Order Data analysis (Joins) => 1 to 4 Questions

from pyspark.sql import SparkSession  
from pyspark.sql.functions import col  
  
spark = SparkSession.builder.master(**"local"**).appName(**"orders"**).getOrCreate()  
  
df = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**)\  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**orders.csv"**)  
  
df.show(1)  
  
print(**"What is the daily product revenue for CLOSED or COMPLETE orders? "**)  
spark.sql(**"SELECT order\_status,o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from orders o join "  
 "orderitems ot on o.order\_id = ot.order\_item\_order\_id join products p on "  
 "p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') "  
 "group by order\_status,o.order\_date, p.product\_name"**).show()  
  
  
print(**"Load the required data in to DF like categories,customer,departments,order\_items,orders and products"**)  
categoriesDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**)\  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**categories**\\**part-00000"**)  
categoriesDFHeaders = categoriesDF.toDF(**"category\_id"**,**"serial\_No"**,**"Games"**)  
categoriesDFHeaders.show()  
  
customerDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**customers**\\**part-00000"**)  
customerDFHeaders = customerDF.toDF(**"customer\_id"**,**"firstname"**,**"lastname"**,**"num1"**,**"num2"**,**"address"**,**"city"**,**"citycode"**,**"zipcode"**)  
customerDFHeaders.show()  
  
departmentDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**departments**\\**part-00000"**)  
departmentDFHeader = departmentDF.toDF(**"dept\_id"**,**"departName"**)  
departmentDFHeader.show()  
  
order\_itemsDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**order\_items**\\**part-00000"**)  
order\_itemsDFHeader = order\_itemsDF.toDF(**"order\_id"**,**"one"**,**"two"**,**"three"**,**"four"**,**"fifth"**)  
order\_itemsDFHeader.show()  
  
ordersDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**orders**\\**part-00000"**)  
ordersDFHeaders = ordersDF.toDF(**"id"**,**"timestamp"**,**"amount"**,**"mode"**)  
ordersDFHeaders.show()  
  
productsDF = spark.read.format(**"csv"**).option(**"header"**,**"true"**).option(**"inferSchema"**,**"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**products**\\**part-00000"**)  
productsDFHeaders = productsDF.toDF(**"Prod\_id"**,**"Serial\_No"**,**"Product\_detail"**,**"values"**,**"amount"**,**"link"**)  
productsDFHeaders.show()  
  
print(**"3. Get the count for each order status"**)  
orderCount = df.groupBy(**"order\_status"**).count()  
orderCount.show()  
  
print(**"4. Filter only complete or closed orders"**)  
filterDF = df.filter((col(**"order\_status"**) == **"CLOSED"**) | (col(**"order\_status"**) == **"COMPLETE"**))  
filterDF.show()

Order Data analysis (Joins) => 5 to 6 Questions

from pyspark.sql import SparkSession  
from pyspark.sql.functions import col, sum, date\_format  
  
spark = SparkSession.builder.master(**"local"**).appName(**"orders"**) \  
 .config(**"spark.sql.warehouse.dir"**, **"/user/hive/warehouse/zakeerdb"**).enableHiveSupport().getOrCreate()  
  
*# Sample data loading from CSV files (replace with your actual paths)*productsDF = spark.read.format(**"csv"**).option(**"header"**, **"true"**).option(**"inferSchema"**, **"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**products**\\**part-00000"**)  
order\_itemsDF = spark.read.format(**"csv"**).option(**"header"**, **"true"**).option(**"inferSchema"**, **"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**order\_items**\\**part-00000"**)  
ordersDF = spark.read.format(**"csv"**).option(**"header"**, **"true"**).option(**"inferSchema"**, **"true"**) \  
 .load(**"C:**\\**Users**\\**zakeerm**\\**Downloads**\\**PysparkLearning**\\**resources**\\**retail\_db**\\**orders**\\**part-00000"**)  
  
productsDF.show()  
productsDF.printSchema()  
order\_itemsDF.show()  
order\_itemsDF.printSchema()  
ordersDF.show()  
ordersDF.printSchema()  
  
*# Join order\_itemsDF with productDF on Product\_ID*joined\_df = order\_itemsDF.join(productsDF, order\_itemsDF.Product\_ID == productsDF.ID)  
  
*# Join the result with ordersDF on Order\_ID*final\_df = joined\_df.join(ordersDF, joined\_df.Order\_ID == ordersDF.Id)  
  
*# Convert the Timestamp column to a Date column*final\_df = final\_df.withColumn(**"Date"**, date\_format(col(**"Timestamp"**), **"yyyy-MM-dd"**))  
  
*# Calculate daily product revenue*daily\_revenue\_df = final\_df.groupBy(**"Date"**, **"Product\_Name"**).agg(  
 sum(col(**"Total\_Price"**)).alias(**"Daily\_Product\_Revenue"**)  
)  
  
*# Show the resulting DataFrame*daily\_revenue\_df.show()  
  
daily\_revenue\_df.write.partitionBy(**"Product\_Name"**).mode(**"overwrite"**).saveAsTable(**"revenue"**)

C:\Users\zakeerm\IdeaProjects\Spark-Scala\venv\Scripts\python.exe C:/Users/zakeerm/Downloads/PysparkLearning/pysparkprograms/order\_5\_Question.py

23/11/11 22:08:47 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP\_HOME and hadoop.home.dir are unset. -see https://wiki.apache.org/hadoop/WindowsProblems

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

23/11/11 22:08:47 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

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

| ID|Quantities| Product\_Name|Price|Image\_URL|

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

| 1| 2|Quest Q64 10 FT. ...| NULL| 59.98|

| 2| 2|Under Armour Men'...| NULL| 129.99|

| 3| 2|Under Armour Men'...| NULL| 89.99|

| 4| 2|Under Armour Men'...| NULL| 89.99|

| 5| 2|Riddell Youth Rev...| NULL| 199.99|

| 6| 2|Jordan Men's VI R...| NULL| 134.99|

| 7| 2|Schutt Youth Recr...| NULL| 99.99|

| 8| 2|Nike Men's Vapor ...| NULL| 129.99|

| 9| 2|Nike Adult Vapor ...| NULL| 50.0|

| 10| 2|Under Armour Men'...| NULL| 129.99|

| 11| 2|Fitness Gear 300 ...| NULL| 209.99|

| 12| 2|Under Armour Men'...| NULL| 139.99|

| 13| 2|Under Armour Men'...| NULL| 89.99|

| 14| 2|Quik Shade Summit...| NULL| 199.99|

| 15| 2|Under Armour Kids...| NULL| 59.99|

| 16| 2|Riddell Youth 360...| NULL| 299.99|

| 17| 2|Under Armour Men'...| NULL| 129.99|

| 18| 2|Reebok Men's Full...| NULL| 29.97|

| 19| 2|Nike Men's Finger...| NULL| 124.99|

| 20| 2|Under Armour Men'...| NULL| 129.99|

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

only showing top 20 rows

root

|-- ID: integer (nullable = true)

|-- Quantities: integer (nullable = true)

|-- Product\_Name: string (nullable = true)

|-- Price: string (nullable = true)

|-- Image\_URL: double (nullable = true)

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

|Order\_ID|Product\_ID|Quantity|Unit\_Price|Total\_Price|

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

| 1| 1| 957| 1| 299.98|

| 2| 2| 1073| 1| 199.99|

| 3| 2| 502| 5| 250.0|

| 4| 2| 403| 1| 129.99|

| 5| 4| 897| 2| 49.98|

| 6| 4| 365| 5| 299.95|

| 7| 4| 502| 3| 150.0|

| 8| 4| 1014| 4| 199.92|

| 9| 5| 957| 1| 299.98|

| 10| 5| 365| 5| 299.95|

| 11| 5| 1014| 2| 99.96|

| 12| 5| 957| 1| 299.98|

| 13| 5| 403| 1| 129.99|

| 14| 7| 1073| 1| 199.99|

| 15| 7| 957| 1| 299.98|

| 16| 7| 926| 5| 79.95|

| 17| 8| 365| 3| 179.97|

| 18| 8| 365| 5| 299.95|

| 19| 8| 1014| 4| 199.92|

| 20| 8| 502| 1| 50.0|

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

only showing top 20 rows

root

|-- Order\_ID: integer (nullable = true)

|-- Product\_ID: integer (nullable = true)

|-- Quantity: integer (nullable = true)

|-- Unit\_Price: integer (nullable = true)

|-- Total\_Price: double (nullable = true)

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

| Id| timestamp|amount| mode|

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

| 1|2013-07-25 00:00:00| 11599| CLOSED|

| 2|2013-07-25 00:00:00| 256|PENDING\_PAYMENT|

| 3|2013-07-25 00:00:00| 12111| COMPLETE|

| 4|2013-07-25 00:00:00| 8827| CLOSED|

| 5|2013-07-25 00:00:00| 11318| COMPLETE|

| 6|2013-07-25 00:00:00| 7130| COMPLETE|

| 7|2013-07-25 00:00:00| 4530| COMPLETE|

| 8|2013-07-25 00:00:00| 2911| PROCESSING|

| 9|2013-07-25 00:00:00| 5657|PENDING\_PAYMENT|

| 10|2013-07-25 00:00:00| 5648|PENDING\_PAYMENT|

| 11|2013-07-25 00:00:00| 918| PAYMENT\_REVIEW|

| 12|2013-07-25 00:00:00| 1837| CLOSED|

| 13|2013-07-25 00:00:00| 9149|PENDING\_PAYMENT|

| 14|2013-07-25 00:00:00| 9842| PROCESSING|

| 15|2013-07-25 00:00:00| 2568| COMPLETE|

| 16|2013-07-25 00:00:00| 7276|PENDING\_PAYMENT|

| 17|2013-07-25 00:00:00| 2667| COMPLETE|

| 18|2013-07-25 00:00:00| 1205| CLOSED|

| 19|2013-07-25 00:00:00| 9488|PENDING\_PAYMENT|

| 20|2013-07-25 00:00:00| 9198| PROCESSING|

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

only showing top 20 rows

root

|-- Id: integer (nullable = true)

|-- timestamp: timestamp (nullable = true)

|-- amount: integer (nullable = true)

|-- mode: string (nullable = true)

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

| Date| Product\_Name|Daily\_Product\_Revenue|

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

|2013-07-26|Under Armour Men'...| 1249.9|

|2013-07-28|Under Armour Wome...| 899.97|

|2013-07-29|Manduka PRO Yoga Mat| 329.98|

|2013-08-07|Perception Sport ...| 569.95|

|2013-08-09|Majestic Men's 20...| 499.93|

|2013-07-30|Under Armour Men'...| 2960.4499999999994|

|2013-08-04|TaylorMade Men's ...| 699.96|

|2013-08-10|Nike Men's Home G...| 299.97|

|2013-08-10|Nike Men's Air Le...| 549.95|

|2013-08-11|Schutt Youth Inte...| 609.94|

|2013-07-26|Elevation Trainin...| 829.95|

|2013-07-27|Nike Men's USA Aw...| 1029.9|

|2013-08-01|Under Armour Wome...| 409.98|

|2013-08-02|Merrell Women's A...| 329.91|

|2013-08-04|Bag Boy Beverage ...| 769.87|

|2013-08-05|Top Flite Kids' 2...| 2843.73|

|2013-08-08|YETI Roadie 20 Ch...| 349.99|

|2013-07-27|Nike Women's Pro ...| 919.94|

|2013-08-01|Nike Men's USA Aw...| 499.9|

|2013-08-07|Columbia Women's ...| 1469.89|

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

only showing top 20 rows