READ ME – execution – silver\_spend\_from\_public.py

Date: 2025-10-08

* Initial Load: spend\_data\_2025-10-06.json (include data 9/30 – 10/6)
* Incremental Load: spend\_data\_2025-10-08.json (include data 9/30 – 10/8)
* Duplication records: 9/30,10/01,10/02,10/03,10/04,10/5,10/06. (the Delta\_log will only record the latest one record as valid, the previous will be inactive history record)
* incremental: 10/07, 10/08

**Argument:**

* Below are the arguments for generating Delta data sets.

spark-submit

--deploy-mode cluster

--packages io.delta:delta-spark\_2.12:3.2.0

--conf spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension

--conf spark.sql.catalog.spark\_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog

--conf spark.delta.logStore.class=org.apache.spark.sql.delta.storage.S3SingleDriverLogStore

s3://dea-calendly-data/code/silver\_spend\_from\_public.py

s3://betty-calendly-test/calendly\_spend\_data

2025-10-06

s3://dea-calendly-data/silver/events\_spend

**NOTE:**

* The source file (spend\_data\_2025010-06.json) will include all previous date records for about 1 month (9/6 – 10/6). I deleted the records older than 9/30 because I have no Webhook data for those dates.
* I copied the source data from s3://dea-data-bucket/calendly\_spend\_data, loaded it to my local laptop, deleted the unnecessary dates (older than 9/30), then uploaded the file (for example, spend\_date\_2025-10-06, which includes 9/30 – 10/06 data only) to s3://betty-calendly-test/calendly\_spend\_data. Thus, the spend\_data will match the data’s dates in Calendly\_events (9/30-10/6)

**Key Codes:**

* If the date is duplicated, for example, if “dt=2025-10-06” exists there. A new data entry will be created within the same folder; however, in the Delta\_log, only the most recent record will be considered valid.

predicate = "dt IN ('2025-09-30','2025-10-01','2025-10-02','2025-10-03','2025-10-04','2025-10-05','2025-10-06','2025-10-07','2025-10-08')"

(df.write

.format("delta")

.mode("overwrite")

.option("replaceWhere", predicate)

.partitionBy("dt")

.save(silver\_out))

**Result:**

1. Delta finds all existing files whose partition value is in that list (i.e., 09-30 … 10-06) and **tombstones** them (marks removed in \_delta\_log).
2. It writes your new df rows (09-30 … 10-08) and **adds** them to the log.
3. Net effect after the commit:
   * **2025-09-30 … 2025-10-06** are **replaced** with your new data.
   * **2025-10-07, 2025-10-08** are **new partitions** (added).
   * Readers of the **Delta** table see only the new snapshot (no duplicates).
   * Old files still exist on S3 but are hidden; remove them later with VACUUM if I want.

For example, I loaded spend\_data\_2025-10-06 on Oct 8, then loaded spend\_data\_2025-10-08 on Oct 09, created 2 entries under dt=2025-10-06:

A screenshot of a computer

AI-generated content may be incorrect.

* predicate → the condition (usually specific dates or a date range) that identifies which partitions/rows you intend to replace.
* replaceWhere → the option that applies that predicate to scope the overwrite (i.e., “overwrite only where predicate is true”). Not a location; it’s a filter.
* mode("overwrite") → perform a replacement instead of append. With Delta, this replacement is atomic

**RESULT 1:**

**data from 9/30 – 10/6 loaded into** s3://dea-calendly-data/silver/events\_spend

A screenshot of a computer

AI-generated content may be incorrect.

**RESULT 2:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**INCREMENTAL:**

* Later, I can modify the code to accept the process date as “current date – 1”, then I don’t need to input the date manually

A screenshot of a computer

AI-generated content may be incorrect.

You will see two records, one is the history, another is the latest.

The Delta log will only record Oct/9 (the latest) as active. Ignore the Oct 8’s record:

A screenshot of a computer

AI-generated content may be incorrect.

Athena:

After new load with 10/7, 10/8 data, check the marketing\_data.events\_spend:

* We only see the LATEST 3 records here for 2025-10-06 (even though there are two entries under dt=2025-10-06

A screenshot of a computer

AI-generated content may be incorrect.