

## ▼ Amazon Reviews ETL

By Emmanuel Martinez

```
import os
# Find the latest version of spark 2.0 from http://www-us.apache.org/dist/spark/ and enter as
# For example:
# spark_version = 'spark-2.4.7'
spark_version = 'spark-2.4.7'
os.environ['SPARK_VERSION']=spark_version

# Install Spark and Java
!apt-get update
!apt-get install openjdk-11-jdk-headless -qq > /dev/null
!wget -q http://www-us.apache.org/dist/spark/\$SPARK\_VERSION/\$SPARK\_VERSION-bin-hadoop2.7.tgz
!tar xf $SPARK_VERSION-bin-hadoop2.7.tgz
!pip install -q findspark

# Set Environment Variables ",
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["SPARK_HOME"] = f"/content/{spark_version}-bin-hadoop2.7"

# Start a SparkSession
import findspark
findspark.init()

Hit:1 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:6 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
Hit:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:8 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Hit:9 http://security.ubuntu.com/ubuntu bionic-security InRelease
Ign:10 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64 InRe
Ign:11 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/
Hit:12 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64 Rele
Hit:13 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/
Reading package lists... Done
```



```
# Download the Postgres driver that will allow Spark to interact with Postgres.
!wget https://jdbc.postgresql.org/download/postgresql-42.2.16.jar

--2021-03-11 15:02:39-- https://jdbc.postgresql.org/download/postgresql-42.2.16.jar
Resolving jdbc.postgresql.org (jdbc.postgresql.org)... 72.32.157.228, 2001:4800:3e1:1::
```

```
Connecting to jdbc.postgresql.org (jdbc.postgresql.org)|72.32.157.228|:443... connected
HTTP request sent, awaiting response... 200 OK
Length: 1002883 (979K) [application/java-archive]
Saving to: 'postgresql-42.2.16.jar'
```

```
postgresql-42.2.16. 100%[=====>] 979.38K 1.64MB/s in 0.6s
```

```
2021-03-11 15:02:40 (1.64 MB/s) - 'postgresql-42.2.16.jar' saved [1002883/1002883]
```



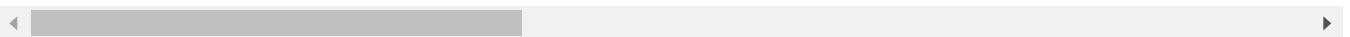
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("BigData-Challenge").config("spark.driver.extraClassPath",
```

## Load Amazon Data into Spark DataFrame

```
from pyspark import SparkFiles
url = "https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Furniture_v1_00.tsv"
spark.sparkContext.addFile(url)
df = spark.read.option("encoding", "UTF-8").csv(SparkFiles.get("amazon_reviews_us_Furniture_v1_00.tsv"))
df.show()
```

marketplace	customer_id	review_id	product_id	product_parent	product_title
US	24509695	R3VR960AHLFKDV	B004HB5E0E	488241329	Shoal Creek Compu...
US	34731776	R16LGVMFKIUT0G	B0042TNMMS	205864445	Dorel Home Produc...
US	1272331	R1AIMEEPPYHMOE4	B0030MPBZ4	124663823	Bathroom Vanity T...
US	45284262	R1892CCSZWZ9SR	B005G02ESA	382367578	Sleep Master Ulti...
US	30003523	R285P679YWVKD1	B005JS8AUA	309497463	1 1/4" GashGuards...
US	18311821	RLB33HJBXHZHU	B00AVUQQGQ	574537906	Serta Bonded Leat...
US	42943632	R1VGTZ94DBAD6A	B00CFY20GQ	407473883	Prepac Shoe Stora...
US	43157304	R168KF82ICSOHD	B00FKC48QA	435120460	HomCom PU Leather...
US	51918480	R20DIYIJ0OCMOG	B00N9IAL9K	356495985	Folding Step Stool
US	14522766	RD46RNV0HNZSC	B001T4XU1C	243050228	Ace Bayou Adult V...
US	43054112	R2JD0CETTM3AXS	B002HRFLBC	93574483	4D Concepts Audio...
US	26622950	R33YMW36IDZ6LE	B006MISZOC	941823468	Zinus SC-SBBK-14N...
US	17988940	R30ZGGUHZ04C1S	B008BMGABC	460567746	Poundex Marble Di...
US	18444952	RS2EZU76IK2BT	B00CO2VH5Y	829613894	Safavieh Lyndhurs...
US	16937084	R1GJC1BP028X09	B00LI4RJQ0	816478187	Sauder Boone Moun...
US	23665632	R2VKJPGXXEK5GP	B0046EC1D0	358594389	Winsome Wood Brea...
US	4110125	R17KS83G3KLT97	B00DQQPL36	312571325	HODEDAH IMPORT Me...
US	107621	R3PQL8SR4NEHWL	B003X7RWB2	402665054	Flash Furniture H...
US	2415090	R2F5WW7WNO5RRG	B001TJYPJ8	854989315	Sleep Revolution...
US	48285966	R3UDJKVWQCFIC9	B000TMHX9A	814079288	Flash Furniture V...

only showing top 20 rows



## Create DataFrames to match tables

```

from pyspark.sql.functions import to_date
# Read in the Review dataset as a DataFrame

# df is already a DataFrame, no need to re-do. Skipping current step.

# Create the customers_table DataFrame
customers_df = df.groupby("customer_id").agg({"customer_id": "count"}).withColumnRenamed("customer_id", "customer_count")
customers_df.show()

```

```

+-----+-----+
|customer_id|customer_count|
+-----+-----+
| 17067926|                2|
| 10714827|                1|
| 42560427|                1|
| 30717305|                1|
| 1178966|                1|
| 10429047|                1|
| 41351814|                1|
| 52541790|                2|
| 52512151|                1|
| 37534120|                1|
| 22555935|                1|
| 18681995|                1|
| 2119235|                2|
| 21846356|                1|
| 42251639|                1|
| 7730812|                1|
| 37666248|                1|
| 43676452|                1|
| 41466760|                1|
| 30403003|                1|
+-----+-----+
only showing top 20 rows

```

```

# Create the products_table DataFrame and drop duplicates.
products_df = df.select(['product_id', 'product_title']).drop_duplicates()
products_df.show()

```

```

+-----+-----+
|product_id|product_title|
+-----+-----+
|B0049H810M|Fun Rugs Surf Tim...|
|B001U0U006|Furniture Repair Set|
|B0076I51JE|Traditional Opera...|
|B0070WPBGU|Frenchi Home Furn...|
|B00DHWGB4M|Circle Design Sid...|
|B00RI6TNJ8|Abrahami Hariz Bu...|
|B00GHZA29Q|nuLOOM Varanas Co...|
|B00GSIIGQS|Milton Greens Sta...|
|B00VZ4RY1I|Dean Shifting San...|
|B007QUM5DM| Charles Petite Sofa|
|B00MN6NTD0|Safavieh Adironda...|

```

```
|B00BK31LDQ|Glass Computer De...|
|B0091SXURW|Altra Parsons Des...|
|B00TBVK0Y0|Best Price Mattre...|
|B00A2XM5QC|Legacy Decor Shoj...|
|B002KE7HTQ|Home Styles 5001-...|
|B00V3LVD20|Roundhill Furnitu...|
|B00PN9YSAG|Baxton Studio Hir...|
|B005VAFFN6|Duro Hanley Silve...|
|B001BX1JSC|Flash Furniture V...|
```

```
+-----+
only showing top 20 rows
```

```
# Create the review_id_table DataFrame.
```

```
# Convert the 'review_date' column to a date datatype with to_date("review_date", 'yyyy-MM-dd')
review_id_df = df.select(['review_id', 'customer_id', 'product_id', 'product_parent', 'review_date'])
review_id_df.show()
```

```
+-----+-----+-----+-----+-----+
| review_id|customer_id|product_id|product_parent|review_date|
+-----+-----+-----+-----+-----+
|R3VR960AHLFKDV|24509695|B004HB5E0E|488241329|2015-08-31|
|R16LGVMFKIUT0G|34731776|B0042TNMMS|205864445|2015-08-31|
|R1AIMEEPPYHMOE4|1272331|B0030MPBZ4|124663823|2015-08-31|
|R1892CCSZWZ9SR|45284262|B005G02ESA|382367578|2015-08-31|
|R285P679YWVKD1|30003523|B005JS8AUA|309497463|2015-08-31|
|RLB33HJBXHZHU|18311821|B00AVUQQGQ|574537906|2015-08-31|
|R1VGTZ94DBAD6A|42943632|B00CFY20GQ|407473883|2015-08-31|
|R168KF82ICSOHD|43157304|B00FKC48QA|435120460|2015-08-31|
|R20DIYIJ00CMOG|51918480|B00N9IAL9K|356495985|2015-08-31|
|RD46RNV0HNZSC|14522766|B001T4XU1C|243050228|2015-08-31|
|R2JDOCETTM3AXS|43054112|B002HRFLBC|93574483|2015-08-31|
|R33YMW36IDZ6LE|26622950|B006MISZOC|941823468|2015-08-31|
|R30ZGGUHZ04C1S|17988940|B008BMGABC|460567746|2015-08-31|
|RS2EZU76IK2BT|18444952|B00C02VH5Y|829613894|2015-08-31|
|R1GJC1BP028X09|16937084|B00LI4RJQ0|816478187|2015-08-31|
|R2VKJPGXXEK5GP|23665632|B0046EC1D0|358594389|2015-08-31|
|R17KS83G3KLT97|4110125|B00DQQL36|312571325|2015-08-31|
|R3PQL8SR4NEHWL|107621|B003X7RWB2|402665054|2015-08-31|
|R2F5WW7WNO5RRG|2415090|B001TJYPJ8|854989315|2015-08-31|
|R3UDJKVWQCFIC9|48285966|B000TMHX9A|814079288|2015-08-31|
```

```
+-----+
only showing top 20 rows
```

```
# Create the vine_table. DataFrame
```

```
vine_df = df.select(['review_id', 'star_rating', 'helpful_votes', 'total_votes', 'vine', 'verified_purchase'])
vine_df.show()
```

```
+-----+-----+-----+-----+-----+
| review_id|star_rating|helpful_votes|total_votes|vine|verified_purchase|
+-----+-----+-----+-----+-----+
|R3VR960AHLFKDV|4|0|0|N|Y|
|R16LGVMFKIUT0G|5|0|0|N|Y|
```

R1AIMEEPYHMOE4	5	1	1	N	Y
R1892CCSZWZ9SR	3	0	0	N	Y
R285P679YWVKD1	3	0	0	N	N
RLB33HJBXHZHU	5	0	0	N	Y
R1VGTZ94DBAD6A	5	2	2	N	Y
R168KF82ICSOHD	5	0	0	N	Y
R20DIYIJ0OCMOG	5	0	0	N	Y
RD46RNVOHNZSC	5	0	0	N	Y
R2JDOCETTM3AXS	5	0	0	N	Y
R33YMW36IDZ6LE	5	0	0	N	Y
R30ZGGUHZ04C1S	5	1	1	N	Y
RS2EZU76IK2BT	5	0	0	N	Y
R1GJC1BP028X09	5	2	3	N	Y
R2VKJPGXXEK5GP	1	0	0	N	Y
R17KS83G3KLT97	3	0	0	N	Y
R3PQL8SR4NEHWL	4	0	0	N	Y
R2F5WW7WNO5RRG	5	0	0	N	Y
R3UDJKVWQCFIC9	5	0	0	N	Y

+-----+-----+-----+-----+-----+-----+

only showing top 20 rows

## Connect to the AWS RDS instance and write each DataFrame to its table

```
# Configure settings for RDS
mode = "append"
jdbc_url="jdbc:postgresql://mychallenge16.cnoe7lzdcmxmt.us-east-1.rds.amazonaws.com:5432/mycha
config = {"user": "postgres",
          "password": "Acc3ss12",
          "driver": "org.postgresql.Driver"}

# Write review_id_df to table in RDS
review_id_df.write.jdbc(url=jdbc_url, table='review_id_table', mode=mode, properties=config)

# Write products_df to table in RDS
# Average of 3 Minutes.
products_df.write.jdbc(url=jdbc_url, table='products_table', mode=mode, properties=config)

# Write customers_df to table in RDS
# Average of 5 Minutes and 17 Seconds
customers_df.write.jdbc(url=jdbc_url, table='customers_table', mode=mode, properties=config)

# Write vine_df to table in RDS
# Average of 12 Minutes
vine_df.write.jdbc(url=jdbc_url, table='vine_table', mode=mode, properties=config)
```

**Amazon Reviews ETL completion by Emmanuel Martinez**

Double-click (or enter) to edit