



**PySpark**  
Learning Hub | Practice Problem



**Akash Mahindrakar**  
Data Engineer  
akashsjce8050@gmail.com

## Step - 1 : Problem Statement

### 14\_Purchasing Activity by Product Type

We have been given purchasing activity DF and we need to find out cumulative purchases of each product over time.

**Difficult Level :** EASY

**DataFrame:**

```
# Define schema for the DataFrame
schema = StructType([
    StructField("order_id", IntegerType(), True),
    StructField("product_type", StringType(), True),
    StructField("quantity", IntegerType(), True),
    StructField("order_date", StringType(), True),
])

# Define data
# Define data
data = [
    (213824, 'printer', 20, "2022-06-27 "),
    (212312, 'hair dryer', 5, "2022-06-28 "),
    (132842, 'printer', 18, "2022-06-28 "),
    (284730, 'standing lamp', 8, "2022-07-05 ")
]
```

## Step - 2 : Identifying The Input Data And Expected

### INPUT

INPUT			
ORDER_ID	PRODUCT_TYPE	QUANTITY	ORDER_DATE
213824	printer	20	2022-06-27 12:00:00
212312	hair dryer	5	2022-06-28 12:00:00
132842	printer	18	2022-06-28 12:00:00
284730	standing lamp	8	2022-07-05 12:00:00

### OUTPUT

OUTPUT		
ORDER_DATE	PRODUCT_TYPE	CUM_PURCHASED
2022-06-27 12:00:00	printer	20
2022-06-28 12:00:00	hair dryer	5
2022-06-28 12:00:00	printer	38
2022-07-05 12:00:00	standing lamp	8

## Step - 3 : Writing the pyspark code to solve

### # Creating Spark Session

```
from pyspark.sql import SparkSession
from pyspark.sql.types import
StructType, StructField, IntegerType, StringType
```

### #creating spark session

```
spark = SparkSession. \
builder. \
config('spark.shuffle.useOldFetchProtocol', 'true'). \
config('spark.ui.port','0'). \
config("spark.sql.warehouse.dir", "/user/itv008042/warehouse"). \
enableHiveSupport(). \
master('yarn'). \
getOrCreate()
```

### # Define schema for the DataFrame

```
schema = StructType([
    StructField("order_id", IntegerType(), True),
    StructField("product_type", StringType(), True),
    StructField("quantity", IntegerType(), True),
    StructField("order_date", StringType(), True),
])
```

### # Define data

### # Define data

```
data = [
    (213824, 'printer', 20, "2022-06-27 "),
    (212312, 'hair dryer', 5, "2022-06-28 "),
    (132842, 'printer', 18, "2022-06-28 "),
    (284730, 'standing lamp', 8, "2022-07-05 ")
]
```

## PYSPARK LEARNING HUB : DAY - 14

]

```
order_df=spark.createDataFrame(data,schema)
order_df.show()
```

```
+-----+-----+-----+-----+
|order_id| product_type|quantity| order_date|
+-----+-----+-----+-----+
|  213824|      printer|      20|2022-06-27|
|  212312|   hair dryer|       5|2022-06-28|
|  132842|      printer|      18|2022-06-28|
|  284730|standing lamp|       8|2022-07-05|
+-----+-----+-----+-----+
```

```
# Define a Window specification based on the 'order_date' column
window_spec =
Window.partitionBy("product_type").orderBy("order_date").rowsBe-
tween(Window.unboundedPreceding, 0)
```

```
# Add a new column 'cumulative_purchases' representing the
cumulative sum
```

```
result_df = order_df.withColumn("cumulative_purchases",
F.sum("quantity").over(window_spec))
result_df.show()
```

```
+-----+-----+-----+-----+-----+
|order_id| product_type|quantity| order_date|cumulative_purchases|
+-----+-----+-----+-----+-----+
|  284730|standing lamp|       8|2022-07-05|          8|
|  212312|   hair dryer|       5|2022-06-28|          5|
|  213824|      printer|      20|2022-06-27|         20|
|  132842|      printer|      18|2022-06-28|         38|
+-----+-----+-----+-----+-----+
```



Save

**Was it  
helpful?**  
follow for more!



**Akash Mahindrakar**

Data Engineer

akashsjce8050@gmail.com



Comment

**SHARE YOUR THOUGHTS  
IN COMMENT BELOW**



Share