Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: M Humza Moeen(2303.KHI.DEG.030)

Assignment 5.1

Based on the data contained in
tasks/4_data_pipelines/day_1_introduction/daily_assignment/data
directory, use PySpark to read, filter and join the data from CSV files and answer the following questions:

What are the daily total sales for the store with id 1?

What are the mean sales for the store with id 2?

What is the email of the client who spent the most when summing up purchases from all of the stores?

Which 5 products are most frequently bought across all stores?

Problem # 1

What are the daily total sales for the store with id 1?

[2]:
```
from pyspark.sql import SparkSession
from pyspark.sql.types import IntegerType
from pyspark.sql.functions import col, to_date
from pyspark.sql.functions import sum as spark_sum
from pyspark.sql.functions import avg as spark_avg
from pyspark.sql.functions import desc
```

[3]:
```
scSpark = SparkSession.builder.appName("Spark Example").getOrCreate()
```

[4]:
```
df_merged = scSpark.read.csv("transaction*.csv", header=True)
```

[68]:
```
df_merged.limit(5).show()
```

```
+-------+------------+----------+---------+--------+------------------+
|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|
+-------+------------+----------+---------+--------+------------------+
```

Type casting Quantity into Integer

[5]:
```
df_merged = df_merged.withColumn(
    "Quantity", df_merged["Quantity"].cast(IntegerType())
)
df_merged
```

[5]:
```
DataFrame[StoreId: string, TransactionId: string, CustomerId: string, ProductId: string, Quantity: int, TransactionTime: string]
```

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: M Humza Moeen(2303.KHI.DEG.030)

Splitting date from transaction time give alias transaction data

```
[6]: df_merged_with_date = df_merged.withColumn("TransactionDate", to_date(col("TransactionTime")))
     df_merged_with_date.show()
```

```
+-------+-------------+----------+---------+--------+-------------------+---------------+
|StoreId|TransactionId|CustomerId|ProductId|Quantity|    TransactionTime|TransactionDate|
+-------+-------------+----------+---------+--------+-------------------+---------------+
|      3|          454|        35|        3|       3|2022-12-23 17:36:11|     2022-12-23|
|      3|          524|        37|        9|      11|2022-12-23 22:02:51|     2022-12-23|
|      3|          562|         4|        3|       4|2022-12-23 02:51:50|     2022-12-23|
|      3|          581|        35|       14|      56|2022-12-23 17:05:54|     2022-12-23|
|      3|          200|        34|       15|      24|2022-12-23 07:15:01|     2022-12-23|
|      3|          506|        41|       24|      19|2022-12-23 21:26:29|     2022-12-23|
|      3|          278|         5|        1|       5|2022-12-23 16:41:42|     2022-12-23|
|      3|          849|        36|       23|      13|2022-12-23 13:22:55|     2022-12-23|
|      3|          992|        34|        7|       3|2022-12-23 16:47:14|     2022-12-23|
|      3|          703|        19|        7|      13|2022-12-23 22:36:48|     2022-12-23|
|      3|          719|        48|       18|      12|2022-12-23 10:11:29|     2022-12-23|
|      3|          526|        13|       14|       3|2022-12-23 11:57:23|     2022-12-23|
|      3|          997|        20|        1|      14|2022-12-23 04:02:30|     2022-12-23|
|      3|          281|        11|       15|      25|2022-12-23 16:07:45|     2022-12-23|
|      3|          691|        48|       23|       2|2022-12-23 08:12:00|     2022-12-23|
|      3|          762|        17|        5|      26|2022-12-23 16:18:27|     2022-12-23|
|      3|          106|        24|       23|      11|2022-12-23 07:41:50|     2022-12-23|
|      3|           21|        32|        9|       2|2022-12-23 21:15:10|     2022-12-23|
|      3|          626|        14|       18|      14|2022-12-23 12:55:02|     2022-12-23|
|      3|          219|        11|       15|       5|2022-12-23 13:00:17|     2022-12-23|
+-------+-------------+----------+---------+--------+-------------------+---------------+
only showing top 20 rows
```


Storing customer and product into variable

```
[30]: df_product = scSpark.read.csv("products.csv", header=True)
```

```
[10]: df_customer = scSpark.read.csv("customers.csv", header=True)
```

Solving the 1st query

```
[12]: ### 1. What are the daily total sales for the store with id 1?
      joined_df = df_product.join(df_merged_with_date, on='ProductId', how='inner')
      joined_df
```

```
[12]: DataFrame[ProductId: string, Name: string, Category: string, UnitPrice: int, StoreId: string, TransactionId: string, CustomerId: string, Quantity: int, TransactionTime: string, T
      ransactionDate: date]
```

```
[13]: filtered_df = joined_df.filter(joined_df.StoreId == 1)
      filtered_df.show()
```

```
+---------+-------------+----------+---------+-------+-------------+----------+--------+-------------------+---------------+
|ProductId|         Name|  Category|UnitPrice|StoreId|TransactionId|CustomerId|Quantity|    TransactionTime|TransactionDate|
+---------+-------------+----------+---------+-------+-------------+----------+--------+-------------------+---------------+
|        2|  White Shorts|    Shorts|       89|      1|          971|        13|      10|2022-12-23 04:13:05|     2022-12-23|
|       10|Black Sneakers|     Shoes|      146|      1|          605|         7|       5|2022-12-23 09:36:22|     2022-12-23|
|        2|  White Shorts|    Shorts|       89|      1|          567|        37|       8|2022-12-23 19:44:43|     2022-12-23|
|        5|  Black Shorts|    Shorts|       74|      1|          607|        38|       4|2022-12-23 04:36:41|     2022-12-23|
|        9| Green Sandals|     Shoes|      137|      1|          141|        17|       7|2022-12-23 19:11:29|     2022-12-23|
|       11|         Watch|Accesories|      179|      1|          248|        17|      12|2022-12-23 06:27:58|     2022-12-23|
|        4|  Green Shorts|    Shorts|      121|      1|          726|        45|      13|2022-12-23 14:12:34|     2022-12-23|
|        9| Green Sandals|     Shoes|      137|      1|          725|         4|       1|2022-12-23 12:15:47|     2022-12-23|
|       10|Black Sneakers|     Shoes|      146|      1|          232|        30|       9|2022-12-23 01:26:10|     2022-12-23|
|        6|   Red Sandals|     Shoes|      138|      1|          954|        47|      14|2022-12-23 06:45:59|     2022-12-23|
|        5|  Black Shorts|    Shorts|       74|      1|           38|         2|       3|2022-12-23 10:19:48|     2022-12-23|
|        3|   Blue Shorts|    Shorts|      118|      1|          701|         3|      11|2022-12-23 13:22:38|     2022-12-23|
|        7| White Sandals|     Shoes|      160|      1|          783|        49|       8|2022-12-23 18:00:04|     2022-12-23|
|        8| Blue Sneakers|     Shoes|      111|      1|          333|        23|       9|2022-12-23 20:18:44|     2022-12-23|
|       11|         Watch|Accesories|      179|      1|          482|         1|       2|2022-12-23 09:05:36|     2022-12-23|
|        1|    Red Shorts|    Shorts|       89|      1|          286|        35|      12|2022-12-23 01:23:31|     2022-12-23|
|        5|  Black Shorts|    Shorts|       74|      1|          734|        43|       1|2022-12-23 23:58:16|     2022-12-23|
|        3|   Blue Shorts|    Shorts|      118|      1|           20|         1|       2|2022-12-23 05:18:30|     2022-12-23|
|        6|   Red Sandals|     Shoes|      138|      1|          203|        18|      10|2022-12-23 23:35:44|     2022-12-23|
|        5|  Black Shorts|    Shorts|       74|      1|          924|        30|       4|2022-12-23 11:35:46|     2022-12-23|
+---------+-------------+----------+---------+-------+-------------+----------+--------+-------------------+---------------+
only showing top 20 rows
```

```
[14]: daily_sales = filtered_df.groupBy("TransactionDate").agg(spark_sum(filtered_df.Quantity*filtered_df.UnitPrice).alias("total_sales"))
      daily_sales.show()
```

```
+---------------+-----------+
|TransactionDate|total_sales|
+---------------+-----------+
|     2022-12-23|      41070|
+---------------+-----------+
```

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: M Humza Moeen(2303.KHI.DEG.030)

Solving the 2nd query

```
[ ]: #What are the mean sales for the store with id 2?

[15]: filtered_df2 = joined_df.filter(joined_df.StoreId == 2)

[27]: mean_sales= filtered_df2.groupBy("TransactionDate").agg(spark_avg(filtered_df2.Quantity*filtered_df2.UnitPrice).alias("mean_sales"))

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

```
+---------------+----------------+
|TransactionDate|      mean_sales|
+---------------+----------------+
|     2022-12-23|511.921568627451|
+---------------+----------------+
```

Solving the 3rd query

```
•[18]: ### 3. What is the email of the client who spent the most when summing up purchases from all of the stores?
joined_df2 = joined_df.join(df_customer, on='CustomerId', how='inner')
joined_df2

[18]: DataFrame[CustomerId: string, ProductId: string, Name: string, Category: string, UnitPrice: int, StoreId: string, TransactionId: string, Quantity: int, TransactionTime: string, T
ransactionDate: date, Name: string, Email: string]

[20]: Email_df = joined_df2.groupBy("Email").agg(spark_sum(joined_df2.Quantity*joined_df2.UnitPrice).alias("Email purchasing"))
Email_df.show()
```

```
+--------------------+----------------+
|               Email|Email purchasing|
+--------------------+----------------+
|emilia.pedraza@ex...|            5614|
|juana.pastor@exam...|             518|
|filomeno.fernande...|            1574|
|alice.morin@examp...|            5048|
|stella.masson@exa...|            3329|
|alevtin.paska@exa...|            1534|
|dominic.lo@exampl...|            4932|
|kyn.aalyzdh@examp...|            3464|
|alexia.renaud@exa...|            2080|
|thies.blumel@exam...|            6782|
|amoli.shenoy@exam...|            1887|
|elizabeth.neal@ex...|            2578|
|dobrik.svida@exam...|            3944|
|brittany.holt@exa...|            3129|
|suzy.gibson@examp...|            4338|
|sylvie.lecomte@ex...|            2320|
|avi.shet@example.com|            5569|
|an.jansen@example...|            3225|
|angelique.vennix@...|            5310|
|bernd.colin@examp...|            2752|
+--------------------+----------------+
only showing top 20 rows
```

```
[24]: sorted_email_df = Email_df.sort(desc("Email purchasing"))
sorted_email_df.show(1)
```

```
+--------------------+----------------+
|               Email|Email purchasing|
+--------------------+----------------+
|dwayne.johnson@gm...|           10598|
+--------------------+----------------+
only showing top 1 row
```

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)
Peer Name: M Humza Moeen(2303.KHI.DEG.030)

Solving the 4rt query

[92]:
```
### Which 5 products are most frequently bought across all stores?
joined_df2
```

[92]:
```
152
```

[120]:
```
top_5_productID = joined_df2.groupBy("ProductId").agg(spark_sum(joined_df2.Quantity).alias("5_most_product_quantity")).sort(desc("5_most_product_quantity"))
top_5_productID.show(5)
```

```
+---------+----------------------+
|ProductId|5_most_product_quantity|
+---------+----------------------+
|       14|                    82|
|       24|                    77|
|       15|                    76|
|        5|                    75|
|       19|                    74|
+---------+----------------------+
only showing top 5 rows
```

[121]:
```
top_product_ids_with_names = top_5_productID.join(df_product, ["ProductId"], "left_outer").sort(desc("5_most_product_quantity"))
top_product_ids_with_names.show(5)
```

```
+---------+----------------------+-------------+--------+---------+
|ProductId|5_most_product_quantity|         Name|Category|UnitPrice|
+---------+----------------------+-------------+--------+---------+
|       14|                    82|  Red t-shirt|T-Shirts|      121|
|       24|                    77|   Blue Jeans|   Pants|      173|
|       15|                    76|White t-shirt|T-Shirts|      131|
|        5|                    75| Black Shorts|  Shorts|       74|
|       19|                    74| Green jacket| Jackets|      223|
+---------+----------------------+-------------+--------+---------+
only showing top 5 rows
```

[124]:
```
selected_data = top_product_ids_with_names.select("ProductId", "Name","5_most_product_quantity" )
selected_data.show(5)
```

```
+---------+-------------+----------------------+
|ProductId|         Name|5_most_product_quantity|
+---------+-------------+----------------------+
|       14|  Red t-shirt|                    82|
|       24|   Blue Jeans|                    77|
|       15|White t-shirt|                    76|
|        5| Black Shorts|                    75|
|       19| Green jacket|                    74|
+---------+-------------+----------------------+
only showing top 5 rows
```