DYNAMIC TABLES IN SNOWFLAKE

Data Sheet | LumenData

Snowflake provides a cost-effective, reliable, and automated way to transform our data. Instead of creating a sequential object and monitoring it to transform our data, we can simply define the end objective of the transformation in dynamic tables, then the pipeline management will be managed by Snowflake.

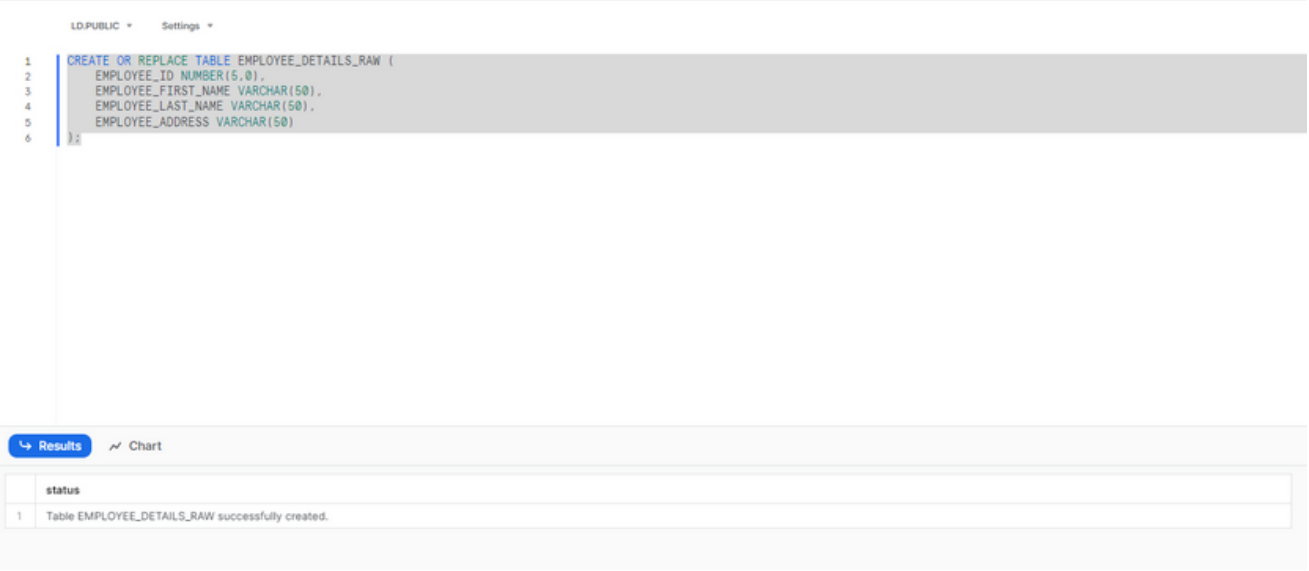
To demonstrate this, let's create two tables and monitor how Change Data Capture is achieved. This can be done in two ways.

- Creating streams and Tasks for Automating Change Data Capture
- Creating Dynamic Tables for Automating Change Data Capture

Creating streams and Tasks for Automating Change Data Capture

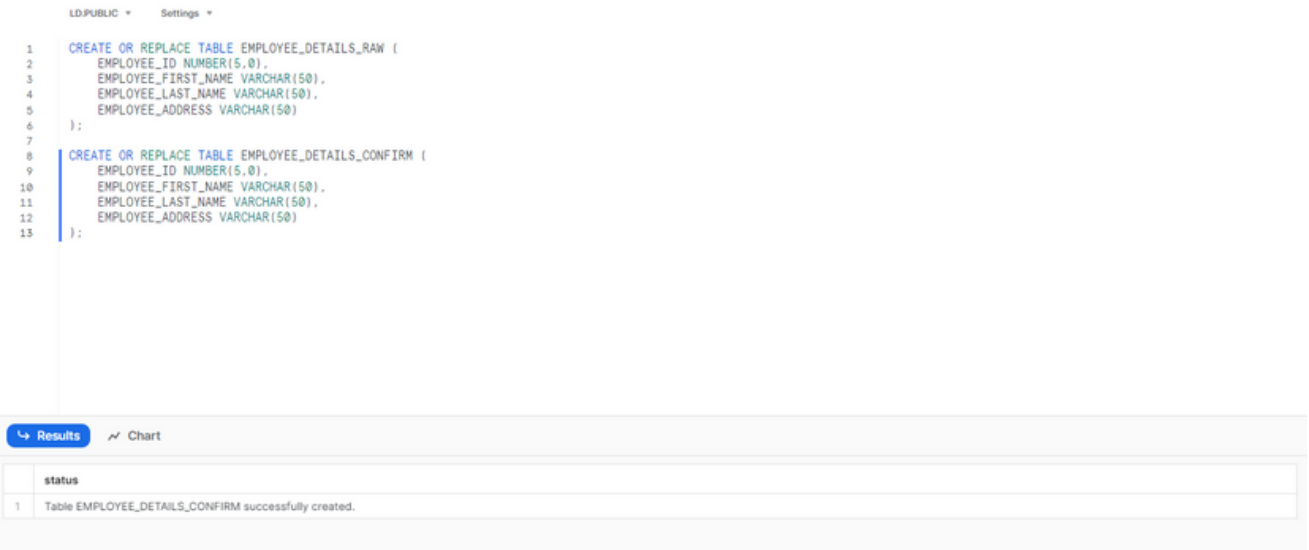
1) Creating a Table named "EMPLOYEE_DETAILS_RAW".

```
CREATE OR REPLACE TABLE EMPLOYEE_DETAILS_RAW (
EMPLOYEE_ID NUMBER(5,0),
EMPLOYEE_FIRST_NAME VARCHAR(50),
EMPLOYEE_LAST_NAME VARCHAR(50),
EMPLOYEE_ADDRESS VARCHAR(50)
);
```



2) Creating a table named “EMPLOYEE_DETAILS_CONFIRM”.

```
CREATE OR REPLACE TABLE EMPLOYEE_DETAILS_CONFIRM (
EMPLOYEE_ID NUMBER(5,0),
EMPLOYEE_FIRST_NAME VARCHAR(50),
EMPLOYEE_LAST_NAME VARCHAR(50),
EMPLOYEE_ADDRESS VARCHAR(50)
);
```



3) Now we try to achieve the Change Data Capture that happened in “EMPLOYEE_DETAILS_RAW” using streams and update the “EMPLOYEE_DETAILS_CONFIRM”.

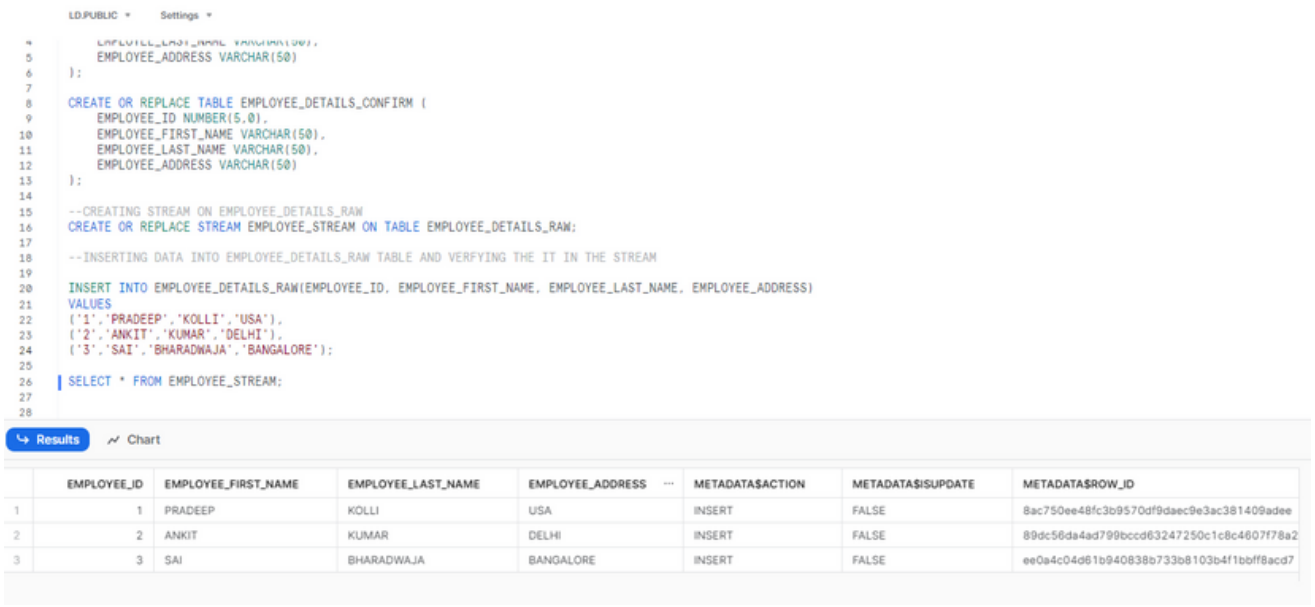
4) Creating Stream named “EMPLOYEE_STREAM” on “EMPLOYEE_DETAILS_RAW”.
CREATE OR REPLACE STREAM EMPLOYEE_STREAM ON TABLE EMPLOYEE_DETAILS_RAW;

5) Inserting the records into “EMPLOYEE_DETAILS_RAW” and verifying the stream.

INSERT INTO EMPLOYEE_DETAILS_RAW(EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
VALUES
('1','PRADEEP','KOLLI','USA'),
('2','ANKIT','KUMAR','DELHI'),
('3','SAI','BHARADWAJA','BANGALORE');

SELECT * FROM EMPLOYEE_STREAM;

6) From the below figure, we can see that the Change Data Capture has been captured by Stream “EMPLOYEE_STREAM”.



```
LD.PUBLIC ▾   Settings ▾

    EMPLOYEE_ADDRESS VARCHAR(50)
);

CREATE OR REPLACE TABLE EMPLOYEE_DETAILS_CONFIRM (
    EMPLOYEE_ID NUMBER(5,0),
    EMPLOYEE_FIRST_NAME VARCHAR(50),
    EMPLOYEE_LAST_NAME VARCHAR(50),
    EMPLOYEE_ADDRESS VARCHAR(50)
);

--CREATING STREAM ON EMPLOYEE_DETAILS_RAW
CREATE OR REPLACE STREAM EMPLOYEE_STREAM ON TABLE EMPLOYEE_DETAILS_RAW;

--INSERTING DATA INTO EMPLOYEE_DETAILS_RAW TABLE AND VERFYING THE IT IN THE STREAM

INSERT INTO EMPLOYEE_DETAILS_RAW(EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
VALUES
('1','PRADEEP','KOLLI','USA'),
('2','ANKIT','KUMAR','DELHI'),
('3','SAI','BHARADWAJA','BANGALORE');

SELECT * FROM EMPLOYEE_STREAM;
```

Results | Chart

	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_ADDRESS	METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
1	1	PRADEEP	KOLLI	USA	INSERT	FALSE	8ac750ee48fc3b9570df9daec9e3ac381409adee
2	2	ANKIT	KUMAR	DELHI	INSERT	FALSE	89dc56da4ad799bccd63247250c1c8c4607f78a2
3	3	SAI	BHARADWAJA	BANGALORE	INSERT	FALSE	ee0a4c04d61b940838b733b8103b4f1bbff8acd7

7) Now we create a Task named “EMPLOYEE_TASK” to automate the Change Data Capture (CDC) captured by the stream and update the “EMPLOYEE_DETAILS_CONFIRM” on a scheduled basis.

```
CREATE OR REPLACE TASK EMPLOYEE_TASK

WAREHOUSE = COMPUTE_WH

SCHEDULE = '1 minute'

WHEN
SYSTEM$STREAM_HAS_DATA('EMPLOYEE_STREAM')
AS
MERGE INTO EMPLOYEE_DETAILS_CONFIRM a
USING (

SELECT * FROM EMPLOYEE_STREAM

) b ON a.EMPLOYEE_ID = b.EMPLOYEE_ID
```

WHEN MATCHED AND b.METADATA$ACTION = 'DELETE' AND b.METADATA$ISUPDATE = 'FALSE' THEN DELETE

WHEN MATCHED AND b.METADATA$ACTION = 'INSERT' AND b.METADATA$ISUPDATE = 'TRUE' THEN

UPDATE SET a.EMPLOYEE_FIRST_NAME = b.EMPLOYEE_FIRST_NAME, a.EMPLOYEE_LAST_NAME = b.EMPLOYEE_LAST_NAME, a.EMPLOYEE_ADDRESS = b.EMPLOYEE_ADDRESS
WHEN NOT MATCHED AND b.METADATA$ACTION = 'INSERT' THEN INSERT (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
VALUES (b.EMPLOYEE_ID, b.EMPLOYEE_FIRST_NAME, b.EMPLOYEE_LAST_NAME, b.EMPLOYEE_ADDRESS);

ALTER TASK EMPLOYEE_TASK RESUME;

8) Once the task was completed, we can see the “EMPLOYEE_DETAILS_CONFIRM” has the data init.

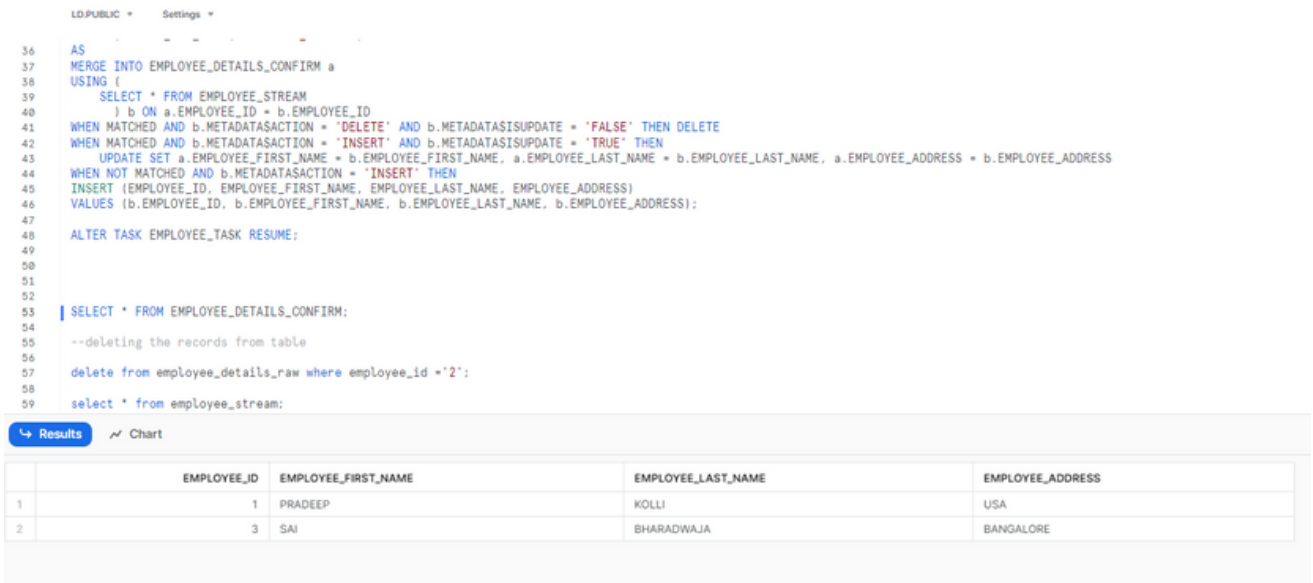
```
CREATE OR REPLACE TASK EMPLOYEE_TASK
WAREHOUSE = COMPUTE_WH
SCHEDULE = '1 minute'
WHEN
SYSTEM$STREAM_HAS_DATA('EMPLOYEE_STREAM')
AS
MERGE INTO EMPLOYEE_DETAILS_CONFIRM a
USING (
    SELECT * FROM EMPLOYEE_STREAM
    ) b ON a.EMPLOYEE_ID = b.EMPLOYEE_ID
WHEN MATCHED AND b.METADATA$ACTION = 'DELETE' AND b.METADATA$ISUPDATE = 'FALSE' THEN DELETE
WHEN MATCHED AND b.METADATA$ACTION = 'INSERT' AND b.METADATA$ISUPDATE = 'TRUE' THEN
    UPDATE SET a.EMPLOYEE_FIRST_NAME = b.EMPLOYEE_FIRST_NAME, a.EMPLOYEE_LAST_NAME = b.EMPLOYEE_LAST_NAME, a.EMPLOYEE_ADDRESS = b.EMPLOYEE_ADDRESS
WHEN NOT MATCHED AND b.METADATA$ACTION = 'INSERT' THEN
INSERT (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
VALUES (b.EMPLOYEE_ID, b.EMPLOYEE_FIRST_NAME, b.EMPLOYEE_LAST_NAME, b.EMPLOYEE_ADDRESS);

ALTER TASK EMPLOYEE_TASK RESUME;



SELECT * FROM EMPLOYEE_DETAILS_CONFIRM;
```

Results | Chart

	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_ADDRESS
1	1	PRADEEP	KOLLI	USA
2	2	ANKIT	KUMAR	DELHI
3	3	SAI	BHARADWAJA	BANGALORE

9) Now let’s perform some delete operation on “EMPLOYEE_DETAILS_RAW” and verify the “EMPLOYEE_DETAILS_CONFIRM”

delete from employee_details_raw where employee_id ='2';
select * from employee_stream;
select * from employee_details_confirm;



```
AS
MERGE INTO EMPLOYEE_DETAILS_CONFIRM a
USING (
    SELECT * FROM EMPLOYEE_STREAM
    ) b ON a.EMPLOYEE_ID = b.EMPLOYEE_ID
WHEN MATCHED AND b.METADATA$ACTION = 'DELETE' AND b.METADATA$ISUPDATE = 'FALSE' THEN DELETE
WHEN MATCHED AND b.METADATA$ACTION = 'INSERT' AND b.METADATA$ISUPDATE = 'TRUE' THEN
    UPDATE SET a.EMPLOYEE_FIRST_NAME = b.EMPLOYEE_FIRST_NAME, a.EMPLOYEE_LAST_NAME = b.EMPLOYEE_LAST_NAME, a.EMPLOYEE_ADDRESS = b.EMPLOYEE_ADDRESS
WHEN NOT MATCHED AND b.METADATA$ACTION = 'INSERT' THEN
INSERT (EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
VALUES (b.EMPLOYEE_ID, b.EMPLOYEE_FIRST_NAME, b.EMPLOYEE_LAST_NAME, b.EMPLOYEE_ADDRESS);

ALTER TASK EMPLOYEE_TASK RESUME;



SELECT * FROM EMPLOYEE_DETAILS_CONFIRM;

--deleting the records from table

delete from employee_details_raw where employee_id ='2';

select * from employee_stream;
```

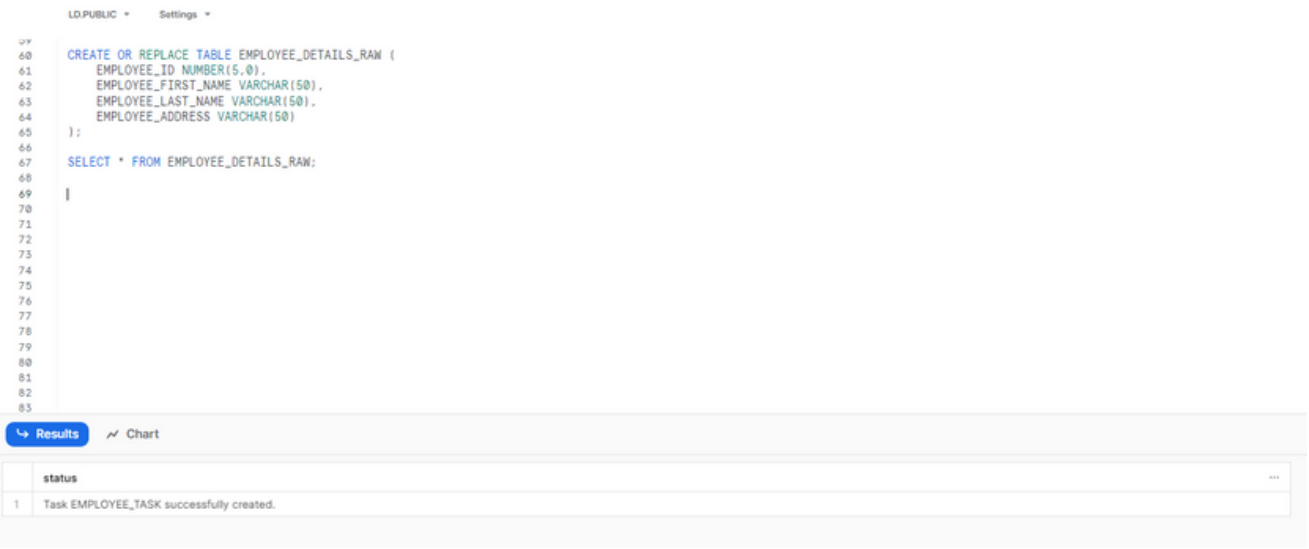
Results | Chart

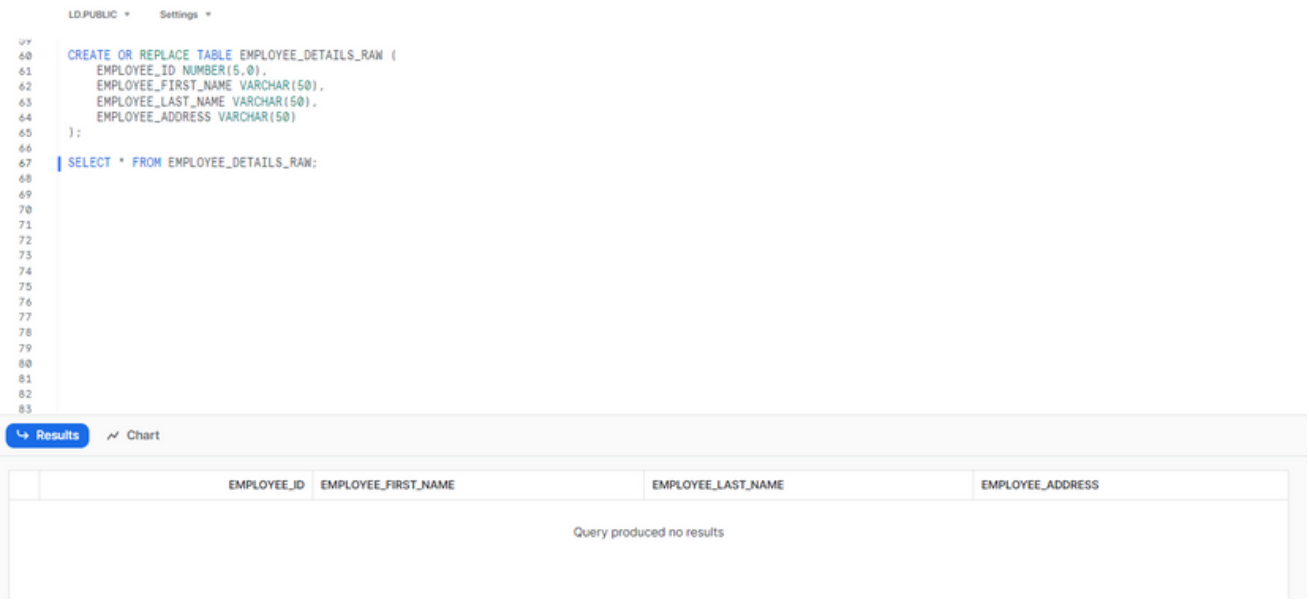
	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_ADDRESS
1	1	PRADEEP	KOLLI	USA
2	3	SAI	BHARADWAJA	BANGALORE

Creating Dynamic Tables for Automating Change Data Capture

10) Creating a Table named “EMPLOYEE_DETAILS_RAW” and view the data in the table.

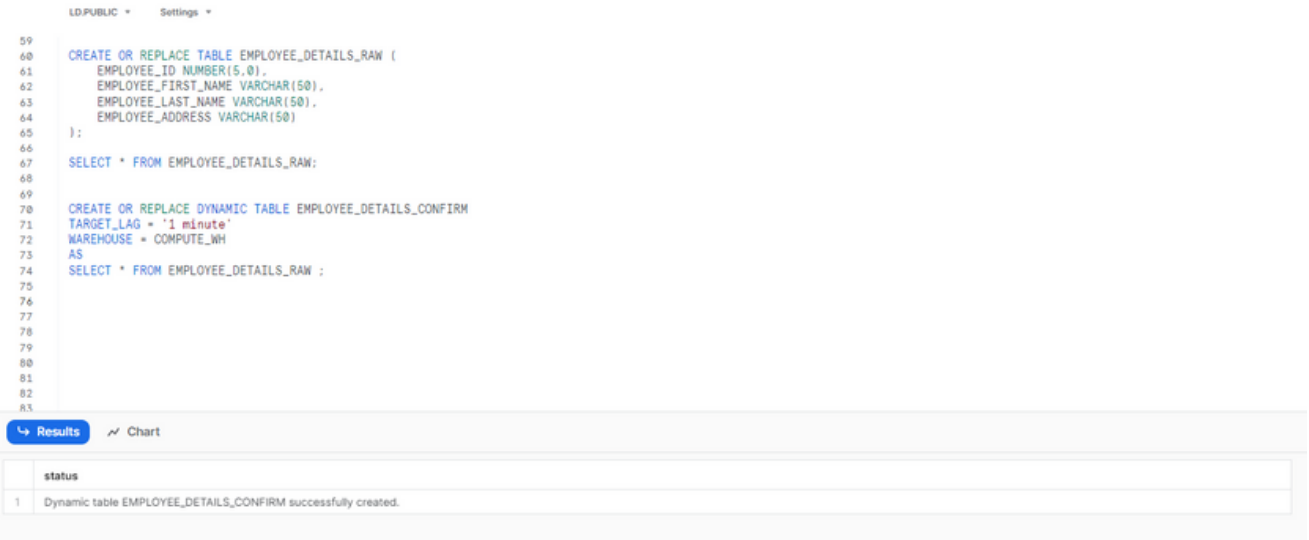
```
CREATE OR REPLACE TABLE EMPLOYEE_DETAILS_RAW (
EMPLOYEE_ID NUMBER(5,0),
EMPLOYEE_FIRST_NAME VARCHAR(50),
EMPLOYEE_LAST_NAME VARCHAR(50),
EMPLOYEE_ADDRESS VARCHAR(50)
);
SELECT * FROM EMPLOYEE_DETAILS_RAW;
```





11) Creating a Dynamic table on a schedule basis to capture the Change Data Capture and update it.

```
CREATE OR REPLACE DYNAMIC TABLE
EMPLOYEE_DETAILS_CONFIRM
TARGET_LAG = '1 minute'
WAREHOUSE = COMPUTE_WH
AS
SELECT * FROM EMPLOYEE_DETAILS_RAW ;
```



12) Now let’s insert the records in “EMPLOYEE_DETAILS_RAW” and verify the Dynamic table “EMPLOYEE_DETAILS_CONFIRM” Table.

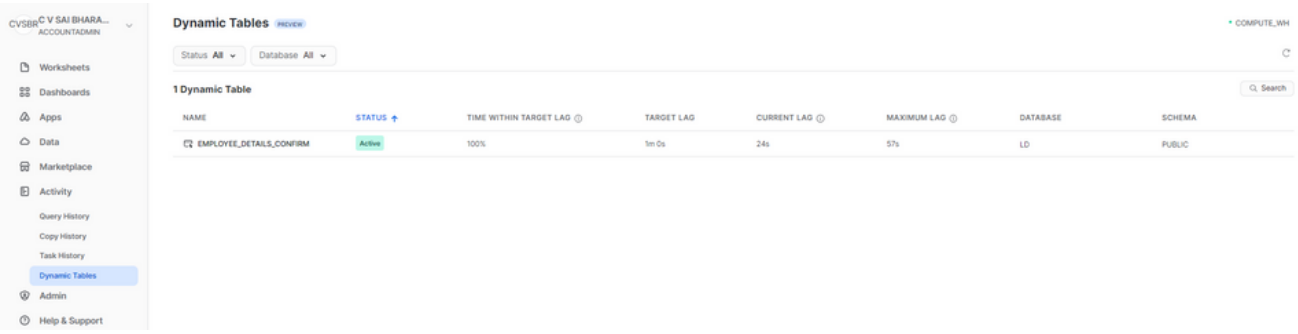
```
INSERT INTO EMPLOYEE_DETAILS_RAW(EMPLOYEE_ID,
EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME,
EMPLOYEE_ADDRESS)
VALUES
('1','PRADEEP','KOLLI','USA'),
('2','ANKIT','KUMAR','DELHI'),
('3','SAI','BHARADWAJA','BANGALORE');

SELECT * FROM EMPLOYEE_DETAILS_CONFIRM;
```

```
LD.PUBLIC   Settings

59
60  CREATE OR REPLACE TABLE EMPLOYEE_DETAILS_RAW (
61      EMPLOYEE_ID NUMBER(5,0),
62      EMPLOYEE_FIRST_NAME VARCHAR(50),
63      EMPLOYEE_LAST_NAME VARCHAR(50),
64      EMPLOYEE_ADDRESS VARCHAR(50)
65  );
66
67  SELECT * FROM EMPLOYEE_DETAILS_RAW;
68
69  CREATE OR REPLACE DYNAMIC TABLE EMPLOYEE_DETAILS_CONFIRM
70  TARGET_LAG = '1 minute'
71  WAREHOUSE = COMPUTE_WH
72  AS
73  SELECT * FROM EMPLOYEE_DETAILS_RAW ;
74
75
76  INSERT INTO EMPLOYEE_DETAILS_RAW(EMPLOYEE_ID, EMPLOYEE_FIRST_NAME, EMPLOYEE_LAST_NAME, EMPLOYEE_ADDRESS)
77  VALUES
78  ('1','PRADEEP','KOLLI','USA'),
79  ('2','ANKIT','KUMAR','DELHI'),
80  ('3','SAI','BHARADWAJA','BANGALORE');
81
82  SELECT * FROM EMPLOYEE_DETAILS_CONFIRM;
```

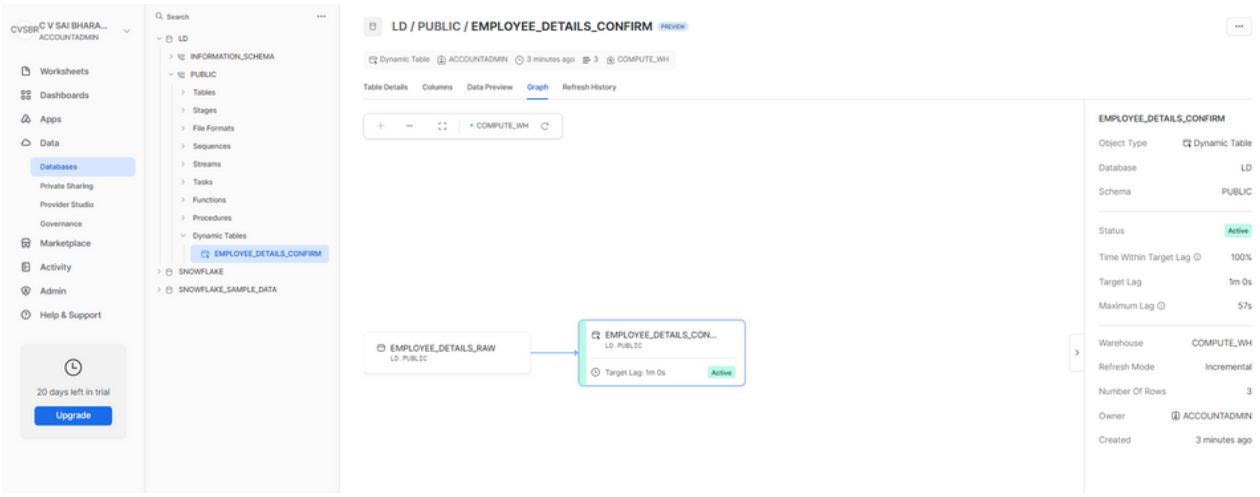
Results | Chart

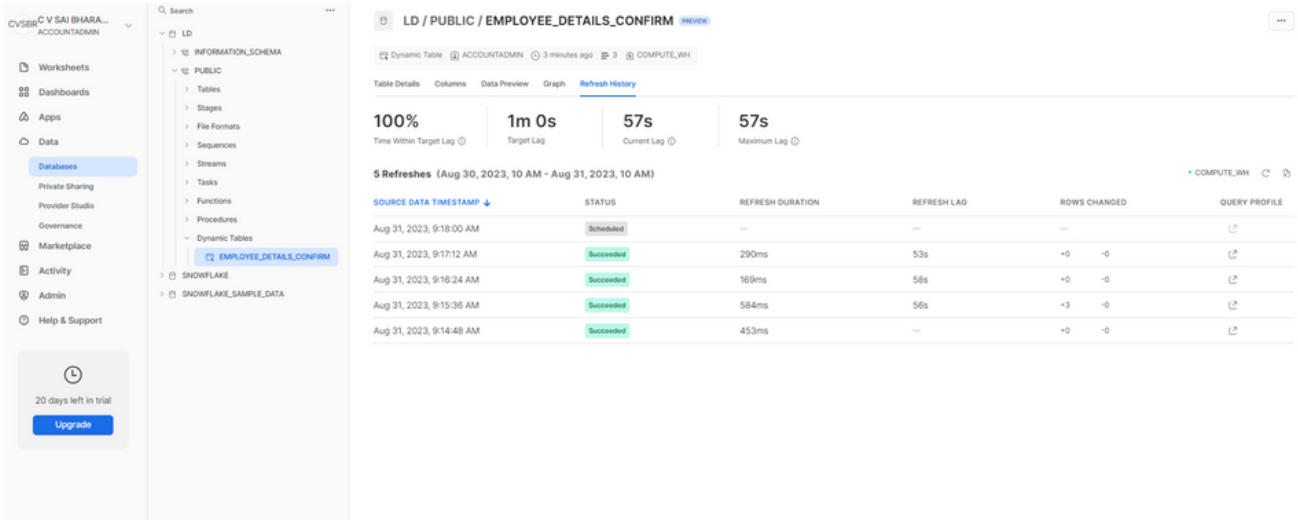
	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_ADDRESS
1	1	PRADEEP	KOLLI	USA
2	2	ANKIT	KUMAR	DELHI
3	3	SAI	BHARADWAJA	BANGALORE

13) We can view the ‘Dynamic Table History’ under ‘Activity Tab’ as shown below from Snowflake UI.

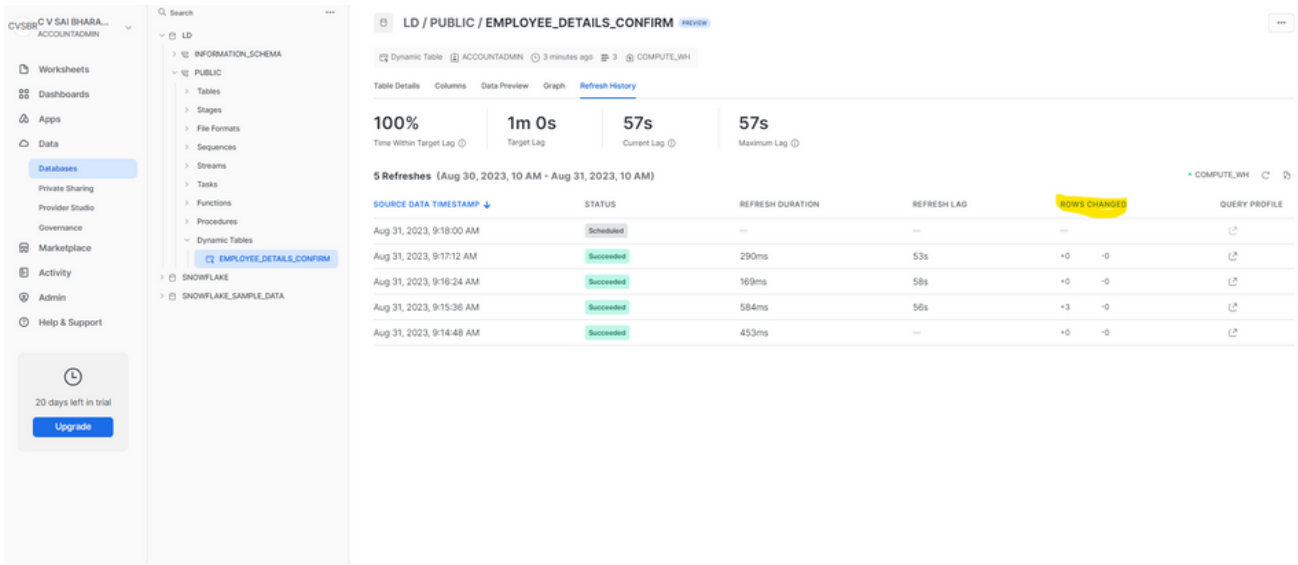


14) When we click the Dynamic Table, we see the Graph of it and the refresh history as shown in below figures.





15) We can see the number of records that got changed or deleted from the refresh history based on the job scheduled as shown in the below figure.



We can conclude that dynamic tables update the results of the query without creating separate objects, writing the code, scheduling tasks, or monitoring them.

ABOUT LUMENDATA:

LumenData is a leading provider of Enterprise Data Management, Cloud & Analytics solutions. We help businesses navigate their data visualization and analytics anxieties and enable them to accelerate their innovation journeys. Founded in 2008, with locations in multiple countries, LumenData is privileged to serve over 100 leading companies, including KwikTrip, Versant Health, US Food & Drug Administration, US Department of Labor, Cummins Engine, BCG, and others. LumenData is SOC2 certified and has instituted extensive controls to protect client data, including adherence to GDPR and CCPA regulations.

Get in touch to discuss how we can facilitate data-driven transformation for your organization.

MEET OUR AUTHORS

Sai Bharadwaja Reddy
Consultant

Ankit Kumar
Technical Lead