Connect to S3 bucket and read data for PET PRODUCTS REVIEWS in Amazon (data is separated by Tabs compressed as GZ file)

```
%pyspark
from pyspark import SparkFiles
# Load in user_data.csv from S3 into a DataFrame
url = "https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Pet_Products_v1_00.tsv.gz"
spark.sparkContext.addFile(url)

df = spark.read.option('header', 'true').csv(SparkFiles.get("amazon_reviews_us_Pet_Products_v1_00.tsv.gz"), inferSchema=True, sep='\t', timestampFormat="mm/dd/yy")

df.show(10)
```

```
+----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+------------+----------+-----+-----------------+--------------------+--------------------+-----------+
|marketplace|customer_id|     review_id|product_id|product_parent|       product_title|product_category|star_rating|helpful_votes|total_votes|vine|verified_purchase|     review_headline|         review_body|review_date|
+----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+------------+----------+-----+-----------------+--------------------+--------------------+-----------+
|        US|   28794885| REAKC26P07MDN|B00Q0K9604|     510387886|(8-Pack) EZwhelp ...|    Pet Products|          5|           0|         0|   N|                Y|A great purchase ...|Best belly bands ...|2015-08-31 00:00:00|
|        US|   11488901|R3NU7OMZ4HQIEG|B00MBW5O9W|     912374672|Warren Eckstein's...|    Pet Products|          2|           0|         1|   N|                Y|My dogs love Hugs...|My dogs love Hugs...|2015-08-31 00:00:00|
|        US|   43214993|R14QJW3XF8QO1P|B0084OHUIO|     902215727|Tyson's True Chew...|    Pet Products|          5|           0|         0|   N|                Y|I have been purch...|I have been purch...|2015-08-31 00:00:00|
|        US|   12835065|R2HB7AX0394ZGY|B001GS71K2|     568880110|Soft Side Pet Cra...|    Pet Products|          5|           0|         0|   N|                Y|it is easy to ope...|It is extremely w...|2015-08-31 00:00:00|
|        US|   26334022| RGKMPDQGSAHR3|B004ABH1LG|     692846826|EliteField 3-Door...|    Pet Products|          5|           0|         0|   N|                Y|           Dog crate|Worked really wel...|2015-08-31 00:00:00|
|        US|   22283621|R1DJCVPQGCV66E|B00AX0LFM4|     590674141|Carlson 68-Inch W...|    Pet Products|          5|           0|         0|   N|                Y|         Five Stars|I love my gates! ...|2015-08-31 00:00:00|
|        US|   14469895|R3V52EAWLPBFQG|B00DQFZGZ0|     688538603|Dog Seat Cover Wi...|    Pet Products|          3|           0|         0|   N|                Y|Seat belt tugs on...|Didn't quite work...|2015-08-31 00:00:00|
|        US|   50896354|R3DKO8J1J28QBI|B00DIRF9US|     742358789|The Bird Catcher ...|    Pet Products|          2|           0|         0|   N|                Y|Great Pole, but S...|I had the origina...|2015-08-31 00:00:00|
|        US|   18440567| R764DBXGRNECG|B00JRCBFUG|     869798483|Cat Bed - Purrfec...|    Pet Products|          5|           1|         1|   N|                N|     My cat loves it|The pad is very s...|2015-08-31 00:00:00|
|        US|   50502362| RW1853GAT0Z9F|B000L3XYZ4|     501118658|PetSafe Drinkwell...|    Pet Products|          5|           0|         0|   N|                Y|         Five Stars|My cat drinks mor...|2015-08-31 00:00:00|
+----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+------------+----------+-----+-----------------+--------------------+--------------------+-----------+
only showing top 10 rows
```

Interpreter: spark.pyspark. FINISHED Took 1 min 43 sec 76 millisec. Updated by Francisco on August 02 2019, 2:03:50 PM (PDT)

Count the number of records read from the data source

```
%pyspark
# Rows
df.count()
```

```
2643619
```

Interpreter: spark.pyspark. FINISHED Took 15 sec 233 millisec. Updated by Francisco on August 02 2019, 2:04:05 PM (PDT)

Count the number of columns available in the data source

```
%pyspark
# Columns
len(df.columns)
```

```
15
```

Interpreter: spark.pyspark. FINISHED Took 160 millisec. Updated by Francisco on August 02 2019, 2:04:06 PM (PDT)

Select PRODUCT ID and PRODUCT TITLE columns for the products table

```
%pyspark
products = df.select(["product_id", "product_title"])
products.show(5)
```

```
+----------+--------------------+
|product_id|       product_title|
+----------+--------------------+
|B00Q0K9604|(8-Pack) EZwhelp ...|
|B00MBW5O9W|Warren Eckstein's...|
|B0084OHUIO|Tyson's True Chew...|
|B001GS71K2|Soft Side Pet Cra...|
|B004ABH1LG|EliteField 3-Door...|
+----------+--------------------+
only showing top 5 rows
```

Interpreter: spark.pyspark. FINISHED Took 176 millisec. Updated by Francisco on August 02 2019, 2:04:06 PM (PDT)

Eliminate duplicated records. A product can be in multiple reviews, we only need 1 record per PRODUCT ID

```
%pyspark
print(products.count())
products = products.dropDuplicates(["product_id"])
print(products.count())
```

```
2643619
239341
```

Interpreter: spark.pyspark. FINISHED Took 31 sec 918 millisec. Updated by Francisco on August 02 2019, 2:04:38 PM (PDT)

Identify the number of reviews by CUSTOMER. Group the records by CUSTOMER ID and count the number of records

```
%pyspark
customers = df.groupby("customer_id").agg({"customer_id":"count"})
customers.show()
```

```
+----------+------------------+
|customer_id|count(customer_id)|
+----------+------------------+
|   10270641|                 1|
|   18365872|                 1|
|   16711087|                 1|
|   10742726|                 2|
|   41169638|                 1|
|   43622307|                 1|
|   24540309|                 2|
|   28258386|                 1|
|   35329257|                 2|
|   14552054|                 1|
|   14529507|                 5|
|   45392827|                 5|
|   47282953|                 1|
|    8201930|                 1|
|   20109760|                 2|
|   16405801|                 4|
|   15056685|                21|
|   20840575|                 2|
|   39048303|                 1|
|    5596610|                 1|
+----------+------------------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 17 sec 337 millisec. Updated by Francisco on August 02 2019, 2:04:55 PM (PDT)

Sort the records grouped by count and rename the column to CUSTOMER_COUNT as defined in the DB schema

```
%pyspark
from pyspark.sql.functions import desc
customers = customers.withColumnRenamed("count(customer_id)", "customer_count")
customers.orderBy(desc("customer_count")).show()
```

```
+----------+--------------+
|customer_id|customer_count|
+----------+--------------+
|   22500722|           258|
|   31511151|           236|
|   16053845|           228|
|   25659082|           210|
|   23375624|           200|
|   43856165|           198|
|   51175061|           163|
|   37446839|           160|
|   16540594|           150|
|   52263513|           134|
|   16916667|           134|
|   33315159|           134|
|   18539854|           132|
|   18164293|           132|
|   29676361|           130|
|   39154578|           121|
|   16052325|           120|
|   38352624|           119|
|   42329785|           115|
|   50991253|           115|
+----------+--------------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 17 sec 287 millisec. Updated by Francisco on August 02 2019, 2:05:12 PM (PDT)

Select columns for table reviews

```
%pyspark
review_id_table = df.select(["review_id", "customer_id", "product_id", "product_parent", "review_date"])
review_id_table.show(5)
```

```
+--------------+-----------+----------+--------------+-------------------+
|     review_id|customer_id|product_id|product_parent|        review_date|
+--------------+-----------+----------+--------------+-------------------+
| REAKC26P07MDN|   28794885|B00Q0K9604|     510387886|2015-08-31 00:00:00|
|R3NU7OMZ4HQIEG|   11488901|B00MBW5O9W|     912374672|2015-08-31 00:00:00|
|R14QJW3XF8QO1P|   43214993|B0084OHUIO|     902215727|2015-08-31 00:00:00|
|R2HB7AX0394ZGY|   12835065|B001GS71K2|     568880110|2015-08-31 00:00:00|
| RGKMPDQGSAHR3|   26334022|B004ABH1LG|     692846826|2015-08-31 00:00:00|
+--------------+-----------+----------+--------------+-------------------+
only showing top 5 rows
```

Interpreter: spark.pyspark. FINISHED Took 210 millisec. Updated by Francisco on August 02 2019, 2:05:12 PM (PDT)

Extract the DATE from the timestamp column as requested in the table schema

```
%pyspark
from pyspark.sql.types import DateType

review_id_table = review_id_table.withColumn("review_date",review_id_table['review_date'].cast(DateType()))
#below is the result
review_id_table.show()
```

```
+--------------+-----------+----------+--------------+-----------+
|     review_id|customer_id|product_id|product_parent|review_date|
+--------------+-----------+----------+--------------+-----------+
| REAKC26P07MDN|   28794885|B00Q0K9604|     510387886| 2015-08-31|
|R3NU7OMZ4HQIEG|   11488901|B00MBW5O9W|     912374672| 2015-08-31|
|R14QJW3XF8QO1P|   43214993|B0084OHUIO|     902215727| 2015-08-31|
|R2HB7AX0394ZGY|   12835065|B001GS71K2|     568880110| 2015-08-31|
| RGKMPDQGSAHR3|   26334022|B004ABH1LG|     692846826| 2015-08-31|
|R1DJCVPQGCV66E|   22283621|B00AX0LFM4|     590674141| 2015-08-31|
|R3V52EAWLPBFQG|   14469895|B00DQFZGZ0|     688538603| 2015-08-31|
|R3DKO8J1J28QBI|   50896354|B00DIRF9US|     742358789| 2015-08-31|
| R764DBXGRNECG|   18440567|B00JRCBFUG|     869798483| 2015-08-31|
| RW1853GAT0Z9F|   50502362|B000L3XYZ4|     501118658| 2015-08-31|
|R33GITXNUF1AD4|   33930128|B00BOEXWFG|     454737777| 2015-08-31|
|R1H7AVM81TAYRV|   43534290|B001HBBQKY|     420905252| 2015-08-31|
|R2ZOYAQZNNZZWV|   45555864|B007O1FHB0|     302588963| 2015-08-31|
|R2FN1H3CGW6J8H|   11147406|B001P3NU30|     525778264| 2015-08-31|
| RJB41Q575XNG4|    6495678|B00ZP6HS6S|     414117299| 2015-08-31|
|R28W8BM1587CPF|    2019416|B00IP05CUA|     833937853| 2015-08-31|
|R1II0M01NIG293|   40459386|B001U8Y598|      85343577| 2015-08-31|
| RMB8N0DBRH34O|   23126800|B011AY4JWO|     499241195| 2015-08-31|
|R24WB6A6WVIPU6|   30238476|B00DDSHE5A|     409532388| 2015-08-31|
| ROCJSH0P9YSRW|   35113999|B00PJW5OR8|     259271919| 2015-08-31|
+--------------+-----------+----------+--------------+-----------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 162 millisec. Updated by Francisco on August 02 2019, 2:05:13 PM (PDT)

Select columns for the Vine Table

```
%pyspark
vine_table = df.select(["review_id", "star_rating", "helpful_votes", "total_votes", "vine"])
vine_table.show(5)
```

```
+--------------+-----------+-------------+-----------+----+
|     review_id|star_rating|helpful_votes|total_votes|vine|
+--------------+-----------+-------------+-----------+----+
| REAKC26P07MDN|          5|            0|          0|   N|
|R3NU7OMZ4HQIEG|          2|            0|          1|   N|
|R14QJW3XF8QO1P|          5|            0|          0|   N|
|R2HB7AX0394ZGY|          5|            0|          0|   N|
| RGKMPDQGSAHR3|          5|            0|          0|   N|
+--------------+-----------+-------------+-----------+----+
only showing top 5 rows
```

Interpreter: spark.pyspark. FINISHED Took 161 millisec. Updated by Francisco on August 02 2019, 2:05:13 PM (PDT)

Configure the connection to the DB server hosted in AWS

```
%pyspark
# Configuration for RDS instance
mode="overwrite"
jdbc_url = "jdbc:postgresql://<Francisco>:5432/my_data_class_db"
config = {"user":"root",
          "password": "******",
          "driver":"org.postgresql.Driver"}
```

Interpreter: spark.pyspark. FINISHED Took 109 millisec. Updated by Francisco on August 02 2019, 2:05:13 PM (PDT)

Write the records to the CUSTOMERS table

```
%pyspark
# Write DataFrame to table

customers.write.jdbc(url=jdbc_url, table='customers', mode=mode, properties=config)
```

Interpreter: spark.pyspark. FINISHED Took 2 min 27 sec 832 millisec. Updated by Francisco on August 02 2019, 2:07:41 PM (PDT)

Write the records to the PRODUCTS table

```
%pyspark
# Write DataFrame to table

products.write.jdbc(url=jdbc_url, table='products', mode=mode, properties=config)
```

Interpreter: spark.pyspark. FINISHED Took 1 min 5 sec 560 millisec. Updated by Francisco on August 02 2019, 2:08:46 PM (PDT)

Write the records to the REVIEW_ID_TABLE table

```
%pyspark
# Write DataFrame to table

review_id_table.write.jdbc(url=jdbc_url, table='review_id_table', mode=mode, properties=config)
```

Interpreter: spark.pyspark. FINISHED Took 13 min 38 sec 221 millisec. Updated by Francisco on August 02 2019, 2:22:24 PM (PDT)

Write the records to the VINE_TABLE table

```
%pyspark
# Write DataFrame to table

products.write.jdbc(url=jdbc_url, table='vine_table', mode=mode, properties=config)
```

Interpreter: spark.pyspark. FINISHED Took 1 min 4 sec 558 millisec. Updated by Francisco on August 02 2019, 2:23:29 PM (PDT)

```

```

Interpreter: spark.