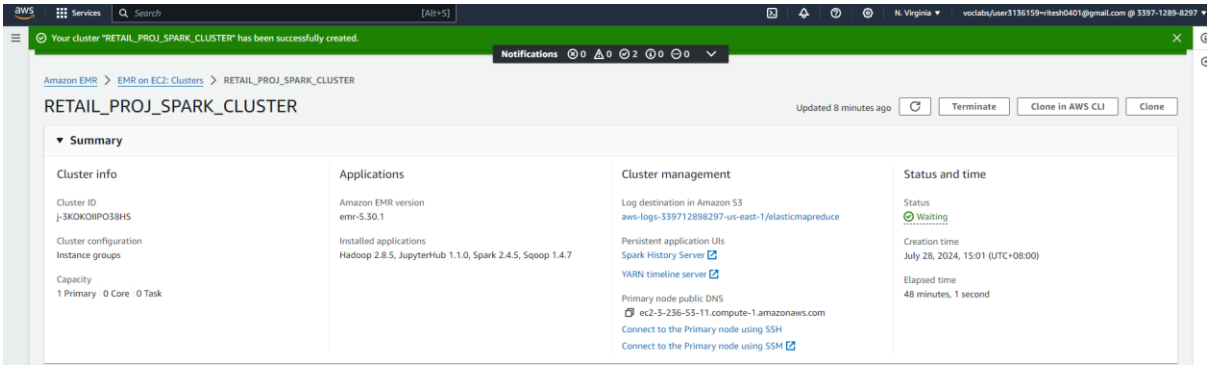
This document explains the code used to do “Retail Data Project” assignment.

1. Pre-requisite:

We need an EMR cluster where we run our spark streaming job(s).



2. <u>spark-streaming.py Code Explanation:</u>

- Importing necessary modules/libraries necessary for spark streaming.

```
## Importing necessary modules

from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.window import Window
```

- Initialising Spark Session.

```
# Initialising SparkSession

spark = SparkSession  \
    .builder  \
    .appName("RetailDataProject")  \
    .getOrCreate()
spark.sparkContext.setLogLevel('ERROR')
```

- Reading retail data info from given Kafka Server

```
# Reading input data from Kafka
raw_input_stream = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers","18.211.252.152:9092") \
    .option("subscribe","real-time-project") \
    .option("startingOffsets", "latest")  \
    .load()
```

- Defining Schema for input stream , Deserialize Kafka Messages and creating Required Order Stream from JSON to a Spark DataFrame.

```
# Defining Schema for Input data

JSON_Schema = StructType() \
    .add("invoice_no", LongType()) \
    .add("country",StringType()) \
    .add("timestamp", TimestampType()) \
    .add("type", StringType()) \
    .add("items", ArrayType(StructType([
        StructField("SKU", StringType()),
        StructField("title", StringType()),
        StructField("unit_price", FloatType()),
        StructField("quantity", IntegerType())
        ])))

order_data_stream =
raw_input_stream.select(from_json(col("value").cast("string"),
JSON_Schema).alias("data")).select("data.*")
```

- Defining User Defined Functions (UDF) and registering them.

```
# Setting up UDFs

# Calculating Total Count of Items in an Order
def total_item_count(items):
    if items is not None:
        item_count =0
        for item in items:
            item_count = item_count + item['quantity']
        return item_count

# Calculating Total Cost of Order
def total_cost(items,type):
    if items is not None:
        total_cost =0
        item_price =0
    for item in items:
        item_price = (item['quantity']*item['unit_price'])
        total_cost = total_cost+ item_price
        item_price=0

    if type  == 'RETURN':
        return total_cost *-1
    else:
        return total_cost
```

```
#Checking if it's an order
def is_a_order(type):
  return 1 if type == 'ORDER' else 0


#Checking if it's a Return order
def is_a_return(type):
  return 1 if type == 'RETURN' else 0

# Registering UDFs

is_order = udf(is_a_order, IntegerType())
is_return = udf(is_a_return, IntegerType())
add_total_item_count = udf(total_item_count, IntegerType())
add_total_cost = udf(total_cost, FloatType())
```

- Now we are calculating some additional columns so that we can calculate required KPIs.

```
# Calculating additional columns for the stream

order_output_stream = order_data_stream \
  .withColumn("total_cost",
add_total_cost(order_data_stream.items,order_data_stream.type)) \
  .withColumn("total_items",
add_total_item_count(order_data_stream.items)) \
  .withColumn("is_order", is_order(order_data_stream.type)) \
  .withColumn("is_return", is_return(order_data_stream.type))
```

- Once we have create additional columns , now we will be writing summarised input table to console.

```
# Writing the summarised input table to the console

order_batch = order_output_stream \
  .select("invoice_no", "country",
"timestamp","total_cost","total_items","is_order","is_return") \
  .writeStream \
  .outputMode("append") \
  .format("console") \
  .option("truncate", "false") \
  .option("path", "/Console_output") \
  .option("checkpointLocation", "/Console_output_checkpoints") \
  .trigger(processingTime="1 minute") \
  .start()
```

- Calculating respective time based and also time & Country based KPIs.

```
# Calculating time-based KPIs for tumbling window of one minute on orders across the globe.

agg_time = order_output_stream \
    .withWatermark("timestamp","1 minutes") \
    .groupby(window("timestamp", "1 minute")) \
    .agg(sum("total_cost").alias("total_volume_of_sales"),
        avg("total_cost").alias("average_transaction_size"),
        count("invoice_no").alias("OPM"),
        avg("is_Return").alias("rate_of_return")) \

.select("window.start","window.end","OPM","total_volume_of_sales","avera
ge_transaction_size","rate_of_return")

# Calculating time- and country-based KPIs for tumbling window of one minute on orders on a per-country basis.

agg_time_country = order_output_stream \
    .withWatermark("timestamp", "1 minutes") \
    .groupBy(window("timestamp", "1 minutes"), "country") \
    .agg(sum("total_cost").alias("total_volume_of_sales"),
        count("invoice_no").alias("OPM"),
        avg("is_Return").alias("rate_of_return")) \
    .select("window.start","window.end","country",
"OPM","total_volume_of_sales","rate_of_return")
```

- Writing both KPIs calculated to HDFS Path(s) in JSON format.

```
# Writing to the HDFS Path : Time based KPI values

KPIByTime = agg_time.writeStream \
    .format("json") \
    .outputMode("append") \
    .option("truncate", "false") \
    .option("path", "timeKPI") \
    .option("checkpointLocation", "timeKPI_checkpoints") \
    .trigger(processingTime="1 minutes") \
    .start()
```

```
# Writing to the HDFS Path: Time and country based KPI values

KPIByTime_country = agg_time_country.writeStream \
    .format("json") \
    .outputMode("append") \
    .option("truncate", "false") \
    .option("path", "time_countryKPI") \
    .option("checkpointLocation", "time_countryKPI_checkpoints") \
    .trigger(processingTime="1 minutes") \
    .start()
```

- Awaiting termination to end the streaming.

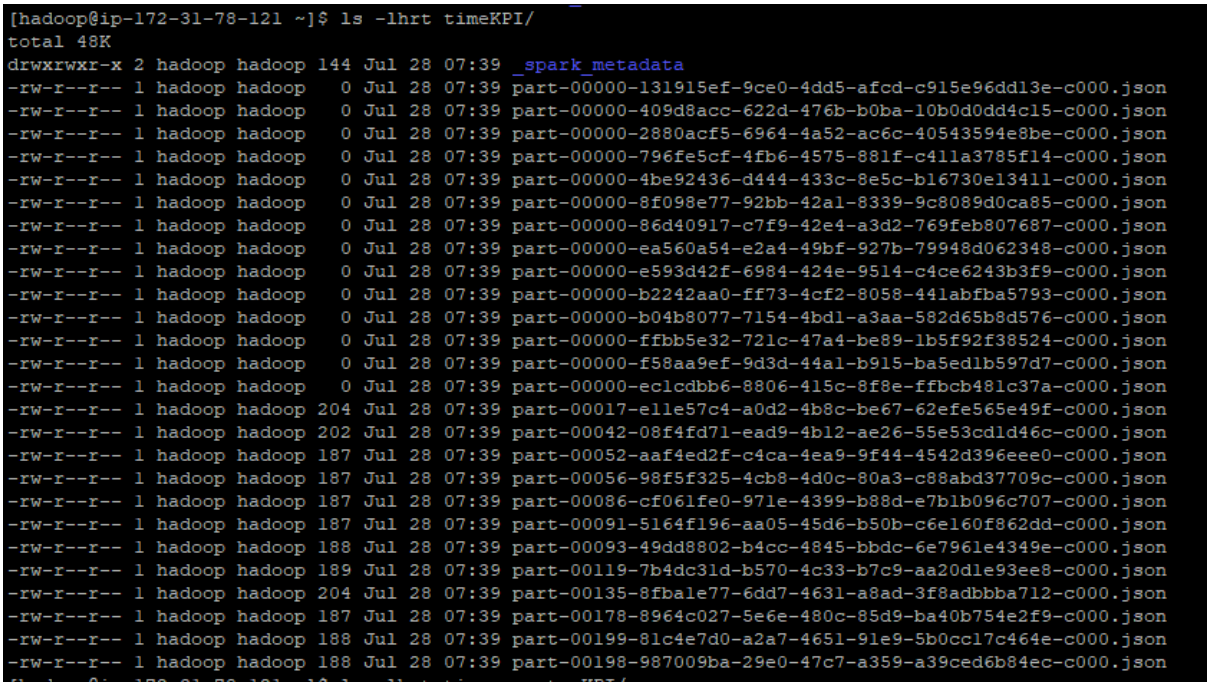
```
order_batch.awaitTermination()
KPIByTime.awaitTermination()
KPIByTime_country.awaitTermination()
```

3. Execution of code.

Code was executed with spark-submit command and console output had been directed to a file.

```
spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark-streaming.py > console-output.txt
```

Code was let run for 10 Minutes and then generated KPIs Json files were collected from HDFS path “timeKPI” and “time_countryKPI”



```
[hadoop@ip-172-31-78-121 ~]$ ls -lhrt timeKPI/
total 48K
drwxrwxr-x 2 hadoop hadoop 144 Jul 28 07:39 _spark_metadata
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-131915ef-9ce0-4dd5-afcd-c915e96dd13e-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-409d8acc-622d-476b-b0ba-10b0d0dd4c15-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-2880acf5-6964-4a52-ac6c-40543594e8be-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-796fe5cf-4fb6-4575-881f-c411a3785f14-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-4be92436-d444-433c-8e5c-b16730e1341l-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-8f098e77-92bb-42a1-8339-9c8089d0ca85-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-86d40917-c7f9-42e4-a3d2-769feb807687-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-ea560a54-e2a4-49bf-927b-79948d062348-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-e593d42f-6984-424e-9514-c4ce6243b3f9-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-b2242aa0-ff73-4cf2-8058-441abfba5793-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-b04b8077-7154-4bd1-a3aa-582d65b8d576-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-ffbb5e32-721c-47a4-be89-1b5f92f38524-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-f58aa9ef-9d3d-44a1-b915-ba5ed1b597d7-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:39 part-00000-ec1cdbb6-8806-415c-8f8e-ffbcb481c37a-c000.json
-rw-r--r-- 1 hadoop hadoop 204 Jul 28 07:39 part-00017-e11e57c4-a0d2-4b8c-be67-62efe565e49f-c000.json
-rw-r--r-- 1 hadoop hadoop 202 Jul 28 07:39 part-00042-08f4fd71-ead9-4b12-ae26-55e53cd1d46c-c000.json
-rw-r--r-- 1 hadoop hadoop 187 Jul 28 07:39 part-00052-aaf4ed2f-c4ca-4ea9-9f44-4542d396eee0-c000.json
-rw-r--r-- 1 hadoop hadoop 187 Jul 28 07:39 part-00056-98f5f325-4cb8-4d0c-80a3-c88abd37709c-c000.json
-rw-r--r-- 1 hadoop hadoop 187 Jul 28 07:39 part-00086-cf061fe0-971e-4399-b88d-e7b1b096c707-c000.json
-rw-r--r-- 1 hadoop hadoop 187 Jul 28 07:39 part-00091-5164f196-aa05-45d6-b50b-c6e160f862dd-c000.json
-rw-r--r-- 1 hadoop hadoop 188 Jul 28 07:39 part-00093-49dd8802-b4cc-4845-bbdc-6e796e4349e-c000.json
-rw-r--r-- 1 hadoop hadoop 189 Jul 28 07:39 part-00119-7b4dc31d-b570-4c33-b7c9-aa20d1e93ee8-c000.json
-rw-r--r-- 1 hadoop hadoop 204 Jul 28 07:39 part-00135-8fba1e77-6dd7-4631-a8ad-3f8adbbba712-c000.json
-rw-r--r-- 1 hadoop hadoop 187 Jul 28 07:39 part-00178-8964c027-5e6e-480c-85d9-ba40b754e2f9-c000.json
-rw-r--r-- 1 hadoop hadoop 188 Jul 28 07:39 part-00199-81c4e7d0-a2a7-4651-91e9-5b0cc17c464e-c000.json
-rw-r--r-- 1 hadoop hadoop 188 Jul 28 07:39 part-00198-987009ba-29e0-47c7-a359-a39ced6b84ec-c000.json
```

```
[hadoop@ip-172-31-78-121 ~]$ ls -lhrt time_countryKPI/
total 92K
drwxrwxr-x 2 hadoop hadoop 154 Jul 28 07:40 _spark_metadata
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-0f5d595f-a7d4-4c72-a4ca-a6264927a43b-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-09d023f6-4f4d-4e9e-a4f5-806683d17d65-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-25f47588-bdcf-40bf-8c48-90486bb50056-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-14063c17-1e68-403f-9e9d-5248f1d72c04-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-54bc1b2c-985b-4b98-9486-752da73701a1-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-4220e8ee-7f0d-4655-b2f3-c1a0402dcfde-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-316da11d-230f-4354-b7d9-7cc0d7d3b281-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-d1b9b4f4-a9fe-4b64-8fe6-9d7df26def76-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-cdb350b1-fa9d-49fa-8aec-b3aa90fa76e2-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-a97dca88-70db-4eab-9a9a-3e9cff741815-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-9d107c41-6920-455d-896e-8c9c82aa6ceb-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-68238f61-053c-402c-8bdf-19cc3d860f98-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-f106ee9f-71ec-43ed-aa38-678efd5c3e05-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-ee6c9e5b-6afc-404e-b2b0-227f3fa09950-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-e909323e-4a4f-4072-810c-14c430f557e9-c000.json
-rw-r--r-- 1 hadoop hadoop   0 Jul 28 07:40 part-00000-dbb255ce-c9ff-4ce9-8454-9ccd9ba115le-c000.json
-rw-r--r-- 1 hadoop hadoop 159 Jul 28 07:40 part-00004-5a23e9ad-43ca-4680-a8e7-7700e788223b-c000.json
-rw-r--r-- 1 hadoop hadoop 161 Jul 28 07:40 part-00008-6af2ae3e-7f74-4828-b9c2-3e609d18efb8-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00009-ffdbe957-8642-4d76-8d1a-5b32836fb00a-c000.json
-rw-r--r-- 1 hadoop hadoop 161 Jul 28 07:40 part-00009-9aafc772-c8f5-4da9-9e66-5414456ceae3-c000.json
-rw-r--r-- 1 hadoop hadoop 159 Jul 28 07:40 part-00018-0936a337-8720-4761-aa67-fe8a72041cb7-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00022-3675db9e-ae90-4f98-8e46-a2287d8d1d37-c000.json
-rw-r--r-- 1 hadoop hadoop 168 Jul 28 07:40 part-00045-6fd47359-f166-4772-a89a-0851aaa32107-c000.json
-rw-r--r-- 1 hadoop hadoop 183 Jul 28 07:40 part-00030-d90209ee-72f2-42af-b51f-b08f58a85d04-c000.json
-rw-r--r-- 1 hadoop hadoop 161 Jul 28 07:40 part-00055-59dea522-b2c6-497e-a501-c20d7e5c7c51-c000.json
-rw-r--r-- 1 hadoop hadoop 170 Jul 28 07:40 part-00056-7e4d511f-2625-4772-a51b-2f9f57970e25-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00056-4014c76f-e75a-4d13-9501-5ef7a79780b0-c000.json
-rw-r--r-- 1 hadoop hadoop 161 Jul 28 07:40 part-00069-a381298d-fa68-434c-afda-0f25043b4c52-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00082-f13a53a4-81a0-4110-b107-a7692c274c3a-c000.json
-rw-r--r-- 1 hadoop hadoop 185 Jul 28 07:40 part-00076-4e3e5e6b-230d-47e2-8c7b-e0e37ae2c89c-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00083-e57836ea-062d-4732-be13-ce69bc2b54d6-c000.json
-rw-r--r-- 1 hadoop hadoop 159 Jul 28 07:40 part-00092-2e647a6a-d460-4583-89d6-81136b813998-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00103-6b048f8d-6aae-4b45-9be7-03b49048e874-c000.json
-rw-r--r-- 1 hadoop hadoop 169 Jul 28 07:40 part-00152-01bd1aa2-9aac-46e6-9f54-22a25c1666c2-c000.json
-rw-r--r-- 1 hadoop hadoop 159 Jul 28 07:40 part-00168-efea95f6-2b77-4c96-8e9a-e96bd079c916-c000.json
-rw-r--r-- 1 hadoop hadoop 168 Jul 28 07:40 part-00162-103557fd-1106-4bf3-b841-c85a0abe9865-c000.json
-rw-r--r-- 1 hadoop hadoop 170 Jul 28 07:40 part-00169-d29d9186-4642-4f92-ab5f-2e66f1114b86-c000.json
-rw-r--r-- 1 hadoop hadoop 165 Jul 28 07:40 part-00179-e57ed487-6bf3-4c5f-8278-5f6fe9532f7f-c000.json
-rw-r--r-- 1 hadoop hadoop 162 Jul 28 07:40 part-00194-07981ac3-8a1d-4bf7-8f67-abdacf251c82-c000.json
```

Console-output:

```
[hadoop@ip-172-31-78-121 ~]$ tail -f console-output.txt
-------------------------------------------
Batch: 0
-------------------------------------------
+----------+-------+---------+----------+-----------+--------+---------+
|invoice_no|country|timestamp|total_cost|total_items|is_order|is_return|
+----------+-------+---------+----------+-----------+--------+---------+
+----------+-------+---------+----------+-----------+--------+---------+

-------------------------------------------
Batch: 1
-------------------------------------------
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|invoice_no     |country       |timestamp          |total_cost|total_items|is_order|is_return|
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|154132561342829|United Kingdom|2024-07-28 07:28:31|1.25      |1          |1       |0        |
|154132561342830|United Kingdom|2024-07-28 07:28:37|5.65      |5          |1       |0        |
|154132561342831|United Kingdom|2024-07-28 07:28:39|182.51    |125        |1       |0        |
|154132561342832|United Kingdom|2024-07-28 07:29:16|4.96      |1          |1       |0        |
+---------------+--------------+-------------------+----------+-----------+--------+---------+

-------------------------------------------
Batch: 2
-------------------------------------------
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|invoice_no     |country       |timestamp          |total_cost|total_items|is_order|is_return|
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|154132561342833|United Kingdom|2024-07-28 07:29:27|31.25     |25         |1       |0        |
|154132561342834|United Kingdom|2024-07-28 07:29:28|300.91    |194        |1       |0        |
|154132561342835|United Kingdom|2024-07-28 07:29:35|3.3       |2          |1       |0        |
|154132561342836|United Kingdom|2024-07-28 07:29:43|6.75      |1          |1       |0        |
|154132561342837|EIRE          |2024-07-28 07:29:49|-651.54   |126        |0       |1        |
|154132561342838|United Kingdom|2024-07-28 07:29:58|65.34     |38         |1       |0        |
|154132561342839|United Kingdom|2024-07-28 07:30:03|9.35      |11         |1       |0        |
|154132561342840|United Kingdom|2024-07-28 07:30:15|22.5      |6          |1       |0        |
+---------------+--------------+-------------------+----------+-----------+--------+---------+

-------------------------------------------
Batch: 3
-------------------------------------------
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|invoice_no     |country       |timestamp          |total_cost|total_items|is_order|is_return|
+---------------+--------------+-------------------+----------+-----------+--------+---------+
|154132561342841|United Kingdom|2024-07-28 07:30:36|5.9       |2          |1       |0        |
|154132561342842|United Kingdom|2024-07-28 07:30:41|14.150001 |13         |1       |0        |
|154132561342843|United Kingdom|2024-07-28 07:30:51|1.7       |2          |1       |0        |
|154132561342844|United Kingdom|2024-07-28 07:31:00|2.95      |1          |1       |0        |
|154132561342845|United Kingdom|2024-07-28 07:31:03|14.1      |19         |1       |0        |
|154132561342846|United Kingdom|2024-07-28 07:31:10|675.24    |420        |1       |0        |
|154132561342847|United Kingdom|2024-07-28 07:31:22|18.03     |7          |1       |0        |
+---------------+--------------+-------------------+----------+-----------+--------+---------+
```