Hands on exercise on pyspark:

from pyspark.sql import SparkSession

from pyspark.sql.functions import col

# Initialize SparkSession

spark = SparkSession.builder \

.appName("Product Sales Analysis") \

.getOrCreate()

# Sample data for products

products = [

(1, "Laptop", "Electronics", 50000),

(2, "Smartphone", "Electronics", 30000),

(3, "Table", "Furniture", 15000),

(4, "Chair", "Furniture", 5000),

(5, "Headphones", "Electronics", 2000),

]

# Sample data for sales transactions

sales = [

(1, 1, 2),

(2, 2, 1),

(3, 3, 3),

(4, 1, 1),

(5, 4, 5),

(6, 2, 2),

(7, 5, 10),

(8, 3, 1),

]

# Define schema for DataFrames

product\_columns = ["ProductID", "ProductName", "Category", "Price"]

sales\_columns = ["SaleID", "ProductID", "Quantity"]

# Create DataFrames

product\_df = spark.createDataFrame(products, schema=product\_columns)

sales\_df = spark.createDataFrame(sales, schema=sales\_columns)

# Show the DataFrames

print("Products DataFrame:")

product\_df.show()

print("Sales DataFrame:")

sales\_df.show()

#### \*\*Step 2: Perform the Following Tasks\*\*

#1. \*\*Join the DataFrames:\*\*

#- Join the `product\_df` and `sales\_df` DataFrames on `ProductID` to create a combined DataFrame with product and sales data.

combined\_df = product\_df.join(sales\_df, on="ProductID")

print("Combined DataFrame:")

combined\_df.show()

#2. \*\*Calculate Total Sales Value:\*\*

# - For each product, calculate the total sales value by multiplying the price by the quantity sold.

total\_sales\_value\_df = combined\_df.withColumn("TotalSalesValue", col("Price") \* col("Quantity"))

print("Total Sales Value DataFrame:")

total\_sales\_value\_df.show()

#3. \*\*Find the Total Sales for Each Product Category:\*\*

# - Group the data by the `Category` column and calculate the total sales value for each product category.

total\_sales\_by\_category\_df = total\_sales\_value\_df.groupBy("Category").agg(\_sum("TotalSalesValue").alias("TotalSalesByCategory"))

print("Total Sales by Category DataFrame:")

total\_sales\_by\_category\_df.show()

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

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

top\_selling\_product\_df = total\_sales\_value\_df.groupBy("ProductID", "ProductName").agg(\_sum("TotalSalesValue").alias("TotalSalesValue")).orderBy(col("TotalSalesValue").desc()).limit(1)

print("Top-Selling Product:")

top\_selling\_product\_df.show()

#5. \*\*Sort the Products by Total Sales Value:\*\*

# - Sort the products by total sales value in descending order.

sorted\_products\_df = total\_sales\_value\_df.groupBy("ProductID", "ProductName").agg(\_sum("TotalSalesValue").alias("TotalSalesValue")).orderBy(col("TotalSalesValue").desc())

print("Sorted Products by Total Sales Value:")

sorted\_products\_df.show()

#6. \*\*Count the Number of Sales for Each Product:\*\*

# - Count the number of sales transactions for each product.

sales\_count\_df = sales\_df.groupBy("ProductID").count()

print("Sales Count for Each Product:")

sales\_count\_df.show()

#7. \*\*Filter the Products with Total Sales Value Greater Than ₹50,000:\*\*

# - Filter out the products that have a total sales value greater than ₹50,000.

filtered\_products\_df = sorted\_products\_df.filter(col("TotalSalesValue") > 50000)

print("Filtered Products with Total Sales Value Greater Than ₹50,000:")

filtered\_products\_df.show()