6_dataframe_quiz_solution

March 28, 2022

1 Answer Key to the Data Wrangling with DataFrames Coding Quiz

Helpful resources: https://spark.apache.org/docs/2.4.0/api/python/pyspark.sql.html

In [1]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import isnan, count, when, col, desc, udf, col, sort_array, a
from pyspark.sql.functions import sum as Fsum
from pyspark.sql.window import Window
from pyspark.sql.types import IntegerType
```

In [2]:
```
# 1) import any other libraries you might need
# 2) instantiate a Spark session
# 3) read in the data set located at the path "data/sparkify_log_small.json"
# 4) write code to answer the quiz questions

spark = SparkSession \
    .builder \
    .appName("Data Frames practice") \
    .getOrCreate()

df = spark.read.json("data/sparkify_log_small.json")
```

2 Question 1

Which page did user id "" (empty string) NOT visit?

In [3]:
```
df.printSchema()
```

```
root
 |-- artist: string (nullable = true)
 |-- auth: string (nullable = true)
 |-- firstName: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- itemInSession: long (nullable = true)
 |-- lastName: string (nullable = true)
 |-- length: double (nullable = true)
 |-- level: string (nullable = true)
```

```
 |-- location: string (nullable = true)
 |-- method: string (nullable = true)
 |-- page: string (nullable = true)
 |-- registration: long (nullable = true)
 |-- sessionId: long (nullable = true)
 |-- song: string (nullable = true)
 |-- status: long (nullable = true)
 |-- ts: long (nullable = true)
 |-- userAgent: string (nullable = true)
 |-- userId: string (nullable = true)
```

```
In [ ]: # filter for users with blank user id
        blank_pages = df.filter(df.userId == '') \
            .select(col('page') \
            .alias('blank_pages')) \
            .dropDuplicates()

        # get a list of possible pages that could be visited
        all_pages = df.select('page').dropDuplicates()

        # find values in all_pages that are not in blank_pages
        # these are the pages that the blank user did not go to
        for row in set(all_pages.collect()) - set(blank_pages.collect()):
            print(row.page)
```

3 Question 2 - Reflect

What type of user does the empty string user id most likely refer to?

Perhaps it represents users who have not signed up yet or who are signed out and are about to log in.

4 Question 3

How many female users do we have in the data set?

```
In [6]: df.filter(df.gender == 'F') \
            .select('userId', 'gender') \
            .dropDuplicates() \
            .count()
```

```
Out[6]: 462
```

5 Question 4

How many songs were played from the most played artist?

```
In [7]: df.filter(df.page == 'NextSong') \
             .select('Artist') \
             .groupBy('Artist') \
             .agg({'Artist':'count'}) \
             .withColumnRenamed('count(Artist)', 'Artistcount') \
             .sort(desc('Artistcount')) \
             .show(1)
```

```
+--------+-----------+
|  Artist|Artistcount|
+--------+-----------+
|Coldplay|         83|
+--------+-----------+
only showing top 1 row
```

6 Question 5 (challenge)

How many songs do users listen to on average between visiting our home page? Please round your answer to the closest integer.

```
In [8]: # TODO: filter out 0 sum and max sum to get more exact answer

        function = udf(lambda ishome : int(ishome == 'Home'), IntegerType())

        user_window = Window \
            .partitionBy('userID') \
            .orderBy(desc('ts')) \
            .rangeBetween(Window.unboundedPreceding, 0)

        cusum = df.filter((df.page == 'NextSong') | (df.page == 'Home')) \
            .select('userID', 'page', 'ts') \
            .withColumn('homevisit', function(col('page'))) \
            .withColumn('period', Fsum('homevisit').over(user_window))

        cusum.filter((cusum.page == 'NextSong')) \
            .groupBy('userID', 'period') \
            .agg({'period':'count'}) \
            .agg({'count(period)':'avg'}).show()
```

```
+------------------+
|avg(count(period))|
+------------------+
| 6.898347107438017|
+------------------+
```

In [ ]: