SaiPrabath Chowdary S

PySpark Coding Challenge 10th Sep 2024

E-commerce Transactions

PySpark setup

```
[1] ! pip install pyspark
    from pyspark.sql import SparkSession
    from pyspark.sql.functions import col, sum, max, avg, count, rank, to_date, round
```

```
# creating session
spark = SparkSession.builder.appName("E-commerce Transactions").getOrCreate()

# loading data
data = '/content/drive/MyDrive/DataEngineering/PysparkCodingAssessment/EcommerceData.csv'
ecommerce_df = spark.read.csv(data, header=True, inferSchema=True)
ecommerce_df.show()
```

```
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
|transaction_id|customer_id|     product|      category|price|quantity|discount_percentage|transaction_date|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
|             1|        101|      Laptop|   Electronics| 1000|       1|                 10|      2023-08-01|
|             2|        102|  Smartphone|   Electronics|  700|       2|                  5|      2023-08-01|
|             3|        103|       Shirt|       Fashion|   40|       3|                  0|      2023-08-02|
|             4|        104|     Blender|Home Appliance|  150|       1|                 15|      2023-08-03|
|             5|        101|  Headphones|   Electronics|  100|       2|                 10|      2023-08-03|
|             6|        105|       Shoes|       Fashion|   60|       1|                 20|      2023-08-04|
|             7|        106|Refrigerator|Home Appliance|  800|       1|                 25|      2023-08-05|
|             8|        107|        Book|         Books|   20|       4|                  0|      2023-08-05|
|             9|        108|     Toaster|Home Appliance|   30|       1|                  5|      2023-08-06|
|            10|        102|      Tablet|   Electronics|  300|       2|                 10|      2023-08-06|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
```

```
[6] # 1. Calculate the Total Revenue per Category
total_revenue_by_category = ecommerce_df.groupBy("category").agg(sum(col("price")*col("quantity")).alias("total_revenue"))
total_revenue_by_category.show()
```

```
+--------------+-------------+
|      category|total_revenue|
+--------------+-------------+
|       Fashion|          180|
|   Electronics|         3200|
|         Books|           80|
|Home Appliance|          980|
+--------------+-------------+
```

```
[7] # 2. Filter Transactions with a Discount Greater Than 10%
discounted_transactions = ecommerce_df.filter(col("discount_percentage") > 10)
discounted_transactions.show()
```

```
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
|transaction_id|customer_id|     product|      category|price|quantity|discount_percentage|transaction_date|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
|             4|        104|     Blender|Home Appliance|  150|       1|                 15|      2023-08-03|
|             6|        105|       Shoes|       Fashion|   60|       1|                 20|      2023-08-04|
|             7|        106|Refrigerator|Home Appliance|  800|       1|                 25|      2023-08-05|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+
```

```
[10] # 3. Find the Most Expensive Product Sold
     most_expensive_product = ecommerce_df.orderBy(col("price").desc()).limit(1)
     most_expensive_product.show()
```

```
+--------------+-----------+-------+-----------+-----+--------+-------------------+----------------+
|transaction_id|customer_id|product|   category|price|quantity|discount_percentage|transaction_date|
+--------------+-----------+-------+-----------+-----+--------+-------------------+----------------+
|             1|        101| Laptop|Electronics| 1000|       1|                 10|      2023-08-01|
+--------------+-----------+-------+-----------+-----+--------+-------------------+----------------+
```

```
0s # 4. Calculate the Average Quantity of Products Sold per Category
   avg_quantity_by_category = ecommerce_df.groupBy("category").agg(avg("quantity").alias("avg_quantity"))
   avg_quantity_by_category.show()
```

```
+--------------+------------+
|      category|avg_quantity|
+--------------+------------+
|       Fashion|         2.0|
|   Electronics|        1.75|
|         Books|         4.0|
|Home Appliance|         1.0|
+--------------+------------+
```

```
1s [13] # 5. Identify Customers Who Purchased More Than One Product
        customers_with_multiple_products = ecommerce_df.groupBy("customer_id").agg(count("*").alias("total_products")).filter(col("total_products") > 1)
        customers_with_multiple_products.show()
```

```
+-----------+--------------+
|customer_id|total_products|
+-----------+--------------+
|        101|             2|
|        102|             2|
+-----------+--------------+
```

```
# 6. Find the Top 3 Highest Revenue Transactions
from pyspark.sql.window import Window

ecommerce_df = ecommerce_df.withColumn("transaction_date", to_date(col("transaction_date"), "yyyy-MM-dd"))
window = Window.orderBy(col("final_price").desc())

highest_revenue_transactions = ecommerce_df.withColumn("final_price", col("price")*col('quantity') - (col("price") * col("discount_percentage") / 100)) \
    .withColumn("rank", rank().over(window)) \
    .filter(col("rank") <= 3)
highest_revenue_transactions.show()
```

```
+--------------+-----------+-----------+--------------+-----+--------+-------------------+----------------+-----------+----+
|transaction_id|customer_id|    product|      category|price|quantity|discount_percentage|transaction_date|final_price|rank|
+--------------+-----------+-----------+--------------+-----+--------+-------------------+----------------+-----------+----+
|             2|        102| Smartphone|   Electronics|  700|       2|                  5|      2023-08-01|     1365.0|   1|
|             1|        101|     Laptop|   Electronics| 1000|       1|                 10|      2023-08-01|      900.0|   2|
|             7|        106|Refrigerator|Home Appliance|  800|       1|                 25|      2023-08-05|      600.0|   3|
+--------------+-----------+-----------+--------------+-----+--------+-------------------+----------------+-----------+----+
```

```
1s [19] # 7. Calculate the Total Number of Transactions per Day
        total_transactions_per_day = ecommerce_df.groupBy("transaction_date").agg(count("*").alias("total_transactions"))
        total_transactions_per_day.show()
```

```
+----------------+------------------+
|transaction_date|total_transactions|
+----------------+------------------+
|      2023-08-03|                 2|
|      2023-08-06|                 2|
|      2023-08-01|                 2|
|      2023-08-05|                 2|
|      2023-08-04|                 1|
|      2023-08-02|                 1|
+----------------+------------------+
```

```python
# 8. Find the Customer Who Spent the Most Money
total_spending_by_customer = ecommerce_df.groupBy("customer_id") \
                            .agg(sum(col("price")*col('quantity') - (col("price") * col("discount_percentage") / 100)).alias("total_spending")) \
                            .orderBy(col("total_spending").desc()).limit(1)
total_spending_by_customer.show()
```

```
+-----------+--------------+
|customer_id|total_spending|
+-----------+--------------+
|        102|        1935.0|
+-----------+--------------+
```

```python
[24] # 9. Calculate the Average Discount Given per Product Category
avg_discount_by_category = ecommerce_df.groupBy("category").agg(avg("discount_percentage").alias("avg_discount"))
avg_discount_by_category.show()
```

```
+--------------+------------+
|      category|avg_discount|
+--------------+------------+
|       Fashion|        10.0|
|   Electronics|        8.75|
|         Books|         0.0|
|Home Appliance|        15.0|
+--------------+------------+
```

```python
# 10. Create a New Column for Final Price After Discount
ecommerce_df = ecommerce_df.withColumn("final_price", col("price")*col('quantity') - (col("price") * col("discount_percentage") / 100))
ecommerce_df.show()
```

```
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+-----------+
|transaction_id|customer_id|     product|      category|price|quantity|discount_percentage|transaction_date|final_price|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+-----------+
|             1|        101|      Laptop|   Electronics| 1000|       1|                 10|      2023-08-01|      900.0|
|             2|        102|  Smartphone|   Electronics|  700|       2|                  5|      2023-08-01|     1365.0|
|             3|        103|       Shirt|       Fashion|   40|       3|                  0|      2023-08-02|      120.0|
|             4|        104|     Blender|Home Appliance|  150|       1|                 15|      2023-08-03|      127.5|
|             5|        101|  Headphones|   Electronics|  100|       2|                 10|      2023-08-03|      190.0|
|             6|        105|       Shoes|       Fashion|   60|       1|                 20|      2023-08-04|       48.0|
|             7|        106|Refrigerator|Home Appliance|  800|       1|                 25|      2023-08-05|      600.0|
|             8|        107|        Book|         Books|   20|       4|                  0|      2023-08-05|       80.0|
|             9|        108|     Toaster|Home Appliance|   30|       1|                  5|      2023-08-06|       28.5|
|            10|        102|      Tablet|   Electronics|  300|       2|                 10|      2023-08-06|      570.0|
+--------------+-----------+------------+--------------+-----+--------+-------------------+----------------+-----------+
```