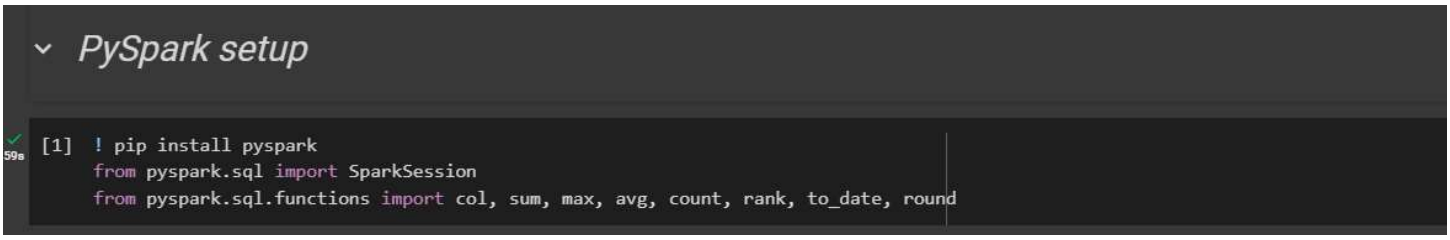
SaiPrabath Chowdary S

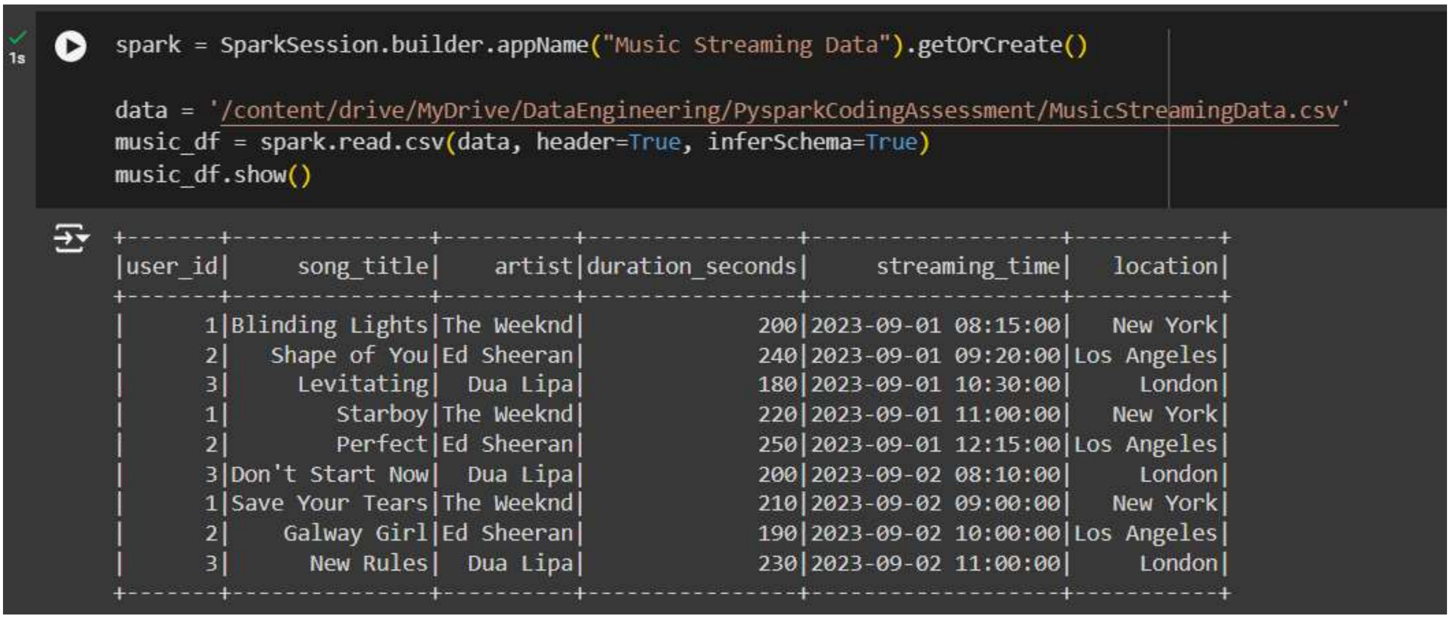
PySpark Coding Challenge 10th Sep 2024

Music Streaming Data



PySpark setup

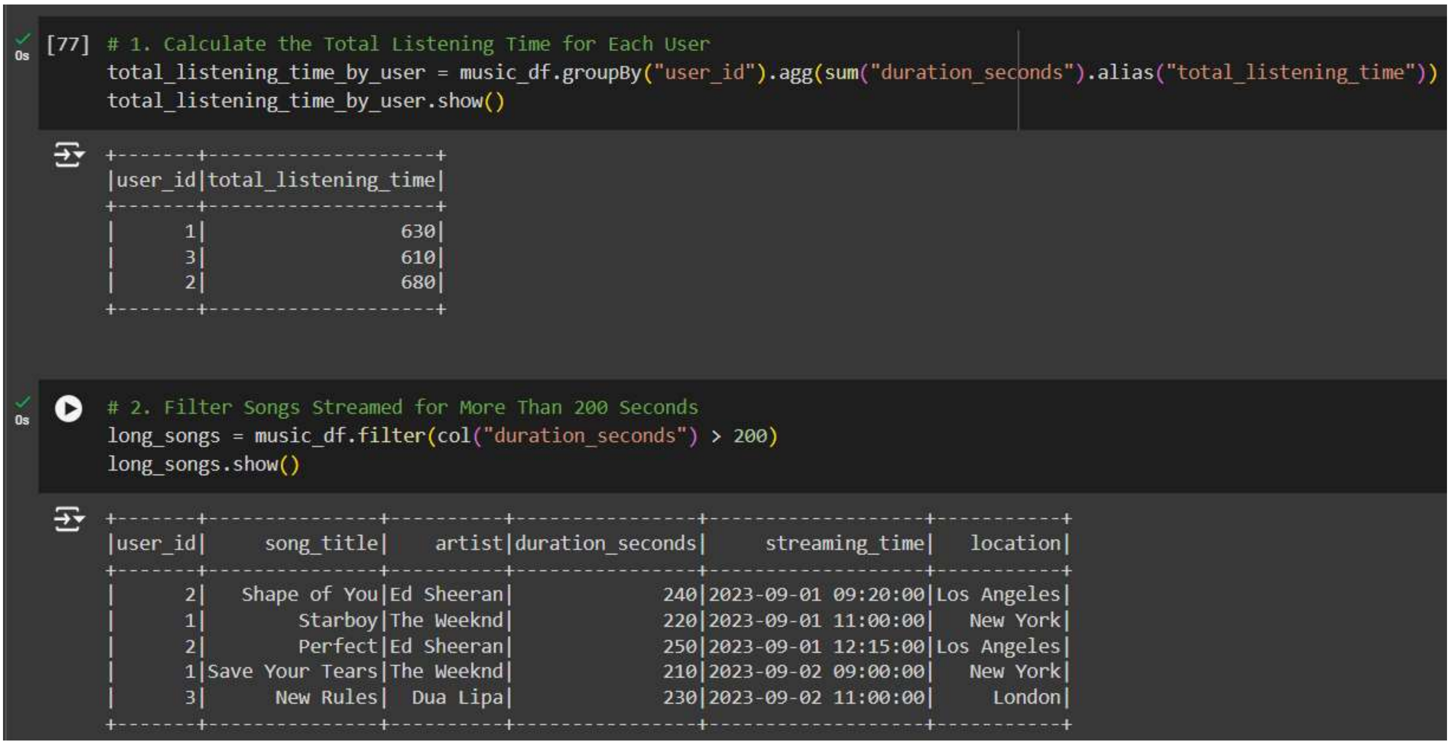
```
[1] ! pip install pyspark
    from pyspark.sql import SparkSession
    from pyspark.sql.functions import col, sum, max, avg, count, rank, to_date, round
```



```
spark = SparkSession.builder.appName("Music Streaming Data").getOrCreate()

data = '/content/drive/MyDrive/DataEngineering/PysparkCodingAssessment/MusicStreamingData.csv'
music_df = spark.read.csv(data, header=True, inferSchema=True)
music_df.show()
```

```
+-------+---------------+----------+----------------+-------------------+-----------+
|user_id|     song_title|    artist|duration_seconds|     streaming_time|   location|
+-------+---------------+----------+----------------+-------------------+-----------+
|      1|Blinding Lights|The Weeknd|             200|2023-09-01 08:15:00|   New York|
|      2|   Shape of You|Ed Sheeran|             240|2023-09-01 09:20:00|Los Angeles|
|      3|      Levitating|  Dua Lipa|             180|2023-09-01 10:30:00|     London|
|      1|        Starboy|The Weeknd|             220|2023-09-01 11:00:00|   New York|
|      2|        Perfect|Ed Sheeran|             250|2023-09-01 12:15:00|Los Angeles|
|      3| Don't Start Now|  Dua Lipa|             200|2023-09-02 08:10:00|     London|
|      1|Save Your Tears|The Weeknd|             210|2023-09-02 09:00:00|   New York|
|      2|     Galway Girl|Ed Sheeran|             190|2023-09-02 10:00:00|Los Angeles|
|      3|      New Rules|  Dua Lipa|             230|2023-09-02 11:00:00|     London|
+-------+---------------+----------+----------------+-------------------+-----------+
```

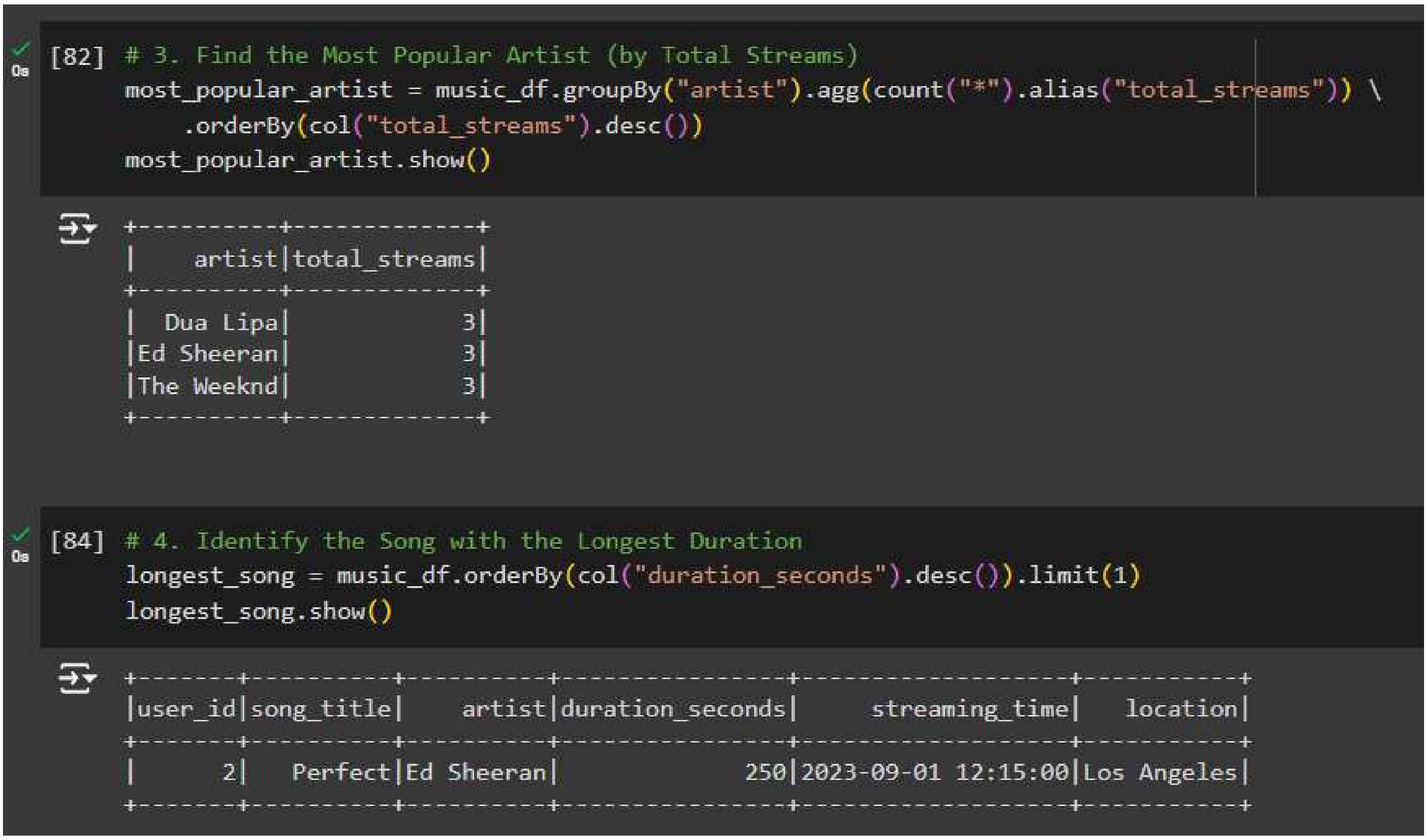


```
[77] # 1. Calculate the Total Listening Time for Each User
     total_listening_time_by_user = music_df.groupBy("user_id").agg(sum("duration_seconds").alias("total_listening_time"))
     total_listening_time_by_user.show()
```

```
+-------+--------------------+
|user_id|total_listening_time|
+-------+--------------------+
|      1|                 630|
|      3|                 610|
|      2|                 680|
+-------+--------------------+
```

```
# 2. Filter Songs Streamed for More Than 200 Seconds
long_songs = music_df.filter(col("duration_seconds") > 200)
long_songs.show()
```

```
+-------+---------------+----------+----------------+-------------------+-----------+
|user_id|     song_title|    artist|duration_seconds|     streaming_time|   location|
+-------+---------------+----------+----------------+-------------------+-----------+
|      2|   Shape of You|Ed Sheeran|             240|2023-09-01 09:20:00|Los Angeles|
|      1|        Starboy|The Weeknd|             220|2023-09-01 11:00:00|   New York|
|      2|        Perfect|Ed Sheeran|             250|2023-09-01 12:15:00|Los Angeles|
|      1|Save Your Tears|The Weeknd|             210|2023-09-02 09:00:00|   New York|
|      3|      New Rules|  Dua Lipa|             230|2023-09-02 11:00:00|     London|
+-------+---------------+----------+----------------+-------------------+-----------+
```

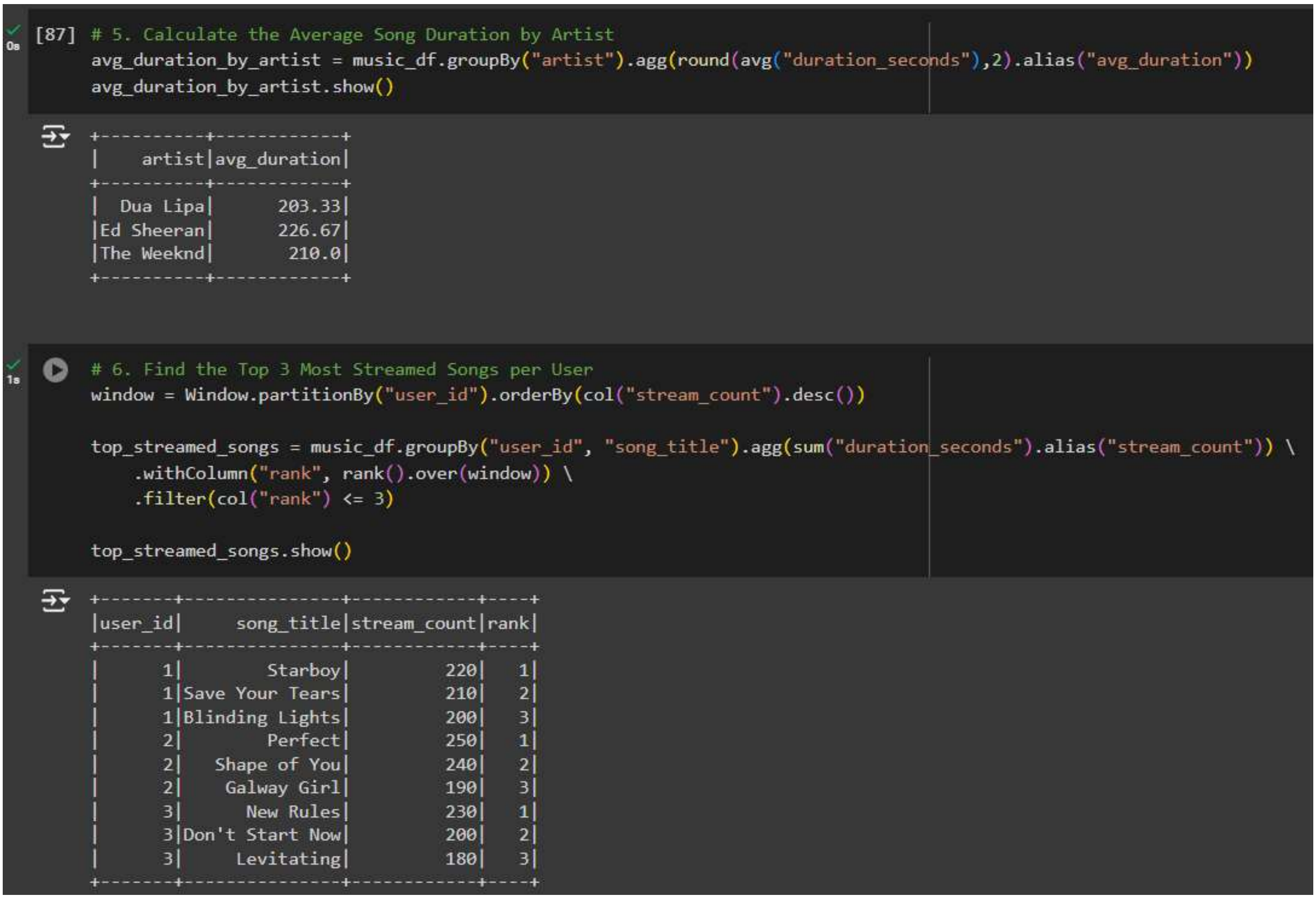


```
[82] # 3. Find the Most Popular Artist (by Total Streams)
     most_popular_artist = music_df.groupBy("artist").agg(count("*").alias("total_streams")) \
         .orderBy(col("total_streams").desc())
     most_popular_artist.show()
```

```
+----------+-------------+
|    artist|total_streams|
+----------+-------------+
|  Dua Lipa|            3|
|Ed Sheeran|            3|
|The Weeknd|            3|
+----------+-------------+
```

```
[84] # 4. Identify the Song with the Longest Duration
     longest_song = music_df.orderBy(col("duration_seconds").desc()).limit(1)
     longest_song.show()
```

```
+-------+----------+----------+----------------+-------------------+-----------+
|user_id|song_title|    artist|duration_seconds|     streaming_time|   location|
+-------+----------+----------+----------------+-------------------+-----------+
|      2|   Perfect|Ed Sheeran|             250|2023-09-01 12:15:00|Los Angeles|
+-------+----------+----------+----------------+-------------------+-----------+
```



```
[87] # 5. Calculate the Average Song Duration by Artist
     avg_duration_by_artist = music_df.groupBy("artist").agg(round(avg("duration_seconds"),2).alias("avg_duration"))
     avg_duration_by_artist.show()
```

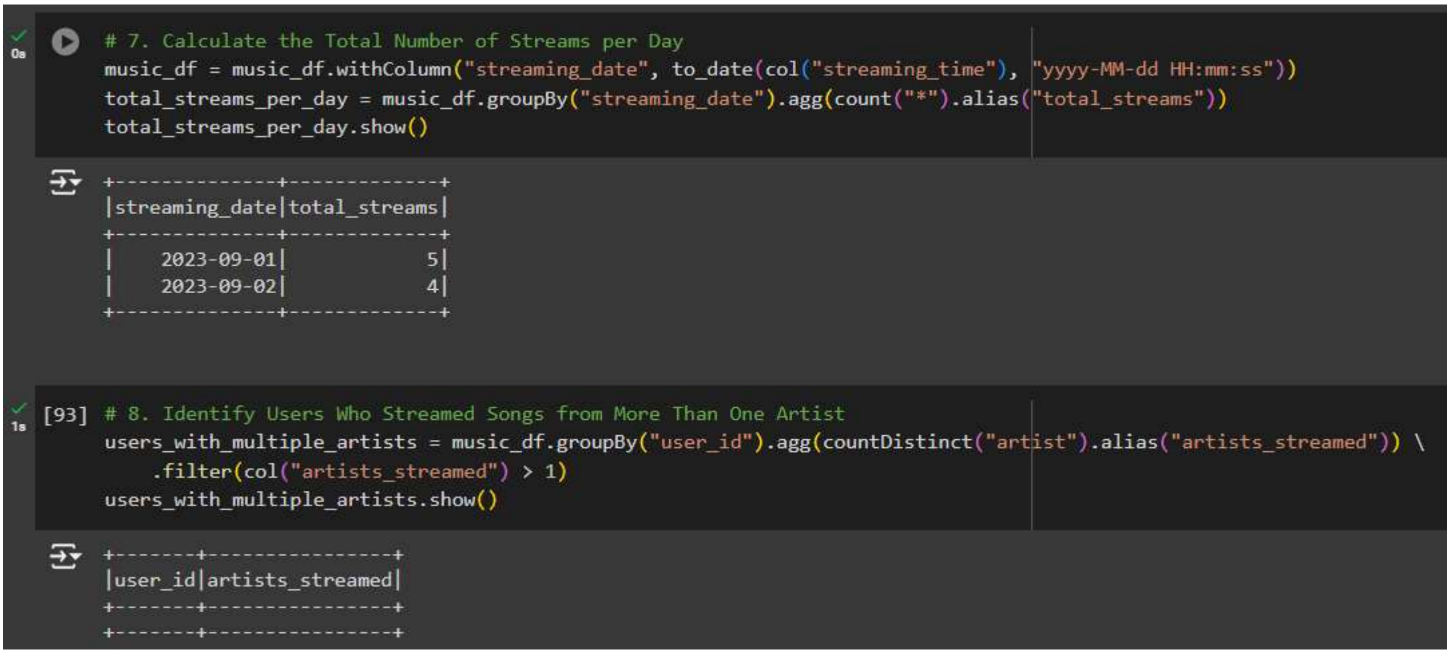
```
+----------+------------+
|    artist|avg_duration|
+----------+------------+
|  Dua Lipa|      203.33|
|Ed Sheeran|      226.67|
|The Weeknd|       210.0|
+----------+------------+
```

```
# 6. Find the Top 3 Most Streamed Songs per User
window = Window.partitionBy("user_id").orderBy(col("stream_count").desc())

top_streamed_songs = music_df.groupBy("user_id", "song_title").agg(sum("duration_seconds").alias("stream_count")) \
    .withColumn("rank", rank().over(window)) \
    .filter(col("rank") <= 3)

top_streamed_songs.show()
```

```
+-------+---------------+------------+----+
|user_id|     song_title|stream_count|rank|
+-------+---------------+------------+----+
|      1|        Starboy|         220|   1|
|      1|Save Your Tears|         210|   2|
|      1|Blinding Lights|         200|   3|
|      2|        Perfect|         250|   1|
|      2|    Shape of You|         240|   2|
|      2|    Galway Girl|         190|   3|
|      3|      New Rules|         230|   1|
|      3| Don't Start Now|         200|   2|
|      3|      Levitating|         180|   3|
+-------+---------------+------------+----+
```

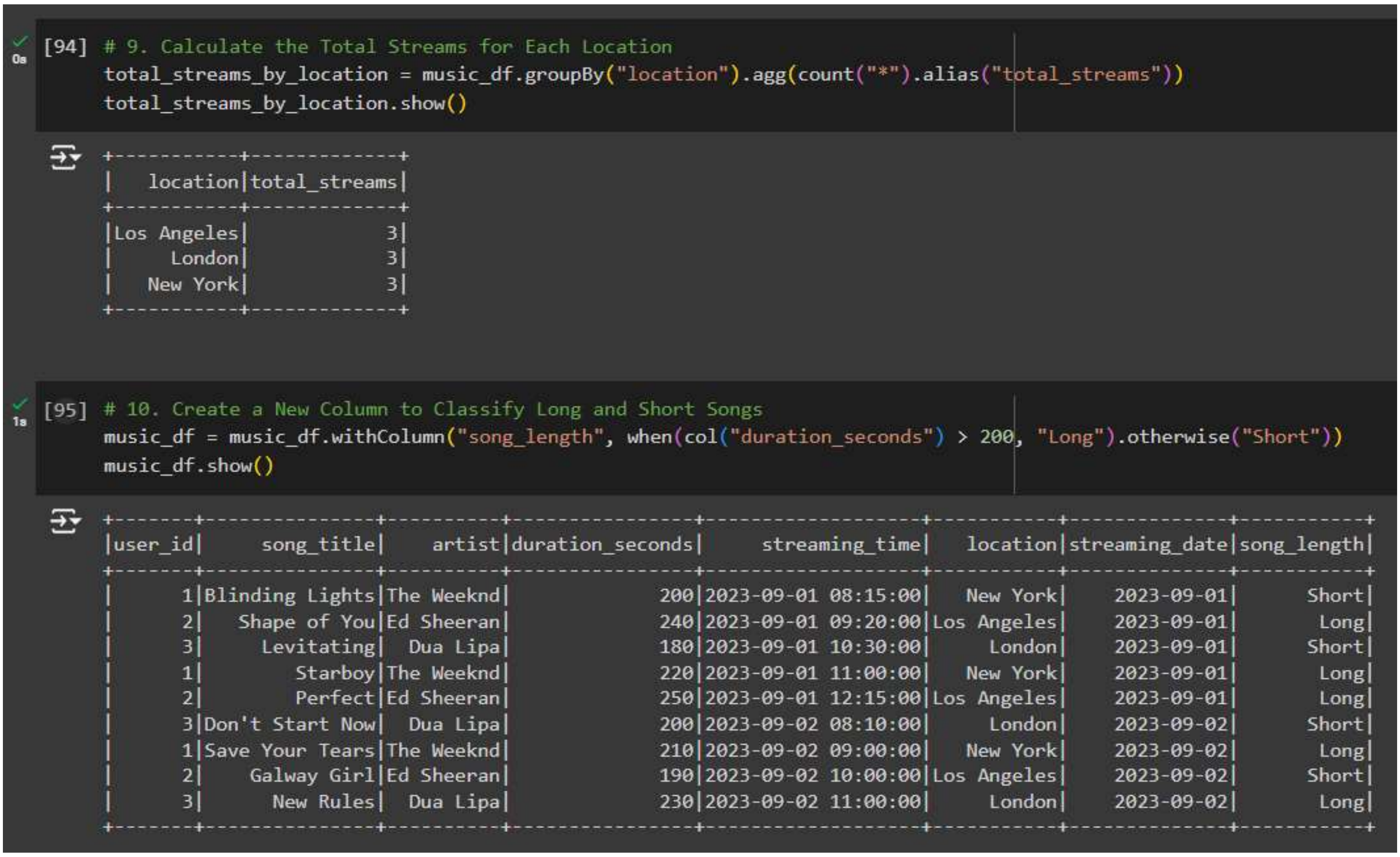


```
# 7. Calculate the Total Number of Streams per Day
music_df = music_df.withColumn("streaming_date", to_date(col("streaming_time"), "yyyy-MM-dd HH:mm:ss"))
total_streams_per_day = music_df.groupBy("streaming_date").agg(count("*").alias("total_streams"))
total_streams_per_day.show()
```

```
+--------------+-------------+
|streaming_date|total_streams|
+--------------+-------------+
|    2023-09-01|            5|
|    2023-09-02|            4|
+--------------+-------------+
```

```
[93] # 8. Identify Users Who Streamed Songs from More Than One Artist
users_with_multiple_artists = music_df.groupBy("user_id").agg(countDistinct("artist").alias("artists_streamed")) \
    .filter(col("artists_streamed") > 1)
users_with_multiple_artists.show()
```

```
+-------+----------------+
|user_id|artists_streamed|
+-------+----------------+
+-------+----------------+
```



```
[94] # 9. Calculate the Total Streams for Each Location
total_streams_by_location = music_df.groupBy("location").agg(count("*").alias("total_streams"))
total_streams_by_location.show()
```

```
+-----------+-------------+
|   location|total_streams|
+-----------+-------------+
|Los Angeles|            3|
|     London|            3|
|   New York|            3|
+-----------+-------------+
```

```
[95] # 10. Create a New Column to Classify Long and Short Songs
music_df = music_df.withColumn("song_length", when(col("duration_seconds") > 200, "Long").otherwise("Short"))
music_df.show()
```

```
+-------+---------------+----------+----------------+-------------------+-----------+--------------+-----------+
|user_id|     song_title|    artist|duration_seconds|     streaming_time|   location|streaming_date|song_length|
+-------+---------------+----------+----------------+-------------------+-----------+--------------+-----------+
|      1|Blinding Lights|The Weeknd|             200|2023-09-01 08:15:00|   New York|    2023-09-01|      Short|
|      2|   Shape of You|Ed Sheeran|             240|2023-09-01 09:20:00|Los Angeles|    2023-09-01|       Long|
|      3|     Levitating|  Dua Lipa|             180|2023-09-01 10:30:00|     London|    2023-09-01|      Short|
|      1|        Starboy|The Weeknd|             220|2023-09-01 11:00:00|   New York|    2023-09-01|       Long|
|      2|        Perfect|Ed Sheeran|             250|2023-09-01 12:15:00|Los Angeles|    2023-09-01|       Long|
|      3|Don't Start Now|  Dua Lipa|             200|2023-09-02 08:10:00|     London|    2023-09-02|      Short|
|      1|Save Your Tears|The Weeknd|             210|2023-09-02 09:00:00|   New York|    2023-09-02|       Long|
|      2|    Galway Girl|Ed Sheeran|             190|2023-09-02 10:00:00|Los Angeles|    2023-09-02|      Short|
|      3|      New Rules|  Dua Lipa|             230|2023-09-02 11:00:00|     London|    2023-09-02|       Long|
+-------+---------------+----------+----------------+-------------------+-----------+--------------+-----------+
```