Big Data Processing Lab-4

Name: Sumit Vishwakarma

Roll No: 202201320

Colab Link:
https://colab.research.google.com/drive/15CLfIprF4Cubc9vsUGJi95E8dzBFDCwC#scrollTo=ZlKIh41VQEqr

Q1: Compute top-10 selling products in terms of numbers
Code and Output:

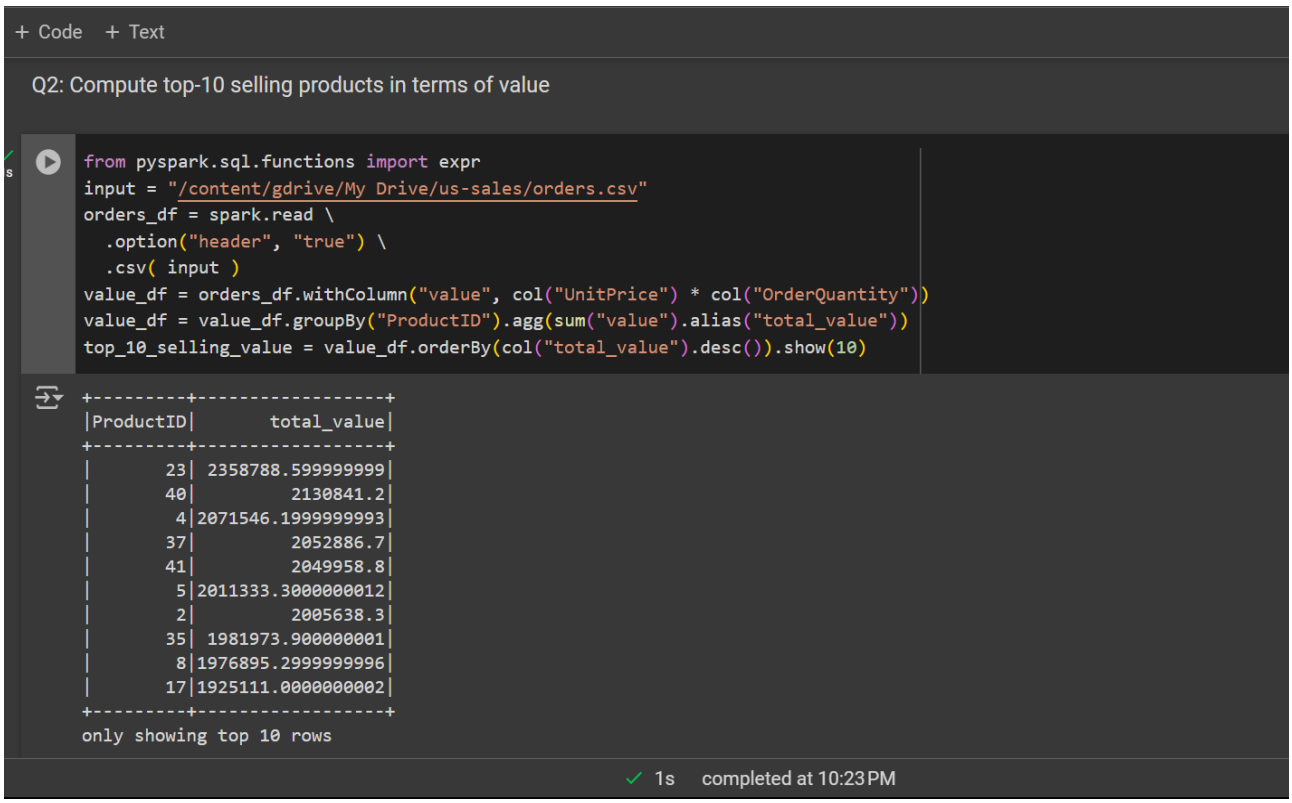
Q1: Compute top-10 selling products in terms of numbers

```
from pyspark.sql.functions import col, sum
orders_df = spark.read \
  .option("header", "true") \
  .csv( input )
qty_df = orders_df.groupBy("ProductID").agg(sum("OrderQuantity").alias("TotalQuantity"))
top_10_selling_qty = qty_df.orderBy(col("TotalQuantity").desc()).limit(10)
top_10_selling_qty.show()
```

```
+---------+-------------+
|ProductID|TotalQuantity|
+---------+-------------+
|       23|        956.0|
|       37|        896.0|
|        8|        879.0|
|        4|        878.0|
|       40|        855.0|
|       41|        854.0|
|       22|        837.0|
|       38|        832.0|
|       27|        830.0|
|       12|        827.0|
+---------+-------------+
```

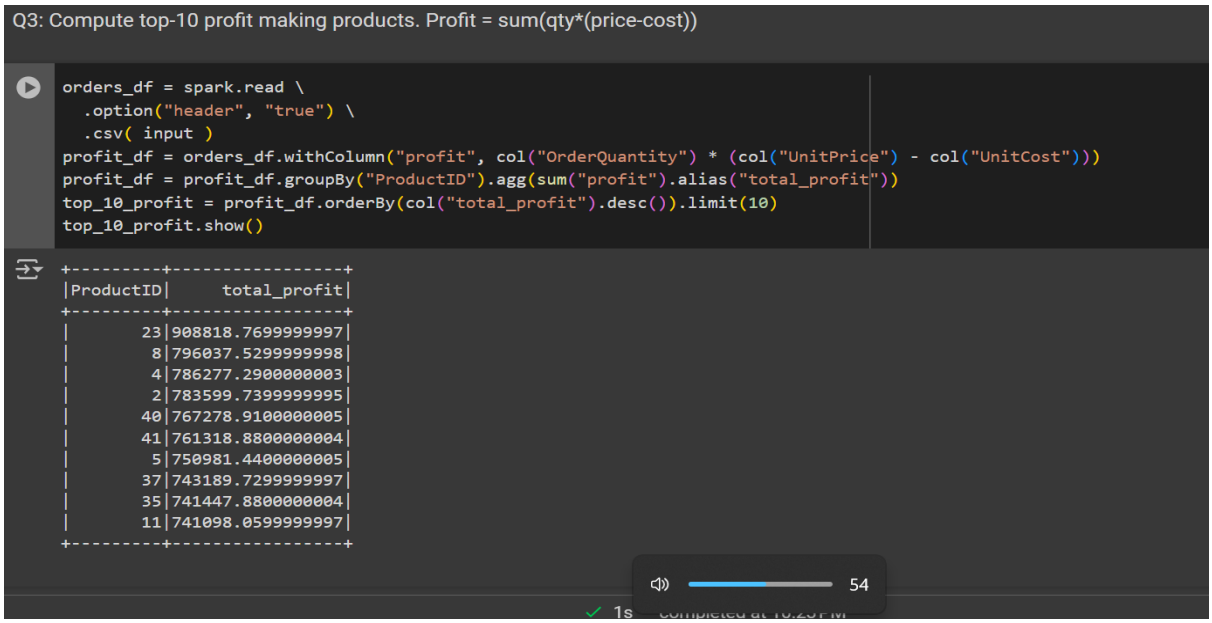
Q2: Compute top-10 selling products in terms of value

Code and Output:



Q3: Compute top-10 profit making products. Profit = sum(qty*(price-cost))

Code and Output:



Q4: Give top-3 stores selling product number 25

Code and Output:

Q4: Give top-3 stores selling product product number 25

```
from pyspark.sql.functions import col, sum
orders_df = spark.read \
  .option("header", "true") \
  .csv( input )
product_25_df = orders_df.filter(col("ProductID") == 25)
store_sales_df = product_25_df.groupBy("StoreID").agg(sum("OrderQuantity").alias("total_qty_sold"))
top_3_stores = store_sales_df.orderBy(col("total_qty_sold").desc()).limit(3)
top_3_stores.show()
```

```
+-------+--------------+
|StoreID|total_qty_sold|
+-------+--------------+
|     56|          16.0|
|     26|          14.0|
|    350|          13.0|
+-------+--------------+
```

Q5: Give top-3 products sold in mid-west region

Code and Output:

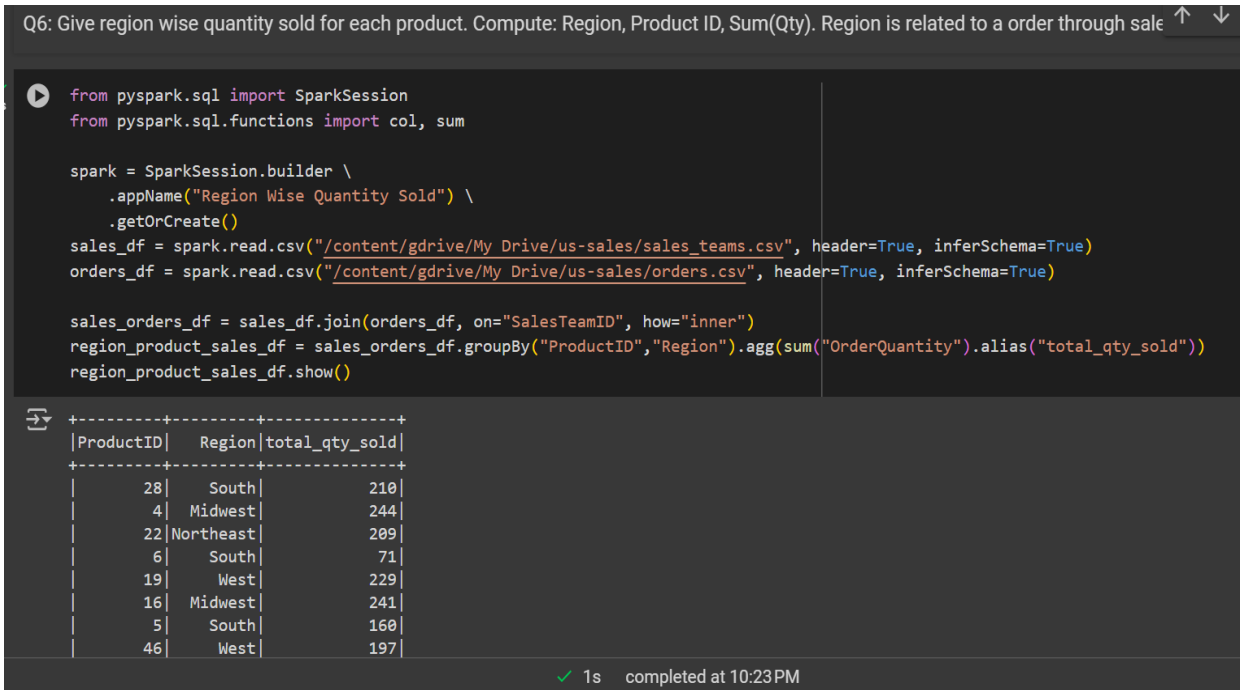
Q5: Give top-3 products sold in midwest region

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, sum

spark = SparkSession.builder \
    .appName("Top Products in Midwest") \
    .getOrCreate()

sales_df = spark.read.csv("/content/gdrive/My Drive/us-sales/sales_teams.csv", header=True, inferSchema=True)
orders_df = spark.read.csv("/content/gdrive/My Drive/us-sales/orders.csv", header=True, inferSchema=True)

sales_orders_df = sales_df.join(orders_df, on="SalesTeamID", how="inner")
midwest_df = sales_orders_df.filter(col("Region") == "Midwest")
product_sales_midwest_df = midwest_df.groupBy("ProductID").agg(sum("OrderQuantity").alias("total_qty_sold"))
top_3_products_midwest = product_sales_midwest_df.orderBy(col("total_qty_sold").desc()).limit(3)
top_3_products_midwest.show()
```

```
+---------+--------------+
|ProductID|total_qty_sold|
+---------+--------------+
|        5|           285|
|       40|           279|
|       27|           276|
+---------+--------------+
```

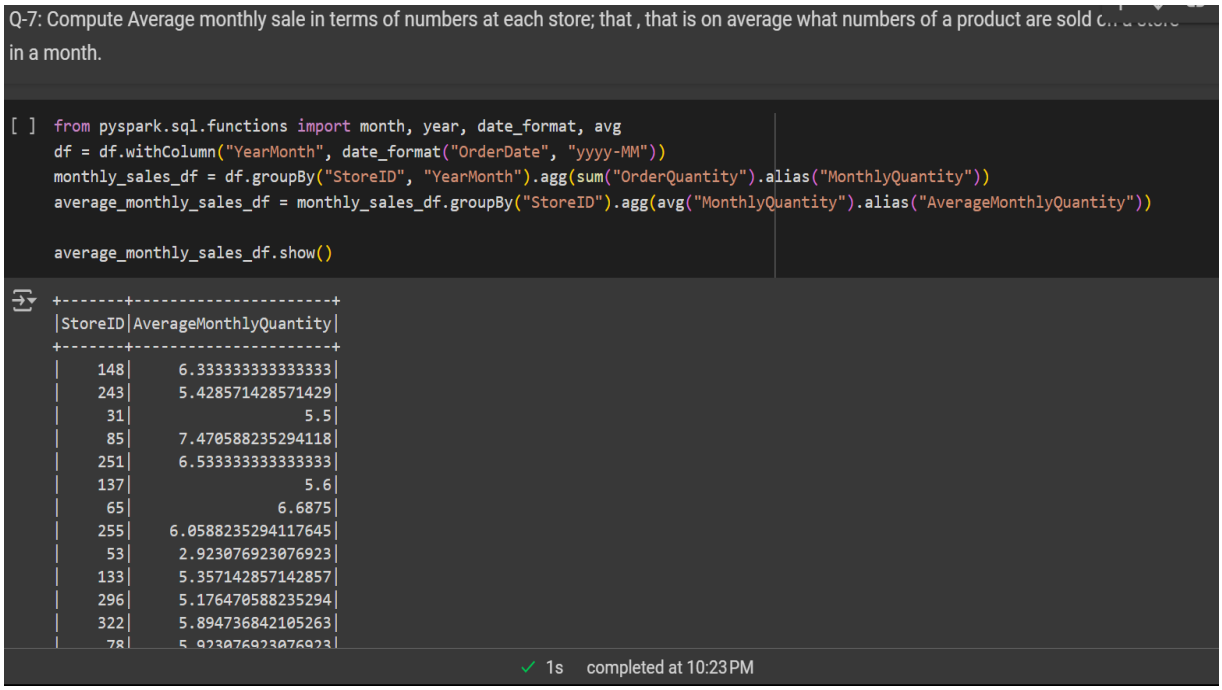
Q6: Give region wise quantity sold for each product. Compute: Region, Product ID, Sum(Qty). Region is related to a order through sales team.

Code and Output:



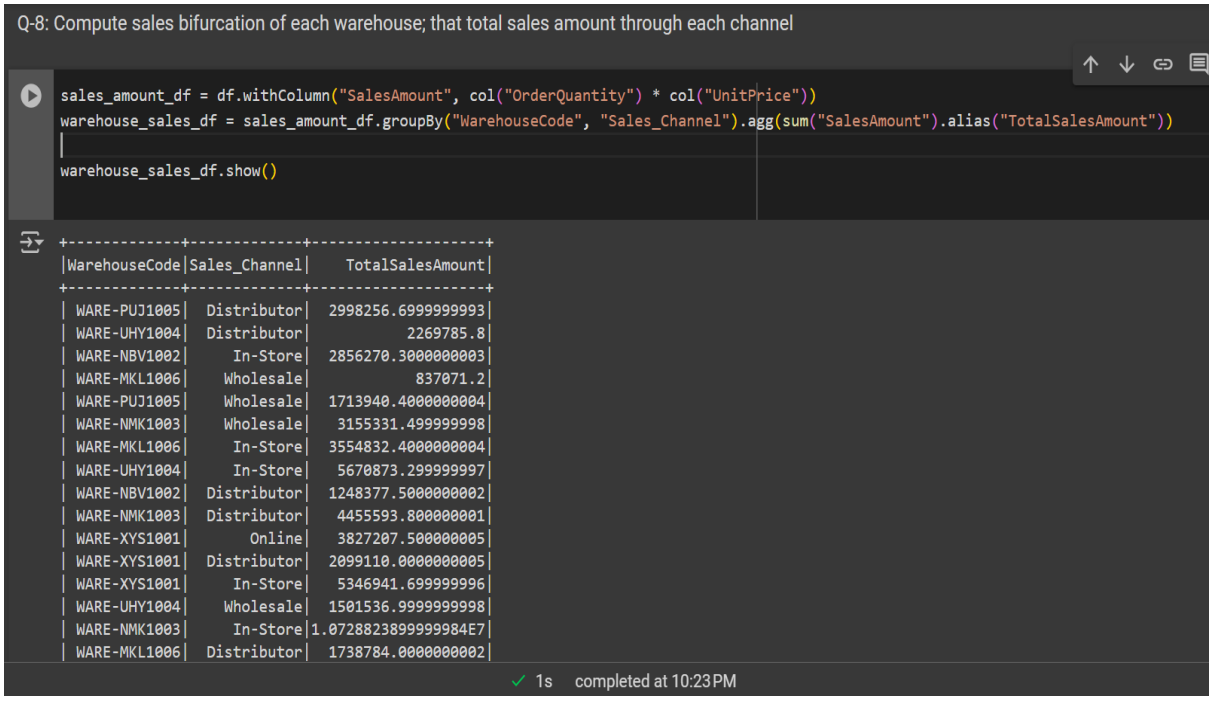
Q-7: Compute Average monthly sale in terms of numbers at each store; that , that is on average what numbers of a product are sold on a store in a month.

Code and Output:



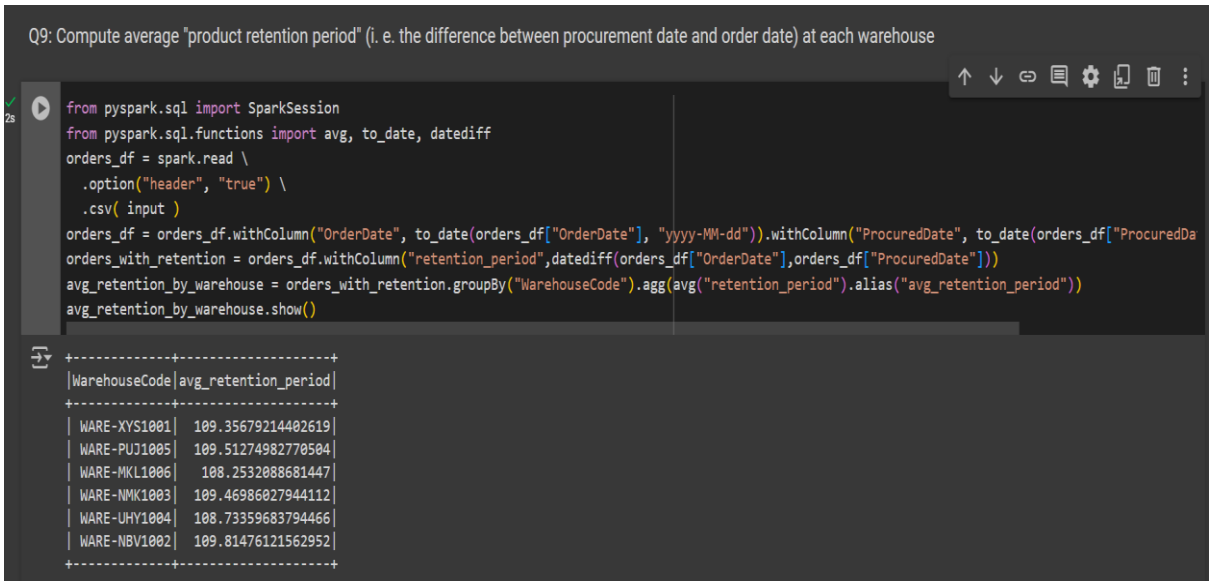
Q-8: Compute sales bifurcation of each warehouse; that total sales amount through each channel

Code and Output:



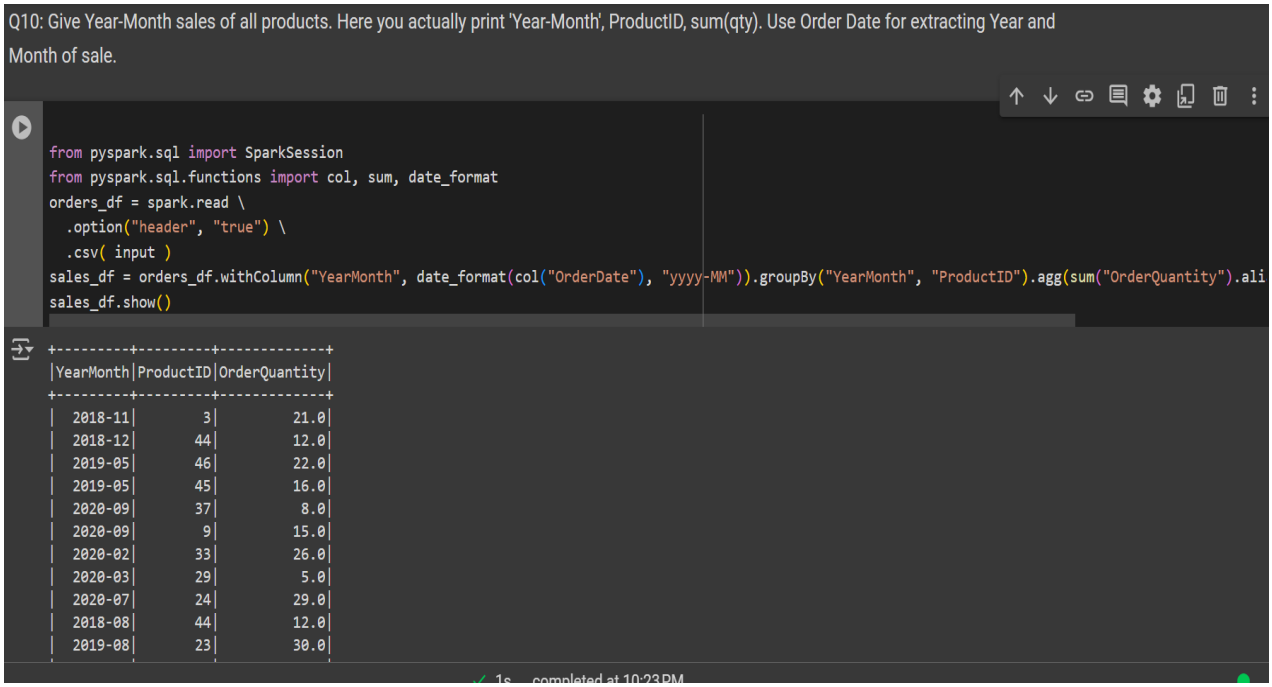
Q9: Compute average "product retention period" (i. e. the difference between procurement date and order date) at each warehouse

Code and Output:



Q10: Give Year-Month sales of all products. Here you actually print 'Year-Month', ProductID, sum(qty).

Code and Output:



Q11: Compute a fact file with the dimensions of "store_id", "product_id", and "month_year".

Code and output: