In [2]:

```python
from pyspark.sql import SparkSession
from pyspark.sql.types import IntegerType
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, date_format, sum, avg, desc
```

In [3]:

```python
# Create a SparkSession
spark = SparkSession.builder.appName("data").getOrCreate()
```

In [4]:

```python
# Read the CSV files
transactions_df = spark.read.csv("transactions_*", inferSchema=True, head
products_df = spark.read.csv("products.csv", inferSchema=True, header=Tru
customers_df = spark.read.csv("customers.csv", inferSchema=True, header=T
```

In [5]:

```python
transactions_df.printSchema()
products_df.printSchema()
customers_df.printSchema()
```

```
root
 |-- StoreId: integer (nullable = true)
 |-- TransactionId: integer (nullable = true)
 |-- CustomerId: integer (nullable = true)
 |-- ProductId: integer (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- TransactionTime: timestamp (nullable = true)

root
 |-- ProductId: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Category: string (nullable = true)
 |-- UnitPrice: double (nullable = true)

root
 |-- CustomerId: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Email: string (nullable = true)
```

In [6]:

```python
transactions_df.show(3)
products_df.show(3)
customers_df.show(3)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|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|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 3 rows

+---------+------------+--------+---------+
|ProductId|        Name|Category|UnitPrice|
+---------+------------+--------+---------+
|        1|  Red Shorts|  Shorts|    89.75|
|        2|White Shorts|  Shorts|    89.27|
|        3| Blue Shorts|  Shorts|   118.88|
+---------+------------+--------+---------+
only showing top 3 rows

+----------+-------------+--------------------+
|CustomerId|         Name|               Email|
+----------+-------------+--------------------+
|         1|Emilia Pedraza|emilia.pedraza@ex...|
|         2|  Thies Blümel|thies.blumel@exam...|
|         3| بهاره علیزاده|bhrh.aalyzdh@exam...|
+----------+-------------+--------------------+
only showing top 3 rows
```

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

In [7]:

```
# Filter the transactions DataFrame to only include rows with storeId 1
store_id_1 = transactions_df.filter(transactions_df.StoreId == 1)
store_id_1.limit(10).show(3)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|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|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 3 rows
```

In [8]:

```
# Add a column to DataFrame with the date of the transaction (for daily t
store_id_1 = store_id_1.withColumn("TransactionDate", date_format("Transa
```

In [9]:

```
# Join the store_id_1 DataFrame with the products DataFrame on ProductId
daily_sales_df = store_id_1.join(products_df, on="ProductId")
```

In [10]:

```
# Calculate the total sales for each row
daily_sales_df = daily_sales_df.withColumn("Daily Sales", col("UnitPrice"
```

In [11]:

```
# Calculate the total sales for each day
daily_sales_df = daily_sales_df.groupBy("TransactionDate").agg(sum("Daily
daily_sales_df.show(5)
```

```
+---------------+------------------+
|TransactionDate|       Total Sales|
+---------------+------------------+
|     2022-12-23|41264.000000000015|
+---------------+------------------+
```

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

In [12]:

```
# Filter the transactions DataFrame to only include rows with storeId 2
store_id_2 = transactions_df.filter(transactions_df.StoreId == 2)
store_id_2.show(3)
```

```
+-------+-------------+----------+---------+--------+-------------------+
|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|
+-------+-------------+----------+---------+--------+-------------------+
only showing top 3 rows
```

In [13]:

```
# Add a column to the store_id_2 DataFrame with the date of the transacti
store_id_2 = store_id_2.withColumn("TransactionDate", date_format("Transa
```

In [14]:

```
# Join the store_id_2 DataFrame with the products DataFrame on ProductId
daily_sales_df = store_id_2.join(products_df, on="ProductId")

# Calculate the total sales for each row
daily_sales_df = daily_sales_df.withColumn("Total_SALES", col("UnitPrice"
```

In [15]:

```
# Calculate the average sales for each day
daily_sales_df = daily_sales_df.groupBy("TransactionDate").agg(avg("Total
daily_sales_df.show()
```

```
+---------------+-----------------+
|TransactionDate|    Average Sales|
+---------------+-----------------+
|     2022-12-23|513.4598039215689|
+---------------+-----------------+
```

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

In [16]:

```
# Join the transactions and customers DataFrames on CustomerId and Produc
customer_purchases_df = transactions_df.join(customers_df, on="CustomerId
```

In [17]:

```
# Calculate the total sales for each customer
customer_purchases_df = customer_purchases_df.withColumn("Customer_purcha
```

In [18]:

```
# Group the customer_purchases_df DataFrame by CustomerId and sum the Cus
customer_purchases_df = customer_purchases_df.groupBy("CustomerId").agg(s
```

In [19]:

```
# Join the customer_purchases_df DataFrame with the customers DataFrame o
customer_purchases_df = customer_purchases_df.join(customers_df, on="Cust
```

In [20]:

```
# Order the customer_purchases_df DataFrame by Total Purchases in descend
customer_purchases_df  = customer_purchases_df.orderBy(desc("Total Purcha
```

In [21]:

```
# Get the email of the customer with the highest Total Purchases
highest_purchaser_email = customer_purchases_df.first()["Email"]
print(highest_purchaser_email)
```

```
dwayne.johnson@gmail.com
```

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

In [22]:

```
#Join the transactions and products DataFrames on ProductId
products = transactions_df.join(products_df, on="ProductId")
```

In [23]:

```
# Group DataFrame by ProductId and product name and count the sum of quan
most_frequently_products = products.groupBy('ProductId','Name').agg(sum('
```

In [24]:

```
# Order the top 5 most frequently products in descending order
most_frequently_products = most_frequently_products.orderBy(desc("Quantit
most_frequently_products.show(5)
```

```
+---------+------------+-----------------+
|ProductId|        Name|QuantityPurchased|
+---------+------------+-----------------+
|       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
```

In [ ]: