Problem Explanation:

Tool: Databricks community edition (Runtime:7.2), AWS S3

Note: Actual problem was detailed in step 11. Rest all is the preface/context for understanding problem

1. My original file was of size 3.5 gb (close to 1 crore observations) which was stored on AWS S3. Accessed this from DataBricks community edition.
2. Both S3 and DataBricks are in the same region (US-west)
3. CSV was read after mounting the S3 in DataBricks. IAM user with programmatic access from AWS, fullaccess role was used
4. Below is the schema of my original DataFrame

```
Measurement_Title: string
Measurement_Description: string
Measurement_Type: string
Measurement_Medium: string
Measurement_Time: string
Measurement_Value: double
Units: string
Units_Abbreviation: string
Measurement_Period_Type: string
Data_Stream_ID: integer
Resource_ID: integer
Measurement_ID: long
Record_ID: decimal(24,0)
Latitude: double
Longitude: double
Location: string
Measurement_TimeStamp: timestamp
```

5. This Dataframe has a column called **Measurement Value** which has temperature, pressure, wind speed, %VWC data etc. These values are in **one column**.

Measurement title	**Measurement Type**	**…**	**…**	**Measurement time**	**…**	**Measurement value**
Argyle temp	Temperature					63.5
Langley wind speed	Windspeed					11
UI pressure	pressure					12345

6. I want these Measurement values in different columns which is simply a transpose of current DataFrame. My required DataFrame will look like below

Date	**Temperature**	**Windspeed**	**Pressure**	**%vwc**

7. Hence i first **filtered** on Argyle stations (this will be individual DataFrame of temperature, pressure…) and **select** measurement value, date, hour => These column names are then **aliased** => These DataFrames were then "**inner joined**" over the date column to get single DataFrame => after which unnecessary date columns were **dropped** => Then data was **aggregated** over day level
8. Below image has schema of my final dataset

```
root
 |-- Date: timestamp (nullable = true)
 |-- Avg_Temperature: double (nullable = true)
 |-- Avg_Pressure: double (nullable = true)
 |-- Avg_Wind_Speed: double (nullable = true)
 |-- Avg_Rainfall: double (nullable = true)
 |-- Avg_VWC: double (nullable = true)
```

9. Final DataFrame has 164 records.
10. This final DataFrame has to be written into either csv/parquet
11. ***So while attempting to write the dataframe as csv/parquet file I am getting an error of stage being skipped and I was unable to write dataframe completely***

```
1  ## Saving Arg_Cum_data dataframe as parquet file
2  permanent_table_name = "Arg_Cum_Data"
3  Arg_Cum_data.write.format("parquet").saveAsTable(permanent_table_name)
```

▾ (6) Spark Jobs
- ▸ Job 2 View (Stages: 1/1)
- ▸ Job 3 View (Stages: 1/1)
- ▸ Job 4 View (Stages: 1/1)
- ▸ Job 5 View (Stages: 1/1)
- ▸ Job 6 View (Stages: 1/1)
- ▸ Job 7 View (Stages: 0/0, 6 skipped)

```
1  %sql
2  Select
3  *
4  From
5  Arg
```

▾ (2) Spark Jobs
- ▸ Job 5 View (Stages: 6/6)
- ▸ Job 6 View (Stages: 0/0, 6 skipped)