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)

       ]

   }

pandas\_df = pd.DataFrame(data)

pandas\_df.to\_csv("sales\_data.csv",index=False)

# 1. Initialize the SparkSession

spark = SparkSession.builder.appName("Sales Dataset Analysis").getOrCreate()

# 2. Load the CSV File into a PySpark DataFrame

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

spark\_df.show()

# Explore the Data

# 1. \*\*Print the Schema:\*\*

#    - Display the schema of the DataFrame to understand the data types.

spark\_df.printSchema()

# 2. \*\*Show the First Few Rows:\*\*

#    - Display the first 5 rows of the DataFrame.

spark\_df.show(5)

# 3. \*\*Get Summary Statistics:\*\*

#    - Get summary statistics for numeric columns (`Quantity` and `Price`).

spark\_df.describe("Quantity","Price").show()

#### \*\*Step 4: Perform Data Transformations and Analysis\*\*

# Perform the following tasks to analyze the data:

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

#    - Add a new column called `TotalSales`, calculated by multiplying `Quantity` by `Price`.

totalSales\_transaction\_df = spark\_df.withColumn("TotalSales", col("Quantity") \* col("Price"))

totalSales\_transaction\_df.show()

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

#    - Group the data by `ProductID` and calculate the total sales for each product.

groupProduct\_sales\_df = totalSales\_transaction\_df.groupBy("ProductID").sum("TotalSales").withColumnRenamed("sum(TotalSales)","TotalSales")

groupProduct\_sales\_df.show()

# 3. \*\*Identify the Top-Selling Product:\*\*

#    - Find the product that generated the highest total sales.

selling\_product\_df = groupProduct\_sales\_df.orderBy(col("TotalSales").desc())

selling\_product\_df.limit(1).show()

# 4. \*\*Calculate the Total Sales by Date:\*\*

#    - Group the data by `Date` and calculate the total sales for each day.

groupDate\_sales\_df = totalSales\_transaction\_df.groupBy("Date").sum("TotalSales").withColumnRenamed("sum(TotalSales)","TotalSales")

groupDate\_sales\_df.show()

# 5. \*\*Filter High-Value Transactions:\*\*

#    - Filter the transactions to show only those where the total sales value is greater than ₹500.

high\_value\_df = totalSales\_transaction\_df.filter(col("TotalSales") > 500)

high\_value\_df.show()

# Additional Challenge

# 1. \*\*Identify Repeat Customers:\*\*

#    - Count how many times each customer has made a purchase and display the customers who have made more than one purchase.

repeat\_customer\_df = spark\_df.groupBy("CustomerID").count().withColumnRenamed("count","PurchaseCount")

repeat\_customer\_df.filter(col("PurchaseCount") > 1).show()

# 2. \*\*Calculate the Average Sale Price Per Product:\*\*

#    - Calculate the average price per unit for each product and display the results.

avg\_price\_product\_df = spark\_df.groupBy("ProductID").avg("Price").withColumnRenamed("avg(Price)","AveragePrice")

avg\_price\_product\_df.show()