------------+
only showing top 20 rows
```

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

```
df_donations = load_dataframe('Donations.csv')

df_donations.createOrReplaceTempView("df_donations")
```

Afficher le DF

```
df_donations.count()
df_donations.show()
```

```
+--------------------+--------------------+--------------------
+-----------------------------------+--------------
+------------------+----------------------+
|          Project ID|         Donation_ID|            Donor_ID|
Donation_Included_Optional_Donation|Donation_Amount|Donor_Cart
Sequence|Donation_Received_Date|
+--------------------+--------------------+--------------------
+-----------------------------------+--------------
+------------------+----------------------+
|000009891526c0ade...|68872912085866622...|1f4b5b6e68445c6c4...|
No|         178.37|                  11|   2016-08-23 13:15:57|
|000009891526c0ade...|dcf1071da3aa3561f...|4aaab6d244bf35996...|
Yes|           25.0|                   2|   2016-06-06 20:05:23|
|000009891526c0ade...|18a234b9d1e538c43...|0b0765dc9c759adc4...|
Yes|           20.0|                   3|   2016-06-06 14:08:46|
|000009891526c0ade...|38d2744bf9138b0b5...|377944ad61f72d800...|
Yes|           25.0|                   1|   2016-05-15 10:23:04|
|000009891526c0ade...|5a032791e31167a70...|6d5b22d39e68c6560...|
Yes|           25.0|                   2|   2016-05-17 01:23:38|
|000009891526c0ade...|8cea27f0cc03f41f6...|896c75c9b8d9a91c7...|
Yes|           15.0|                   1|   2016-06-04 17:58:55|
|00000ce845c00cbf0...|39af862cb04e4f938...|8a1875762c85932ff...|
Yes|           50.0|                   1|   2013-02-27 09:07:51|
|00000ce845c00cbf0...|c47f78571f62bcf10...|a3f070e439d52de72...|
Yes|           50.0|                   2|   2013-02-27 09:53:12|
|00000ce845c00cbf0...|19351e1d9ae0bccab...|bd323208dc78b1c74...|
Yes|          200.0|                   2|   2013-02-17 21:36:24|
|00000ce845c00cbf0...|d5364b1bb3b145948...|6dd6113f89f2766d3...|
Yes|           10.0|                  44|   2013-02-27 10:32:22|
|00000ce845c00cbf0...|84d4bd0c34c8c28f9...|391f14831940fc7bc...|
```

```
Yes|          100.0|                   1|   2013-02-27 09:55:18|
|00000ce845c00cbf0...|987eecef69373f0d7...|531ed26f1a5052823...|
Yes|           25.0|                   1|   2013-02-27 09:57:57|
|00000ce845c00cbf0...|72f8a2bf2a996b287...|499496888e927737a...|
Yes|           50.0|                   1|   2013-02-27 10:56:48|
|00002d44003ed46b0...|3dc5237cf215a2bdc...|3fa001f7a31563bb2...|
Yes|           25.0|                   1|   2017-06-20 22:45:41|
|00002d44003ed46b0...|3fb12c4ea45461531...|c77b27c9837573aae...|
Yes|           20.0|                   1|   2017-07-05 12:19:02|
|00002d44003ed46b0...|1abb69e9f91e80a4c...|43ca9835ccb5c7c24...|
Yes|          100.0|                   2|   2017-06-28 22:56:05|
|00002d44003ed46b0...|3f878f6ea8afe42b2...|6243c0acf1dc9a4d7...|
Yes|           25.0|                   1|   2017-06-29 19:49:13|
|00002d44003ed46b0...|7b28925a3c4c768da...|3fa001f7a31563bb2...|
No|         117.92|                   3|   2017-07-31 08:48:01|
|00002d44003ed46b0...|aee61d191d3dcaf58...|3fa001f7a31563bb2...|
No|          250.0|                   2|   2017-07-30 23:16:53|
|00002d44003ed46b0...|c40e75f11f570cd59...|344ad0a72366a27bd...|
Yes|           10.0|                   9|   2017-07-24 08:40:35|
+--------------------+--------------------+--------------------
+-----------------------------------+--------------
+--------------------+----------------------+
only showing top 20 rows
```

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".

```
spark.sql("""
select Donation_Amount, max(Donation_Amount) as maxMontant,
min(Donation_Amount) as minMontant, round(avg(Donation_Amount)) as
avgMontant
from df_donations GROUP BY Donation_Amount
""").show()
```

```
+---------------+----------+----------+----------+
|Donation_Amount|maxMontant|minMontant|avgMontant|
+---------------+----------+----------+----------+
|           0.02|      0.02|      0.02|       0.0|
|           0.07|      0.07|      0.07|       0.0|
|           0.11|      0.11|      0.11|       0.0|
|           0.14|      0.14|      0.14|       0.0|
|           0.15|      0.15|      0.15|       0.0|
|           0.16|      0.16|      0.16|       0.0|
```

```
|            0.2|       0.2|       0.2|       0.0|
|           0.21|      0.21|      0.21|       0.0|
|           0.22|      0.22|      0.22|       0.0|
|           0.31|      0.31|      0.31|       0.0|
|           0.32|      0.32|      0.32|       0.0|
|           0.33|      0.33|      0.33|       0.0|
|           0.35|      0.35|      0.35|       0.0|
|           0.38|      0.38|      0.38|       0.0|
|           0.39|      0.39|      0.39|       0.0|
|           0.42|      0.42|      0.42|       0.0|
|           0.45|      0.45|      0.45|       0.0|
|           0.49|      0.49|      0.49|       0.0|
|            0.5|       0.5|       0.5|       1.0|
|           0.53|      0.53|      0.53|       1.0|
+---------------+----------+----------+----------+
only showing top 20 rows
```

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

```
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

```
spark.sql(""" select * from df_donors inner join df_donations on
df_donors.Donor_ID = df_donations.Donor_ID """ )
```

```
DataFrame[Donor_ID: string, Donor_City: string, Donor_State: string,
Donor_Is_Teacher: string, Donor_Zip: string, Project ID: string,
Donation_ID: string, Donor_ID: string,
Donation_Included_Optional_Donation: string, Donation_Amount: string,
Donor_Cart Sequence: string, Donation_Received_Date: string]
```

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" ')

```
spark.sql("""select sum(df_donations.Donation_Amount)
 as amountProf from df_donations inner join df_donors on
df_donations.Donor_ID
 = df_donors.Donor_ID and df_donors.Donor_Is_Teacher
 = "Yes" """).show()
```

```
+-----------------+
|       amountProf|
+-----------------+
|6765183.590000001|
```