Get SparkContext

In [1]:
```
import pyspark
```

In [2]:
```
from pprint import pprint
```

In [3]:
```
sc = pyspark.SparkContext(appName="aas")
```

In [4]:
```
sc
```

Out[4]:

SparkContext

Spark UI (http://192.168.1.6:4041)

Version

v2.3.1

Master

local[*]

AppName

aas

In [5]:
```
from pyspark.sql import SparkSession
```

In [6]:
```
spark = SparkSession.builder \
        .master("local").appName("aas").getOrCreate()
```

Load a file

In [11]:
```
rawblocks = sc.textFile("../Dropbox/pj_ss/donation/block_1.csv")
```

RDD

In [12]:
```
rawblocks
```

Out[12]:
```
../Dropbox/pj_ss/donation/block_1.csv MapPartitionsRDD[3] at textFile a
t NativeMethodAccessorImpl.java:0
```

In [13]:
```
rawblocks.first()
```

Out[13]:
```
'"id_1","id_2","cmp_fname_c1","cmp_fname_c2","cmp_lname_c1","cmp_lname_
c2","cmp_sex","cmp_bd","cmp_bm","cmp_by","cmp_plz","is_match"'
```

In [14]:
```
head = rawblocks.take(10)
```

In [15]:
```
pprint(head)
```

```
['"id_1","id_2","cmp_fname_c1","cmp_fname_c2","cmp_lname_c1","cmp_lname
_c2","cmp_sex","cmp_bd","cmp_bm","cmp_by","cmp_plz","is_match"',
 '37291,53113,0.833333333333333,?,1,?,1,1,1,1,0,TRUE',
 '39086,47614,1,?,1,?,1,1,1,1,1,TRUE',
 '70031,70237,1,?,1,?,1,1,1,1,1,TRUE',
 '84795,97439,1,?,1,?,1,1,1,1,1,TRUE',
 '36950,42116,1,?,1,1,1,1,1,1,1,TRUE',
 '42413,48491,1,?,1,?,1,1,1,1,1,TRUE',
 '25965,64753,1,?,1,?,1,1,1,1,1,TRUE',
 '49451,90407,1,?,1,?,1,1,1,1,0,TRUE',
 '39932,40902,1,?,1,?,1,1,1,1,1,TRUE']
```

In [17]:
```
rawblocks.take(5)
```

Out[17]:
```
['"id_1","id_2","cmp_fname_c1","cmp_fname_c2","cmp_lname_c1","cmp_lname
_c2","cmp_sex","cmp_bd","cmp_bm","cmp_by","cmp_plz","is_match"',
 '37291,53113,0.833333333333333,?,1,?,1,1,1,1,0,TRUE',
 '39086,47614,1,?,1,?,1,1,1,1,1,TRUE',
 '70031,70237,1,?,1,?,1,1,1,1,1,TRUE',
 '84795,97439,1,?,1,?,1,1,1,1,1,TRUE']
```

In [19]:
```
head = sc.parallelize(head)
```

In [23]:
```
head.foreach(print)
# console에 표시됨
```

In [24]:
```
def isHeader(line):
    if line.find("id_1")==-1: return False
    else: return True

head.filter(isHeader).foreach(print)
```

In [25]:
```
head.filter(lambda x: not isHeader(x)).foreach(print)
```

Data Frame

In [26]:
```
parsed = spark.read.option("header", "true") \
                   .option("nullValue", "?") \
                   .option("inferSchema", "true") \
                   .csv('../Dropbox/pj_ss/donation/block_1.csv')
```

In [27]:
```
parsed.printSchema()
```

```
root
 |-- id_1: integer (nullable = true)
 |-- id_2: integer (nullable = true)
 |-- cmp_fname_c1: double (nullable = true)
 |-- cmp_fname_c2: double (nullable = true)
 |-- cmp_lname_c1: double (nullable = true)
 |-- cmp_lname_c2: double (nullable = true)
 |-- cmp_sex: integer (nullable = true)
 |-- cmp_bd: integer (nullable = true)
 |-- cmp_bm: integer (nullable = true)
 |-- cmp_by: integer (nullable = true)
 |-- cmp_plz: integer (nullable = true)
 |-- is_match: boolean (nullable = true)
```

In [28]:
```
parsed.count()
```

Out[28]:
```
574913
```

In [29]:
```
parsed.cache()
```

Out[29]:
```
DataFrame[id_1: int, id_2: int, cmp_fname_c1: double, cmp_fname_c2: dou
ble, cmp_lname_c1: double, cmp_lname_c2: double, cmp_sex: int, cmp_bd:
int, cmp_bm: int, cmp_by: int, cmp_plz: int, is_match: boolean]
```

In [31]:
```
parsed.take(10)
```

Out[31]:
```
[Row(id_1=37291, id_2=53113, cmp_fname_c1=0.833333333333333, cmp_fname_
c2=None, cmp_lname_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_
bm=1, cmp_by=1, cmp_plz=0, is_match=True),
 Row(id_1=39086, id_2=47614, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=70031, id_2=70237, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=84795, id_2=97439, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=36950, id_2=42116, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=1.0, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=1,
cmp_plz=1, is_match=True),
 Row(id_1=42413, id_2=48491, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=25965, id_2=64753, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=49451, id_2=90407, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=0, is_match=True),
 Row(id_1=39932, id_2=40902, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True),
 Row(id_1=46626, id_2=47940, cmp_fname_c1=1.0, cmp_fname_c2=None, cmp_l
name_c1=1.0, cmp_lname_c2=None, cmp_sex=1, cmp_bd=1, cmp_bm=1, cmp_by=
1, cmp_plz=1, is_match=True)]
```

In [30]:
```
parsed.rdd.map(lambda x: x.is_match).countByValue()
```

Out[30]:
```
defaultdict(int, {True: 2093, False: 572820})
```

In [32]:
```
parsed.groupBy("is_match") \
    .count() \
    .orderBy("count", ascending=False) \
    .show()
```

```
+--------+------+
|is_match| count|
+--------+------+
|   false|572820|
|    true|  2093|
+--------+------+
```

In [33]:
```
parsed.agg({"cmp_sex": "avg", "cmp_sex":"stddev"})
```

Out[33]:
```
DataFrame[stddev(cmp_sex): double]
```

In [34]:
```
parsed.agg({"cmp_sex": "avg", "cmp_sex":"stddev"}).show()
```

```
+-------------------+
|    stddev(cmp_sex)|
+-------------------+
|0.20710152240504381|
+-------------------+
```

In [35]:
```
parsed.agg({"cmp_bd": "avg", "cmp_sex":"stddev"}).show()
```

```
+-------------------+-------------------+
|    stddev(cmp_sex)|        avg(cmp_bd)|
+-------------------+-------------------+
|0.20710152240504381|0.22475563232907309|
+-------------------+-------------------+
```

In [36]:
```
parsed.createOrReplaceTempView("linkage")
```

In [37]:
```
spark.sql("""
SELECT is_match, COUNT(*) cnt
FROM linkage
GROUP BY is_match
ORDER BY cnt DESC
""").show()
```

```
+--------+------+
|is_match|   cnt|
+--------+------+
|   false|572820|
|    true|  2093|
+--------+------+
```

In [38]:
```
summary = parsed.describe()
```

In [39]:
```
summary.show()
```

```
+-------+------------------+----------------+------------------+-----
------------+------------------+------------------+-----------------
--+------------------+-----------------+------------------+---------
-----------+
|summary|              id_1|            id_2|       cmp_fname_c1|
cmp_fname_c2|       cmp_lname_c1|       cmp_lname_c2|            cmp_se
x|            cmp_bd|           cmp_bm|            cmp_by|
   cmp_plz|
+-------+------------------+----------------+------------------+-----
------------+------------------+------------------+-----------------
--+------------------+-----------------+------------------+---------
-----------+
|  count|            574913|          574913|            574811|
      10325|            574913|               239|             5749
13|            574851|           574851|            574851|
     573618|
|   mean|33271.962171667714| 66564.6636865056| 0.7127592938251666|0.897
7586763518972|0.31557245780995347|0.32691554145529045| 0.95509233570992
48|0.22475563232907309|0.4886361857246487|0.22266639529199742|0.0054949
46113964...|
| stddev|23622.669425933625|23642.00230967225|0.38892864524635457| 0.27
4257752043053|0.33424946875542494|  0.378309202054067|0.207101522405043
81|0.41742165872355663|0.4998712818281624|0.41603650416456145| 0.073924
02321301918|
|    min|                 1|               6|               0.0|
         0.0|               0.0|               0.0|
0|                 0|                0|                 0|
         0|
|    max|             99894|          100000|               1.0|
         1.0|               1.0|               1.0|
1|                 1|                1|                 1|
         1|
+-------+------------------+----------------+------------------+-----
------------+------------------+------------------+-----------------
--+------------------+-----------------+------------------+---------
-----------+
```

In [40]:
```
summary.select("summary", "cmp_fname_c1", "cmp_fname_c2").show()
```

```
+-------+------------------+------------------+
|summary|       cmp_fname_c1|      cmp_fname_c2|
+-------+------------------+------------------+
|  count|            574811|             10325|
|   mean| 0.7127592938251666|0.8977586763518972|
| stddev|0.38892864524635457| 0.274257752043053|
|    min|               0.0|               0.0|
|    max|               1.0|               1.0|
+-------+------------------+------------------+
```

In [41]:
```
matches = parsed.where("is_match = true")
matchSummary =  matches.describe()
```

In [43]:
```
matchSummary.select("summary", "cmp_fname_c1", "cmp_fname_c2").show()
```

```
+-------+------------------+------------------+
|summary|      cmp_fname_c1|      cmp_fname_c2|
+-------+------------------+------------------+
|  count|              2091|               128|
|   mean|0.9970329792424486|0.9955357142857143|
| stddev|0.03979189523588238|0.05050762722761048|
|    min|               0.0| 0.428571428571429|
|    max|               1.0|               1.0|
+-------+------------------+------------------+
```

In [44]:
```
misses = parsed.filter(parsed.is_match == False)
```

In [45]:
```
misses.describe().select("summary", "cmp_fname_c1", "cmp_fname_c2").show
()
```

```
+-------+------------------+------------------+
|summary|      cmp_fname_c1|      cmp_fname_c2|
+-------+------------------+------------------+
|  count|            572720|             10197|
|   mean|0.7117214109570877|0.8965313093953877|
| stddev|0.3892503865780529|0.27569600395266136|
|    min|               0.0|               0.0|
|    max|               1.0|               1.0|
+-------+------------------+------------------+
```