# **PYSPARK HANDS ON 2**

#Part 1: Dataset Preparation

import pandas as pd

from datetime import datetime

# Sample sales data

data = {

"TransactionID": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

"CustomerID": [101, 102, 103, 101, 104, 102, 103, 104, 101, 105],

"ProductID": [501, 502, 501, 503, 504, 502, 503, 504, 501, 505],

"Quantity": [2, 1, 4, 3, 1, 2, 5, 1, 2, 1],

"Price": [150.0, 250.0, 150.0, 300.0, 450.0, 250.0, 300.0, 450.0, 150.0, 550.0],

"Date": [

datetime(2024, 9, 1),

datetime(2024, 9, 1),

datetime(2024, 9, 2),

datetime(2024, 9, 2),

datetime(2024, 9, 3),

datetime(2024, 9, 3),

datetime(2024, 9, 4),

datetime(2024, 9, 4),

datetime(2024, 9, 5),

datetime(2024, 9, 5)

]

}

# Create a DataFrame

df = pd.DataFrame(data)

# Save the DataFrame to a CSV file

df.to\_csv('sales\_data.csv', index=False)

print("Sample sales dataset has been created and saved as 'sales\_data.csv'.")

**OUTPUT:**



#Part 2: Load and Analyze the Dataset Using PySpark

#Step 2: Load the Dataset into PySpark

#1. Initialize the SparkSession:

from pyspark.sql import SparkSession

spark = SparkSession.builder \

.appName("Sales Dataset Analysis") \

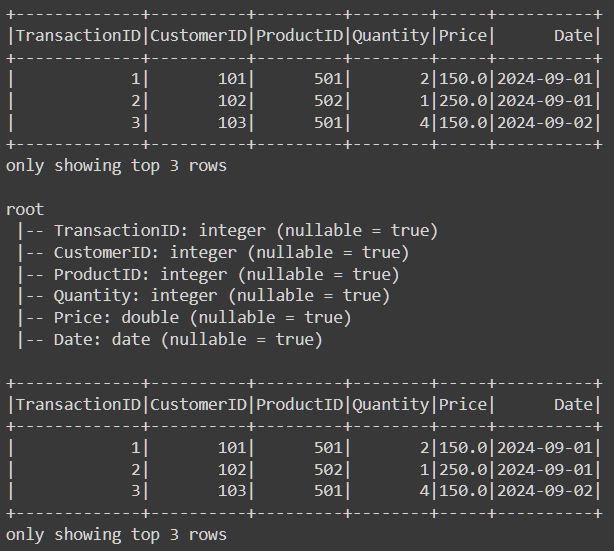
.getOrCreate()

#2. Load the CSV File into a PySpark DataFrame:

sales\_df = spark.read.csv('sales\_data.csv', header=True, inferSchema=True)

sales\_df.show(3)

**OUTPUT:**

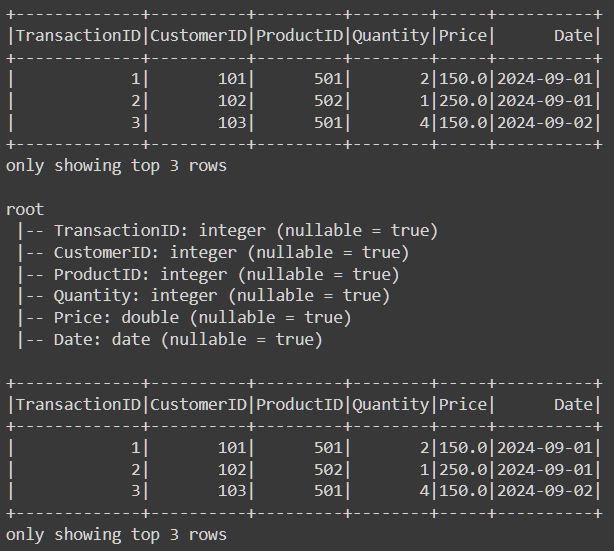


#Step 3: Explore the Data

#1. Print the Schema:

sales\_df.printSchema()

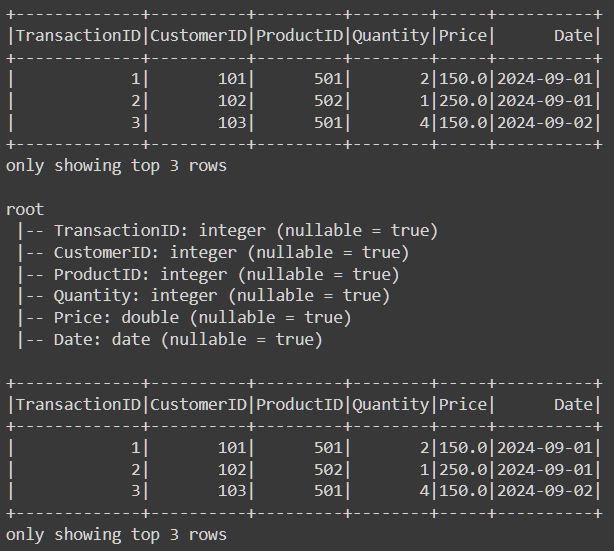
**OUTPUT:**



#2. Show the First Few Rows:

sales\_df.show(3)

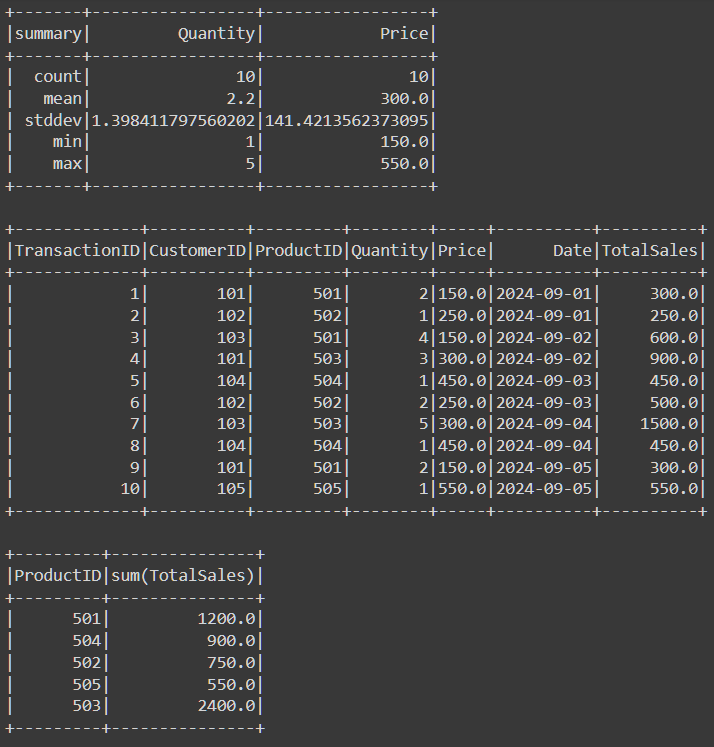
**OUTPUT:**



#3. Get Summary Statistics:

sales\_df.describe(['Quantity', 'Price']).show()

**OUTPUT:**



#Step 4: Perform Data Transformations and Analysis

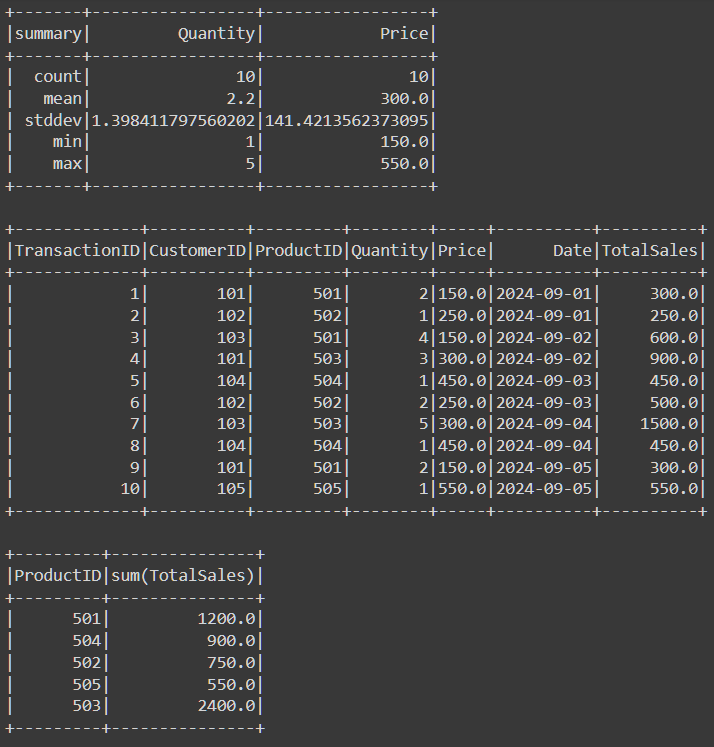
#1. Calculate the Total Sales Value for Each Transaction:

from pyspark.sql.functions import col

sales\_df = sales\_df.withColumn('TotalSales', col('Quantity') \* col('Price'))

sales\_df.show()

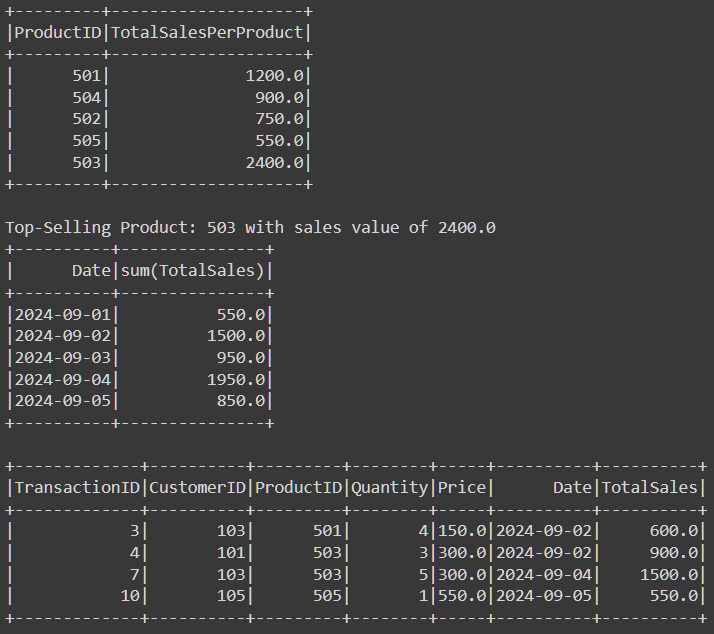
**OUTPUT:**



#2. Group By ProductID and Calculate Total Sales Per Product:

sales\_df.groupBy('ProductID').sum('TotalSales').alias('TotalSalesPerProduct').show()

**OUTPUT:**



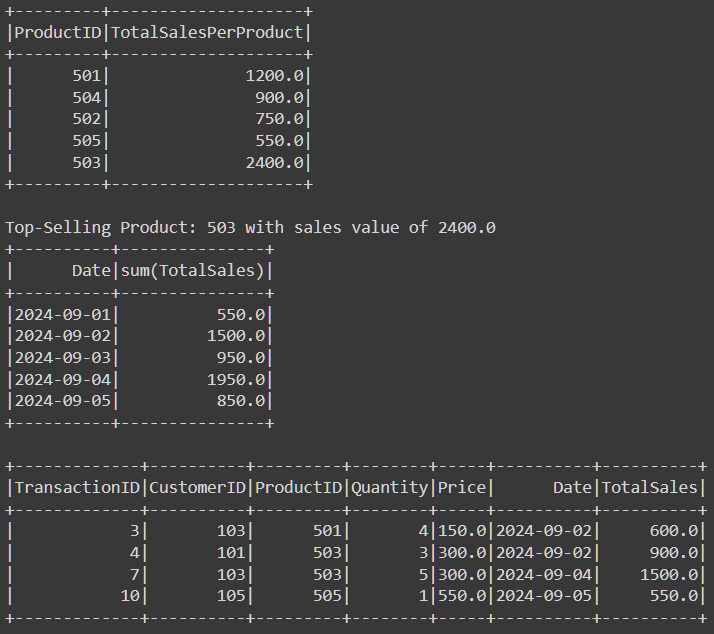
#3. Identify the Top-Selling Product:

from pyspark.sql.functions import desc

top\_selling\_product = sales\_df.groupBy('ProductID').sum('TotalSales').orderBy(desc('sum(TotalSales)')).first()

print(f"Top-Selling Product: {top\_selling\_product['ProductID']} with sales value of {top\_selling\_product['sum(TotalSales)']}")

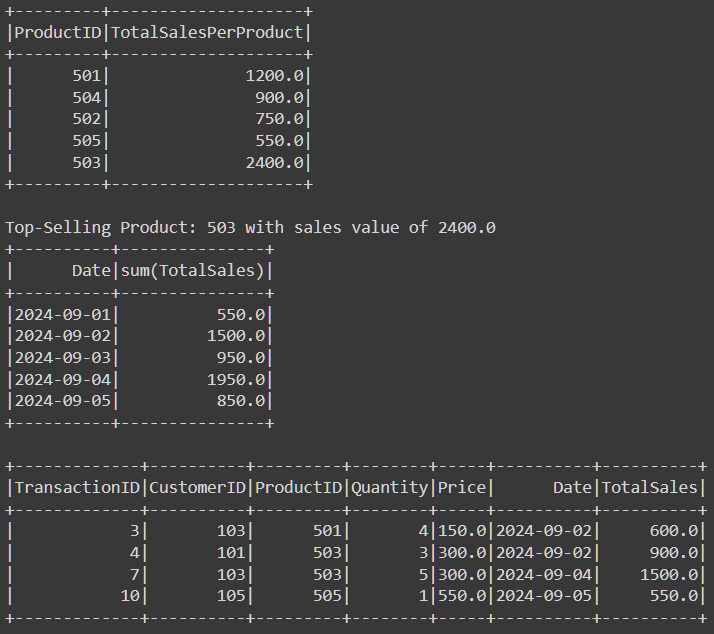
**OUTPUT:**



#4. Calculate the Total Sales by Date:

sales\_df.groupBy('Date').sum('TotalSales').orderBy('Date').show()

**OUTPUT:**

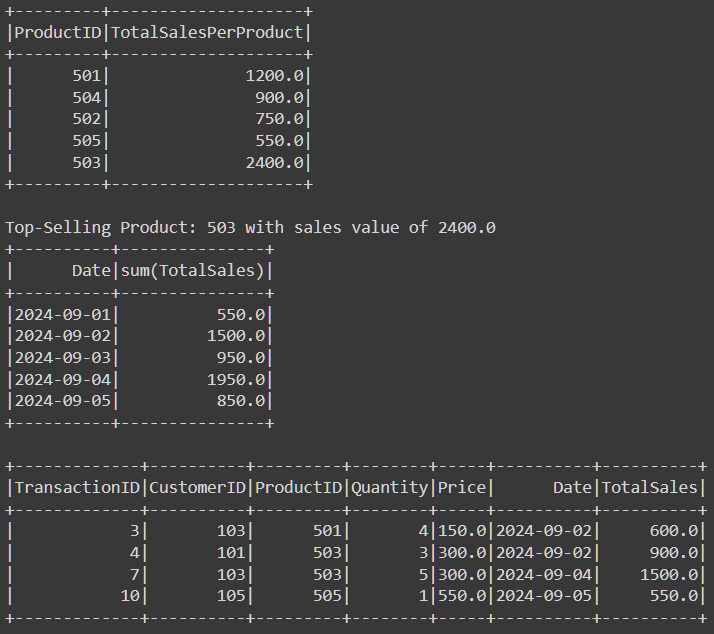


#5. Filter High-Value Transactions:

high\_value\_transactions = sales\_df.filter(col('TotalSales') > 500)

high\_value\_transactions.show()

**OUTPUT:**



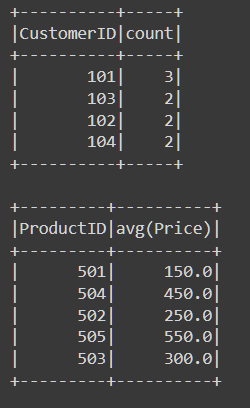
#Additional Challenge

#1. Identify Repeat Customers:

repeat\_customers = sales\_df.groupBy('CustomerID').count().filter(col('count') > 1)

repeat\_customers.show()

**OUTPUT:**



#2. Calculate the Average Sale Price Per Product:

average\_price\_per\_product = sales\_df.groupBy('ProductID').avg('Price')

average\_price\_per\_product.show()

**OUTPUT:**

