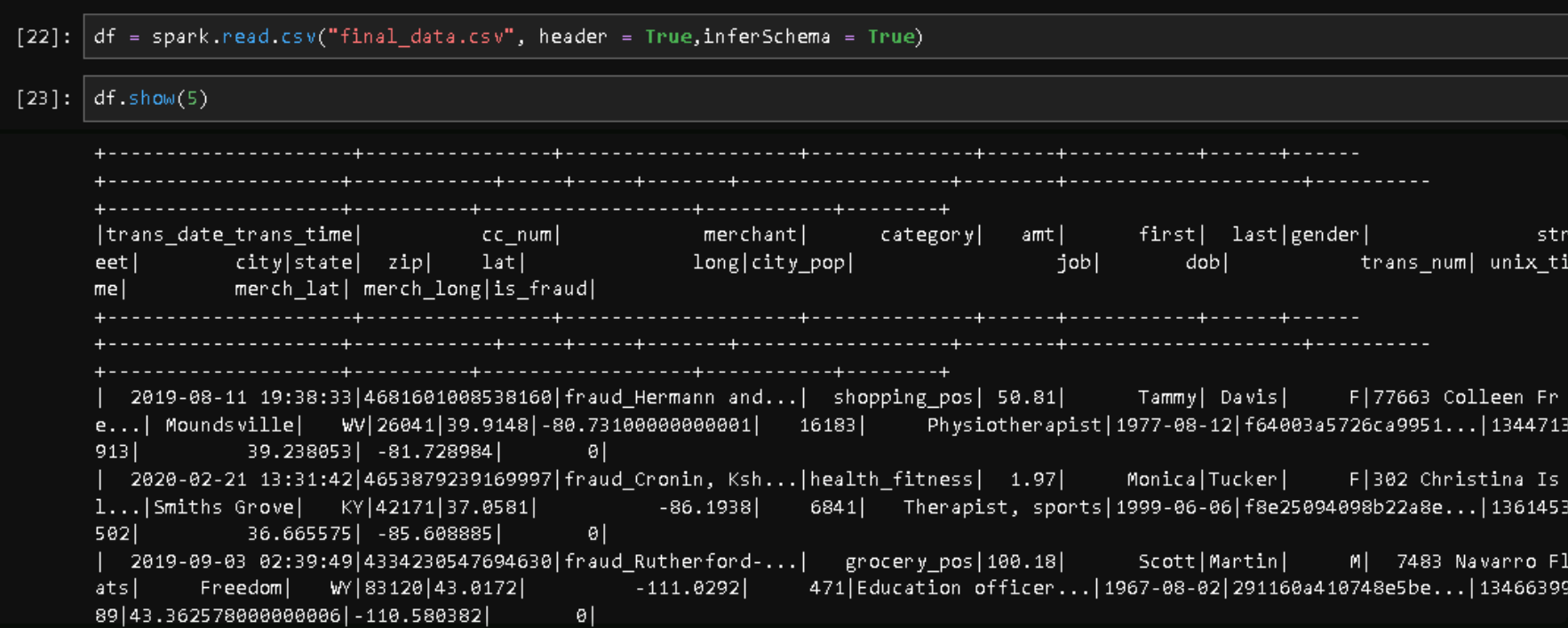
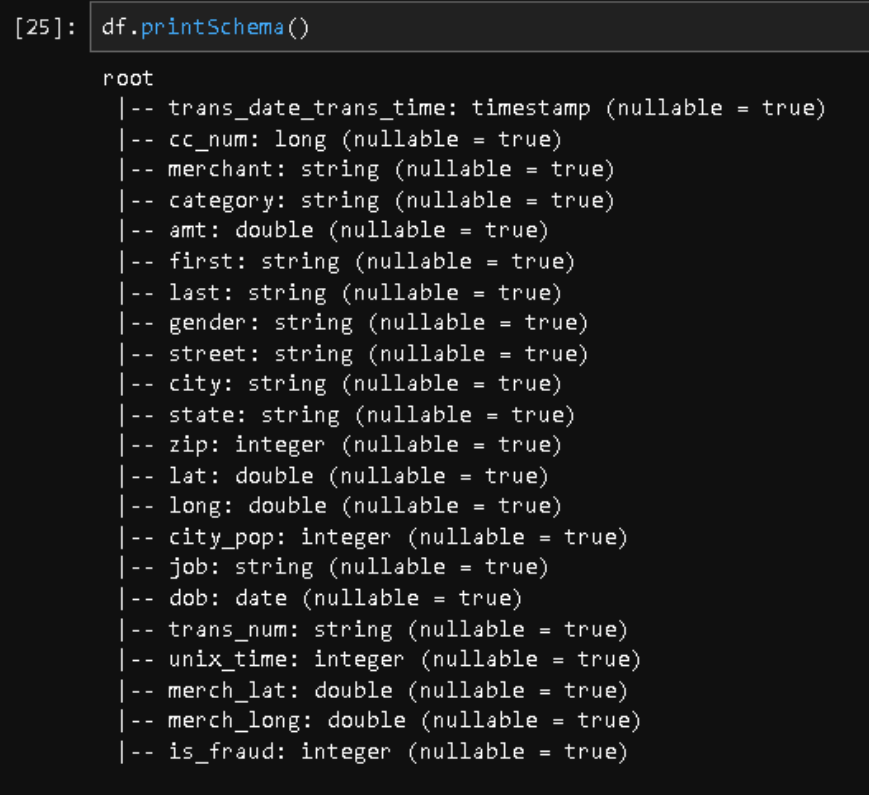
# PySpark Assessment

TEAM B

# Fraud Transaction Data



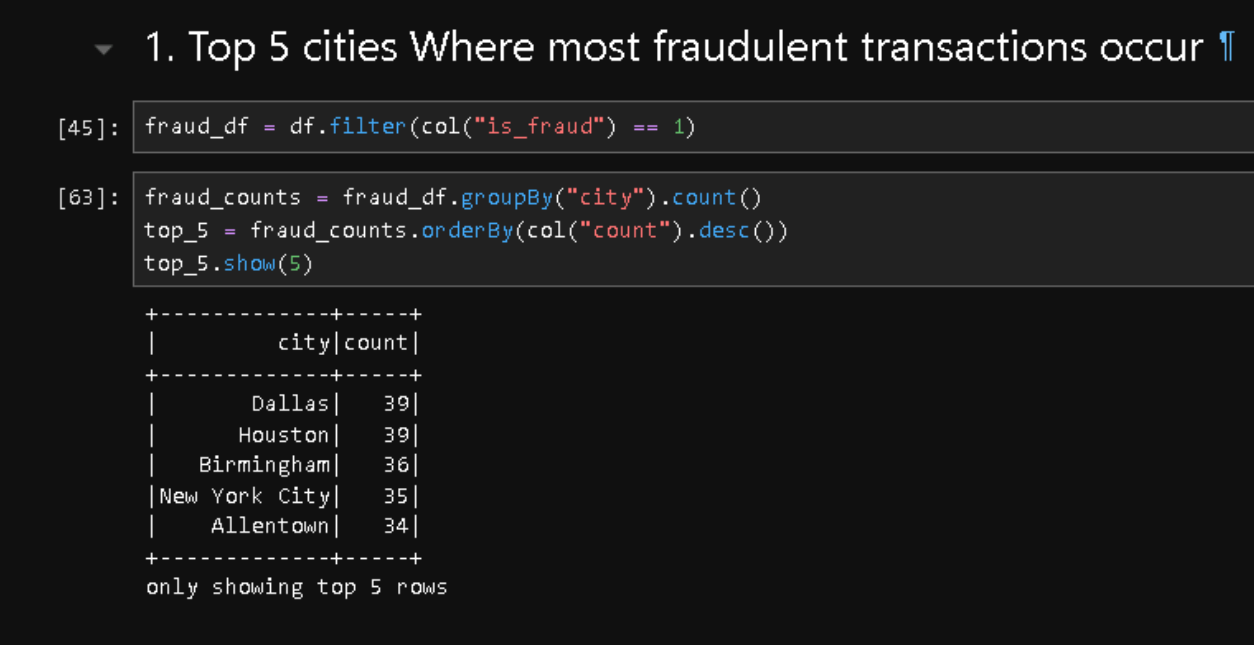
## Schema

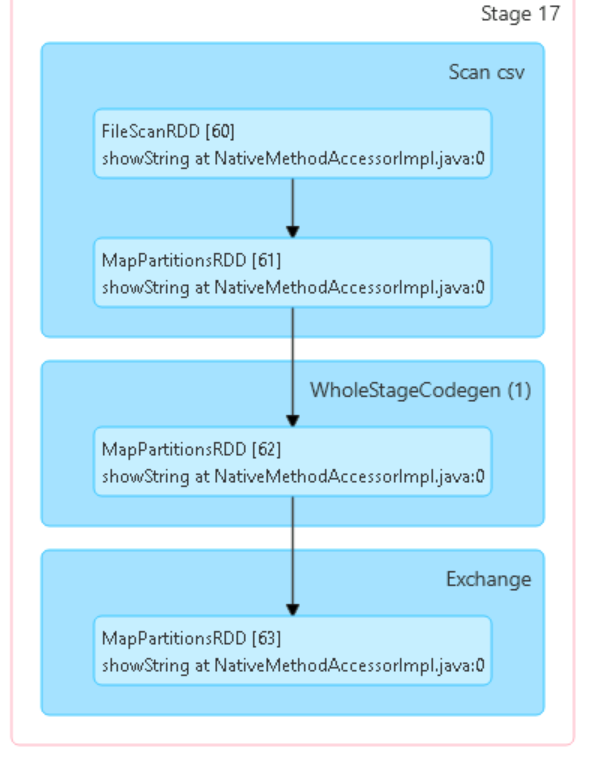


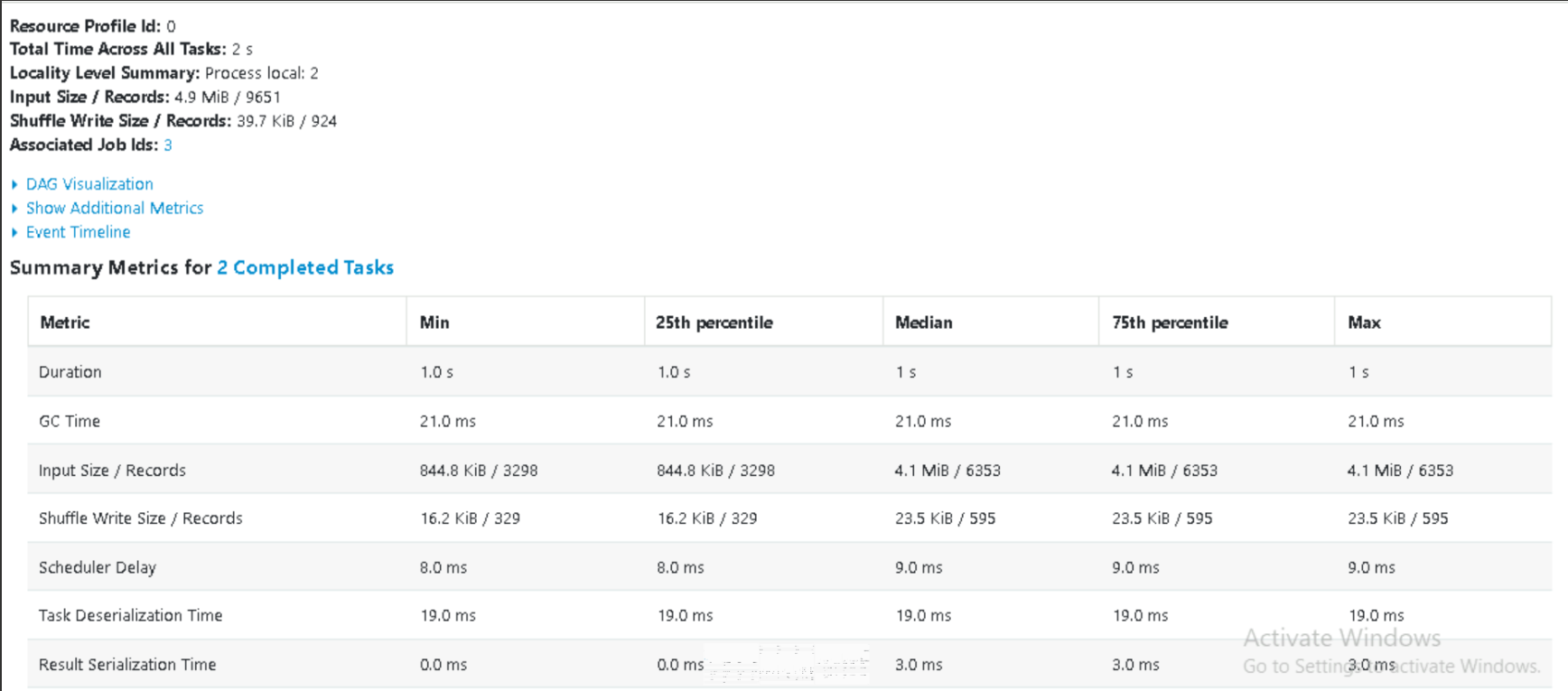
Coding

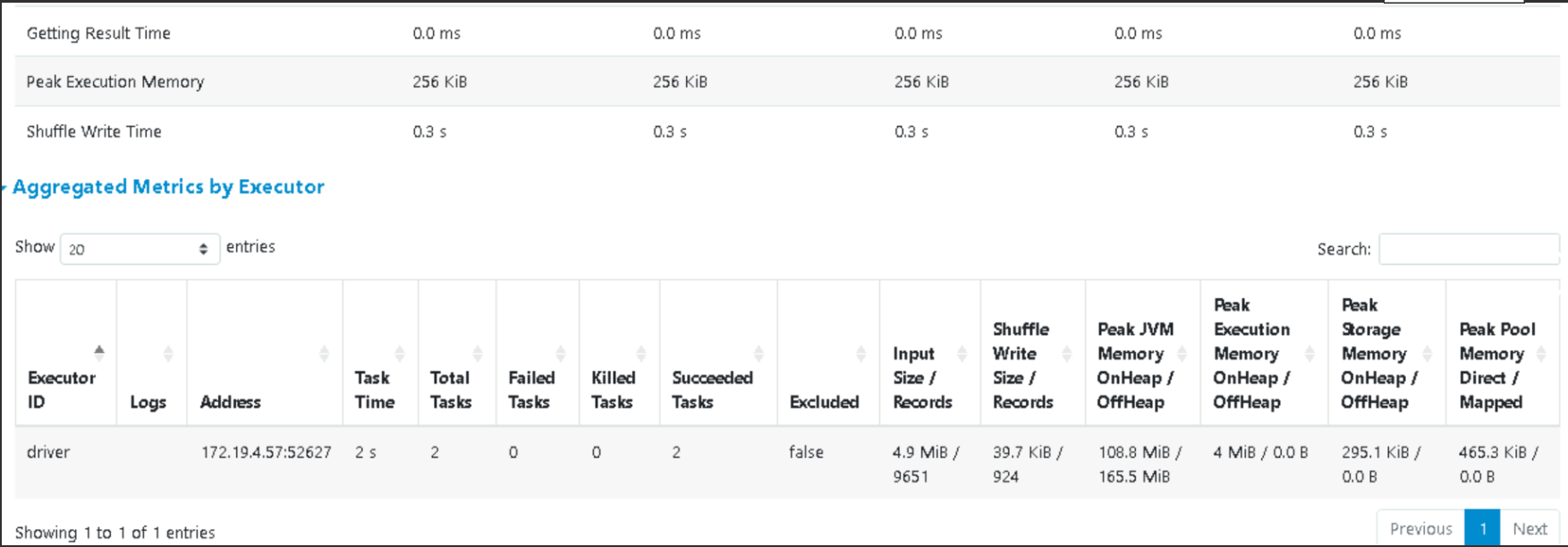
* **filter(col("is\_fraud") == 1)** : Filters fraudulent transactions.
* **groupBy("column\_name").count()** : Counts occurrences per group.
* **groupBy("column\_name").agg(avg(col("amt")).alias("avg\_amount\_spent")):** Calculates the average amount spent per group.
* **orderBy(col("count").desc()):** Orders results in descending order.
* **to\_date(col("dob"), "yyyy-MM-dd HH:mm:ss") :** Converts a date string into a proper date format.
* **year(col("dob")) :** Extracts the year from a date column.
* **current\_date() :** Gets the current date.
* **month(col("trans\_date\_trans\_time")) :** Extracts the month from the transaction timestamp.
* **withColumn("age\_group", …)** : This function is used to add a new column "age\_group" to the DataFrame. It applies conditional logic to categorize customers into age groups.
* **when(condition, value) :** This function is used for conditional expressions in PySpark.
* **col("age").between(start, end) :** This checks whether the value in "age" falls within a specified range.

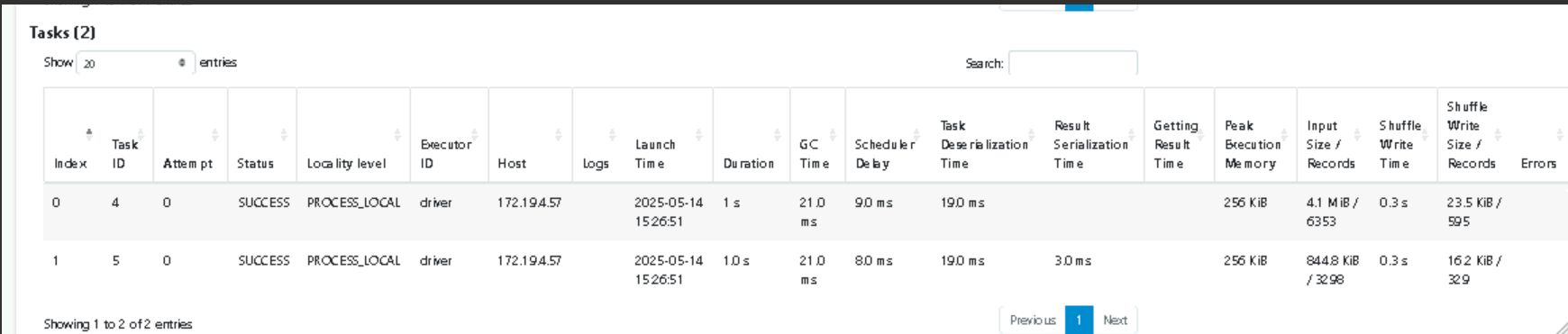
1.



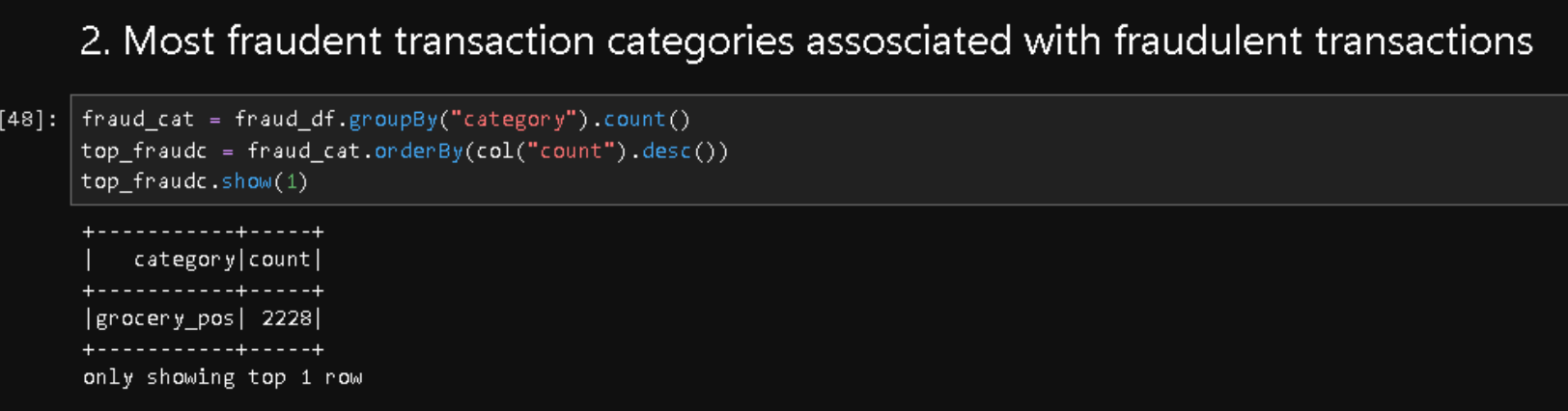


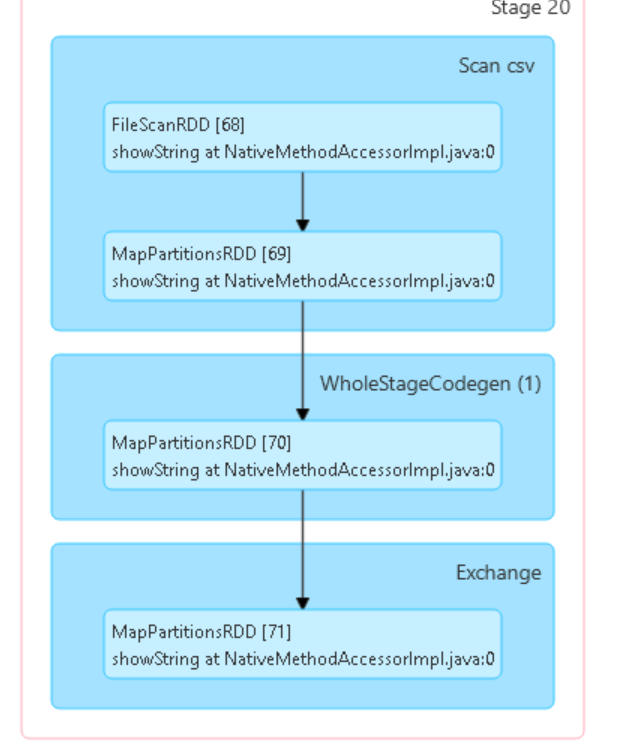


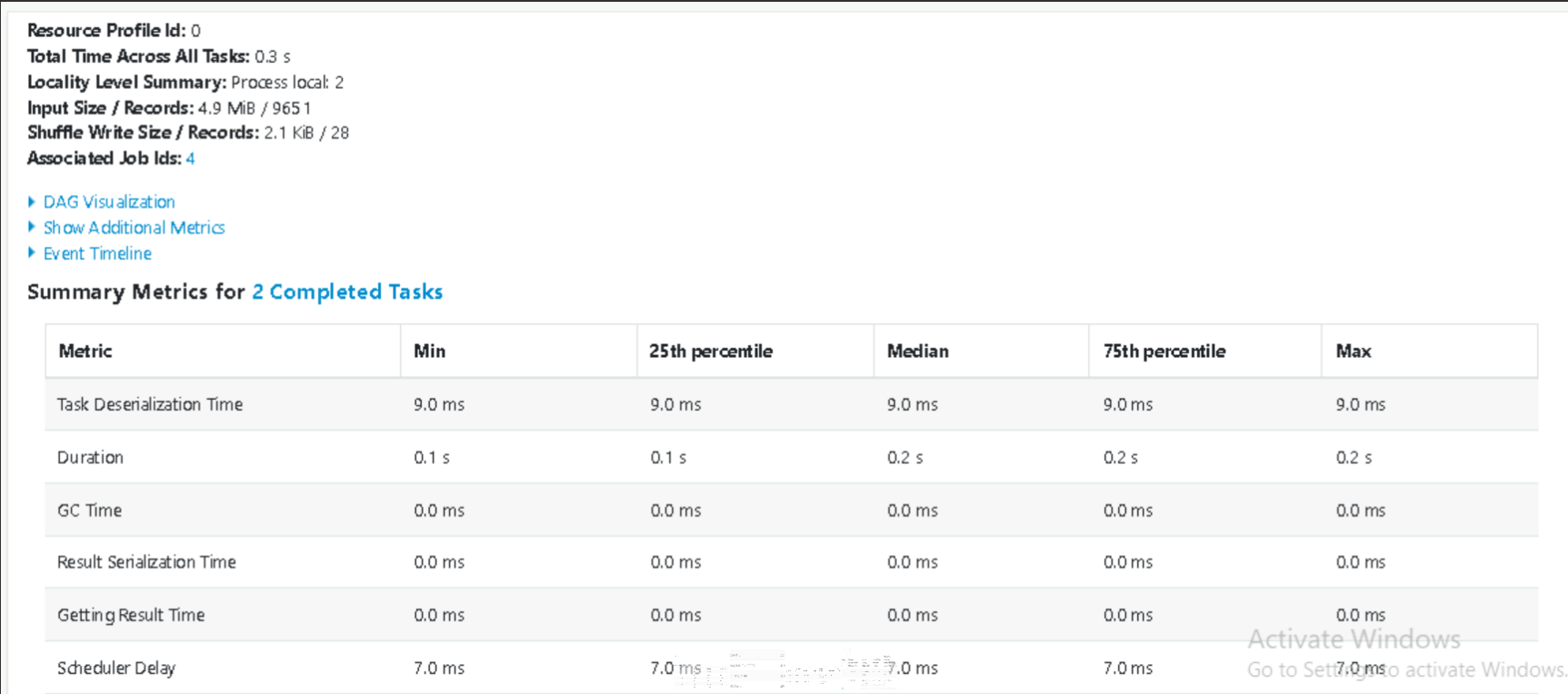


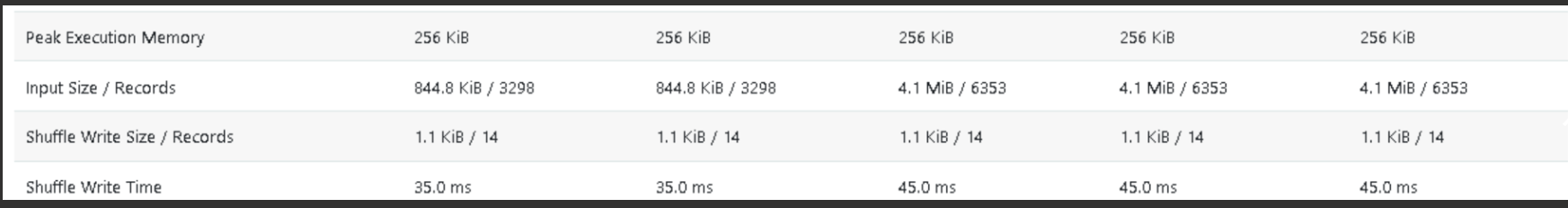


2.

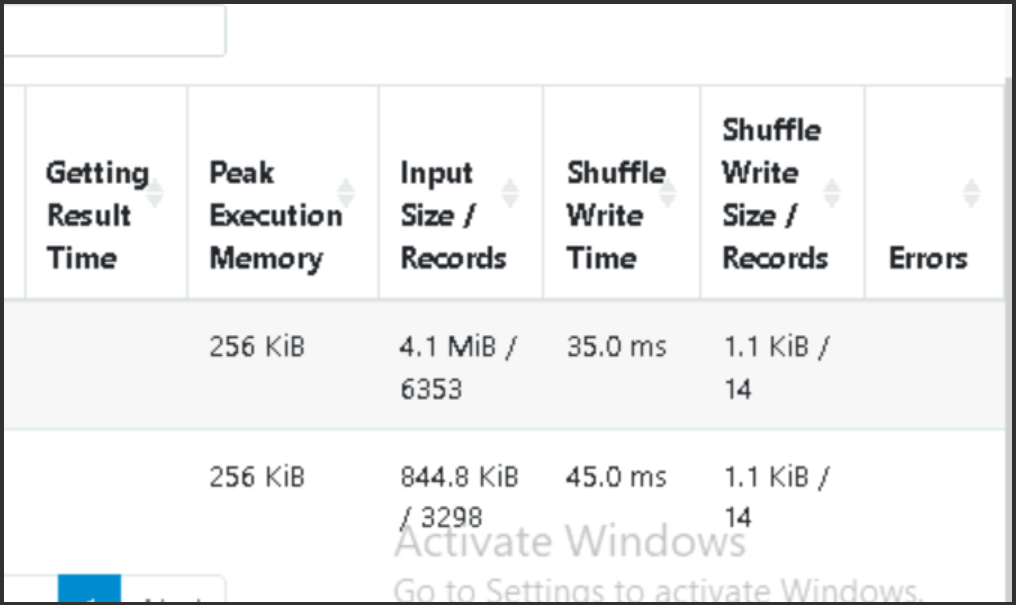




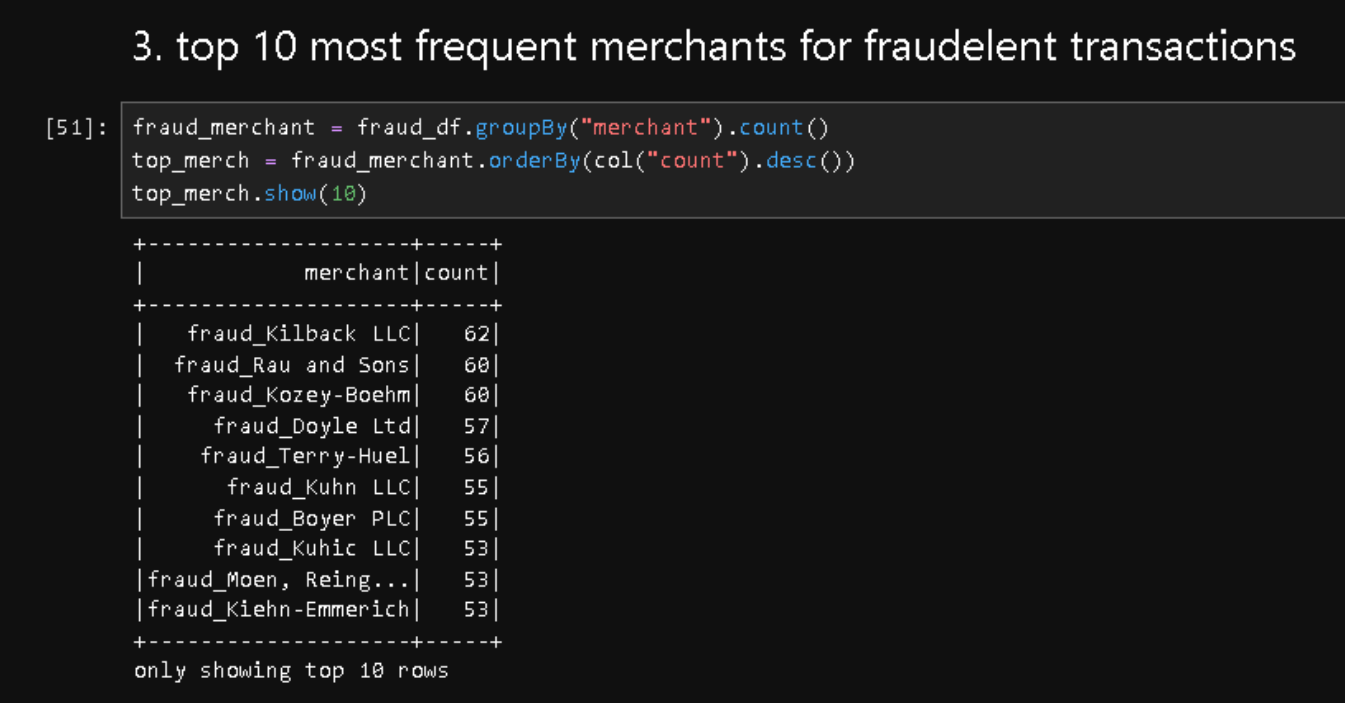


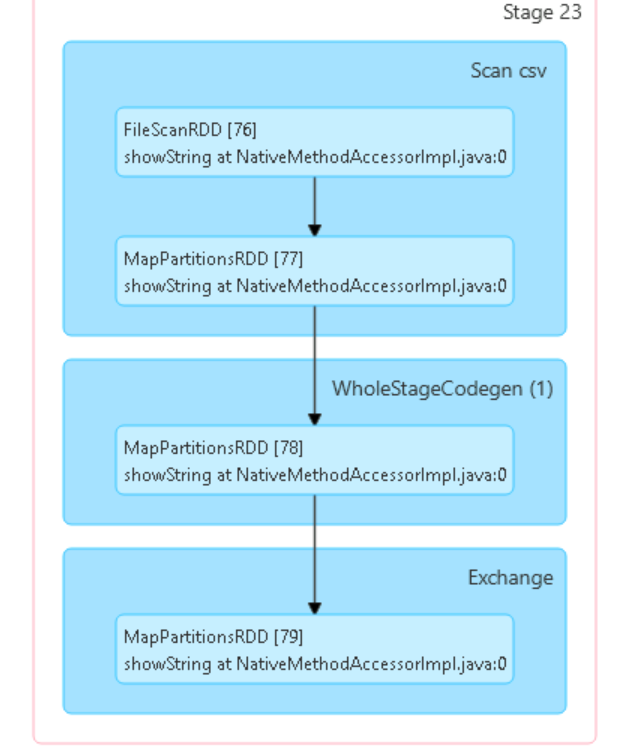


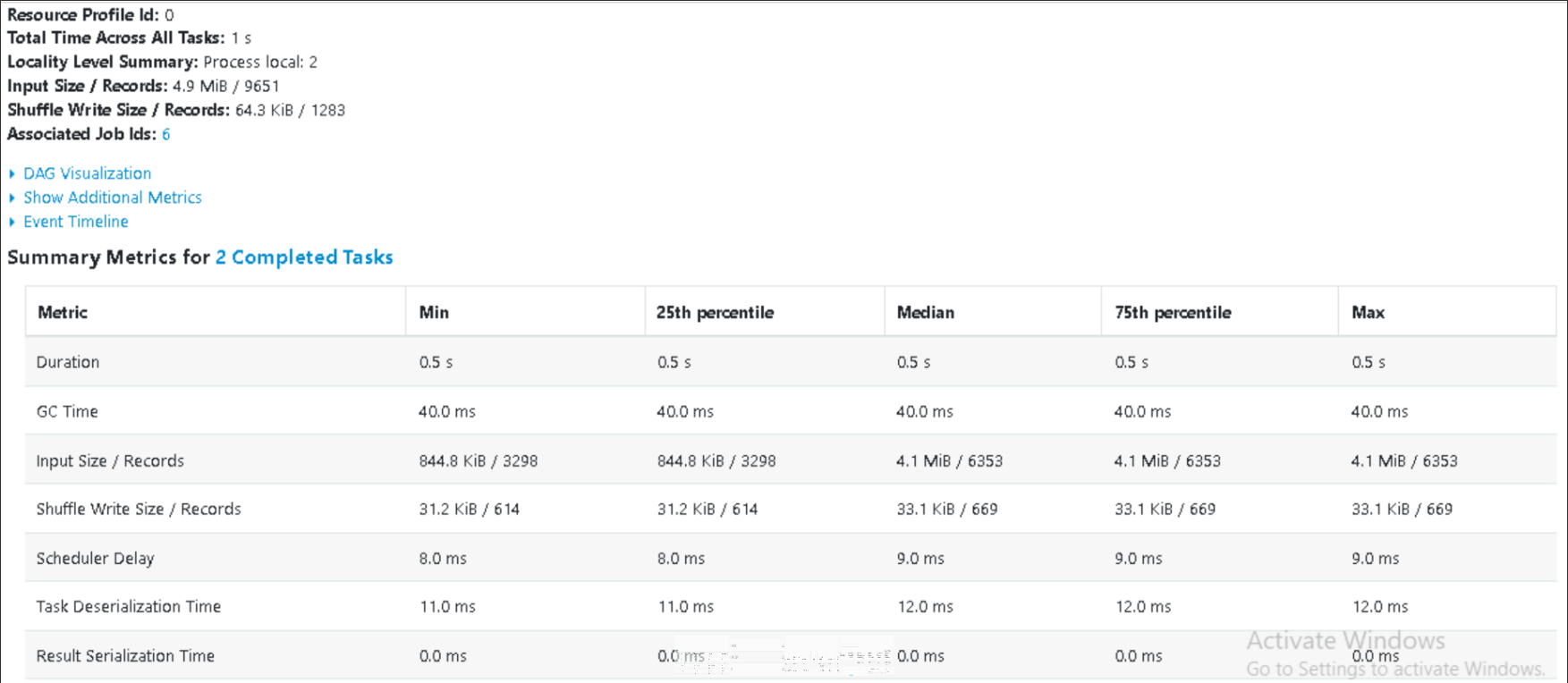


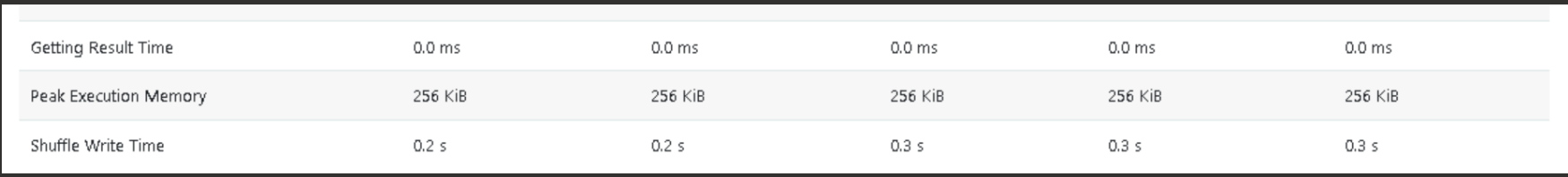


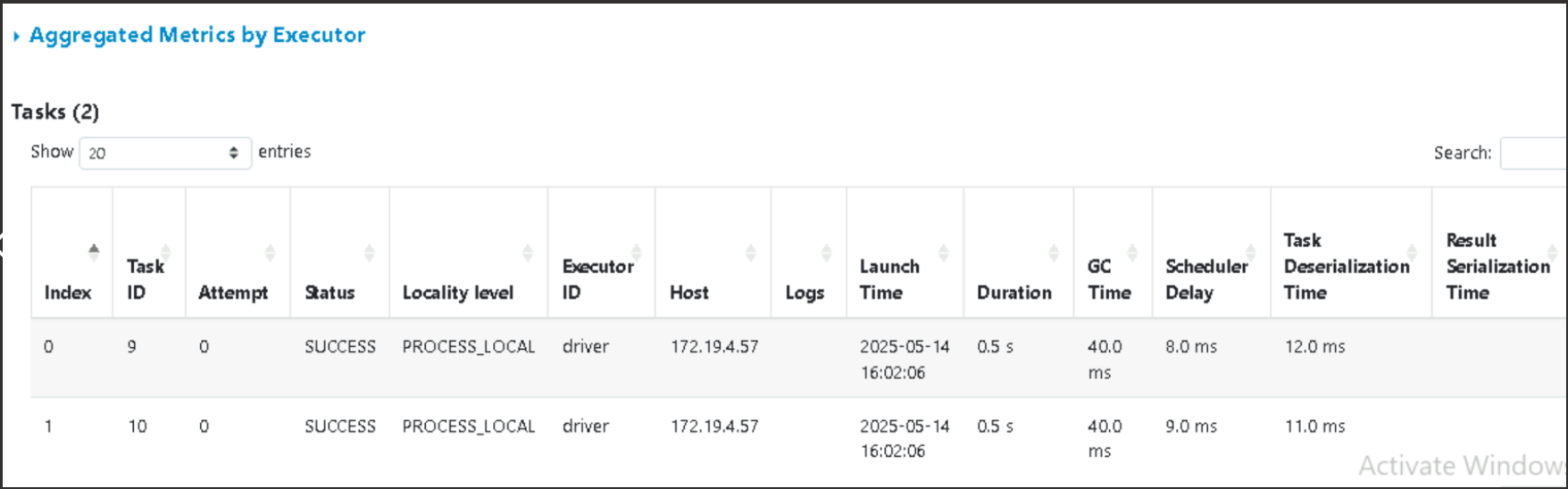
3.

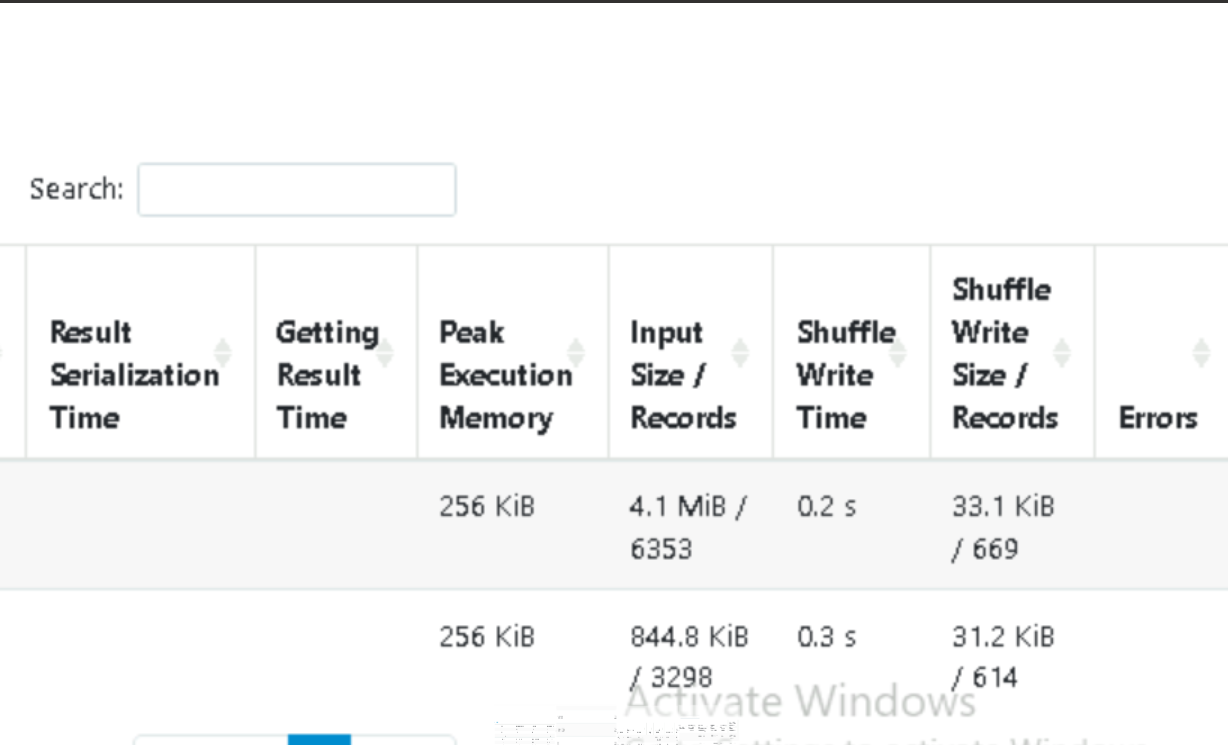




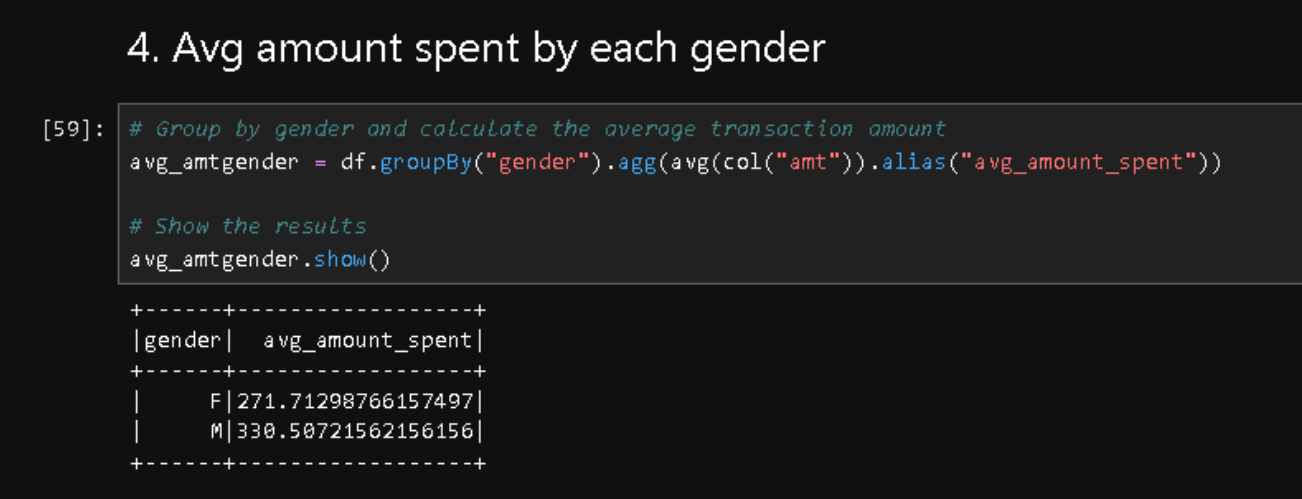


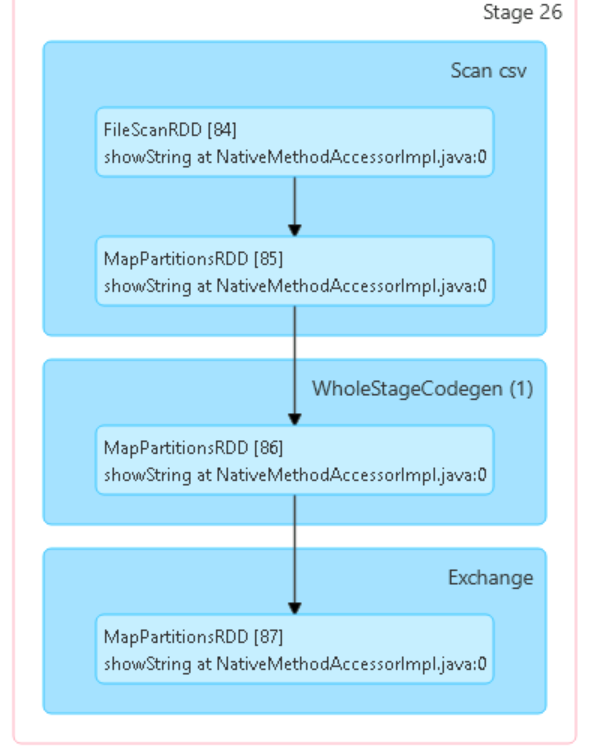


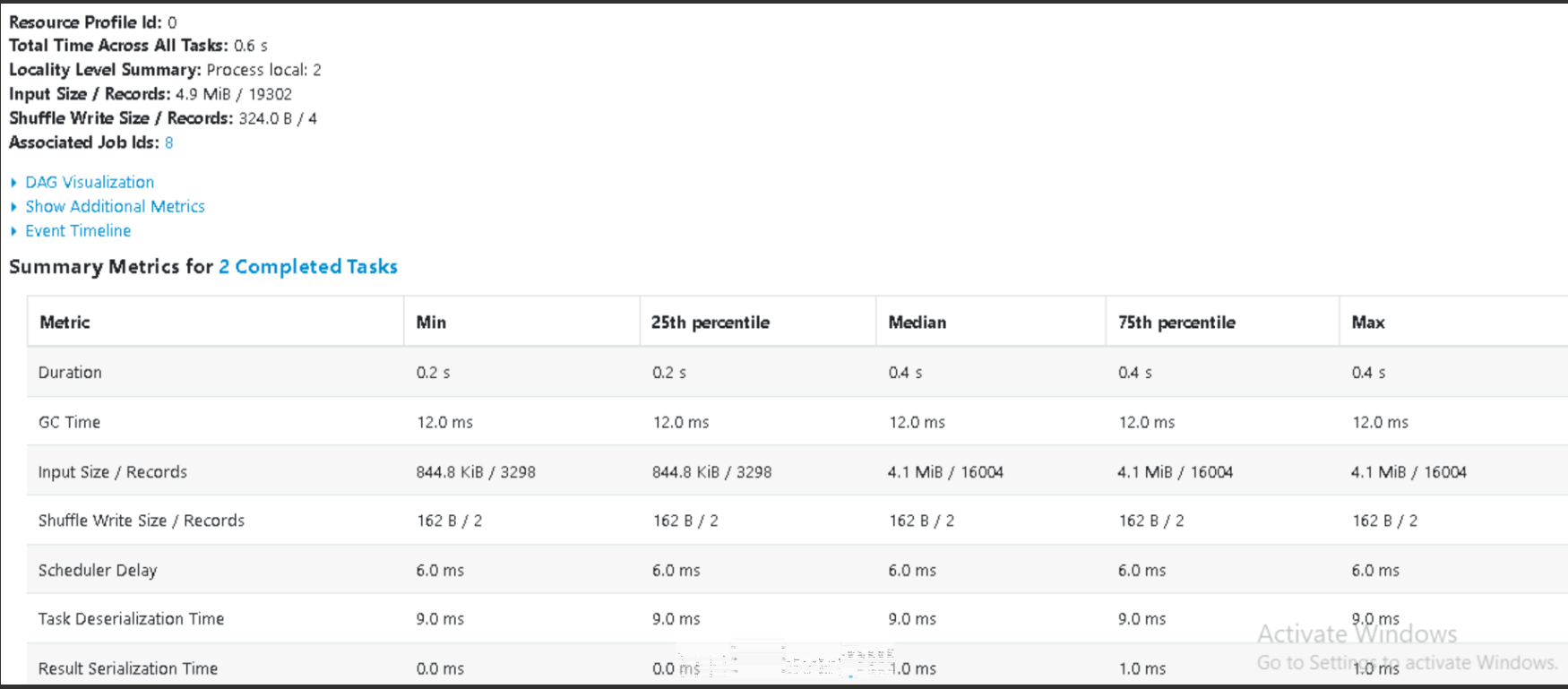




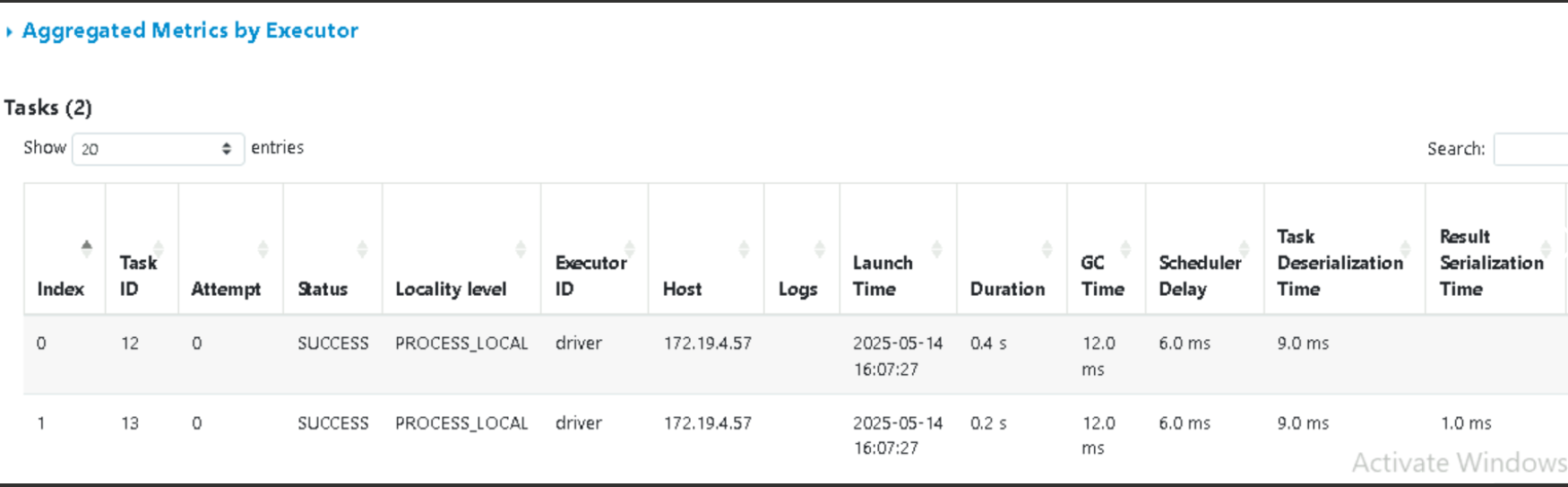
4.

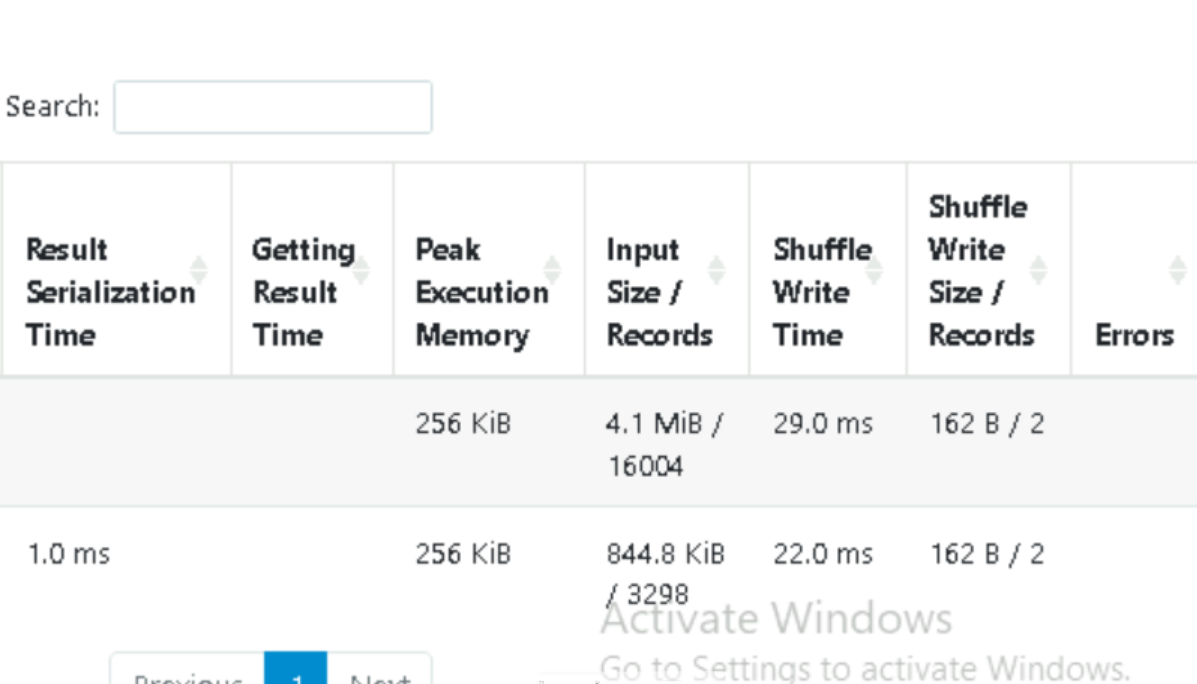




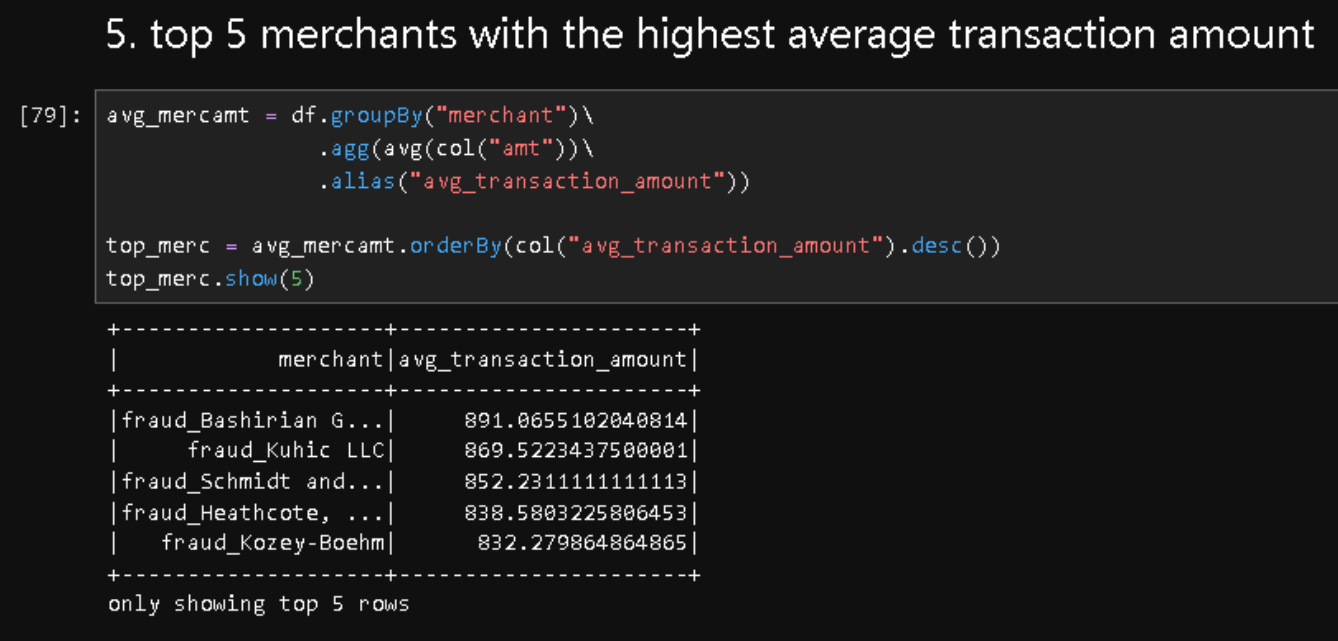


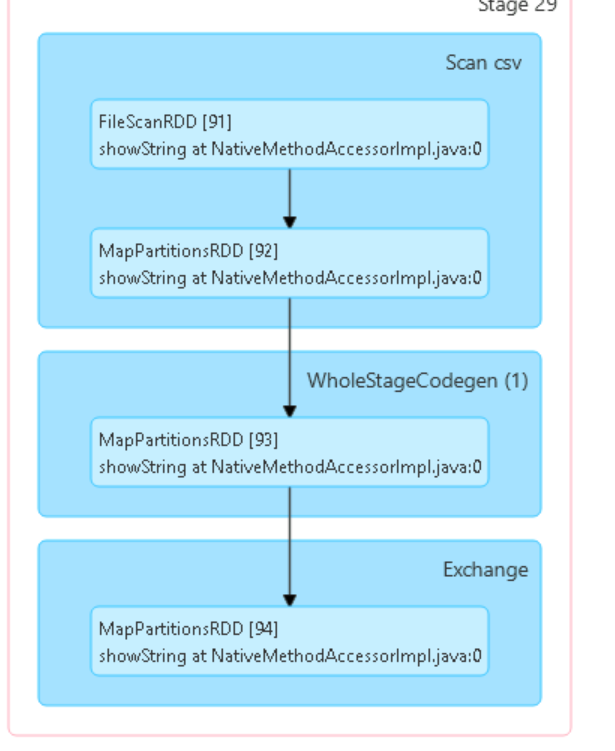


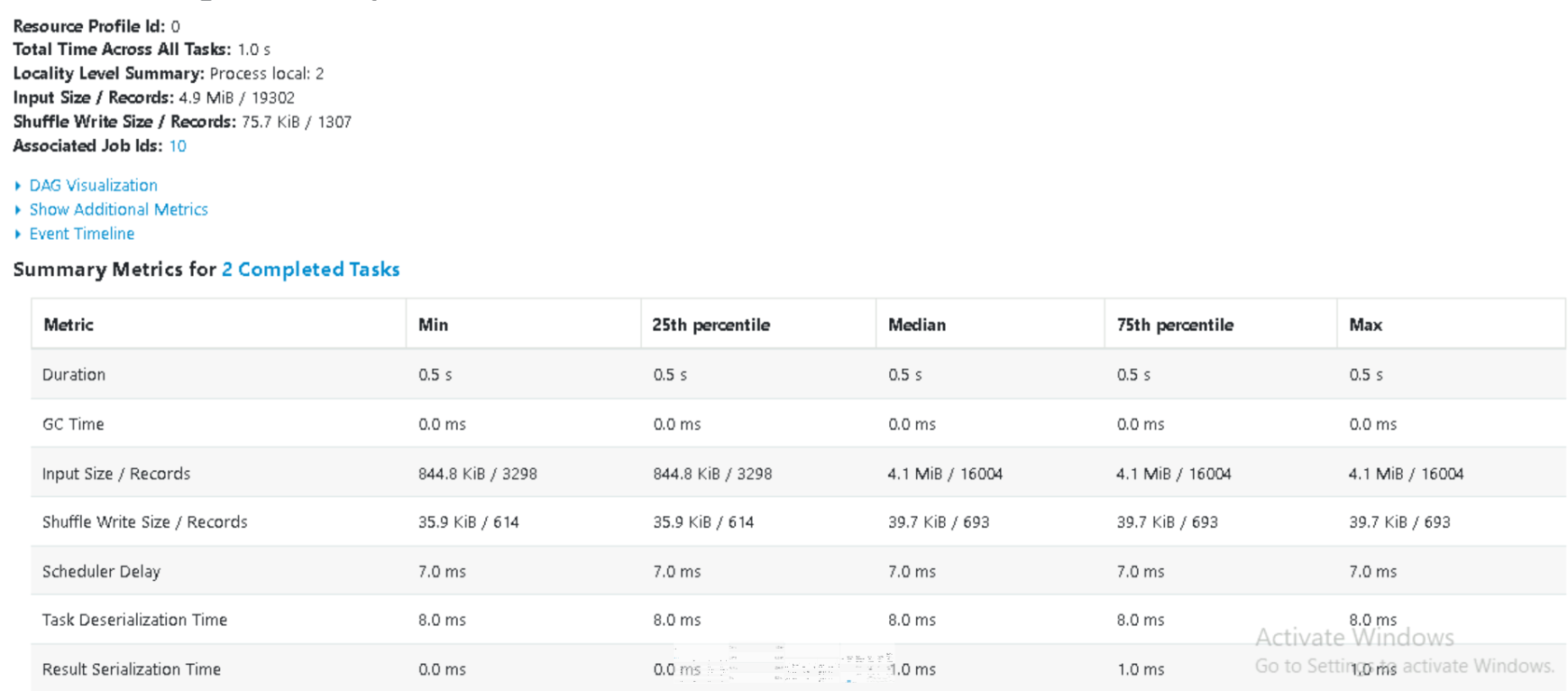




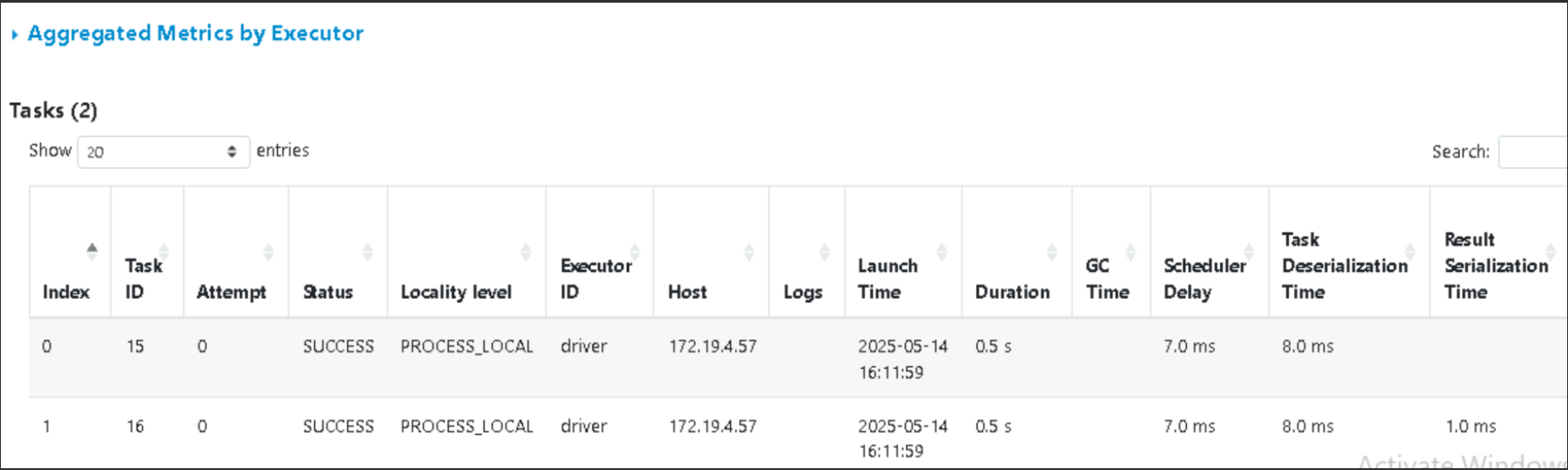
5.

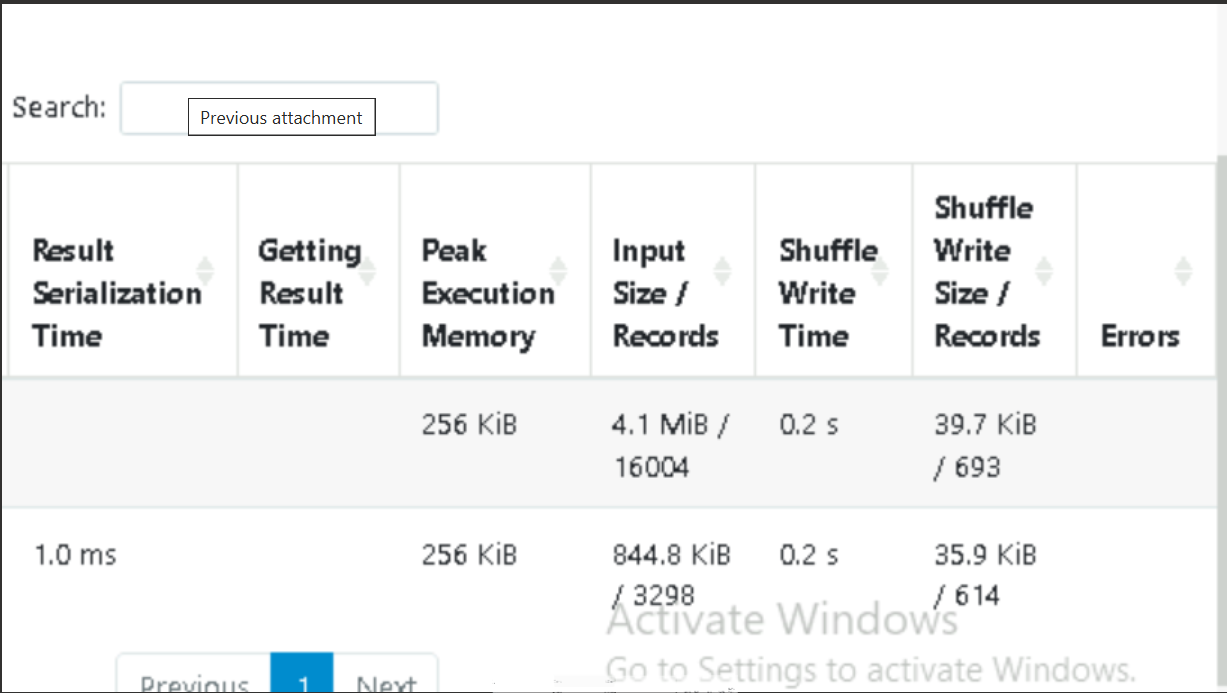




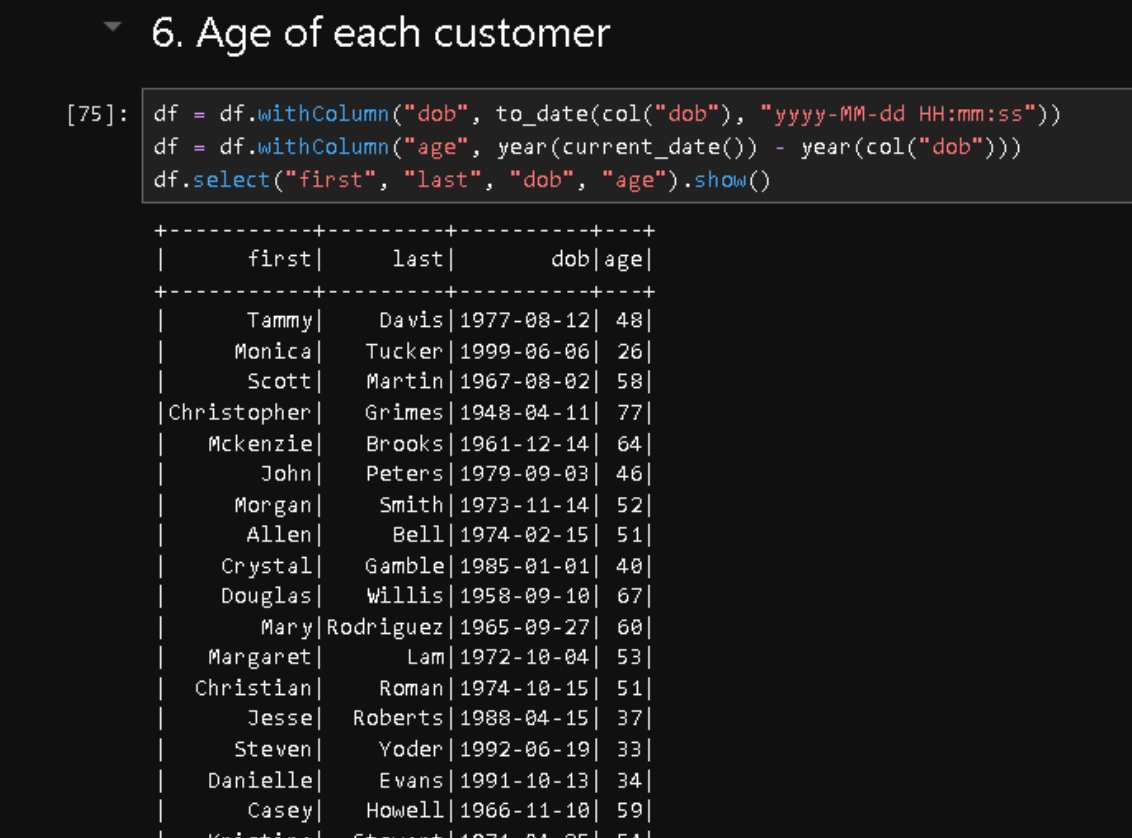


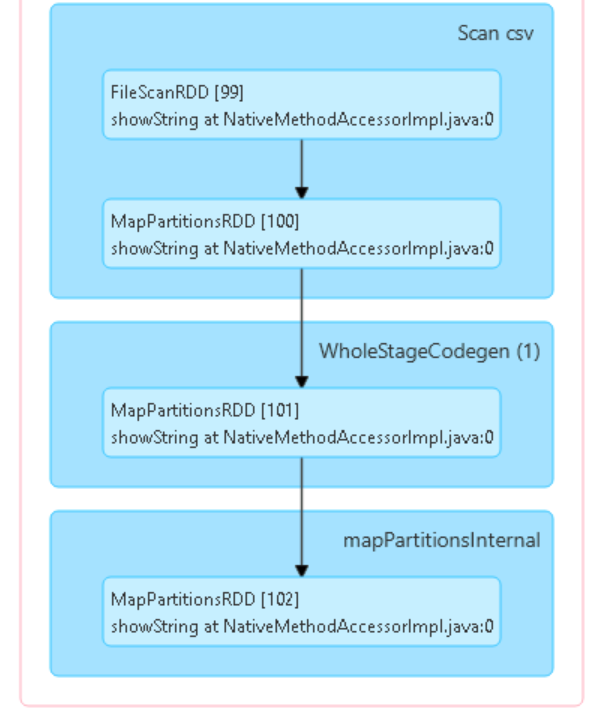


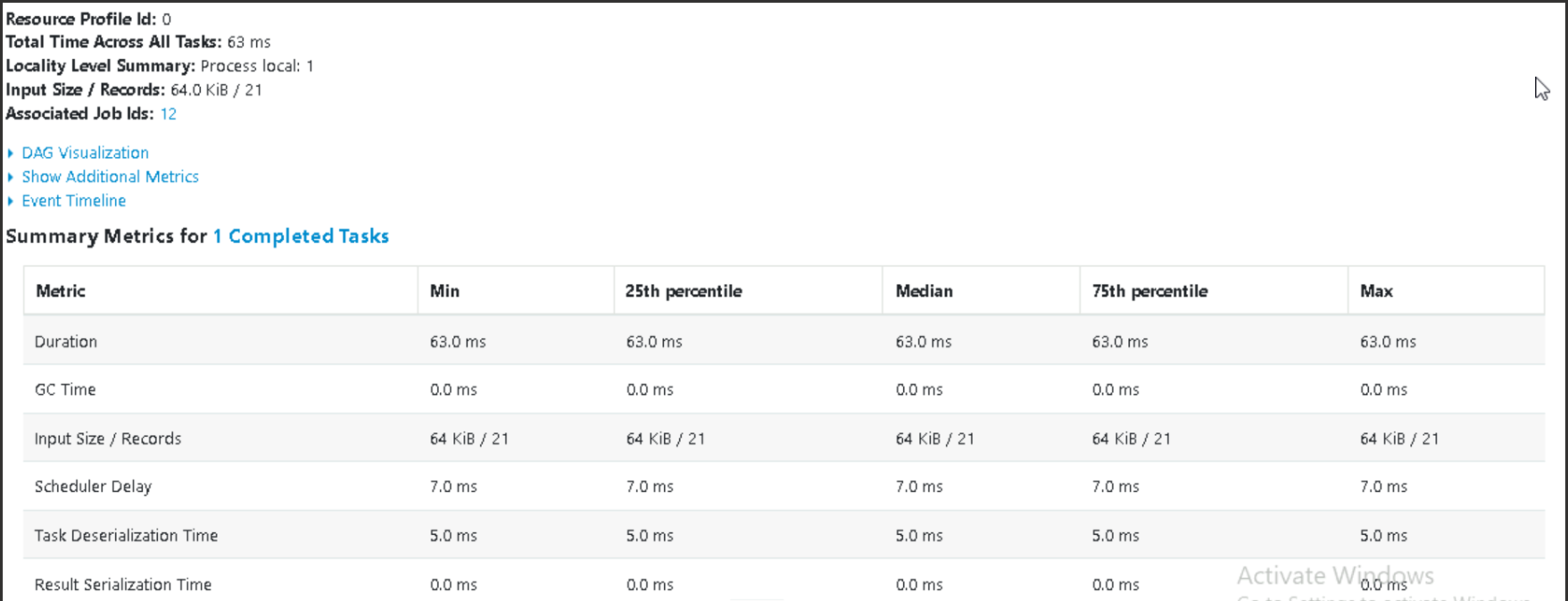




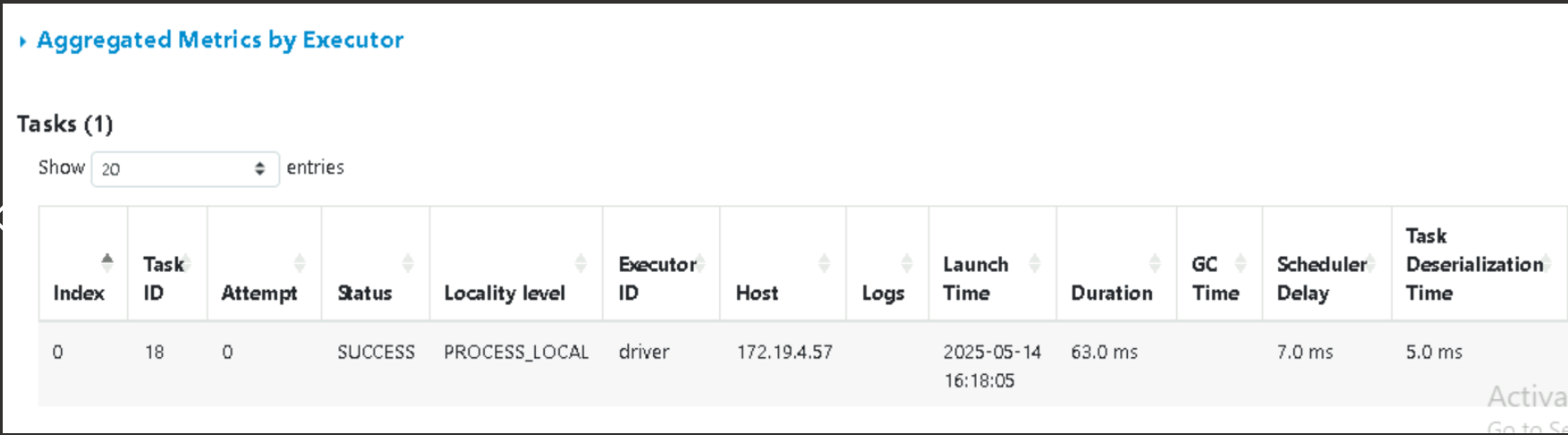
6.

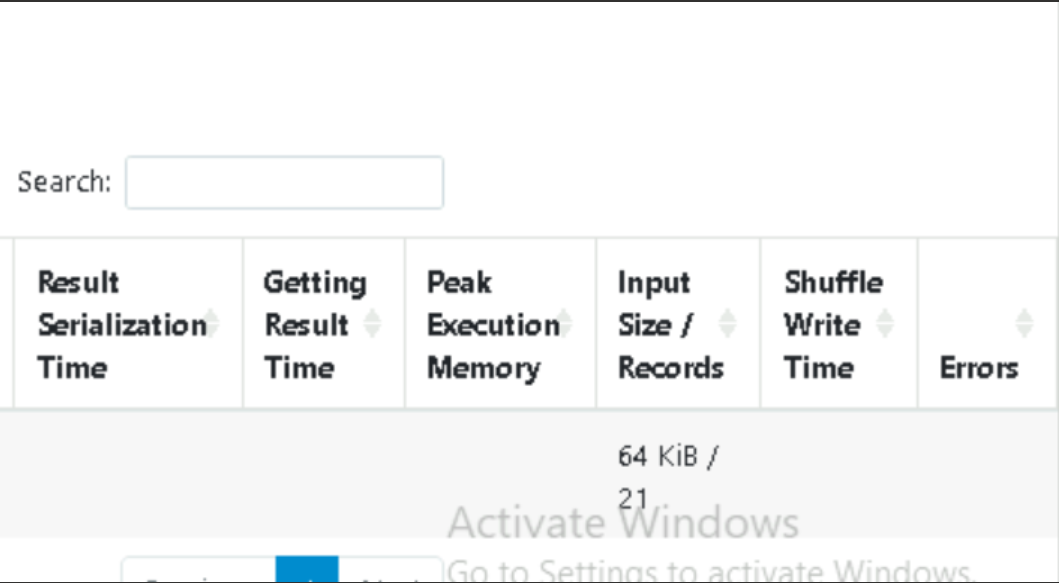




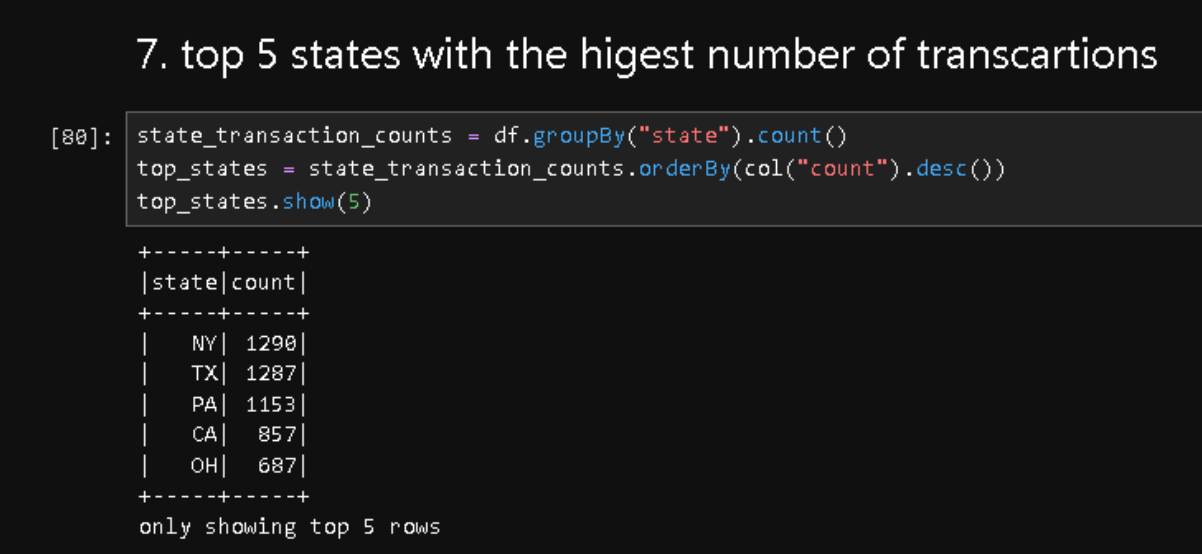


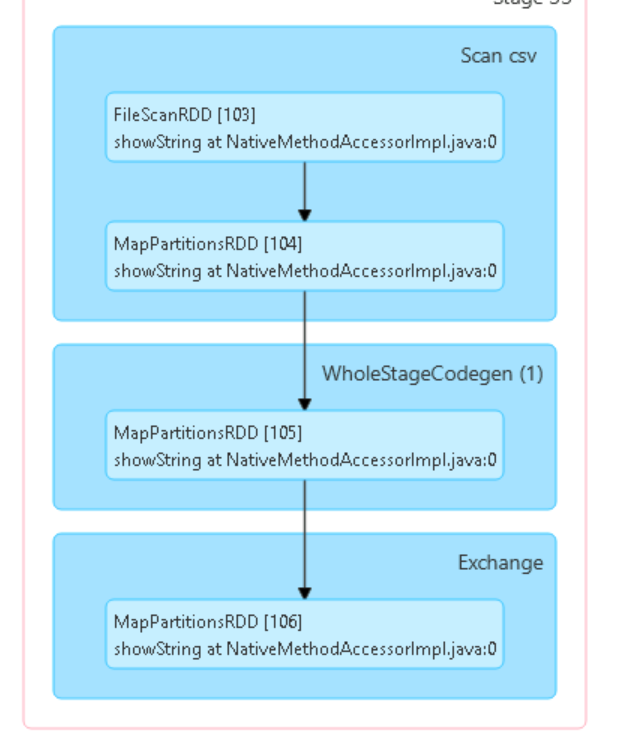


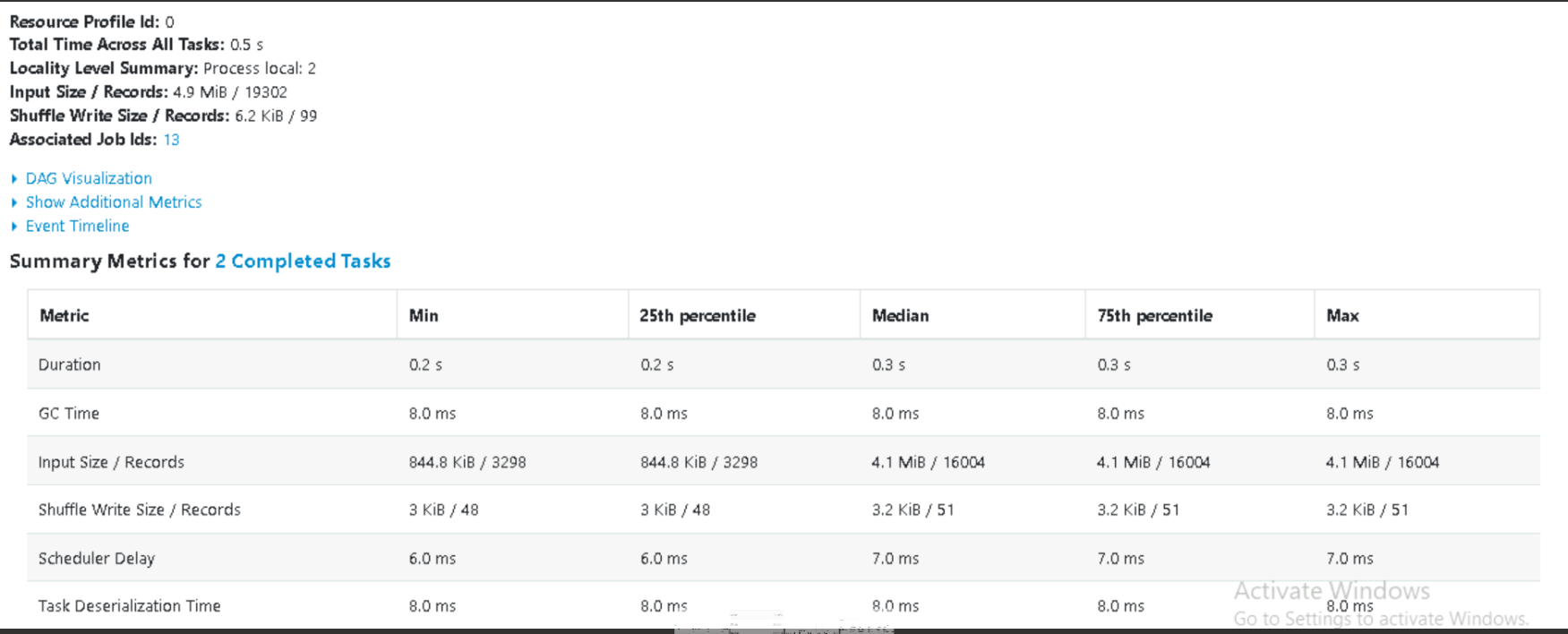


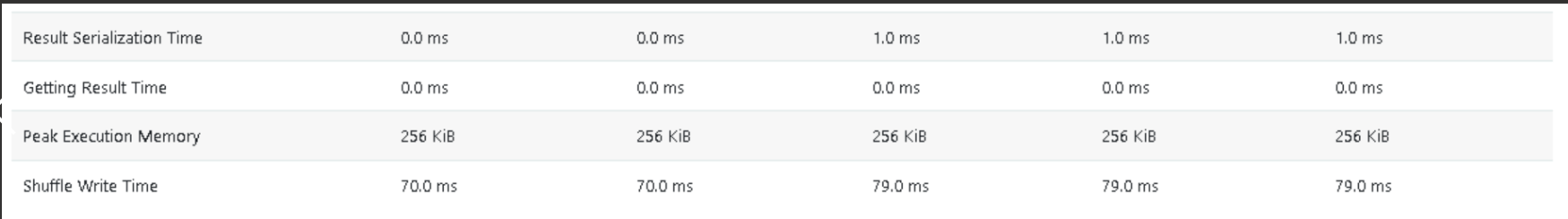


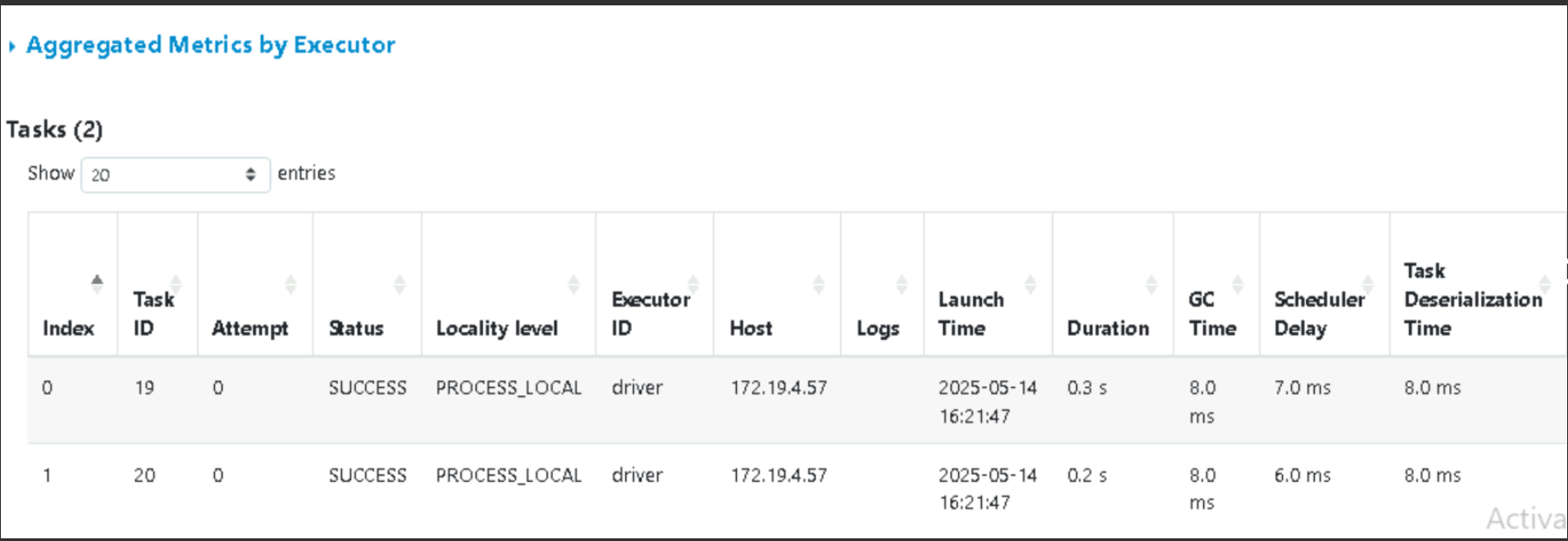
7.

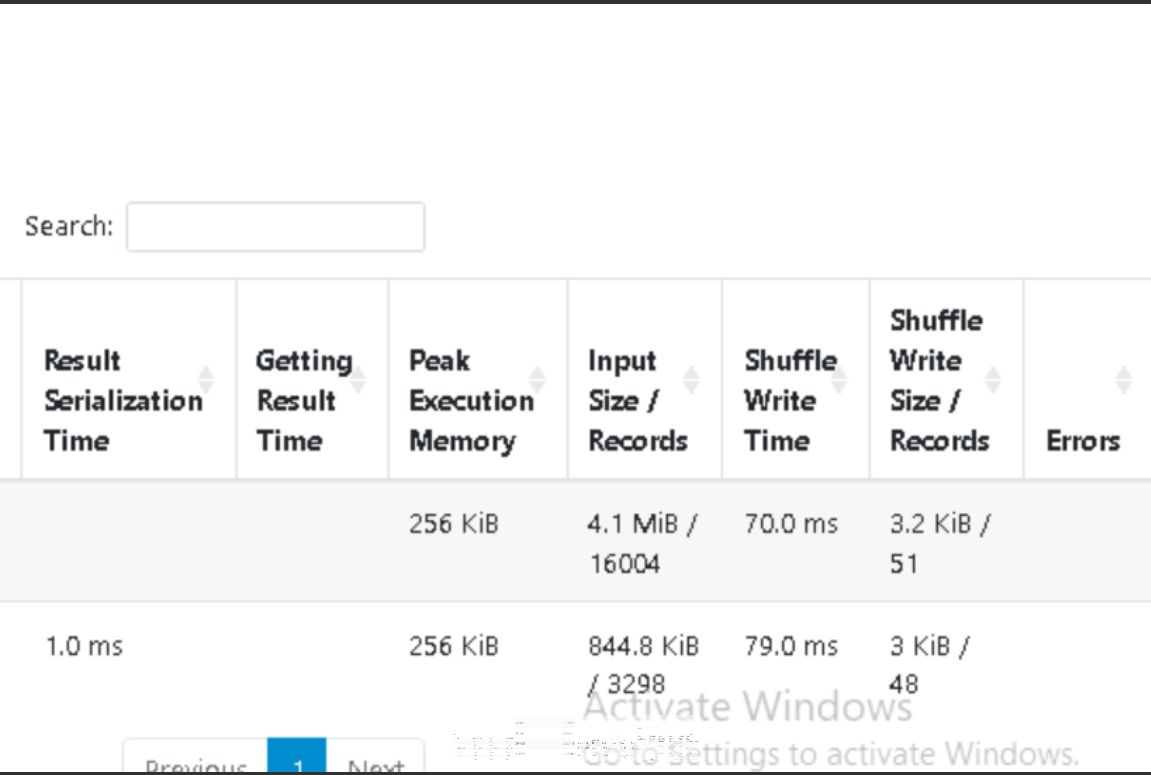




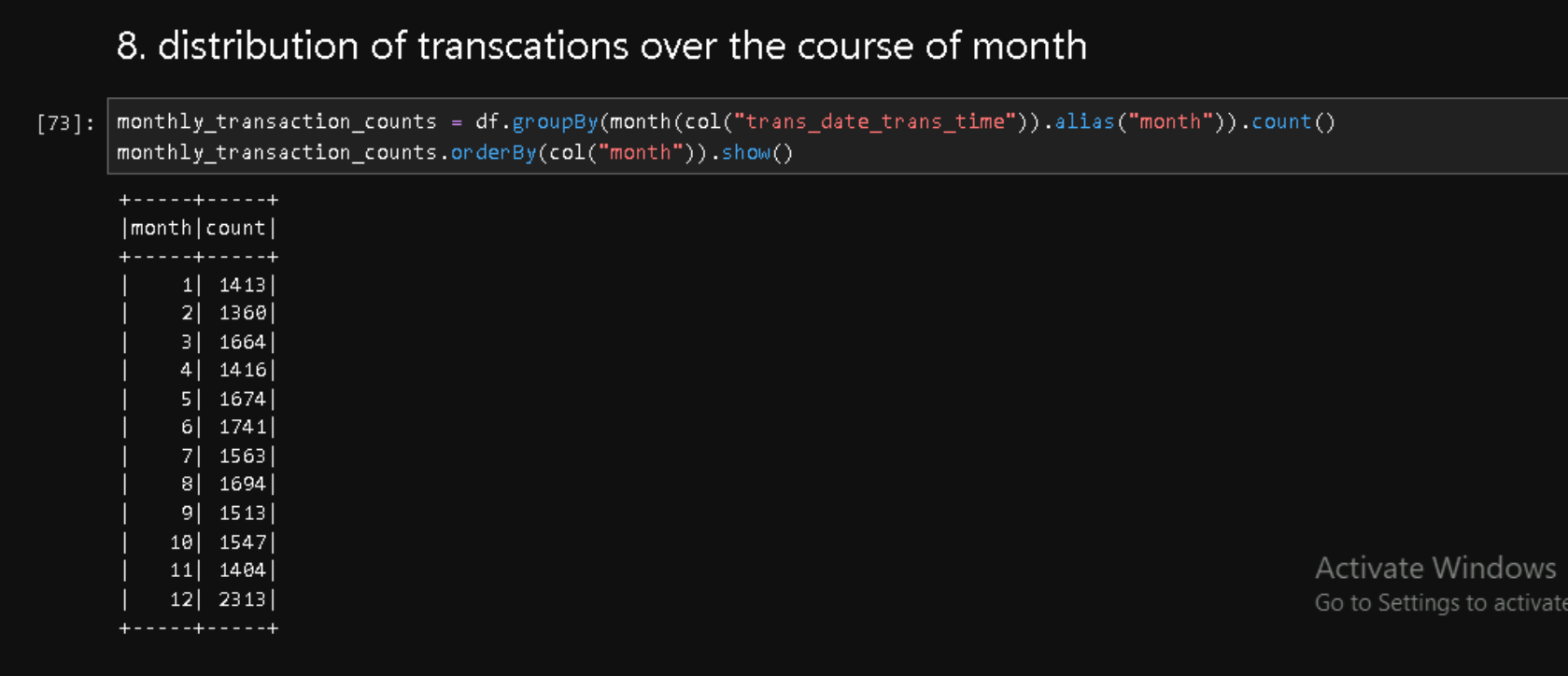


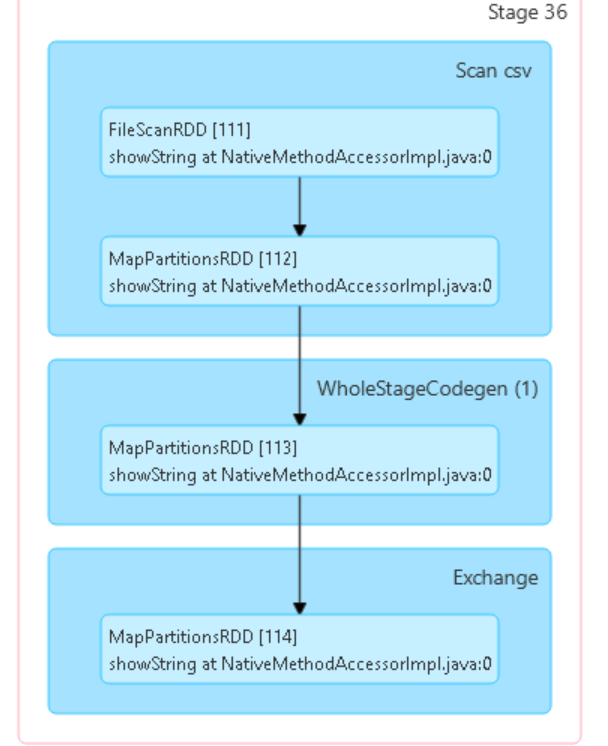


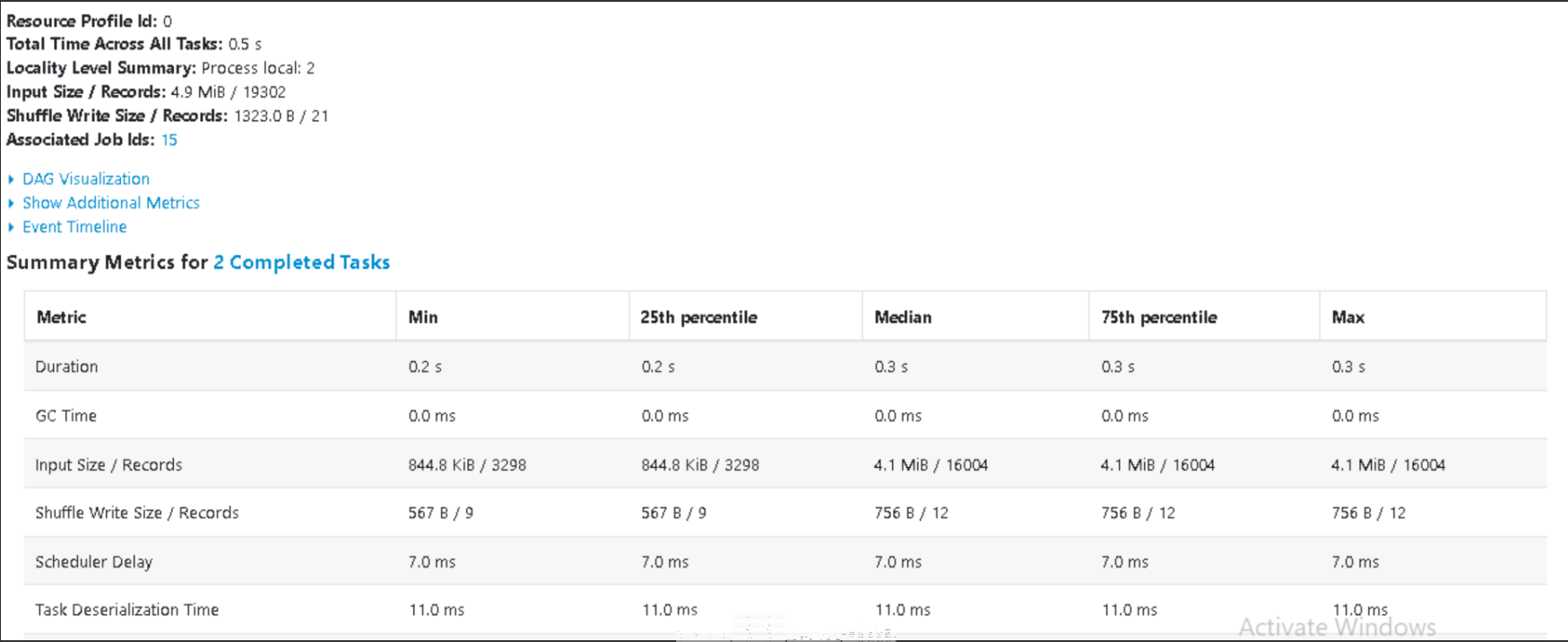




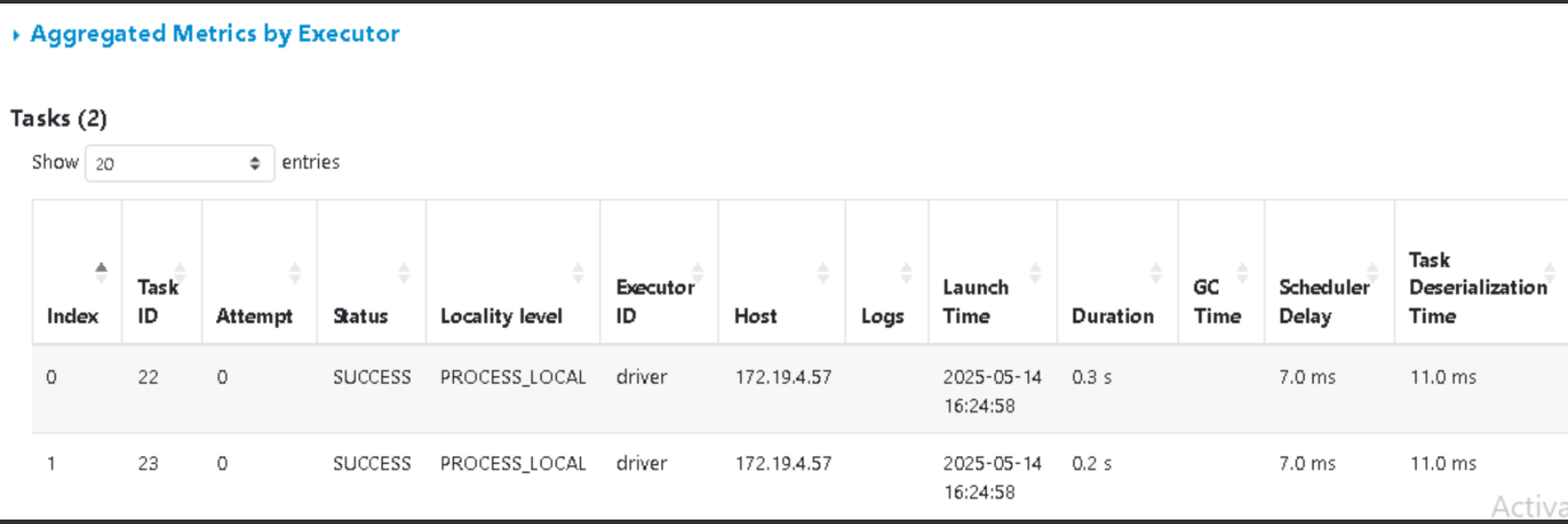
8.

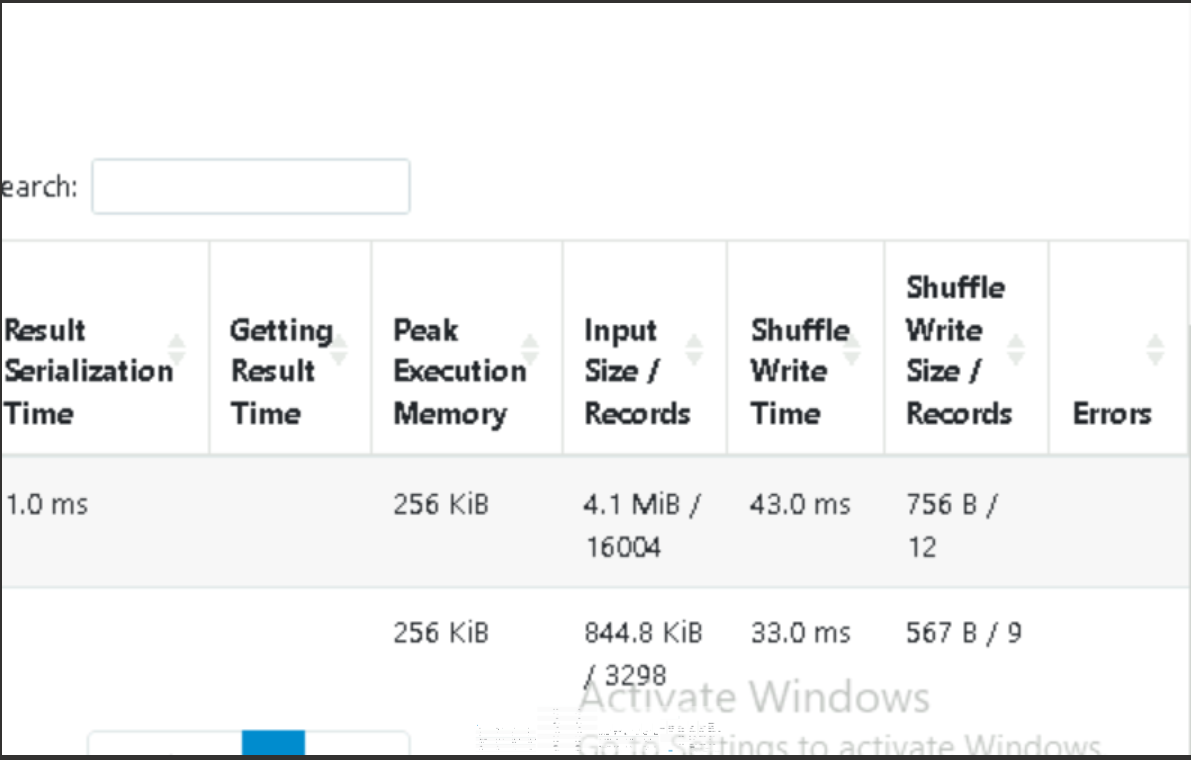












10.

