from pyspark.sql import SparkSession

def customerData():

data = [

(1, 101, 5001, 'Laptop', 'Electronics', 1000.0, 1),

(2, 102, 5002, 'Headphones', 'Electronics', 50.0, 2),

(3, 101, 5003, 'Book', 'Books', None, 3),

(4, 103, 5004, 'Laptop', 'Electronics', 1000.0, 1),

(5, 102, 5005, 'Chair', 'Furniture', None, 1)

]

columns = ["transaction\_id", "customer\_id", "product\_id", "product\_name", "category", "price", "quantity"]

# Initialize SparkSession

spark = SparkSession.builder.appName("E-Commerce Analysis").getOrCreate()

# Create DataFrame

df = spark.createDataFrame(data, columns)

# print(df.show())

# Filter transactions where quantity is greater than 1

#df\_filtered = df.filter(df.quantity > 1)

#df\_filtered.show()

# Filling null values in price column with the average price

#average\_price = df.selectExpr("avg(price)").collect()[0][0]

#df\_filled = df.na.fill({"price": average\_price})

#print(df\_filled.show())

# Drop duplicate rows based on customer\_id and product\_id

#df\_no\_duplicates = df.dropDuplicates(["customer\_id", "product\_id"])

#df\_no\_duplicates.show()

# Select specific columns

#df\_selected = df.select("customer\_id", "product\_name", "price")

#df\_selected.show()

# Calculate the total spending per customer

df\_grouped = df.groupBy("customer\_id").agg({"price": "sum"})

df\_grouped.show()

# Assume we have another DataFrame with customer details

customer\_data = [

(101, "John Doe", "john@example.com"),

(102, "Jane Smith", "jane@example.com"),

(103, "Alice Johnson", "alice@example.com")

]

customer\_columns = ["customer\_id", "customer\_name", "email"]

df\_customers = spark.createDataFrame(customer\_data, customer\_columns)

# Join on customer\_id

#df\_joined = df.join(df\_customers, on="customer\_id", how="inner")

#df\_joined.show()

# Create another DataFrame with similar schema

new\_data = [

(6, 104, 5006, 'Table', 'Furniture', 200.0, 1)

]

df\_new = spark.createDataFrame(new\_data, columns)

# Union the DataFrames

#df\_union = df.union(df\_new)

#df\_union.show()

# Create a temporary view

#df.createOrReplaceTempView("transactions")

# Run SQL query

#sql\_result = spark.sql(

# "SELECT customer\_id, SUM(price \* quantity) as total\_spent FROM transactions GROUP BY customer\_id")

#sql\_result.show()

if \_\_name\_\_ == '\_\_main\_\_':

customerData()