

Assignment: Structured Streaming and Transformations on Streams

SaiPrabath Chowdary S

```
10: folder for streaming

# folder for streaming
csv_path = 'file:/Workspace/Shared/assignment17sep/transactions.csv'
streaming_path = 'dbfs:/FileStore/assignment17sep/streaming/input/'
dbutils.fs.cp(csv_path, f"{streaming_path}transactions.csv")

True

11: Set up a structured streaming source

from pyspark.sql import SparkSession
from pyspark.sql.functions import col

spark = SparkSession.builder.appName("Streaming").getOrCreate()

transaction_schema = "TransactionID STRING, TransactionDate DATE, ProductID STRING, Quantity INT, Price DOUBLE"
transaction_stream_df = spark.readStream.format("csv").option("header", True).schema(transaction_schema) \
    .load("dbfs:/FileStore/assignment17sep/streaming/input/")

transaction_stream_df.printSchema()

transaction_stream_df: pyspark.sql.dataframe.DataFrame = [TransactionID: string, TransactionDate: date ... 3 more fields]
root
|-- TransactionID: string (nullable = true)
|-- TransactionDate: date (nullable = true)
|-- ProductID: string (nullable = true)
|-- Quantity: integer (nullable = true)
|-- Price: double (nullable = true)
```

12: transformation Python

```
transformed_stream_df = transaction_stream_df.withColumn("TotalAmount", transaction_stream_df["Quantity"] * transaction_stream_df["Price"]).filter(
(transaction_stream_df["Quantity"] > 1)

query = transformed_stream_df.writeStream \
    .format("memory") \
    .queryName("transformed_data") \
    .outputMode("append") \
    .start()

# To view the data from memory
spark.sql("SELECT * FROM transformed_data").show()
```

(1) Spark Jobs

transformed_data (id: 56300314-ad54-4d9c-96ac-19695abda93c) Last updated: 1 hour ago

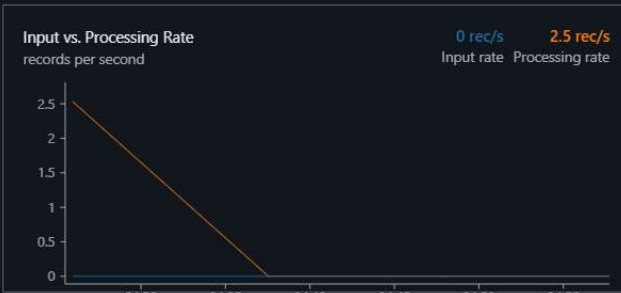
Dashboard Raw Data

Input vs. Processing Rate

records per second

0 rec/s 2.5 rec/s

Input rate Processing rate



Batch Duration

in milliseconds

480.2 ms 1178 ms

Average Latest

