Task 3 – PySpark Data Pipeline

-- Task 2 Steps --

-- Please try to solve the task first without using this file as a guidance --

First Approach:

1) Use the same Databricks account you created on Task 2.

2) You will find a script in PySpark folder in our respository called pipeline.py

3) The previously mentioned script contains a PySpark code that: Reads Data -> Processes Data (Cleaning and Transformation) -> Saves same Data in two Files (CSV and Parquet).

3) Your script should be similar to it but this is not a must, you might end up with another solution to share.

4) Assuming our source CSV files were already ingested from a DataWarehous (ex. Snowflake, the 3 source CSV files were placed in a newly created volume in the Catalog under jandj schema.

5) Create a new Databricks job to use your pipeline python script and provide any parameters if necessary. Based on our solution, the parameters were provided as follows:

["--mfg","/Volumes/workspace/jandj/jnj-volume/manufacturing_factory_dataset.csv","--events","/Volumes/workspace/jandj/jnj-volume/maintenance_events.csv","--operators","/Volumes/workspace/jandj/jnj-volume/operators_roster.csv","--out","/Volumes/workspace/jandj/jnj-volume/out","--csv-coalesce","1"]

6) Run your job and make sure that the output files were created successfully

Optional Step: create a simple Databricks Notebook to display your results by reading the output file (CSV or Parquet) and using the display() funtion.

Second Approach:

- You can test your pipeline script locally using "spark-submit" command but make sure you provide the right directory paths for your files on your local

machine along with the options (ex. spark-submit pipeline.py --mfg manufacturing_factory_dataset.csv --events maintenance_events.csv --operators operators_roster.csv --out out --csv-coalesce 1) AND make sure you installed and configured PySpark correctly to avoid any issues.