BIG DATA ANALYTICS

Introduction:

The objective of this project was to perform some analysis on the queries created by ingesting the "Food and Goods order in Brazil batch data from Kaggle". Using Apache Nifi to Apache Hadoop and then perform analysis on it using Apache Spark. The report contains the snapshots of source codes and brief explanation about the codes and the project.

Importing the required libraries:

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.functions import count
from pyspark.sql.types import Row
from pyspark.sql.functions import countDistinct
```

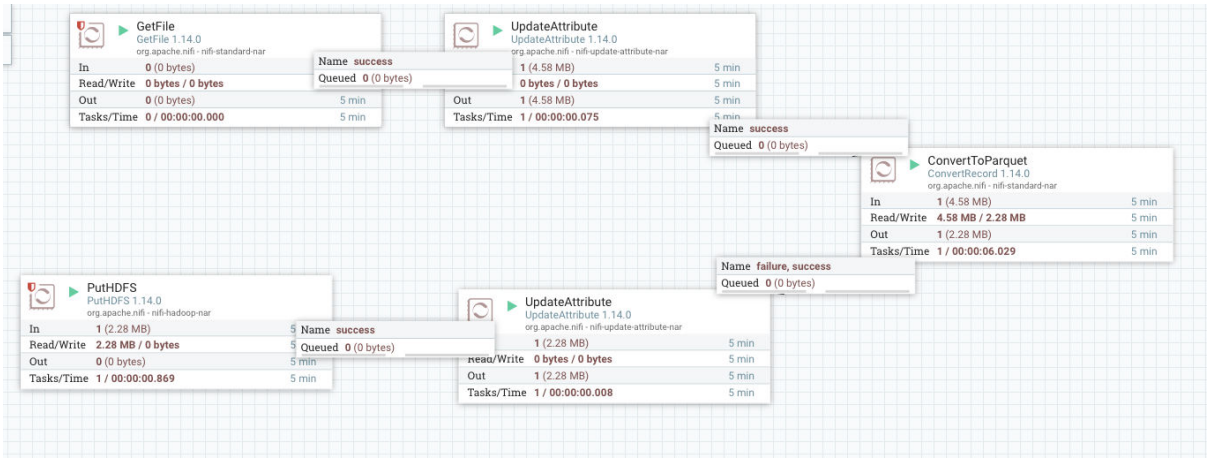
Importing the batch data to perform join on it. So that we can combine those files which are required for the analysis.

```
spark = SparkSession \
    .builder \
    .appName("BDA project") \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()
file_name = '/user/vagrant/stores.csv'
df1 = spark.read.csv(file_name, header='true', inferSchema='true', sep=",")
file_name = '/user/vagrant/hubs.csv'
df2 = spark.read.csv(file_name, header='true', inferSchema='true', sep=",")
df3 = df1.join(df2, "hub_id")
df3.write.option("header",True) \
.csv("final")
```

After joining all the required files using the primary key, I uploaded the resulting dataset i.e. final.csv to Apache Nifi in order to convert it into Apache Parquet format.

Apache Parquet : It is a free and open-source column-oriented data storage format in the Apache Hadoop ecosystem. It is similar to RCFile and ORC, the other columnar-storage file formats in Hadoop, and is compatible with most of the data processing frameworks around Hadoop.

Here is the snapshot for the configuration used in Nifi:



Once it was finished, I was able to see the converted file in the hdfs.

```
Last login: Tue Feb 14 23:00:15 2023 from 10.0.2.2
vagrant@node1:~$ hdfs dfs -ls /user
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.7.6/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/apache-tez-0.9.1-bin/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Found 10 items
drwxr-xr-x   - anonymous supergroup          0 2021-12-31 17:18 /user/anonymous
-rw-r--r--   1 root      supergroup    2386140 2023-02-15 00:18 /user/final.parquet
drwxr-xr-x   - root      supergroup          0 2023-02-14 21:51 /user/hbase
drwxr-xr-x   - anonymous supergroup          0 2021-12-31 17:57 /user/hive
```

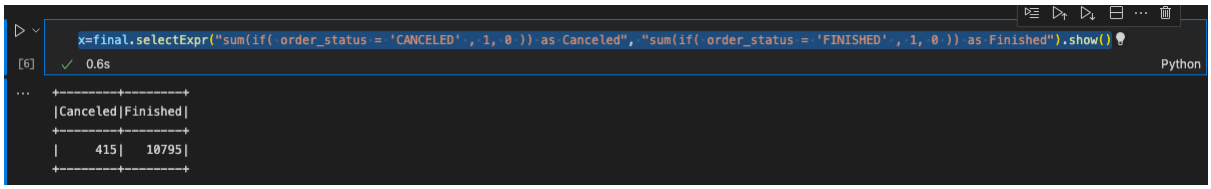
To do some analysis on the converted data, I used pyspark to perform some queries. Here are the queries which were obtained from the data:

1. The first query contains the list of all different types of payment methods that were used to pay the riders in year 2021. We can clearly observe that the apart from cash and card, there are plenty of options offered by the courier services in order to provide ease to the customers and to expand their business.

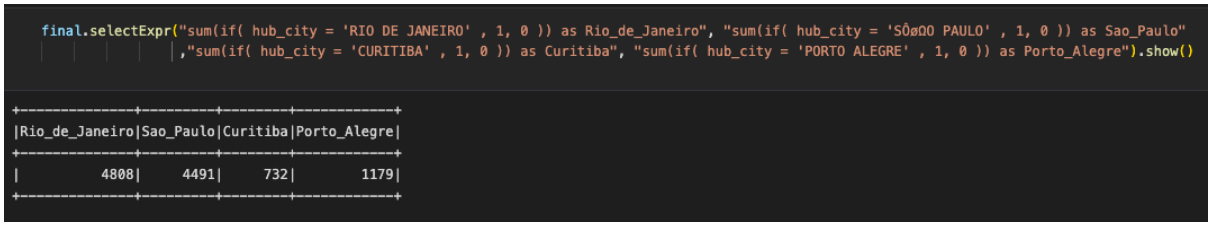
```
dropDisDF = final.dropDuplicates(["payment_method"]).select("payment_method")
dropDisDF.show(truncate=False)

print(final.dropDuplicates(["payment_method"]).select("payment_method").collect())
```

```
+----------------------+
|payment_method        |
+----------------------+
|VOUCHER_DC            |
|CREDIT_STORE          |
|INSTALLMENT_CREDIT_STORE|
|VOUCHER_STORE         |
|DEBIT_STORE           |
|MONEY                 |
|VOUCHER_OL            |
|VOUCHER               |
|STORE_DIRECT_PAYMENT  |
|BANK_TRANSFER_DC      |
|DEBIT                 |
|MEAL_BENEFIT          |
|ONLINE                |
|PAYMENT_LINK          |
|CREDIT                |
+----------------------+
```

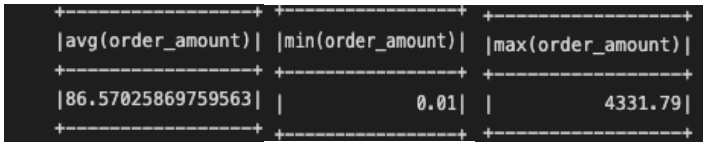
2. The second query contains the count of number of orders that were canceled and one that got completed. We can clearly observe the number of orders which got canceled are almost negligible.



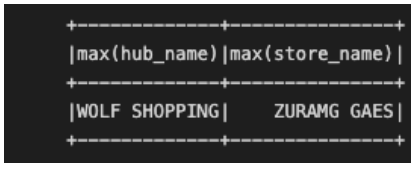
3. The third query was meant to observe in which popular city of Brazil the maximum number of orders were delivered. The results shows that Rio de Janeiro is on the top and Curitiba is at the bottom.



4. In the fourth query I tried to find out the average, minimum and maximum amount of orders. The results here are astonishing because the minimum amount is almost zero. Also, one can find out the average food rates using this data.



5. The fifth query shows about the most and least popular shop and most popular hub.



```
+--------------+------------------+
|  min(hub_name)|    min(store_name)|
+--------------+------------------+
|AVENUE SHOPPING|AIEMAMIMU IU PRUSEM|
+---------------+------------------+
```

6. In the sixth query we measured the delivery distance here also the results are surprising because the maximum deliver distance is 1283 Km.

```
final.agg({'delivery_distance_meters': 'max'}).show() ,final.agg({'delivery_distance_meters': 'min'}).show(), final.agg({'delivery_distance_meters': 'avg'})
```

Python

```
+----------------------------+
|max(delivery_distance_meters)|
+----------------------------+
|                     1283093|
+----------------------------+

+----------------------------+
|min(delivery_distance_meters)|
+----------------------------+
|                          21|
+----------------------------+

+----------------------------+
|avg(delivery_distance_meters)|
+----------------------------+
|           2436.247190008921|
+----------------------------+
```

Once all the queries were finished, I uploaded the analyzed data to hdfs again using command final.write.parquet (“queries.parquet”).

```
drwxr-xr-x   - vagrant   supergroup          0 2023-02-15 00:41 /user/vagrant
vagrant@node1:~$ hdfs dfs -ls /user/vagrant
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.7.6/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/apache-tez-0.9.1-bin/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Found 2 items
drwxr-xr-x   - vagrant supergroup          0 2022-01-02 15:16 /user/vagrant/.sparkStaging
drwxr-xr-x   - vagrant supergroup          0 2023-02-15 00:41 /user/vagrant/queries.parquet
vagrant@node1:~$
```