SaiPrabath Chowdary S

PySpark Coding Challenge 10th Sep 2024

Retail Store Sales Data

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
```

```
spark = SparkSession.builder.appName("Retail Store Sales Data").getOrCreate()

data = '/content/drive/MyDrive/DataEngineering/PysparkCodingAssessment/RetailStoreSalesData.csv'
retail_df = spark.read.csv(data, header=True, inferSchema=True)
retail_df.show()
```

```
+--------------+------------+-----------+-----+--------+----------+
|transaction_id|product_name|   category|price|quantity|sales_date|
+--------------+------------+-----------+-----+--------+----------+
|             1|       Apple|  Groceries|  0.5|      10|2023-09-01|
|             2|     T-shirt|   Clothing| 15.0|       2|2023-09-01|
|             3|    Notebook| Stationery|  2.0|       5|2023-09-02|
|             4|      Banana|  Groceries|  0.3|      12|2023-09-02|
|             5|      Laptop|Electronics|800.0|       1|2023-09-03|
|             6|       Pants|   Clothing| 25.0|       3|2023-09-03|
|             7|  Headphones|Electronics|100.0|       2|2023-09-04|
|             8|         Pen| Stationery|  1.0|      10|2023-09-04|
|             9|      Orange|  Groceries|  0.6|       8|2023-09-05|
|            10|    Sneakers|   Clothing| 50.0|       1|2023-09-05|
+--------------+------------+-----------+-----+--------+----------+
```

```
[97] # 1. Calculate the Total Revenue per Category
     total_revenue_by_category = retail_df.withColumn("total_sales", col("price") * col("quantity")) \
         .groupBy("category").agg(sum("total_sales").alias("total_revenue"))
     total_revenue_by_category.show()
```

```
+-----------+------------------+
|   category|     total_revenue|
+-----------+------------------+
| Stationery|              20.0|
|  Groceries|13.399999999999999|
|Electronics|            1000.0|
|   Clothing|             155.0|
+-----------+------------------+
```

```
[98] # 2. Filter Transactions Where the Total Sales Amount is Greater Than $100
     high_value_transactions = retail_df.filter(col("price") * col("quantity") > 100)
     high_value_transactions.show()
```

```
+--------------+------------+-----------+-----+--------+----------+
|transaction_id|product_name|   category|price|quantity|sales_date|
+--------------+------------+-----------+-----+--------+----------+
|             5|      Laptop|Electronics|800.0|       1|2023-09-03|
|             7|  Headphones|Electronics|100.0|       2|2023-09-04|
+--------------+------------+-----------+-----+--------+----------+
```

```
# 3. Find the Most Sold Product
most_sold_product = retail_df.groupBy("product_name").agg(sum("quantity").alias("total_quantity")) \
    .orderBy(col("total_quantity").desc()).limit(1)
most_sold_product.show()
```

```
+------------+--------------+
|product_name|total_quantity|
+------------+--------------+
|      Banana|            12|
+------------+--------------+
```

```
# 4. Calculate the Average Price per Product Category
avg_price_by_category = retail_df.groupBy("category").agg(round(avg("price"),2).alias("avg_price"))
avg_price_by_category.show()
```

```
+-----------+---------+
|   category|avg_price|
+-----------+---------+
| Stationery|      1.5|
|  Groceries|     0.47|
|Electronics|    450.0|
|   Clothing|     30.0|
+-----------+---------+
```

```
# 5. Find the Top 3 Highest Grossing Products
window = Window.orderBy(col("total_revenue").desc())

top_grossing_products = retail_df.withColumn("total_sales", col("price") * col("quantity")) \
    .groupBy("product_name").agg(sum("total_sales").alias("total_revenue")) \
    .withColumn("rank", rank().over(window)) \
    .filter(col("rank") <= 3)
top_grossing_products.show()
```

```
+------------+-------------+----+
|product_name|total_revenue|rank|
+------------+-------------+----+
|      Laptop|        800.0|   1|
|  Headphones|        200.0|   2|
|       Pants|         75.0|   3|
+------------+-------------+----+
```

```
# 6. Calculate the Total Number of Items Sold per Day
retail_df = retail_df.withColumn("sales_date", to_date(col("sales_date"), "yyyy-MM-dd"))

total_items_sold_per_day = retail_df.groupBy("sales_date").agg(sum("quantity").alias("total_items_sold"))
total_items_sold_per_day.show()
```

```
+----------+----------------+
|sales_date|total_items_sold|
+----------+----------------+
|2023-09-03|               4|
|2023-09-01|              12|
|2023-09-05|               9|
|2023-09-02|              17|
|2023-09-04|              12|
+----------+----------------+
```

```
[112] # 7. Identify the Product with the Lowest Price in Each Category

      lowest_price_product = retail_df.withColumn("min_price", min("price").over(Window.partitionBy("category"))) \
                                .filter(col("price") == col("min_price")) \
                                .select("category", "product_name", "price").distinct()
      lowest_price_product.show()
```

```
+-----------+------------+-----+
|   category|product_name|price|
+-----------+------------+-----+
|   Clothing|     T-shirt| 15.0|
|Electronics|  Headphones|100.0|
|  Groceries|      Banana|  0.3|
| Stationery|         Pen|  1.0|
+-----------+------------+-----+
```

```
# 8. Calculate the Total Revenue for Each Product
total_revenue_by_product = retail_df.groupBy("product_name").agg((sum("price") * sum("quantity")).alias("total_revenue"))
total_revenue_by_product.show()
```

```
+------------+------------------+
|product_name|     total_revenue|
+------------+------------------+
|     T-shirt|              30.0|
|    Sneakers|              50.0|
|      Orange|               4.8|
|      Banana|3.5999999999999996|
|         Pen|              10.0|
|       Pants|              75.0|
|      Laptop|             800.0|
|    Notebook|              10.0|
|       Apple|               5.0|
|  Headphones|             200.0|
+------------+------------------+
```

```
# 9. Find the Total Sales per Day for Each Category
total_sales_per_day_category = retail_df.groupBy("sales_date", "category").agg((sum("price") * sum("quantity")).alias("total_sales"))\
                                        .orderBy("category","sales_date")
total_sales_per_day_category.show()
```

```
+----------+-----------+------------------+
|sales_date|   category|       total_sales|
+----------+-----------+------------------+
|2023-09-01|   Clothing|              30.0|
|2023-09-03|   Clothing|              75.0|
|2023-09-05|   Clothing|              50.0|
|2023-09-03|Electronics|             800.0|
|2023-09-04|Electronics|             200.0|
|2023-09-01|  Groceries|               5.0|
|2023-09-02|  Groceries|3.5999999999999996|
|2023-09-05|  Groceries|               4.8|
|2023-09-02| Stationery|              10.0|
|2023-09-04| Stationery|              10.0|
+----------+-----------+------------------+
```

```
[120] # 10. Create a New Column for Discounted Price
      retail_df = retail_df.withColumn("discounted_price", col("price") * 0.9)
      retail_df.show()
```

```
+--------------+------------+-----------+-----+--------+----------+----------------+
|transaction_id|product_name|   category|price|quantity|sales_date|discounted_price|
+--------------+------------+-----------+-----+--------+----------+----------------+
|             1|       Apple|  Groceries|  0.5|      10|2023-09-01|            0.45|
|             2|     T-shirt|   Clothing| 15.0|       2|2023-09-01|            13.5|
|             3|    Notebook| Stationery|  2.0|       5|2023-09-02|             1.8|
|             4|      Banana|  Groceries|  0.3|      12|2023-09-02|            0.27|
|             5|      Laptop|Electronics|800.0|       1|2023-09-03|           720.0|
|             6|       Pants|   Clothing| 25.0|       3|2023-09-03|            22.5|
|             7|  Headphones|Electronics|100.0|       2|2023-09-04|            90.0|
|             8|         Pen| Stationery|  1.0|      10|2023-09-04|             0.9|
|             9|      Orange|  Groceries|  0.6|       8|2023-09-05|            0.54|
|            10|    Sneakers|   Clothing| 50.0|       1|2023-09-05|            45.0|
+--------------+------------+-----------+-----+--------+----------+----------------+
```