ASSIGNMENT 5.1

on

Introduction to Data Pipelines

Submitted by:

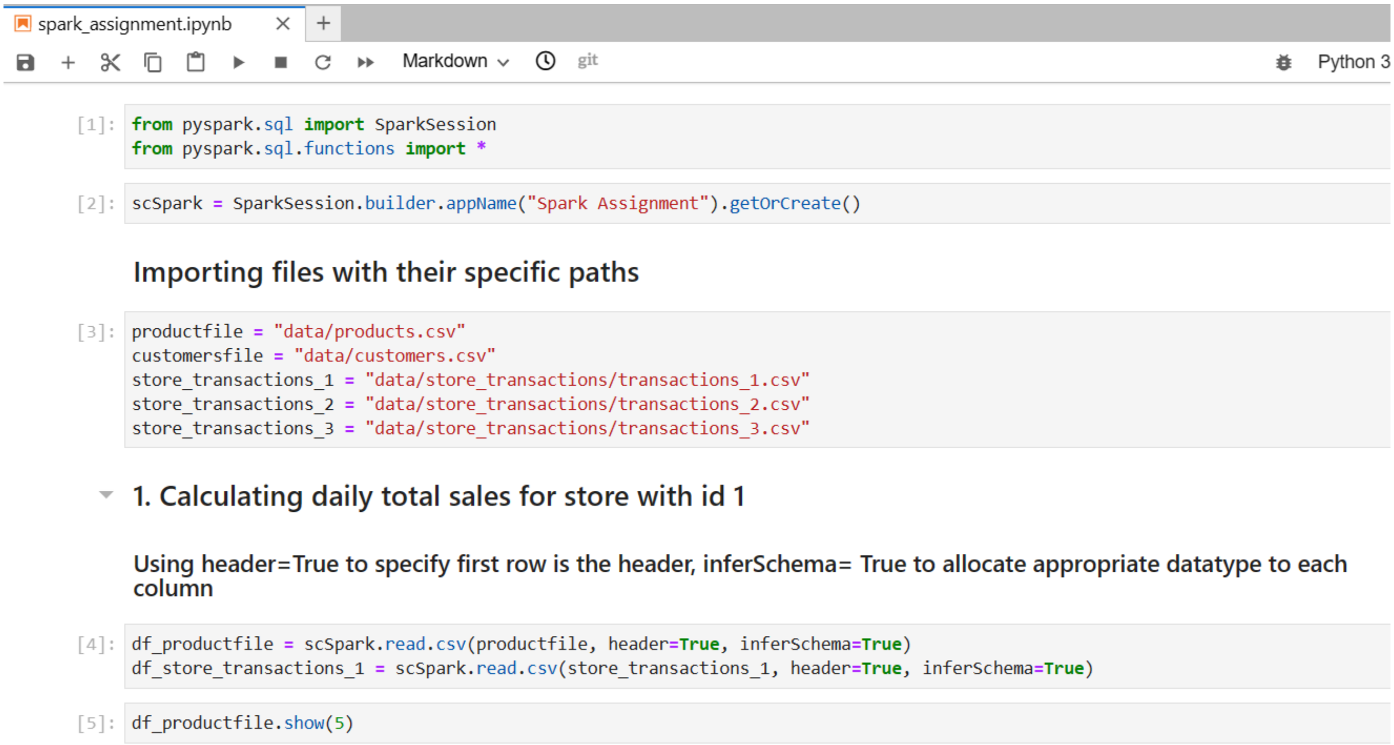
Haseebullah Shaikh (2303.KHI.DEG.015)

and

Faiza Gulzar Ahmed (2303.khi.deg.001)

Dated: 16th May 2023

Solution:



```
[1]: from pyspark.sql import SparkSession
     from pyspark.sql.functions import *
```

```
[2]: scSpark = SparkSession.builder.appName("Spark Assignment").getOrCreate()
```

Importing files with their specific paths

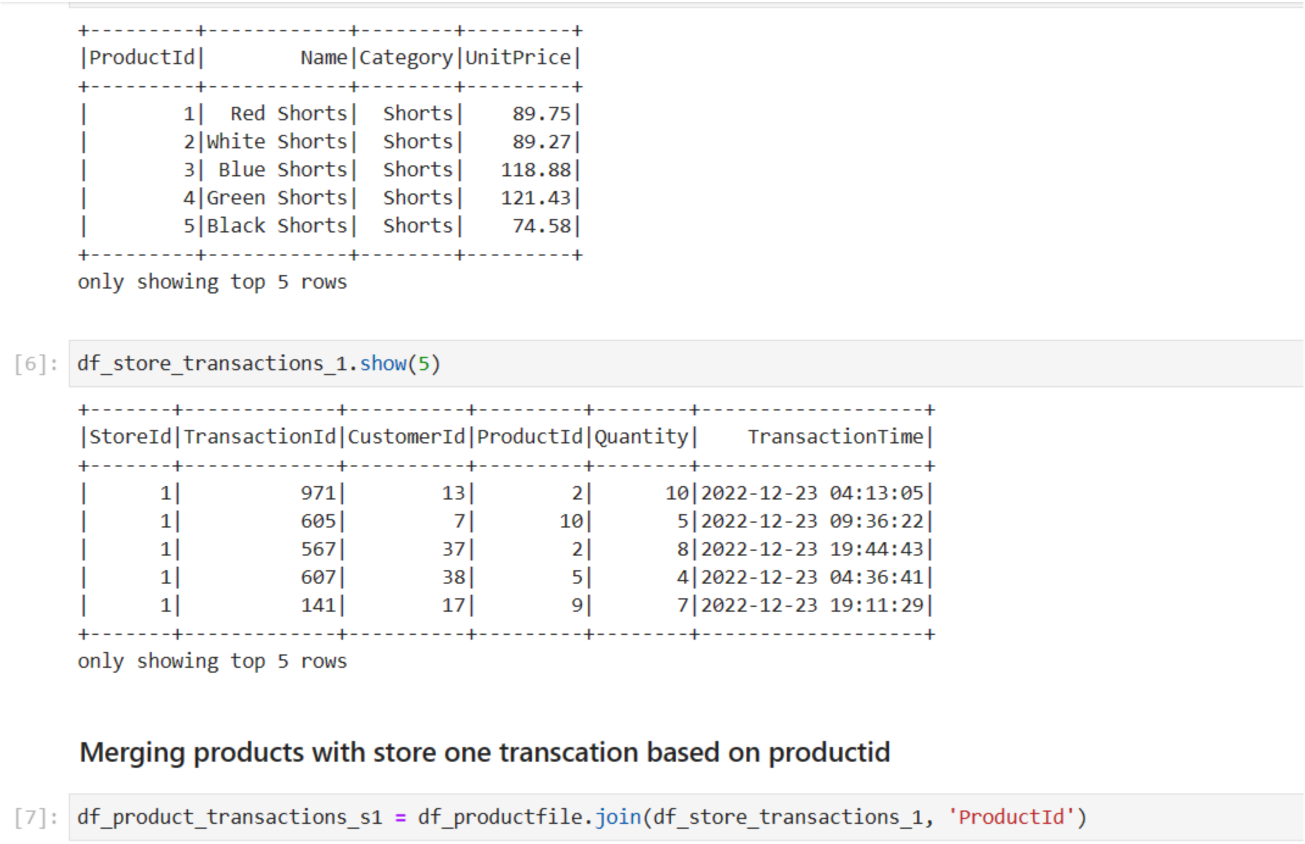
```
[3]: productfile = "data/products.csv"
     customersfile = "data/customers.csv"
     store_transactions_1 = "data/store_transactions/transactions_1.csv"
     store_transactions_2 = "data/store_transactions/transactions_2.csv"
     store_transactions_3 = "data/store_transactions/transactions_3.csv"
```

1. Calculating daily total sales for store with id 1

Using header=True to specify first row is the header, inferSchema= True to allocate appropriate datatype to each column

```
[4]: df_productfile = scSpark.read.csv(productfile, header=True, inferSchema=True)
     df_store_transactions_1 = scSpark.read.csv(store_transactions_1, header=True, inferSchema=True)
```

```
[5]: df_productfile.show(5)
```



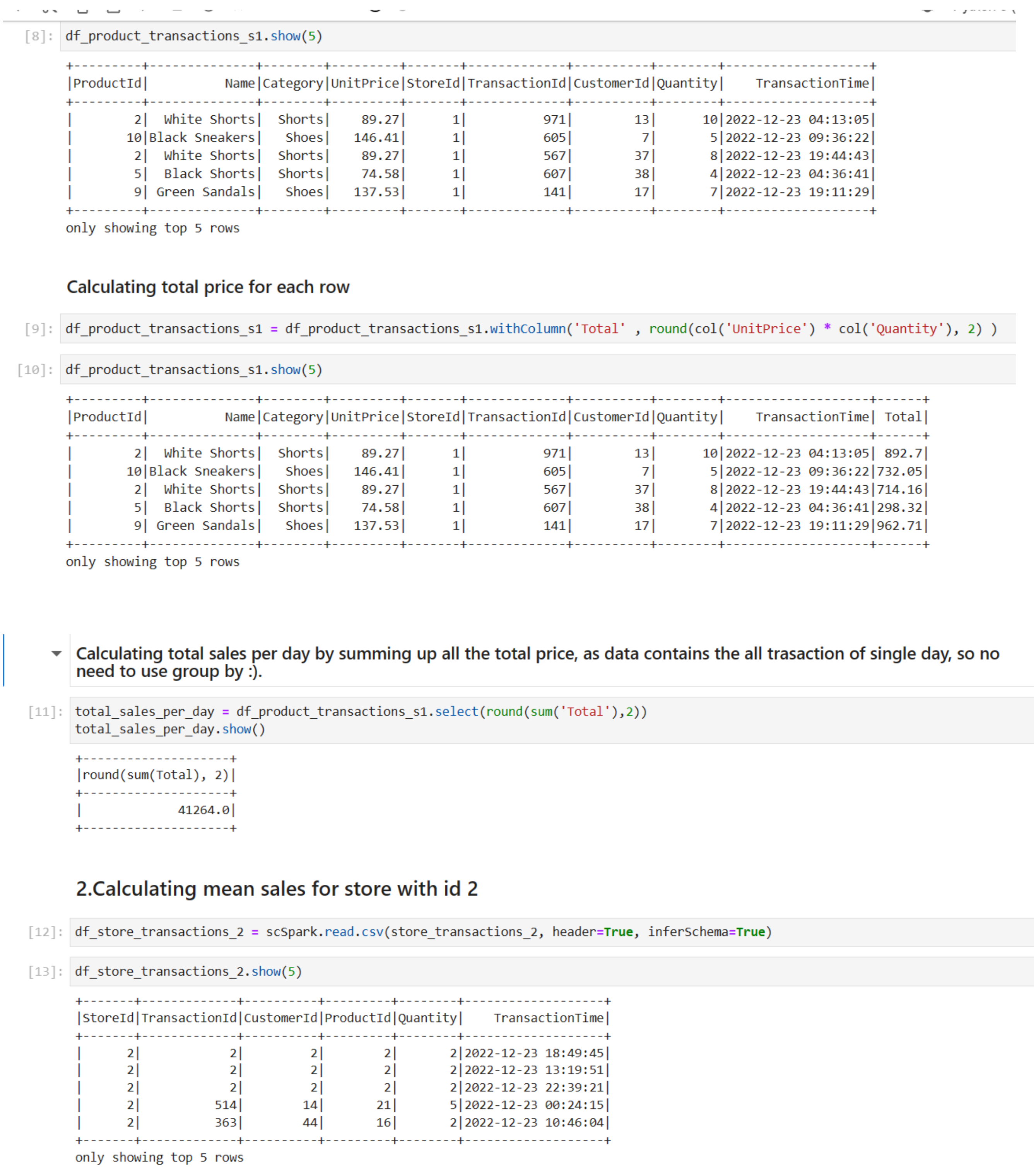
```
+---------+------------+--------+---------+
|ProductId|        Name|Category|UnitPrice|
+---------+------------+--------+---------+
|        1|  Red Shorts|  Shorts|    89.75|
|        2|White Shorts|  Shorts|    89.27|
|        3| Blue Shorts|  Shorts|   118.88|
|        4|Green Shorts|  Shorts|   121.43|
|        5|Black Shorts|  Shorts|    74.58|
+---------+------------+--------+---------+
only showing top 5 rows
```

```
[6]: df_store_transactions_1.show(5)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|StoreId|TransactionId|CustomerId|ProductId|Quantity|    TransactionTime|
+-------+-------------+----------+---------+--------+-------------------+
|      1|          971|        13|        2|      10|2022-12-23 04:13:05|
|      1|          605|         7|       10|       5|2022-12-23 09:36:22|
|      1|          567|        37|        2|       8|2022-12-23 19:44:43|
|      1|          607|        38|        5|       4|2022-12-23 04:36:41|
|      1|          141|        17|        9|       7|2022-12-23 19:11:29|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 5 rows
```

Merging products with store one transcation based on productid

```
[7]: df_product_transactions_s1 = df_productfile.join(df_store_transactions_1, 'ProductId')
```



```
[8]: df_product_transactions_s1.show(5)
```

```
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+
|ProductId|         Name|Category|UnitPrice|StoreId|TransactionId|CustomerId|Quantity|    TransactionTime|
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+
|        2| White Shorts|  Shorts|    89.27|      1|          971|        13|      10|2022-12-23 04:13:05|
|       10|Black Sneakers|   Shoes|   146.41|      1|          605|         7|       5|2022-12-23 09:36:22|
|        2| White Shorts|  Shorts|    89.27|      1|          567|        37|       8|2022-12-23 19:44:43|
|        5| Black Shorts|  Shorts|    74.58|      1|          607|        38|       4|2022-12-23 04:36:41|
|        9| Green Sandals|   Shoes|   137.53|      1|          141|        17|       7|2022-12-23 19:11:29|
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+
only showing top 5 rows
```

Calculating total price for each row

```
[9]: df_product_transactions_s1 = df_product_transactions_s1.withColumn('Total' , round(col('UnitPrice') * col('Quantity'), 2) )
```

```
[10]: df_product_transactions_s1.show(5)
```

```
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
|ProductId|         Name|Category|UnitPrice|StoreId|TransactionId|CustomerId|Quantity|    TransactionTime| Total|
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
|        2| White Shorts|  Shorts|    89.27|      1|          971|        13|      10|2022-12-23 04:13:05| 892.7|
|       10|Black Sneakers|   Shoes|   146.41|      1|          605|         7|       5|2022-12-23 09:36:22|732.05|
|        2| White Shorts|  Shorts|    89.27|      1|          567|        37|       8|2022-12-23 19:44:43|714.16|
|        5| Black Shorts|  Shorts|    74.58|      1|          607|        38|       4|2022-12-23 04:36:41|298.32|
|        9| Green Sandals|   Shoes|   137.53|      1|          141|        17|       7|2022-12-23 19:11:29|962.71|
+---------+-------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
only showing top 5 rows
```

Calculating total sales per day by summing up all the total price, as data contains the all trasaction of single day, so no need to use group by :).

```
[11]: total_sales_per_day = df_product_transactions_s1.select(round(sum('Total'),2))
      total_sales_per_day.show()
```

```
+--------------------+
|round(sum(Total), 2)|
+--------------------+
|             41264.0|
+--------------------+
```

2.Calculating mean sales for store with id 2

```
[12]: df_store_transactions_2 = scSpark.read.csv(store_transactions_2, header=True, inferSchema=True)
```

```
[13]: df_store_transactions_2.show(5)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|StoreId|TransactionId|CustomerId|ProductId|Quantity|    TransactionTime|
+-------+-------------+----------+---------+--------+-------------------+
|      2|            2|         2|        2|       2|2022-12-23 18:49:45|
|      2|            2|         2|        2|       2|2022-12-23 13:19:51|
|      2|            2|         2|        2|       2|2022-12-23 22:39:21|
|      2|          514|        14|       21|       5|2022-12-23 00:24:15|
|      2|          363|        44|       16|       2|2022-12-23 10:46:04|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 5 rows
```

Merging products with store two transcation based on productid

[14]:
```
df_product_transactions_s2 = df_productfile.join(df_store_transactions_2, 'ProductId')
df_product_transactions_s2.show(5)
```

```
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+
|ProductId|        Name|Category|UnitPrice|StoreId|TransactionId|CustomerId|Quantity|    TransactionTime|
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 18:49:45|
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 13:19:51|
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 22:39:21|
|       21|  Red Chinos|   Pants|   134.42|      2|          514|        14|       5|2022-12-23 00:24:15|
|       16|Blue t-shirt|T-Shirts|   140.68|      2|          363|        44|       2|2022-12-23 10:46:04|
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+
only showing top 5 rows
```

Calculating total price for each row

[15]:
```
df_product_transactions_s2 = df_product_transactions_s2.withColumn('Total',round(col('UnitPrice') * col('Quantity'), 2))
df_product_transactions_s2.show(5)
```

```
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
|ProductId|        Name|Category|UnitPrice|StoreId|TransactionId|CustomerId|Quantity|    TransactionTime| Total|
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 18:49:45|178.54|
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 13:19:51|178.54|
|        2|White Shorts|  Shorts|    89.27|      2|            2|         2|       2|2022-12-23 22:39:21|178.54|
|       21|  Red Chinos|   Pants|   134.42|      2|          514|        14|       5|2022-12-23 00:24:15| 672.1|
|       16|Blue t-shirt|T-Shirts|   140.68|      2|          363|        44|       2|2022-12-23 10:46:04|281.36|
+---------+------------+--------+---------+-------+-------------+----------+--------+-------------------+------+
only showing top 5 rows
```

Calculating mean sales for store id 2

[16]:
```
mean_sales = df_product_transactions_s2.agg(round(mean('Total'), 2))
```

[17]:
```
mean_sales.show()
```

```
+-------------------+
|round(avg(Total), 2)|
+-------------------+
|             513.46|
+-------------------+
```

3. Finding email of the client who spent the most by summing up his purchases from all of the stores

[18]:
```
df_store_transactions_3 = scSpark.read.csv(store_transactions_3, header=True, inferSchema=True)
df_store_transactions_3.show(5)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|StoreId|TransactionId|CustomerId|ProductId|Quantity|    TransactionTime|
+-------+-------------+----------+---------+--------+-------------------+
|      3|          454|        35|        3|       3|2022-12-23 17:36:11|
|      3|          524|        37|        9|      11|2022-12-23 22:02:51|
|      3|          562|         4|        3|       4|2022-12-23 02:51:50|
|      3|          581|        35|       14|      56|2022-12-23 17:05:54|
|      3|          200|        34|       15|      24|2022-12-23 07:15:01|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 5 rows
```

Merging the all stores transactions df using union function, as each df contained same columns :)

[19]:
```
df_all_store_transactions = df_store_transactions_1.union(df_store_transactions_2).union(df_store_transactions_3)
```

[20]:
```
df_all_store_transactions.show(5)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|StoreId|TransactionId|CustomerId|ProductId|Quantity|    TransactionTime|
+-------+-------------+----------+---------+--------+-------------------+
|      1|          971|        13|        2|      10|2022-12-23 04:13:05|
|      1|          605|         7|       10|       5|2022-12-23 09:36:22|
|      1|          567|        37|        2|       8|2022-12-23 19:44:43|
|      1|          607|        38|        5|       4|2022-12-23 04:36:41|
|      1|          141|        17|        9|       7|2022-12-23 19:11:29|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 5 rows
```

[21]:
```
df_all_store_transactions.count()
```

[21]:
```
152
```

[22]:
```
df_customersfile = scSpark.read.csv(customersfile, header=True, inferSchema=True)
df_customersfile.show(5)
```

```
+----------+--------------+--------------------+
|CustomerId|          Name|               Email|
+----------+--------------+--------------------+
|         1|Emilia Pedraza|emilia.pedraza@ex...|
|         2|  Thies Blümel|thies.blumel@exam...|
|         3| بهاره علیزاده|bhrh.aalyzdh@exam...|
|         4| Alevtin Paska|alevtin.paska@exa...|
|         5|Charlotte Wong|charlotte.wong@ex...|
+----------+--------------+--------------------+
only showing top 5 rows
```

Merging the customers to all store transactions based on CustomerId

[23]:
```
df_customer_transactions = df_customersfile.join(df_all_store_transactions, 'CustomerId')
df_customer_transactions.show(5)
```

```
+----------+--------------------+--------------------+-------+-------------+---------+--------+-------------------+
|CustomerId|                Name|               Email|StoreId|TransactionId|ProductId|Quantity|    TransactionTime|
+----------+--------------------+--------------------+-------+-------------+---------+--------+-------------------+
|        13|      Elizabeth Neal|elizabeth.neal@ex...|      1|          971|        2|      10|2022-12-23 04:13:05|
|         7|          Dominic Lo|dominic.lo@exampl...|      1|          605|       10|       5|2022-12-23 09:36:22|
|        37|       Brittany Holt|brittany.holt@exa...|      1|          567|        2|       8|2022-12-23 19:44:43|
|        38|  Filomeno Fernandes|filomeno.fernande...|      1|          607|        5|       4|2022-12-23 04:36:41|
|        17|Sevastiana Nester...|sevastiana.nester...|      1|          141|        9|       7|2022-12-23 19:11:29|
+----------+--------------------+--------------------+-------+-------------+---------+--------+-------------------+
only showing top 5 rows
```

Changing the column name as it will conflict with product name because both df have the same column name :)

[24]:
```
df_customer_transactions = df_customer_transactions.withColumnRenamed('Name', "CustomerName")
df_customer_transactions.show(5)
```

Merging the all customer transactions with products based on product id, also renaming the column name.

[25]:

```
df_customer_products_transactions = df_customer_transactions.join(df_productfile, 'ProductId').withColumnRenamed('Name','ProductName')
df_customer_products_transactions.show(5)
```

```
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+
|ProductId|CustomerId|        CustomerName|               Email|StoreId|TransactionId|Quantity|    TransactionTime|   ProductName|Category|UnitPrice|
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+
|        2|        13|      Elizabeth Neal|elizabeth.neal@ex...|      1|          971|      10|2022-12-23 04:13:05|  White Shorts|  Shorts|    89.27|
|       10|         7|          Dominic Lo|dominic.lo@exampl...|      1|          605|       5|2022-12-23 09:36:22|Black Sneakers|   Shoes|   146.41|
|        2|        37|       Brittany Holt|brittany.holt@exa...|      1|          567|       8|2022-12-23 19:44:43|  White Shorts|  Shorts|    89.27|
|        5|        38|  Filomeno Fernandes|filomeno.fernande...|      1|          607|       4|2022-12-23 04:36:41|  Black Shorts|  Shorts|    74.58|
|        9|        17|Sevastiana Nester...|sevastiana.nester...|      1|          141|       7|2022-12-23 19:11:29| Green Sandals|   Shoes|   137.53|
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+
only showing top 5 rows
```

Calculating total price for each row to find maximum purchase

[26]:

```
df_customer_products_transactions = df_customer_products_transactions.withColumn('Total',round(col('UnitPrice') * col('Quantity'), 2))
df_customer_products_transactions.show(5)
```

```
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+------+
|ProductId|CustomerId|        CustomerName|               Email|StoreId|TransactionId|Quantity|    TransactionTime|   ProductName|Category|UnitPrice| Total|
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+------+
|        2|        13|      Elizabeth Neal|elizabeth.neal@ex...|      1|          971|      10|2022-12-23 04:13:05|  White Shorts|  Shorts|    89.27| 892.7|
|       10|         7|          Dominic Lo|dominic.lo@exampl...|      1|          605|       5|2022-12-23 09:36:22|Black Sneakers|   Shoes|   146.41|732.05|
|        2|        37|       Brittany Holt|brittany.holt@exa...|      1|          567|       8|2022-12-23 19:44:43|  White Shorts|  Shorts|    89.27|714.16|
|        5|        38|  Filomeno Fernandes|filomeno.fernande...|      1|          607|       4|2022-12-23 04:36:41|  Black Shorts|  Shorts|    74.58|298.32|
|        9|        17|Sevastiana Nester...|sevastiana.nester...|      1|          141|       7|2022-12-23 19:11:29| Green Sandals|   Shoes|   137.53|962.71|
+---------+----------+--------------------+--------------------+-------+-------------+--------+-------------------+--------------+--------+---------+------+
only showing top 5 rows
```

Calculating total purchased sum for each client

[27]:

```
customer_purchased_sum = df_customer_products_transactions.groupBy('Email').agg(round(sum('Total'), 2).alias("PurchasedSum"))
customer_purchased_sum.show(5)
```

```
+--------------------+------------+
|               Email|PurchasedSum|
+--------------------+------------+
|emilia.pedraza@ex...|     5633.58|
|flenn.henderson@e...|     3279.46|
|filomeno.fernande...|     1580.47|
|lucas.christianse...|      744.78|
|kiara.brun@exampl...|      1383.8|
+--------------------+------------+
only showing top 5 rows
```

Calculating max sum from all purchased sum of each, storing value in max_purchased at row one, item first, instead of whole row

[28]:

```
max_purchased = customer_purchased_sum.agg(max('PurchasedSum')).collect()[0][0]
```

[29]:

```
max_purchased
```

[29]:

```
10653.08
```

Fetching the email of client who has purchased maximum.

[30]:

```
max_buyer_email = customer_purchased_sum.filter(customer_purchased_sum.PurchasedSum == max_purchased).collect()
```

[31]:

```
max_buyer_email
```

[31]:

```
[Row(Email='dwayne.johnson@gmail.com', PurchasedSum=10653.08)]
```

4. Fining 5 products that are most frequently bought across all stores in both terms based on transactions and quantity sold.

Finding 5 products based on transactions

couting the products according to their number of transaction grouping them by their name.

[32]:

```
products_bought = df_customer_products_transactions.groupBy('ProductName').count()
products_bought.show()
```

```
+--------------+-----+
|   ProductName|count|
+--------------+-----+
|  Blue Sneakers|    4|
|Grey Sweatpants|    1|
|   Green Shorts|    6|
|     Red Shorts|    7|
| Black Sneakers|    5|
|    Red Sandals|    6|
|  White Sandals|    3|
|       Bracelet|    4|
|   White Shorts|   20|
|   Black Shorts|    9|
|  Green Sandals|    6|
|    Blue Shorts|    6|
|          Watch|    5|
|     Red Chinos|    4|
|  Green t-shirt|    4|
|    Red t-shirt|    6|
|     Blue Jeans|    7|
|    Black Jeans|    4|
|   White Chinos|    3|
|       Earrings|    5|
+--------------+-----+
only showing top 20 rows
```

5 products that are most frequently bought based on transactions

```
33]: max_5_products_bought = products_bought.orderBy(desc('count')).limit(5)
     max_5_products_bought.show()
```

```
[33]: max_5_products_bought = products_bought.orderBy(desc('count')).limit(5)
      max_5_products_bought.show()
```

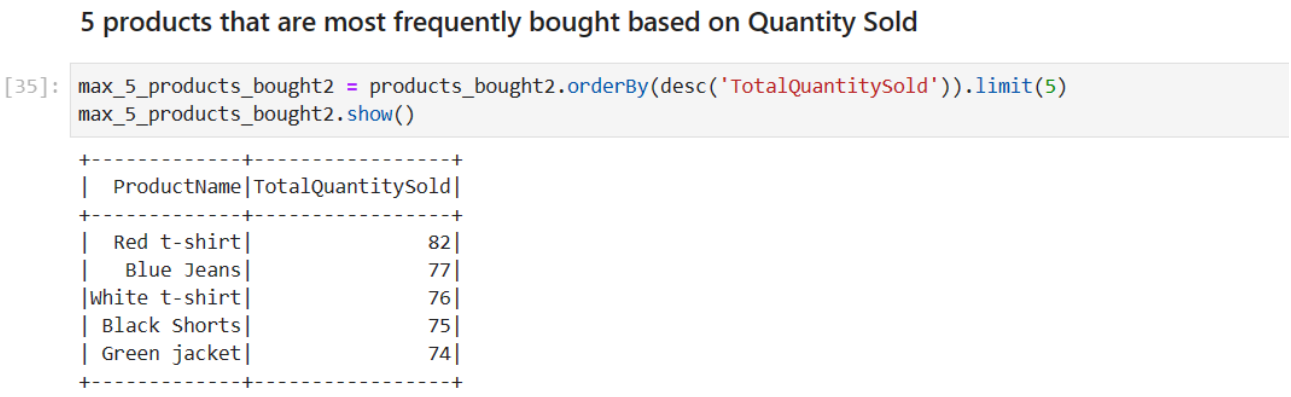
```
+-------------+-----+
|  ProductName|count|
+-------------+-----+
| White Shorts|   20|
| Black Shorts|    9|
| Green jacket|    9|
|White t-shirt|    8|
|   Red Shorts|    7|
+-------------+-----+
```

Finding 5 products based on Quantity Sold

Calculating total quanty sold for each product

```
[34]: products_bought2 = df_customer_products_transactions.groupBy('ProductName').agg(sum("Quantity").alias("TotalQuantitySold"))
      products_bought2.show()
```

```
+---------------+-----------------+
|    ProductName|TotalQuantitySold|
+---------------+-----------------+
|  Blue Sneakers|               21|
|Grey Sweatpants|                1|
|   Green Shorts|               30|
|     Red Shorts|               65|
| Black Sneakers|               30|
|    Red Sandals|               63|
|  White Sandals|               24|
|       Bracelet|               24|
|   White Shorts|               73|
```



5 products that are most frequently bought based on Quantity Sold

```
[35]: max_5_products_bought2 = products_bought2.orderBy(desc('TotalQuantitySold')).limit(5)
      max_5_products_bought2.show()
```

```
+-------------+-----------------+
|  ProductName|TotalQuantitySold|
+-------------+-----------------+
|  Red t-shirt|               82|
|   Blue Jeans|               77|
|White t-shirt|               76|
| Black Shorts|               75|
| Green jacket|               74|
+-------------+-----------------+
```

The End ☺