

Results from the data quality expectations can be analyzed further by querying the event log. The event log contains detailed metrics about all expectations defined for the workflow pipeline. The query below provides an example for viewing key metrics from the last pipeline update, including the number of records that passed or failed expectations:

```
1  SELECT
2      row_expectations.dataset AS dataset,
3      row_expectations.name AS expectation,
4      SUM(row_expectations.passed_records) AS passing_records,
5      SUM(row_expectations.failed_records) AS failing_records
6  FROM
7  (
8      SELECT
9          explode(
10             from_json(
11                 details :flow_progress :data_quality :expectations,
12                 "array<struct<name: string, dataset: string, passed_records: int,
13                 failed_records: int>>"
14             )
15         )
16     ) row_expectations
17  FROM
18      event_log_raw
19  WHERE
20      event_type = 'flow_progress'
21      AND origin.update_id = '${latest_update.id}'
22  )
23 GROUP BY
24      row_expectations.dataset,
25      row_expectations.name;
```

Again, we can view the complete history of changes made to each DLT table by looking at the Delta history logs (see Figure 12). It allows us to understand how tables evolve over time and investigate complete threads of updates if a pipeline fails.

Columns	Sample Data	Details	Permissions	History										
Version	Timestamp	User Id	User Name	Operation	Operation Parameters	Job	Notebook	Cluster Id	Read Version	Isolation Level	Is Blind Append	Operation Metrics	User Metadata	Engine Info
12	2023-01-22T19:10:09	Null	Null	WRITE	> { ... } // 1 item	Null	Null	Null	11	WriteSerializable	false	> { ... } // 3 items	Null	Databricks-Runtime/dlt:11.0-delta-pipelines-1ecad9-750b289-9ea72db-custom-local
11	2022-12-09T11:48:23	Null	Null	WRITE	> { ... } // 1 item	Null	Null	Null	10	WriteSerializable	false	> { ... } // 3 items	Null	Databricks-Runtime/dlt:11.0-delta-pipelines-ed5cc83-e81c5c7-17c692e-custom-local
10	2022-11-08T19:48:31	Null	Null	WRITE	> { ... } // 1 item	Null	Null	Null	9	WriteSerializable	false	> { ... } // 3 items	Null	Databricks-Runtime/dlt:11.0-delta-pipelines-de92f9e-8a33b70-a333f54-custom-local

Figure 12 – View the history of changes made to a resulting Delta Live Tables (DLT) table entity.

We can further use change data capture (CDC) to update tables based on changes in the source datasets. DLT CDC supports updating tables with slow-changing dimensions (SCD) types 1 and 2.

We have one of two options for our batch process to trigger the DLT pipeline. We can use the Databricks **Auto Loader** to incrementally process new data as it arrives in the source tables or create scheduled jobs that trigger at set times or intervals. In this example, we opted for the latter with a scheduled job that executes the DLT pipeline every five minutes.

