```
%pip install pypandoc
%pip install pyspark
```

```
    Requirement already satisfied: pypandoc in /usr/local/lib/python3.7/dist-packages (1.7.5)
    Requirement already satisfied: pyspark in /usr/local/lib/python3.7/dist-packages (3.2.1)
    Requirement already satisfied: py4j==0.10.9.3 in /usr/local/lib/python3.7/dist-packages (from pyspark) (0.10.9.3)
```

```
import os
# Find the latest version of spark 3.0 from http://www.apache.org/dist/spark/ and enter as the spark version
# For example:
# spark_version = 'spark-3.0.3'
spark_version = 'spark-3.0.3'
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.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 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64  InRelease
    Hit:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
    Ign:3 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  InRelease
    Hit:4 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64  Release
    Hit:5 http://archive.ubuntu.com/ubuntu bionic InRelease
    Hit:6 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
    Get:7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
    Get:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
    Hit:9 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
    Get:10 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
    Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
    Hit:13 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
    Fetched 252 kB in 3s (74.8 kB/s)
    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
```

```
    --2022-05-04 02:05:52--  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::228
    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.2’

    postgresql-42.2.16. 100%[===================>] 979.38K  4.92MB/s    in 0.2s

    2022-05-04 02:05:53 (4.92 MB/s) - ‘postgresql-42.2.16.jar.2’ saved [1002883/1002883]
```

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("M16-Amazon-Challenge").config("spark.driver.extraClassPath","/content/postgresql-42.2.
```

Load Amazon Data into Spark DataFrame

```
from pyspark import SparkFiles
url = "https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_us_Video_Games_v1_00.tsv.gz"
spark.sparkContext.addFile(url)
df = spark.read.option("encoding", "UTF-8").csv(SparkFiles.get(""), sep="\t", header=True, inferSchema=True)
df.show()
```

```
    +-----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+---
    |marketplace|customer_id|     review_id|product_id|product_parent|       product_title|product_category|star_rating|hel
    +-----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+---
```

```
|         US|   12039526| RTIS3L2M1F5SM|B001CXYMFS|     737716809|Thrustmaster T-Fl...|     Video Games|          5|
|         US|    9636577| R1ZV7R40OLHKD|B00M920ND6|     569686175|Tonsee 6 buttons ...|     Video Games|          5|
|         US|    2331478|R3BH071QLH8QMC|B0029CSOD2|      98937668|Hidden Mysteries:...|     Video Games|          1|
|         US|   52495923|R127K9NTSXA2YH|B00GOOSV98|      23143350|GelTabz Performan...|     Video Games|          3|
|         US|   14533949|R32ZWUXDJPW27Q|B00Y074JOM|     821342511|Zero Suit Samus a...|     Video Games|          4|
|         US|    2377552|R3AQQ4YUKJWBA6|B002UBI6W6|     328764615|Psyclone Recharge...|     Video Games|          1|
|         US|   17521011|R2F0POU5K6F73F|B008XHCLFO|      24234603|Protection for yo...|     Video Games|          5|
|         US|   19676307|R3VNR804HYSMR6|B00BRA9R6A|     682267517|   Nerf 3DS XL Armor|     Video Games|          5|
|         US|     224068| R3GZTM72WA2QH|B009EPWJLA|     435241890|One Piece: Pirate...|     Video Games|          5|
|         US|   48467989| RNQOY62705W1K|B0000AV7GB|     256572651|Playstation 2 Dan...|     Video Games|          4|
|         US|     106569|R1VTIA3JTYBY02|B00008KTNN|     384411423|Metal Arms: Glitc...|     Video Games|          5|
|         US|   48269642|R29DOU8791QZL8|B000A3IA0Y|     472622859|72 Pin Connector ...|     Video Games|          1|
|         US|   52738710|R15DUT1VIJ9RJZ|B0053BQN34|     577628462|uDraw Gametablet ...|     Video Games|          2|
|         US|   10556786|R3IMF2MQ3OU9ZM|B002I0HIMI|     988218515|NBA 2K12(Covers M...|     Video Games|          4|
|         US|    2963837|R23H79DHOZTYAU|B0081EH12M|     770100932|New Trigger Grips...|     Video Games|          1|
|         US|   23092109| RIV24EQAIXA4O|B005FMLZQQ|      24647669|Xbox 360 Media Re...|     Video Games|          5|
|         US|   23091728|R3UCNGYDVN24YB|B002BSA388|      33706205|Super Mario Galaxy 2|     Video Games|          5|
|         US|   10712640| RUL4H4XTTN2DY|B00BUSLSAC|     829667834|Nintendo 3DS XL -...|     Video Games|          5|
|         US|   17455376|R20JF7Z4DHTNX5|B00KWF38AW|     110680188|Captain Toad:  Tr...|     Video Games|          5|
|         US|   14754850|R2T1AJ5MFI2260|B00BRQJYA8|     616463426|Lego Batman 2: DC...|     Video Games|          4|
+-----------+-----------+--------------+----------+--------------+--------------------+----------------+-----------+---
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.printSchema()
```

```
root
 |-- marketplace: string (nullable = true)
 |-- customer_id: integer (nullable = true)
 |-- review_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- product_parent: integer (nullable = true)
 |-- product_title: string (nullable = true)
 |-- product_category: string (nullable = true)
```

```
     |-- star_rating: integer (nullable = true)
     |-- helpful_votes: integer (nullable = true)
     |-- total_votes: integer (nullable = true)
     |-- vine: string (nullable = true)
     |-- verified_purchase: string (nullable = true)
     |-- review_headline: string (nullable = true)
     |-- review_body: string (nullable = true)
     |-- review_date: string (nullable = true)
```

```
# Create the customers_table DataFrame
customers_df = df.groupby("customer_id").agg({"customer_id":"count"}).withColumnRenamed("count(customer_id)", "customer_coun
customers_df.show(5)
```

```
+-----------+--------------+
|customer_id|customer_count|
+-----------+--------------+
|   48670265|             1|
|   49103216|             2|
|    1131200|             1|
|   43076447|             2|
|   46261368|             1|
+-----------+--------------+
only showing top 5 rows
```

```
# Create the products_table DataFrame and drop duplicates.
products_df = df.select(["product_id", "product_title"]).drop_duplicates()
products_df.show(5)
```

```
+----------+--------------------+
|product_id|       product_title|
+----------+--------------------+
|B00CJ7IUI6|The Elder Scrolls...|
|B00DHF39KS|Wolfenstein: The ...|
|B00MUTAVH6|Under Night In-Bi...|
|B001AZSEUW|              Peggle|
|B00KVOVBGM|PlayStation 4 Con...|
+----------+--------------------+
    only showing top 5 rows
```

```
# Create the review_id_table DataFrame.
review_id_df = df.select(["review_id", "customer_id", "product_id", "product_parent", to_date("review_date", 'yyyy-MM-dd').a
review_id_df.show(5)
```

```
    +--------------+-----------+----------+--------------+-----------+
    |     review_id|customer_id|product_id|product_parent|review_date|
    +--------------+-----------+----------+--------------+-----------+
    | RTIS3L2M1F5SM|   12039526|B001CXYMFS|     737716809| 2015-08-31|
    | R1ZV7R40OLHKD|    9636577|B00M920ND6|     569686175| 2015-08-31|
    |R3BH071QLH8QMC|    2331478|B0029CSOD2|      98937668| 2015-08-31|
    |R127K9NTSXA2YH|   52495923|B00GOOSV98|      23143350| 2015-08-31|
    |R32ZWUXDJPW27Q|   14533949|B00Y074JOM|     821342511| 2015-08-31|
    +--------------+-----------+----------+--------------+-----------+
    only showing top 5 rows
```

```
# Create the vine_table. DataFrame
vine_df = df.select(["review_id", "star_rating", "helpful_votes", "total_votes", "vine", "verified_purchase"])
vine_df.show(5)
```

```
    +--------------+-----------+-------------+-----------+----+-----------------+
    |     review_id|star_rating|helpful_votes|total_votes|vine|verified_purchase|
    +--------------+-----------+-------------+-----------+----+-----------------+
    | RTIS3L2M1F5SM|          5|            0|          0|   N|                Y|
    | R1ZV7R40OLHKD|          5|            0|          0|   N|                Y|
    |R3BH071QLH8QMC|          1|            0|          1|   N|                Y|
    |R127K9NTSXA2YH|          3|            0|          0|   N|                Y|
    |R32ZWUXDJPW27Q|          4|            0|          0|   N|                Y|
    +--------------+-----------+-------------+-----------+----+-----------------+
    only showing top 5 rows
```

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

```
# Configure settings for RDS
```

```
mode = "append"
jdbc_url="jdbc:postgresql://dataviz-nealbhatia.cskmr8qoeo9i.us-east-1.rds.amazonaws.com:5432/AmazonVineAnalysis"
config = {"user":"postgres",
          "password": "Module16!",
          "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
# about 3 min
products_df.write.jdbc(url=jdbc_url, table='products_table', mode=mode, properties=config)


# Write customers_df to table in RDS
# 5 min 14 s
customers_df.write.jdbc(url=jdbc_url, table='customers_table', mode=mode, properties=config)


# Write vine_df to table in RDS
# 11 minutes
vine_df.write.jdbc(url=jdbc_url, table='vine_table', mode=mode, properties=config)
```

✓ 8m 28s completed at 9:39 PM ● ×