For this notebook to work, the prompt command code needs to be "pyspark" instead of "jupyter notebook"

Documentation:

- PySpark RDD: https://spark.apache.org/docs/2.2.0/api/python/pyspark.html#pyspark.RDD (https://spark.apache.org/docs/2.2.0/api/python/pyspark.html#pyspark.RDD)
- PySpark SQL: https://spark.apache.org/docs/latest/sql-programming-guide.html (https://spark.apache.org/docs/latest/sql-programming-guide.html)
- PySpark SQL DataFrame: https://spark.apache.org/docs/2.2.0/api/python/pyspark.sql.html#pyspark.sql.DataFrame (https://spark.apache.org/docs/2.2.0/api/python/pyspark.sql.html#pyspark.sql.DataFrame)

In [1]:
```
# Import pyspark.sql modules
from pyspark.sql import SparkSession, SQLContext, Row

# Packages to upload to firebase
import json
import requests
```

In [2]:
```
# URL for the Firebase database
database_url = 'https://neighborhood-score-la.firebaseio.com/.json'
```

In [3]:
```
# Initiate the sqlSession, sqlContext and get the link for the Spark UI
sqlSession = SparkSession.builder.master("local").appName("Neighborhood-Scorer").
sqlContext = SQLContext(sc)
sqlContext.sparkSession
```

Out[3]:

SparkSession - hive

SparkContext

Spark UI (http://Matheus-PC:4040)

Version

v2.4.2

Master

local[*]

AppName

PySparkShell

Load the Crime Dataset to PySpark and MapReduce to Aggregate Data

In [4]:
```
# Read the csv with neighborhood crime data
df1 = sqlContext.read.csv('Crime_Data_2019_Neighborhoods_v5.csv', header=True)
df1.show()
```

```
+---------+----------+--------------------+-------------------+
|    DR_NO|  DATE OCC|        Neighborhood|Crime_Weighted_Norm|
+---------+----------+--------------------+-------------------+
|191907191|  3/8/2019|              Sylmar|        0.113706949|
|190125334|10/17/2019|Downtown Los Angeles|        0.246650906|
|191920961|12/22/2019|    North Hills East|        0.113706949|
|190604395|  1/9/2019|          Central LA|        0.113706949|
|191310615| 4/30/2019|   South Los Angeles|        0.113706949|
|191419522|  9/3/2019|         Westchester|        0.113706949|
|190123550| 9/21/2019|Downtown Los Angeles|        0.246650906|
|191915118| 8/18/2019|              Sylmar|        0.113706949|
|190811511| 6/20/2019|           Brentwood|        0.113706949|
|191605013| 1/29/2019|          Sun Valley|        0.113706949|
|190129579|12/18/2019|Downtown Los Angeles|        0.113706949|
|191116816| 9/18/2019|Northeast Los Ang...|        0.057744638|
|190220657|10/25/2019|            Westlake|        0.113706949|
|191307997| 3/14/2019|Downtown Los Angeles|        0.246650906|
|190211654| 5/19/2019|            Westlake|        0.113706949|
|191919183|11/12/2019|              Arleta|        0.113706949|
|191106587| 2/20/2019|       Highland Park|        0.113706949|
|191309607| 4/10/2019|   South Los Angeles|        0.113706949|
|190412442| 7/17/2019|     Lincoln Heights|         0.27095249|
|191822844| 11/8/2019|Harbor Gateway North|        0.113706949|
+---------+----------+--------------------+-------------------+
only showing top 20 rows
```

In [5]:
```
# MapReduce to get the count of crimes per neighborhood
df2 = df1.groupBy('Neighborhood').count()
df2 = df2.withColumnRenamed('count', 'CrimeCount')
df2.show()
```

```
+--------------------+----------+
|        Neighborhood|CrimeCount|
+--------------------+----------+
|           Mar Vista|      1364|
|          West Hills|       793|
|           Hollywood|       165|
|       Glassell Park|       232|
|    Pico - Robertson|       844|
|              Harbor|       902|
|            Mid City|      8345|
|Downtown Los Angeles|     13409|
|              Reseda|      2511|
| Crescenta Highlands|         4|
|    North of Montana|         2|
|          Central LA|     24753|
|          McLaughlin|       231|
|       Culver - West|       303|
|       Baldwin Hills|       142|
|      Elysian Valley|        67|
|          Eagle Rock|        10|
|        Sunset Strip|        14|
|        North Arroyo|         1|
|        Century City|       465|
+--------------------+----------+
only showing top 20 rows
```

In [6]:
```
# MapReduce to the the average crime score per neighborhood
df3 = df1.groupBy('Neighborhood').agg({'Crime_Weighted_Norm': 'avg'})
df3 = df3.withColumnRenamed('avg(Crime_Weighted_Norm)', 'CrimeScore')
df3.show()
```

```
+--------------------+--------------------+
|        Neighborhood|          CrimeScore|
+--------------------+--------------------+
|           Mar Vista|  0.1655460595469206|
|          West Hills|  0.1577881623682219|
|           Hollywood| 0.17064889600606062|
|       Glassell Park| 0.13016096131896554|
|    Pico - Robertson| 0.16039390652369653|
|              Harbor| 0.16315045023059846|
|            Mid City| 0.15981678633337423|
|Downtown Los Angeles| 0.16958895540107172|
|              Reseda| 0.14990767685384304|
| Crescenta Highlands|        0.1956636565|
|    North of Montana|0.022944957999999998|
|          Central LA| 0.16590776860829162|
|          McLaughlin| 0.13280403066666668|
|       Culver - West|  0.1599755282871288|
|       Baldwin Hills| 0.15898202638732392|
|      Elysian Valley| 0.22014647929850756|
|          Eagle Rock|        0.1659616953|
|        Sunset Strip| 0.18806723907142858|
|        North Arroyo|         0.037952846|
|        Century City|  0.1710934649311828|
+--------------------+--------------------+
only showing top 20 rows
```

Load the House Dataset to PySpark and MapReduce to Aggregate Data

In [ ]:

In [ ]:

Load the School Dataset to PySpark and MapReduce to Aggregate Data

In [ ]:

In [ ]:

Merge All Datasets Indexing by Neighborhood

In [7]:
```
# Join all dataframes by neighborhood
df4 = df2.join(df3, 'Neighborhood')
df4.show()
```

```
+--------------------+----------+--------------------+
|        Neighborhood|CrimeCount|          CrimeScore|
+--------------------+----------+--------------------+
|           Mar Vista|      1364|  0.1655460595469206|
|          West Hills|       793|  0.1577881623682219|
|           Hollywood|       165| 0.17064889600606062|
|       Glassell Park|       232| 0.13016096131896554|
|    Pico - Robertson|       844| 0.16039390652369653|
|              Harbor|       902| 0.16315045023059846|
|            Mid City|      8345| 0.15981678633337423|
|Downtown Los Angeles|     13409| 0.16958895540107172|
|              Reseda|      2511| 0.14990767685384304|
| Crescenta Highlands|         4|        0.1956636565|
|    North of Montana|         2|0.022944957999999998|
|          Central LA|     24753| 0.16590776860829162|
|          McLaughlin|       231| 0.13280403066666668|
|       Culver - West|       303|  0.1599755282871288|
|       Baldwin Hills|       142| 0.15898202638732392|
|      Elysian Valley|        67| 0.22014647929850756|
|          Eagle Rock|        10|        0.1659616953|
|        Sunset Strip|        14| 0.18806723907142858|
|        North Arroyo|         1|         0.037952846|
|        Century City|       465|  0.1710934649311828|
+--------------------+----------+--------------------+
only showing top 20 rows
```

MapReduce the PySpark DataFrame into a Dictionary

This will be used as the data source for the creation of the jsons to be uploaded

In [9]:
```
# Convert the PySpark Distributed Dataframe to a list of dictionaries to the uplo
dict_by_neighborhood = df4.rdd.map(lambda x: {x['Neighborhood']: {'CrimeCount': x
                                                                  'CrimeScore': x
```

Create the Dictionaries to be Uploaded to Firebase

In [10]:
```
# Create dataset with all data for each neighborhood
neighborhood_dict = {}
for i in dict_by_neighborhood:
    neighborhood_dict[list(i.keys())[0]] = list(i.values())[0]
final_neighborhood = {'NeighborhoodData': neighborhood_dict}
```

In [11]:
```
# Create a dictionary with CrimeCount per Neighborhood
crimecount_dict = {}
for i in range(len(dict_by_neighborhood)):
    crimecount_dict[list(dict_by_neighborhood[i].keys())[0]] = dict_by_neighborhc
final_crimecount = {'CrimeCount': crimecount_dict}
```

In [12]:
```
# Create a dictionary with CrimeScore per Neighborhood
crimescore_dict = {}
for i in range(len(dict_by_neighborhood)):
    crimescore_dict[list(dict_by_neighborhood[i].keys())[0]] = dict_by_neighborhc
final_crimescore = {'CrimeScore': crimescore_dict}
```

Upload Data to Firebase

In [13]:
```
# Patch the Neighborhood Data
patch_neighborhood = requests.patch(database_url, data=json.dumps(final_neighborh
print(f'Patching Neighborhood Data: {patch_neighborhood.reason}')
```

```
Patching Neighborhood Data: OK
```

In [14]:
```
# Patch the CrimeCount Dictionary
patch_crimecount = requests.patch(database_url, data=json.dumps(final_crimecount)
print(f'Patching CrimeCount Data: {patch_crimecount.reason}')
```

```
Patching CrimeCount Data: OK
```

In [15]:
```
# Patch the CrimeScore Dictionary
patch_crimescore = requests.patch(database_url, data=json.dumps(final_crimescore)
print(f'Patching CrimeScore Data: {patch_crimescore.reason}')
```

```
Patching CrimeScore Data: OK
```