# Compute Daily Product Revenue (Using Databricks):

The purpose of this project is to use two tables from the retail database (orders and order items tables) to calculate the daily product revenue per product and store the results in parquet format.

The data for both files (tables) is stored in text format using comma as a delimiter. Each file does not have headings. The headings are stored in a schema JSON file. The file will be read and based on the table that is being processed the correct headings will be attached to the data frame during processing.

## Notebooks:

The project uses four notebooks. The details of each notebook are listed below.

### 01 Cleanup Database and Datasets:

First the notebook will drop the tables in the database to make sure that any existing data will not being considered for this process.

Then the bronze layer folder and gold layer folder will be removed respectively. Each removal statement will receive the folder path dynamically.

### 02 File Format Converter:

The notebook will read the csv files from given dynamic file paths.

Apply the schema from a JSON file as the CSV files do not have column names, then create the data frame.

Takes the data frame and stores the data in parquet format.

There are 3 variables that will be captured dynamically:

1. The name of the dataset to be converted.

2. The source (file path) base directory.

3. The target path where the parquet file will be stored.

### 03 Create Spark SQL Tables:

The notebook will check first if the database exists, creating one if it is not there.

Using the database, it will create a temporary view with a dynamic name, using the parquet file that was stored in the previous notebook (02 File Format Converter).

Then create a table using the temporary view.

### 04 Daily Product Revenue:

The notebook will use the created database as the source.

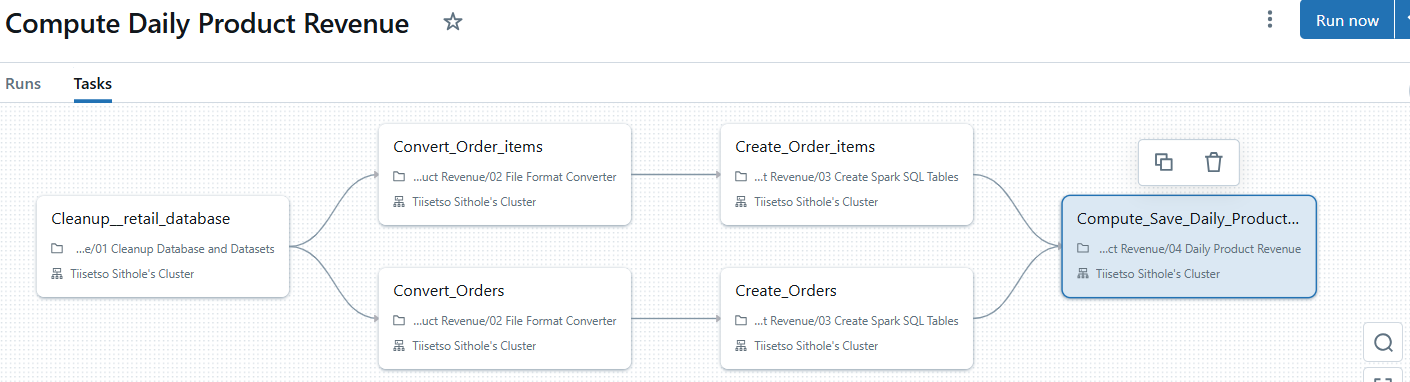
Calculate the daily product revenue using orders and order items tables (tables that are created from previous notebooks.

Store the results in parquet format in a dynamic location (gold layer).

Then the results will be read and can be validated.

## ELT Compute Daily Product Revenue Pipeline:

The image below shows the pipeline which has 6 tasks to handle the whole process of extracting the data from the comma separated files, loading the data and transforming the data and storing results in a folder in parquet format.

Tasks

Cleanup\_\_retail\_database: uses the “01 Cleanup Database and Datasets” notebook for resetting the storage and database before prcocessing.

Convert\_Order\_items: uses the “02 File Format Converter” notebook for converting the order items file into parquet file.

Convert\_Orders: uses the “02 File Format Converter” notebook for converting the orders file into parquet file.

Create\_Order\_items: uses the “03 Create Spark SQL Tables” notebook for creating the order items table.

Create\_Orders: uses the “03 Create Spark SQL Tables” notebook for creating the order table.

Compute\_Save\_Daily\_Product\_Revenue: uses the “04 Daily Product Revenue” notebook for computing the revenue and storing results in parquet format.

## Testing:

The details below outline how the pipeline is run and the parameters needed for a desired outcome.

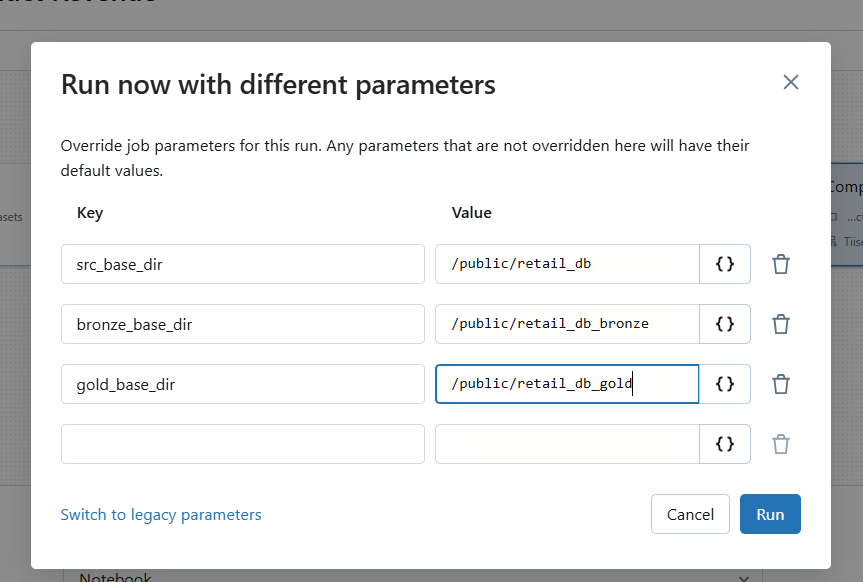
When running the pipeline, use the “run now with different parameters” option as the project needs to know which folder to use as the source, transform and target.

As the folder names are used by several tasks in the pipeline, they are specified at run time.

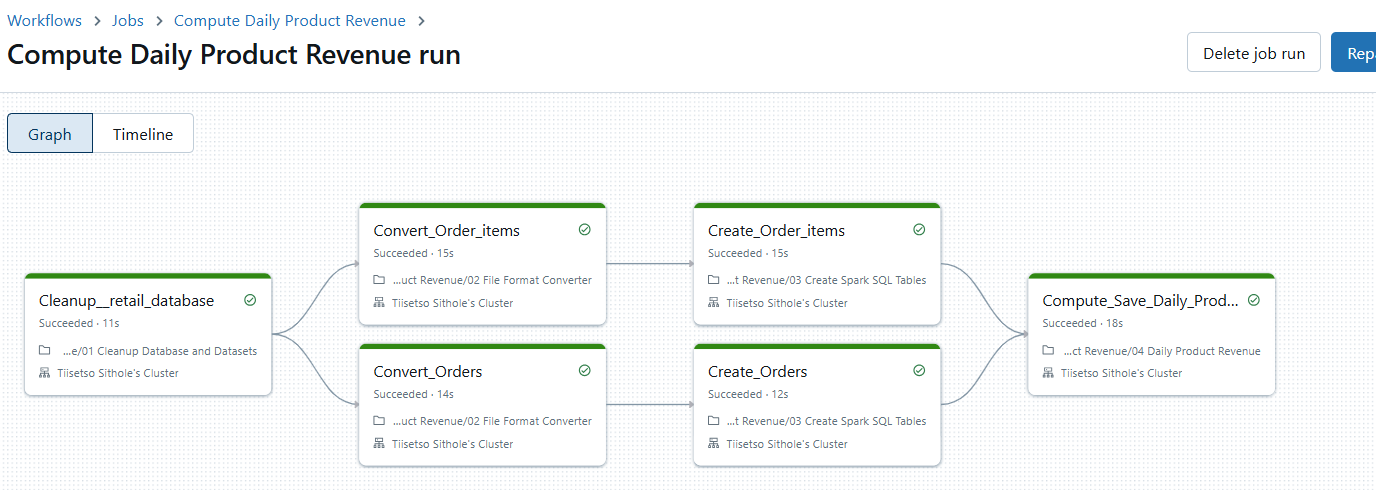
Src\_base\_dir: the directory where the source files are stored.

Bronze\_base\_dir: the directory where the parquet files will be stored after applying the schema for the file.

Gold\_base\_dir: the directory where the overall results of the computed revenue will be stored in parquet format.



The image below shows the results of the executed pipeline. The green bar on each task indicates a successful run. Each task shows the status and time taken per task.



The image below shows the results of the last task, the statement reads the file stored in the target directory which is the revenue calculated per product. 