In [1]:

```
! pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publi
c/simple/
Collecting pyspark
  Downloading pyspark-3.4.0.tar.gz (310.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 310.8/310.8 MB 4.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages
(from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.4.0-py2.py3-none-any.whl size=311317145 s
ha256=ef33712ddcbf5f94e135651fd91eda3b86c5af9c4e0db25a01d1cbc45c1fd6f0
  Stored in directory: /root/.cache/pip/wheels/7b/1b/4b/3363a1d04368e7ff0d408e57ff57966fc
df00583774e761327
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.4.0
```

In [2]:

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.mllib.random import RandomRDDs
from pyspark.sql.types import*
```

In [3]:

```
if 'spark' in dir():
    print("spark context is already created for you!")
else: print("You need to create your own SparkSession object")
```

```
You need to create your own SparkSession object
```

In [4]:

```
spark = SparkSession.builder.appName('lab3').getOrCreate()
sc = spark.sparkContext
```

In [5]:

```
path = "/content/Lab3_view_data.csv"
data_rdd = sc.textFile(path)
```

In [9]:

```
header = data_rdd.first()
new_data_rdd = data_rdd.filter(lambda row: row != header)\
                        .map(lambda x: x.split(","))
```

In [11]:

```
prog_device_day_rdd = new_data_rdd.filter(lambda x: (int(x[3]) >= 200000) and (int(x[3])
< 230000))\
                                   .map(lambda x: ((x[1], x[2]), 1))\
                                   .reduceByKey(lambda  x, y: x + y)\
                                   .sortBy(lambda t: t[1], ascending=False)
```

In [14]:

```
prog_device_rdd = prog_device_day_rdd.map(lambda x: (x[0][0], x[1]))\
                                      .reduceByKey(lambda  x, y: x + y)\
                                      .sortBy(lambda t: t[1], ascending=False)
```

In [18]:

In [18]:

```
dates_num = new_data_rdd.map(lambda row: row[2]).distinct().count()
dates_num
```

Out[18]:

```
14
```

In [21]:

```
average_device_rdd = prog_device_rdd.map(lambda x: (x[0], x[1]/dates_num))\
                        .sortBy(lambda t: t[1], ascending=False)\
                        .collect()
```

In [22]:

```
for i in range(5):
  print('"', end='')
  print(average_device_rdd[i][0], end='')
  print('"', end=' ')
  print(average_device_rdd[i][1])
```

```
"7.5E+14" 97.64285714285714
"7.46E+14" 11.714285714285714
"7.503E+14" 9.357142857142858
"8.00001E+11" 7.5
"8.4843E+14" 5.5
```