Connect to S3 bucket and read data for MAJOR APPLIANCES 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_Major_Appliances_v1_00.tsv.gz"
spark.sparkContext.addFile(url)

df = spark.read.option('header', 'true').csv(SparkFiles.get("amazon_reviews_us_Major_Appliances_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|   16199106|R203HPW78Z7N4K|B0067WNSZY|     633038551|FGGF3032MW Galler...|Major Appliances|          5|           0|          0|   N|               Y|If you need a new...|What a great stov...|2015-08-31 00:00:00|
|        US|   16374060|R2EAIGVLEALSP3|B002QSXK60|     811766671|Best Hand Clothes...|Major Appliances|          5|           1|          1|   N|               Y|          Five Stars|       worked great|2015-08-31 00:00:00|
|        US|   15322085|R1K1CD73HHLILA|B00EC452R6|     345562728|Supco SET184 Ther...|Major Appliances|          5|           0|          0|   N|               Y|      Fast Shipping|Part exactly what...|2015-08-31 00:00:00|
|        US|   32004835|R2KZBMOFRMYOPO|B00MVVIF2G|     563052763|Midea WHS-160RB1 ...|Major Appliances|          5|           1|          1|   N|               Y|          Five Stars|Love my refrigera...|2015-08-31 00:00:00|
|        US|   25414497| R6BIZOZY6UD01|B00IY7BNUW|     874236579|Avalon Bay Portab...|Major Appliances|          5|           0|          0|   N|               Y|          Five Stars|No more running t...|2015-08-31 00:00:00|
|        US|   36311751|R1MCXZFNF8E7Y0|B0033X29CI|     294467812|Danby  Freestandi...|Major Appliances|          1|           0|          0|   N|               Y|       Piece of Junk|It would not cool...|2015-08-31 00:00:00|
|        US|   30920961|R3EMB3E3ODR6BW|B005R597HA|     183784715|Avanti 110-Volt A...|Major Appliances|          5|           2|          2|   N|               Y|Works awesome for...|Works awesome for...|2015-08-31 00:00:00|
|        US|   52491265| RJTONVTTOPJ5S|B00MO6V8Y0|     960251524|     Danby products|Major Appliances|          5|           0|          0|   N|               Y|          Five Stars|exactly what I wa...|2015-08-31 00:00:00|
|        US|   48166169|R21U5QZ2CQECUM|B00HT39QDI|     992475314|3 Pack Tier1 MSWF...|Major Appliances|          4|           0|          0|   N|               Y|          Four Stars|       AS advertised|2015-08-31 00:00:00|
|        US|   50394924| RL2BBC51H89DH|B00LESFZ52|       1641606|True TSSU-60-16 6...|Major Appliances|          4|           0|          0|   N|               Y|but has poor insu...|It works as adver...|2015-08-31 00:00:00|
+----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+------------+-----------+----+----------------+--------------------+--------------------+--------------------+
only showing top 10 rows
```

Interpreter: spark.pyspark. FINISHED Took 34 sec 454 millisec. Updated by Francisco on August 02 2019, 2:28:03 PM (PDT) (outdated)

Count the number of records read from the data source

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

```
96901
```

Interpreter: spark.pyspark. FINISHED Took 1 sec 212 millisec. Updated by Francisco on August 02 2019, 2:28:04 PM (PDT)

Count the number of columns available in the data source

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

```
15
```

Interpreter: spark.pyspark. FINISHED Took 159 millisec. Updated by Francisco on August 02 2019, 2:28:04 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|
+----------+--------------------+
|B0067WNSZY|FGGF3032MW Galler...|
|B002QSXK60|Best Hand Clothes...|
|B00EC452R6|Supco SET184 Ther...|
|B00MVVIF2G|Midea WHS-160RB1 ...|
|B00IY7BNUW|Avalon Bay Portab...|
+----------+--------------------+
only showing top 5 rows
```

Interpreter: spark.pyspark. FINISHED Took 174 millisec. Updated by Francisco on August 02 2019, 2:28:05 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())
```

```
96901
11694
```

Interpreter: spark.pyspark. FINISHED Took 2 sec 515 millisec. Updated by Francisco on August 02 2019, 2:28:07 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)|
+----------+------------------+
|   13326061|                 1|
|   52512151|                 1|
|    8968933|                 1|
|   39416583|                 1|
|   21737233|                 1|
|   44119972|                 1|
|   13947800|                 1|
|   21482558|                 1|
|    2802853|                 1|
|   18518845|                 1|
|     122484|                 1|
|   14035069|                 1|
|   20004191|                 1|
|   16699322|                 1|
|   24768141|                 1|
|   24424556|                 1|
|   44293588|                 1|
|   13188682|                 1|
|   23298840|                 1|
|   50731398|                 1|
+----------+------------------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 1 sec 312 millisec. Updated by Francisco on August 02 2019, 2:28:08 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|
+----------+--------------+
|   32962808|            16|
|   15489471|            15|
|    5695524|            15|
|   30396577|            14|
|   34553562|            13|
|   29748841|            12|
|   51139148|            12|
|   32230187|            12|
|   51042275|            12|
|   30544694|            11|
|   13649055|            11|
|   40382895|            11|
|   20862753|            11|
|   12914327|            11|
|   48159861|            11|
|   45671123|             9|
|   49444310|             9|
|   24715941|             9|
|   14384334|             9|
|   53074513|             9|
+----------+--------------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 1 sec 312 millisec. Updated by Francisco on August 02 2019, 2:28:10 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|
+--------------+-----------+----------+--------------+-------------------+
|R203HPW78Z7N4K|   16199106|B0067WNSZY|     633038551|2015-08-31 00:00:00|
|R2EAIGVLEALSP3|   16374060|B002QSXK60|     811766671|2015-08-31 00:00:00|
|R1K1CD73HHLILA|   15322085|B00EC452R6|     345562728|2015-08-31 00:00:00|
|R2KZBMOFRMYOPO|   32004835|B00MVVIF2G|     563052763|2015-08-31 00:00:00|
| R6BIZOZY6UD01|   25414497|B00IY7BNUW|     874236579|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, 2:28:10 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|
+--------------+-----------+----------+--------------+-----------+
|R203HPW78Z7N4K|   16199106|B0067WNSZY|     633038551| 2015-08-31|
|R2EAIGVLEALSP3|   16374060|B002QSXK60|     811766671| 2015-08-31|
|R1K1CD73HHLILA|   15322085|B00EC452R6|     345562728| 2015-08-31|
|R2KZBMOFRMYOPO|   32004835|B00MVVIF2G|     563052763| 2015-08-31|
| R6BIZOZY6UD01|   25414497|B00IY7BNUW|     874236579| 2015-08-31|
|R1MCXZFNF8E7Y0|   36311751|B0033X29CI|     294467812| 2015-08-31|
|R3EMB3E3ODR6BW|   30920961|B005R597HA|     183784715| 2015-08-31|
| RJTONVTTOPJ5S|   52491265|B00MO6V8Y0|     960251524| 2015-08-31|
|R21U5QZ2CQECUM|   48166169|B00HT39QDI|     992475314| 2015-08-31|
| RL2BBC51H89DH|   50394924|B00LESFZ52|       1641606| 2015-08-31|
|R3RNEPHF3WIRSZ|    3915552|B0149IJVPI|     838108342| 2015-08-31|
|R38DNT9KML2PF3|   17068589|B002HT0958|     387104338| 2015-08-31|
|R2ECMBJM8KNNC8|   52081068|B006WOBNX6|     180688127| 2015-08-31|
|R2F3F92PRN9T7S|   13783713|B00NLPMOU0|     570132358| 2015-08-31|
|R35XNT88XVEMK6|   14555366|B00KJ07SEM|     313983847| 2015-08-31|
| RZRXLHQS71FUP|   48417244|B001AT2ALM|     364403440| 2015-08-31|
|R3J9T4MZRAURV5|   19010760|B00JG8B42K|     672722354| 2015-08-31|
|R3CLDHHI0ZSQLO|   43532555|B00FRJ5EWS|     137548022| 2015-08-31|
| RYS2KZABZK8QF|   21879631|B0052G14E8|     423421857| 2015-08-31|
|R25RXV5X29M3UC|     810098|B00LOVVE9A|     386722389| 2015-08-31|
+--------------+-----------+----------+--------------+-----------+
only showing top 20 rows
```

Interpreter: spark.pyspark. FINISHED Took 211 millisec. Updated by Francisco on August 02 2019, 2:28:10 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|
+--------------+-----------+------------+-----------+----+
|R203HPW78Z7N4K|          5|           0|          0|   N|
|R2EAIGVLEALSP3|          5|           1|          1|   N|
|R1K1CD73HHLILA|          5|           0|          0|   N|
|R2KZBMOFRMYOPO|          5|           1|          1|   N|
| R6BIZOZY6UD01|          5|           0|          0|   N|
+--------------+-----------+------------+-----------+----+
only showing top 5 rows
```

Interpreter: spark.pyspark. FINISHED Took 161 millisec. Updated by Francisco on August 02 2019, 2:28:10 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, 2:28:10 PM (PDT) (outdated)

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 34 sec 312 millisec. Updated by Francisco on August 02 2019, 2:28:45 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 27 sec 752 millisec. Updated by Francisco on August 02 2019, 2:29:12 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 31 sec 755 millisec. Updated by Francisco on August 02 2019, 2:29:44 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 27 sec 199 millisec. Updated by Francisco on August 02 2019, 2:30:11 PM (PDT)

```

```

Interpreter: spark.