Music Streaming:

```
from pyspark.sql import SparkSession

from pyspark.sql.functions import col

from pyspark.sql import functions as F

from pyspark.sql.window import Window


spark = SparkSession.builder.appName("MusicStreaming").getOrCreate()


music_df =
spark.read.format("csv").option("header","true").option("inferSchema","true")
.load("/content/sample_data/music_streaming.csv")
```

 # 1. Calculate the Total Listening Time for Each User

```
total_listening_time =
music_df.groupBy("user_id").agg(F.sum("duration_seconds").alias("total_listen
ing_time"))

total_listening_time.show()
```

 # 2. Filter Songs Streamed for More Than 200 Seconds

```
high_duration = music_df.filter(col("duration_seconds") > 200)

high_duration.show()
```

 # 3. Find the Most Popular Artist (by Total Streams)

```
popular_artist = music_df.groupBy("artist").agg(F.count("*").alias("total_streams")).orderBy(col("total_streams").desc()).limit(1)

popular_artist.show()
```

 # 4. Identify the Song with the Longest Duration

```
longest_duration_song = music_df.orderBy(col("duration_seconds").desc()).select("song_title","artist","duration_seconds").limit(1)

longest_duration_song.show()
```

 # 5. Calculate the Average Song Duration by Artist

```
avg_artist_duration = music_df.groupBy("artist").agg(F.avg("duration_seconds").alias("average_duration"))

avg_artist_duration.show()
```

 # 6. Find the Top 3 Most Streamed Songs per User

```
grouped_df = music_df.groupBy("user_id","song_title").agg(F.count("*").alias("play_count"))


window_spec = Window.partitionBy("user_id").orderBy(col("play_count").desc())
```

```
ranked_df =
grouped_df.withColumn("rank",F.row_number().over(window_spec))


top_3_df = ranked_df.filter(col("rank") <=3).orderBy(col("user_id"),col("rank"))

top_3_df.show()
```

 # 7. Calculate the Total Number of Streams per Day

```
streams_per_day =
music_df.withColumn("stream_date",F.to_date("streaming_time")).groupBy("s
tream_date").agg(F.count("*").alias("total_streams"))

streams_per_day.show()
```

 # 8. Identify Users Who Streamed Songs from More Than One Artist

```
more_than_oneArtist =
music_df.groupBy("user_id").agg(F.countDistinct("artist").alias("artist_count"))
.filter(col("artist_count")>1)

more_than_oneArtist.show()
```

 # 9. Calculate the Total Streams for Each Location

```
streams_per_location =
music_df.groupBy("location").agg(F.count("*").alias("total_streams"))

streams_per_location.show()
```

 # 10. Create a New Column to Classify Long and Short Songs

```
music_df = music_df.withColumn("song_length",
F.when(col("duration_seconds") > 200, "Long").otherwise("Short"))

music_df.show()
```