Policy Lifecycle Tracking Project | End-to-End Azure Pipeline with Databricks & Snowflake

Project Summary

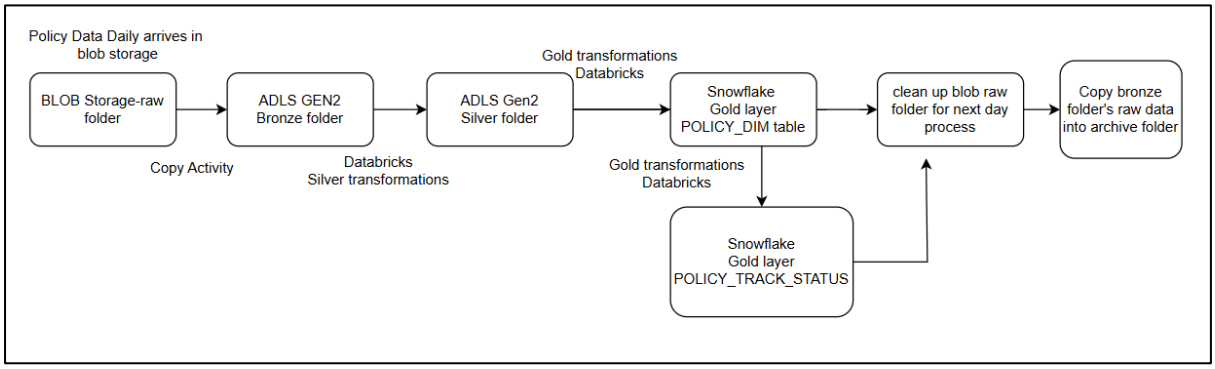
- **Objective**: To build a data pipeline for tracking policy status transitions over time by ingesting, transforming, and storing policy data from raw sources to a refined analytical state.

- **Source**: Daily policy data stored in a *Blob Storage (Raw Folder)*.

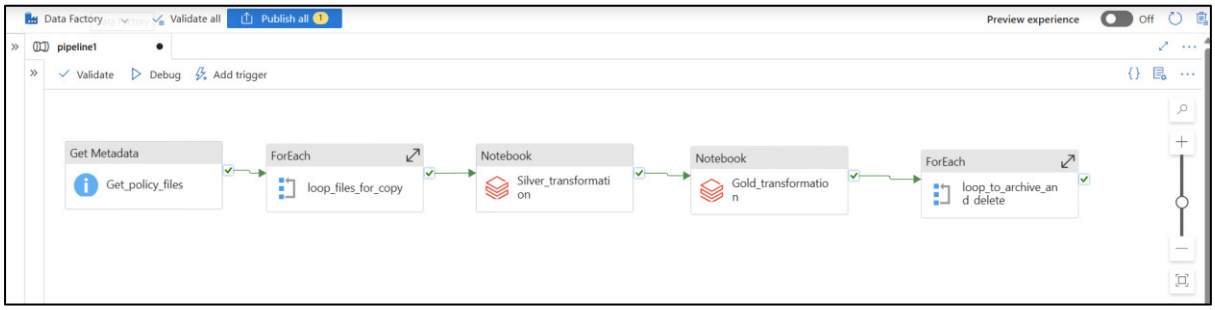
- **Destination**: A *Snowflake data warehouse*, with intermediate transformations and data quality layers handled in *Azure Data Lake Gen2* and *Databricks*.

- **Purpose**: Maintain a full lifecycle view of each policy, track changes in status, and compute the duration spent in each status phase.

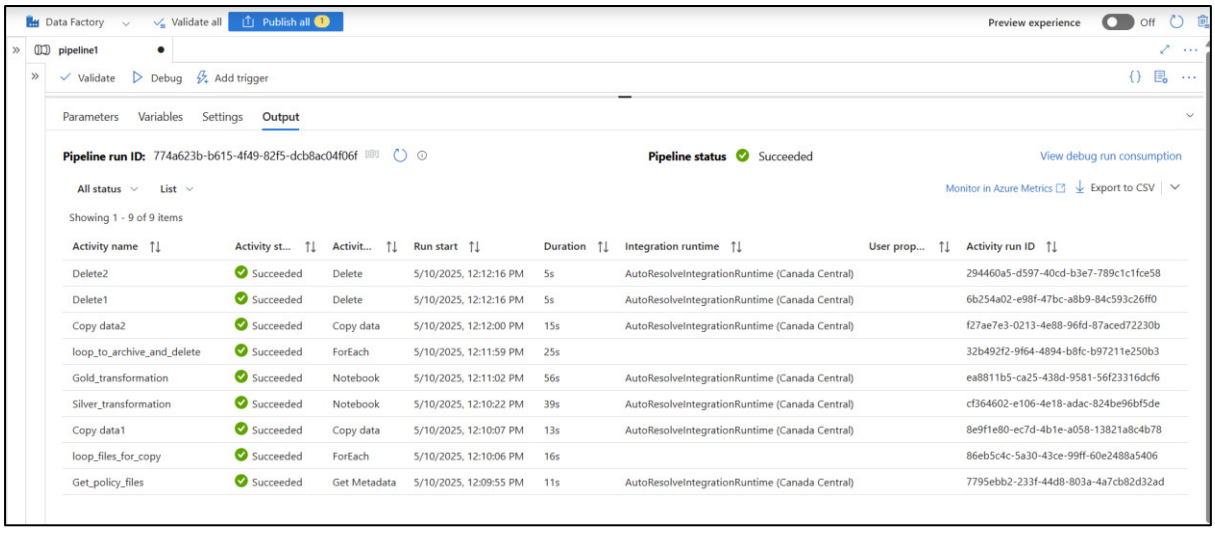
Architecture diagram:



The end-to-End pipeline with dynamic parameters



Pipeline debug result for day-1 file



Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Delete2	Succeeded	Delete	5/10/2025, 12:12:16 PM	5s	AutoResolveIntegrationRuntime (Canada Central)		294460a5-d597-40cd-b3e7-789c1c1fce58
Delete1	Succeeded	Delete	5/10/2025, 12:12:16 PM	5s	AutoResolveIntegrationRuntime (Canada Central)		6b254a02-e98f-47bc-a8b9-84c593c26ff0
Copy data2	Succeeded	Copy data	5/10/2025, 12:12:00 PM	15s	AutoResolveIntegrationRuntime (Canada Central)		f27ae7e3-0213-4e88-96fd-87aced72230b
loop_to_archive_and_delete	Succeeded	ForEach	5/10/2025, 12:11:59 PM	25s			32b492f2-9f64-4894-b8fc-b97211e250b3
Gold_transformation	Succeeded	Notebook	5/10/2025, 12:11:02 PM	56s	AutoResolveIntegrationRuntime (Canada Central)		ea8811b5-ca25-438d-9581-56f23316dcf6
Silver_transformation	Succeeded	Notebook	5/10/2025, 12:10:22 PM	39s	AutoResolveIntegrationRuntime (Canada Central)		cf364602-e106-4e18-adac-824be96bf5de
Copy data1	Succeeded	Copy data	5/10/2025, 12:10:07 PM	13s	AutoResolveIntegrationRuntime (Canada Central)		8e9f1e80-ec7d-4b1e-a058-13821a8c4b78
loop_files_for_copy	Succeeded	ForEach	5/10/2025, 12:10:06 PM	16s			86eb5c4c-5a30-43ce-99ff-60e2488a5406
Get_policy_files	Succeeded	Get Metadata	5/10/2025, 12:09:55 PM	11s	AutoResolveIntegrationRuntime (Canada Central)		7795ebb2-233f-44d8-803a-4a7cb82d32ad

Data Transformation & Workflow

1. Bronze Layer (Raw Ingestion)

- Ingest daily policy data from Azure Blob Storage into ADLS Gen2 Bronze folder.
- Raw data is stored as-is, preserving original structure for audit and traceability.

2. Silver Layer (Cleansing & Standardization)

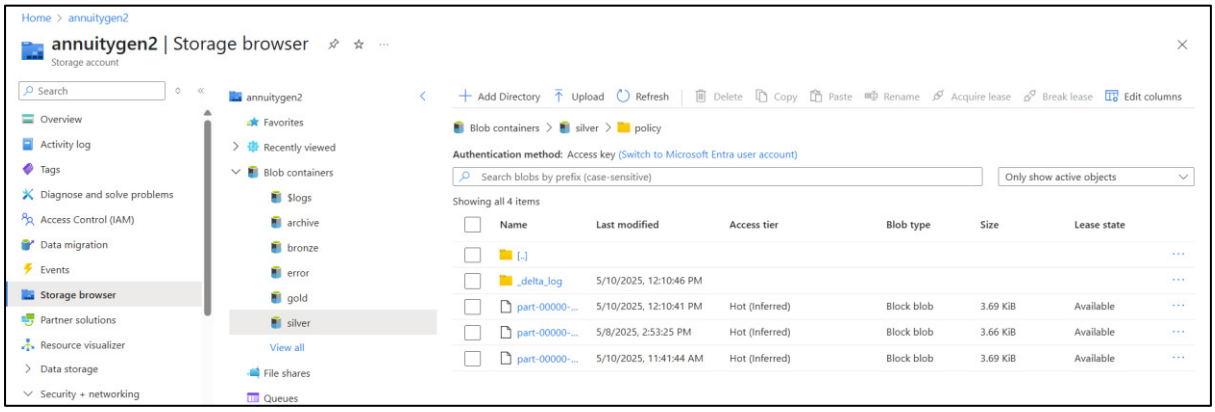
- Use Databricks to process Bronze layer data.
- Apply cleansing (null handling, schema enforcement) and standardization (field renaming, type casting).
- Store cleaned and standardized data in the Silver folder in ADLS Gen2.

Silver transformation logic is

```
# Import the required functions and types

from pyspark.sql.types import *

from pyspark.sql.functions import *

#Read policy file

policy_df = spark.read.format("csv").option("header", "true").option("inferSchema",
"true").load("/mnt/bronze/policy_snapshot_*.csv")

# data cleansing

policy_clean_df=policy_df.withColumn("policy_status",trim(col("policy_status"))) \

.withColumn("policy_status",
when(col("policy_status").isin("submited","Submited"),"Submitted").

when(col("policy_status").isin("Actve","actve"),"Active").


when(col("policy_status").isin("Canceld","canceld"),"Cancelled").

when(col("policy_status").isin("Mature","mature"),"Mature").

otherwise(initcap(col("policy_status")))) \

.withColumn("submission_date",col("submission_date").cast(DateType())) \

.withColumn("status_update_date",to_date(col("status_update_date"))) \

.withColumn("agent_id",trim(col("agent_id"))) \

.withColumn("agent_id", when(col("agent_id").isNull(),
lit("UNKNOWN")).otherwise(col("agent_id"))) \

.fillna("UNKNOWN","policy_status") \

.dropna(subset=["submission_date", "status_update_date"])
```

```
#Read region file

region_df=spark.read.format("csv") \

.option("header", "true") \

.option("inferSchema", "true") \

.load("/mnt/region/regions.csv")

#join policy and region files

policy_region_df=policy_clean_df.join(region_df,policy_clean_df.region
==region_df.region_id,"inner").drop("region_id")

# validation rule

valid_statuses = ["Submitted","Active","Cancelled","Mature"]

policy_validated_df=policy_region_df.filter(col("policy_status").isin(valid_statuses))

# Add audit information

df_silver = policy_validated_df.withColumn("ingesttime", current_timestamp())

# Write to silver table

df_silver.write.mode("overwrite").format("delta").save("/mnt/silver/policy")
```

The transformed file is in silver folder as shown below



3. Gold Layer (Business Transformations)

- Read Silver data into Databricks.
- Apply business logic to generate the Policy Dimension (policy_dim) table:
  - Append daily data to maintain historical versions of policy records.
  - Capture all phase changes for each policy.
- Store policy_dim table into Snowflake as a dimensional table.

4. Policy Status Tracking

- Read policy_dim table from Snowflake.
- Compute previous status, current status, and date difference between transitions using window functions or lag operations.
- Store the output as policy_track_status table in Snowflake

Gold layer transformation logic is

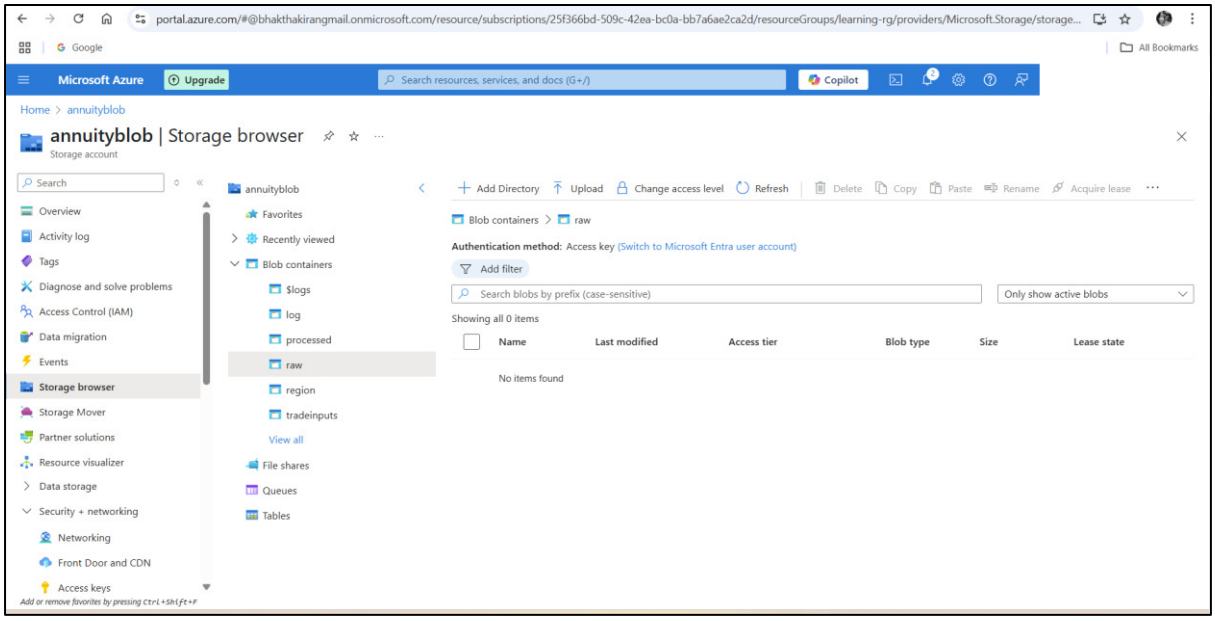
```
# Import the required functions and types
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.window import Window
#read Policy file
policy_df = spark.read.format("delta").load("/mnt/silver/policy")
#Access snowflake account
sfOptions = {
"sfURL": "https://lbnfhza-et12707.snowflakecomputing.com",
"sfDatabase": "POLICY_DATA_DB",
"sfSchema": "GOLD_LAYER",
"sfWarehouse": "POLICY_WH",
"sfRole": "ACCOUNTADMIN",
"sfUser": "AAAAAAAAAAAA",
"sfPassword": "XXXXXXXXXXXX",
}
#write dim current table
policy_df.write \
.format("snowflake") \
.options(**sfOptions) \
.option("dbtable", "policy_dim") \
.mode("append") \
.save()
```

```
#Read the latest policy dim table

policy_dim_current = spark.read \

.format("snowflake") \

.options(**sfOptions) \

.option("dbtable", "policy_dim") \

.load()

#Lifecycle Status Tracking | get previous and next status

windowfunc=Window.partitionBy("policy_id").orderBy("status_update_date")

policy_status_df =
policy_dim_current.withColumn("prev_status",lag("policy_status").over(windowfunc)) \

.withColumn("prev_status_update_date",lag("status_update_date").over(windowfunc))

policy_diff_df =
policy_status_df.withColumn("No_of_day_in_status",datediff(col("status_update_date"),col("p
rev_status_update_date")))

policy_track_status_df = policy_diff_df \

.select("Policy_ID","Prev_status","policy_status","status_update_date","No_of_day_in_status")

#write dim current table

policy_track_status_df.write \

.format("snowflake") \

.options(**sfOptions) \

.option("dbtable", "policy_track_status") \

.mode("overwrite") \

.save()
```

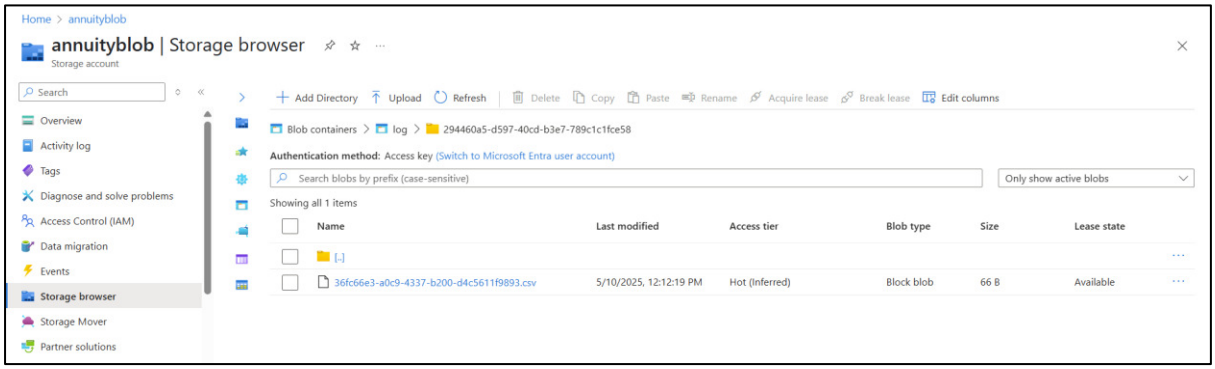
The transformed files are in snowflake as shown in below screenshots

Post-Processing Logic:

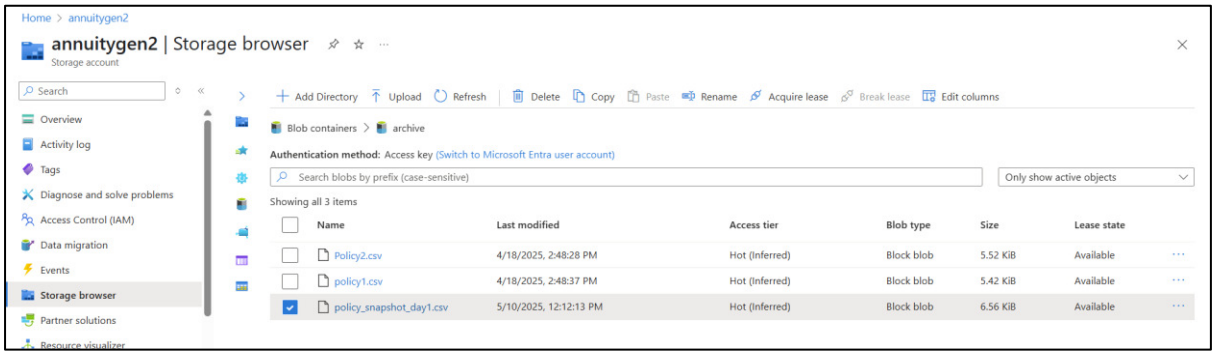
- After processing each day's data: ○ Files are deleted from the raw folder in Blob Storage.
- Metadata/logs of processed files are stored in a log folder.
- Bronze-level files are moved to an archive folder for backup and auditing.



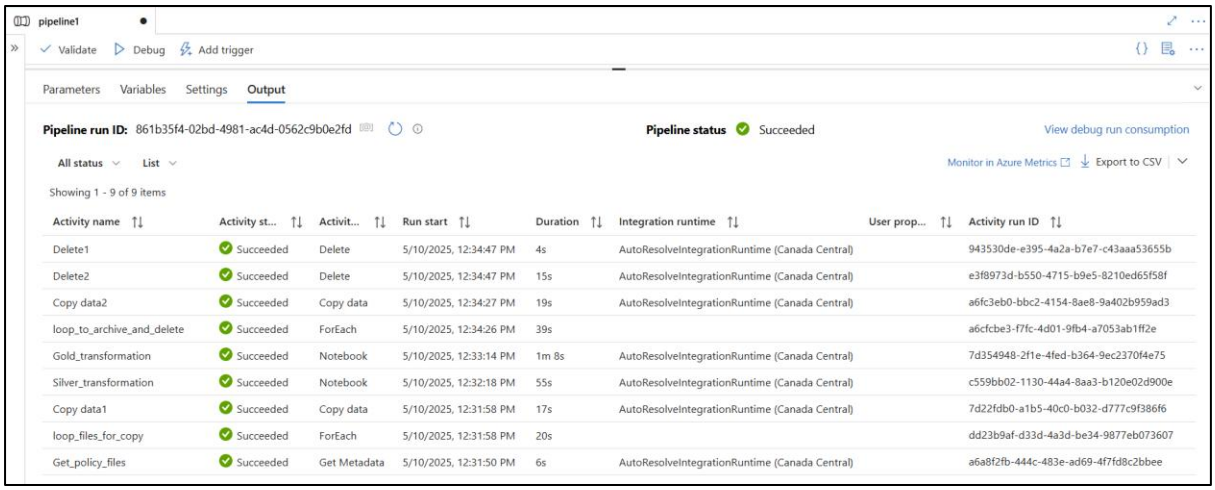
Logged in log file regarding file deletion from raw folder



File was moved to archive folder

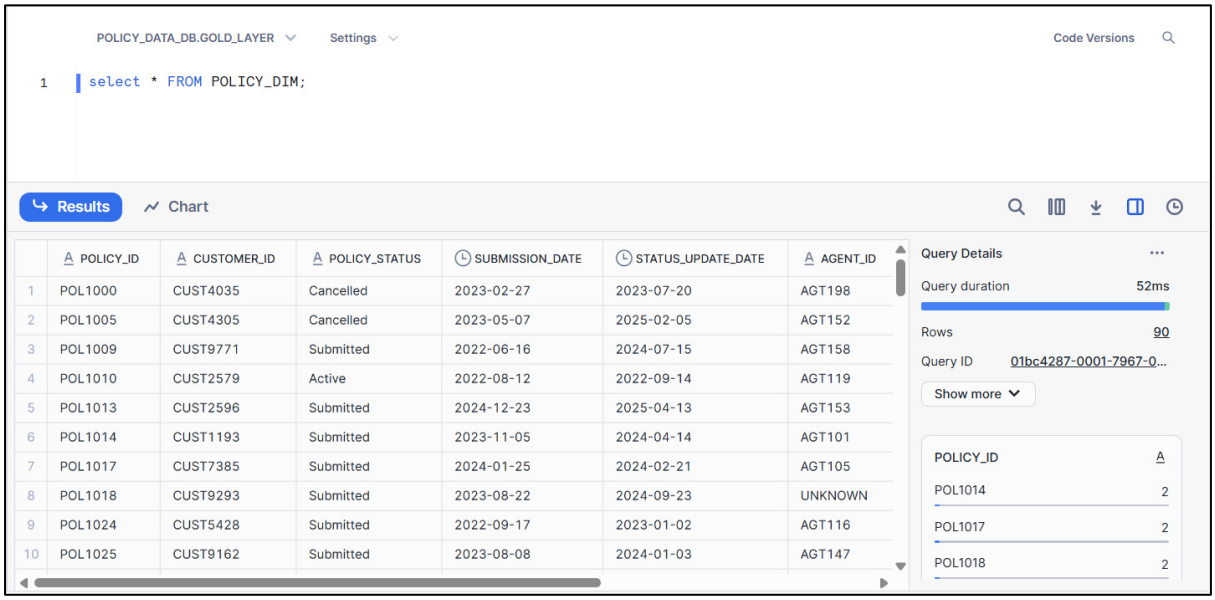


Processing day-2 file:Day-2 file was processed and debug result as shown below



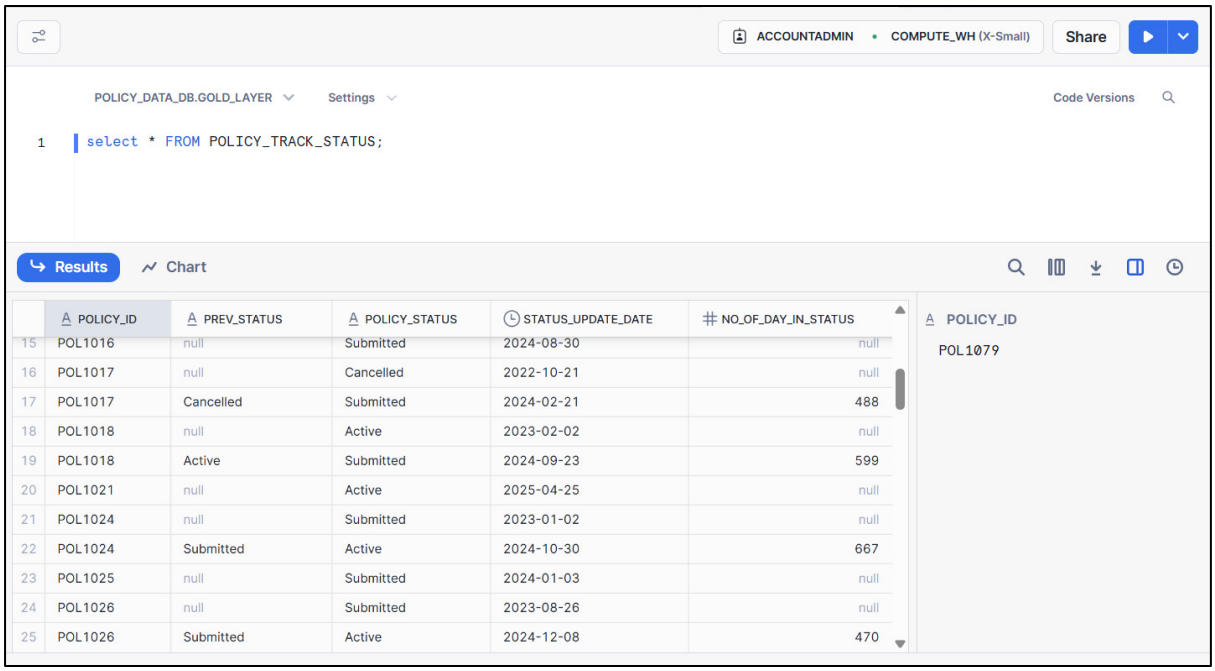
Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Delete1	Succeeded	Delete	5/10/2025, 12:34:47 PM	4s	AutoResolveIntegrationRuntime (Canada Central)		943530de-e395-4a2a-b7e7-c43aaa53655b
Delete2	Succeeded	Delete	5/10/2025, 12:34:47 PM	15s	AutoResolveIntegrationRuntime (Canada Central)		e3f8973d-b550-4715-b9e5-8210ed65f58f
Copy data2	Succeeded	Copy data	5/10/2025, 12:34:27 PM	19s	AutoResolveIntegrationRuntime (Canada Central)		a6fc3eb0-bbc2-4154-8ae8-9a402b959ad3
loop_to_archive_and_delete	Succeeded	ForEach	5/10/2025, 12:34:26 PM	39s			a6cfcbe3-f7fc-4d01-9fb4-a7053ab1ff2e
Gold_transformation	Succeeded	Notebook	5/10/2025, 12:33:14 PM	1m 8s	AutoResolveIntegrationRuntime (Canada Central)		7d354948-2f1e-4fed-b364-9ec2370f4e75
Silver_transformation	Succeeded	Notebook	5/10/2025, 12:32:18 PM	55s	AutoResolveIntegrationRuntime (Canada Central)		c559bb02-1130-44a4-8aa3-b120e02d900e
Copy data1	Succeeded	Copy data	5/10/2025, 12:31:58 PM	17s	AutoResolveIntegrationRuntime (Canada Central)		7d22fdb0-a1b5-40c0-b032-d777c9f386f6
loop_files_for_copy	Succeeded	ForEach	5/10/2025, 12:31:58 PM	20s			dd23b9af-d33d-4a3d-be34-9877eb073607
Get_policy_files	Succeeded	Get Metadata	5/10/2025, 12:31:50 PM	6s	AutoResolveIntegrationRuntime (Canada Central)		a6a8f2fb-444c-483e-ad69-4f7fd8c2bbee

Day-2 data is appended in snowflake Policy_dim table



	POLICY_ID	CUSTOMER_ID	POLICY_STATUS	SUBMISSION_DATE	STATUS_UPDATE_DATE	AGENT_ID
1	POL1000	CUST4035	Cancelled	2023-02-27	2023-07-20	AGT198
2	POL1005	CUST4305	Cancelled	2023-05-07	2025-02-05	AGT152
3	POL1009	CUST9771	Submitted	2022-06-16	2024-07-15	AGT158
4	POL1010	CUST2579	Active	2022-08-12	2022-09-14	AGT119
5	POL1013	CUST2596	Submitted	2024-12-23	2025-04-13	AGT153
6	POL1014	CUST1193	Submitted	2023-11-05	2024-04-14	AGT101
7	POL1017	CUST7385	Submitted	2024-01-25	2024-02-21	AGT105
8	POL1018	CUST9293	Submitted	2023-08-22	2024-09-23	UNKNOWN
9	POL1024	CUST5428	Submitted	2022-09-17	2023-01-02	AGT116
10	POL1025	CUST9162	Submitted	2023-08-08	2024-01-03	AGT147

The below table has the previous status and no of days that Policy was in previous status



	POLICY_ID	PREV_STATUS	POLICY_STATUS	STATUS_UPDATE_DATE	NO_OF_DAY_IN_STATUS
15	POL1016	null	Submitted	2024-08-30	null
16	POL1017	null	Cancelled	2022-10-21	null
17	POL1017	Cancelled	Submitted	2024-02-21	488
18	POL1018	null	Active	2023-02-02	null
19	POL1018	Active	Submitted	2024-09-23	599
20	POL1021	null	Active	2025-04-25	null
21	POL1024	null	Submitted	2023-01-02	null
22	POL1024	Submitted	Active	2024-10-30	667
23	POL1025	null	Submitted	2024-01-03	null
24	POL1026	null	Submitted	2023-08-26	null
25	POL1026	Submitted	Active	2024-12-08	470

POLICY_ID
POL1079

Conclusion

- End-to-end pipeline ensures clean, historical, and traceable policy data.
- Intelligent logging, archival, and deletion mechanisms ensure efficient storage management and audit readiness.
- Gold layer delivers value-added insights into policy lifecycle stages, enabling strategic decision-making.