UNIT 5.1 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)

Umaima Siddiqui (2023.KHI.DEG.033)

UNIT 5.1 GRADED ASSIGNMENT

Task:

- 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?

Solution:

---

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

(To do this, join transactions from store 1 and products tables, multiply `Quantity` and `UnitPrice` columns and find the sum)

[2]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
transactions_1 = spark.read.format("csv").option("header", "true").load("transactions_1.csv")
products = spark.read.format("csv").option("header", "true").load("products.csv")

joined_df = transactions_1.join(products, transactions_1["ProductId"] == products["ProductId"])
joined_df = joined_df.withColumn("Total", joined_df["Quantity"] * joined_df["UnitPrice"])

total_sum = joined_df.selectExpr("sum(Total) as TotalSum").collect()[0]["TotalSum"]
print(total_sum)
```

```
41264.000000000015
```

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

(To do this, join transactions from store 2 and products tables, multiply `Quantity` and `UnitPrice` columns and find the mean)

*[3]:
```
from pyspark.sql.functions import col

transactions_2 = spark.read.format("csv").option("header", "true").load("transactions_2.csv")
products = spark.read.format("csv").option("header", "true").load("products.csv")

joined_df = transactions_2.join(products, transactions_2["ProductId"] == products["ProductId"])
store_2_transactions = joined_df.filter(col("StoreId") == 2)
store_2_transactions = store_2_transactions.withColumn("TotalSales", col("Quantity") * col("UnitPrice"))

mean_sales = store_2_transactions.selectExpr("avg(TotalSales) as MeanSales").collect()[0]["MeanSales"]
print(mean_sales)
```

```
513.4598039215689
```

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

[3]:
```
transactions_1 = spark.read.format("csv").option("header", "true").load("transactions_1.csv")
transactions_2 = spark.read.format("csv").option("header", "true").load("transactions_2.csv")
transactions_3 = spark.read.format("csv").option("header", "true").load("transactions_3.csv")
all_stores_transactions = transactions_1.union(transactions_2).union(transactions_3)
all_stores_transactions.show()
```

```
+-------+-------------+----------+---------+--------+-------------------+
|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|
|      1|          248|        17|       11|      12|2022-12-23 06:27:58|
|      1|          726|        45|        4|      13|2022-12-23 14:12:34|
|      1|          725|         4|        9|       1|2022-12-23 12:15:47|
|      1|          232|        30|       10|       9|2022-12-23 01:26:10|
|      1|          954|        47|        6|      14|2022-12-23 06:45:59|
|      1|           38|         2|        5|       3|2022-12-23 10:19:48|
|      1|          701|         3|        3|      11|2022-12-23 13:22:38|
|      1|          783|        49|        7|       8|2022-12-23 18:00:04|
|      1|          333|        23|        8|       9|2022-12-23 20:18:44|
|      1|          482|         1|       11|       2|2022-12-23 09:05:36|
|      1|          286|        35|        1|      12|2022-12-23 01:23:31|
|      1|          734|        43|        5|       1|2022-12-23 23:58:16|
|      1|           20|         1|        3|       2|2022-12-23 05:18:30|
|      1|          203|        18|        6|      10|2022-12-23 23:35:44|
|      1|          924|        30|        5|       4|2022-12-23 11:35:46|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 20 rows
```

[7]:
```
customers = spark.read.format("csv").option("header", "true").load("customers.csv")
customer_joined_df = all_stores_transactions.join(customers, "CustomerId")
customer_joined_df.show()
```

```
+----------+-------+-------------+---------+--------+-------------------+--------------------+--------------------+
|CustomerId|StoreId|TransactionId|ProductId|Quantity|    TransactionTime|                Name|               Email|
+----------+-------+-------------+---------+--------+-------------------+--------------------+--------------------+
|        13|      1|          971|        2|      10|2022-12-23 04:13:05|      Elizabeth Neal|elizabeth.neal@ex...|
|         7|      1|          605|       10|       5|2022-12-23 09:36:22|         Dominic Lo|dominic.lo@exampl...|
|        37|      1|          567|        2|       8|2022-12-23 19:44:43|       Brittany Holt|brittany.holt@exa...|
|        38|      1|          607|        5|       4|2022-12-23 04:36:41|  Filomeno Fernandes|filomeno.fernande...|
|        17|      1|          141|        9|       7|2022-12-23 19:11:29|Sevastiana Nester...|sevastiana.nester...|
|        17|      1|          248|       11|      12|2022-12-23 06:27:58|Sevastiana Nester...|sevastiana.nester...|
|        45|      1|          726|        4|      13|2022-12-23 14:12:34|   Melissa Patterson|melissa.patterson...|
|         4|      1|          725|        9|       1|2022-12-23 12:15:47|       Alevtin Paska|alevtin.paska@exa...|
|        30|      1|          232|       10|       9|2022-12-23 01:26:10|    Raymonde Riviere|raymonde.riviere@...|
|        47|      1|          954|        6|      14|2022-12-23 06:45:59|     Flenn Henderson|flenn.henderson@e...|
|         2|      1|           38|        5|       3|2022-12-23 10:19:48|        Thies Blümel|thies.blumel@exam...|
|         3|      1|          701|        3|      11|2022-12-23 13:22:38|      بهاره علیزاده|bhrh.aalyzdh@exam...|
|        49|      1|          783|        7|       8|2022-12-23 18:00:04|   Jonathan Carrasco|jonathan.carrasco...|
|        23|      1|          333|        8|       9|2022-12-23 20:18:44|    Ceyhun Hamzaoğlu|ceyhun.hamzaoglu@...|
|         1|      1|          482|       11|       2|2022-12-23 09:05:36|      Emilia Pedraza|emilia.pedraza@ex...|
|        35|      1|          286|        1|      12|2022-12-23 01:23:31|      Dwayne Johnson|dwayne.johnson@gm...|
|        43|      1|          734|        5|       1|2022-12-23 23:58:16|  Lucas Christiansen|lucas.christianse...|
|         1|      1|           20|        3|       2|2022-12-23 05:18:30|      Emilia Pedraza|emilia.pedraza@ex...|
|        18|      1|          203|        6|      10|2022-12-23 23:35:44|         Kiara Brun|kiara.brun@exampl...|
|        30|      1|          924|        5|       4|2022-12-23 11:35:46|    Raymonde Riviere|raymonde.riviere@...|
+----------+-------+-------------+---------+--------+-------------------+--------------------+--------------------+
only showing top 20 rows
```

[9]:
```
customers_purchases = customer_joined_df.groupBy("CustomerId", "Email").agg(sum("Quantity").alias("TotalPurchases"))
customer_most_spent = customers_purchases.orderBy(desc("TotalPurchases")).first()
email_most_spent = customer_most_spent["Email"]
print("Email of the client who spent the most:", email_most_spent)
```

```
Email of the client who spent the most: dwayne.johnson@gmail.com
```

4. Which 5 products are most frequently bought across all stores?

[10]:
```
frequent_products = all_stores_transactions.groupBy("ProductId").count()
frequent_products.show()
```

```
+---------+-----+
|ProductId|count|
+---------+-----+
|        7|    3|
|       11|    5|
|        3|    6|
|        8|    4|
|        5|    9|
|       26|    1|
|        6|    6|
|        9|    6|
|        1|    7|
|       10|    5|
|        4|    6|
|       12|    4|
|        2|   20|
|       15|    8|
|       22|    3|
|       16|    6|
|       18|    3|
|       27|    1|
|       17|    4|
|       19|    9|
+---------+-----+
only showing top 20 rows
```

[11]:
```
sorted_products = frequent_products.orderBy(desc("count"))
sorted_products.show()
```

```
+---------+-----+
|ProductId|count|
+---------+-----+
|        2|   20|
|        5|    9|
|       19|    9|
|       15|    8|
|        1|    7|
|       24|    7|
|        3|    6|
|        6|    6|
|        9|    6|
|        4|    6|
|       16|    6|
|       23|    6|
|       14|    6|
|       11|    5|
|       10|    5|
|       13|    5|
|        8|    4|
|       12|    4|
|       17|    4|
|       25|    4|
+---------+-----+
only showing top 20 rows
```

[12]:
```
top_5_products = sorted_products.limit(5)
top_5_products.show()
```

```
+---------+-----+
|ProductId|count|
+---------+-----+
|        2|   20|
|        5|    9|
|       19|    9|
|       15|    8|
|        1|    7|
+---------+-----+
```