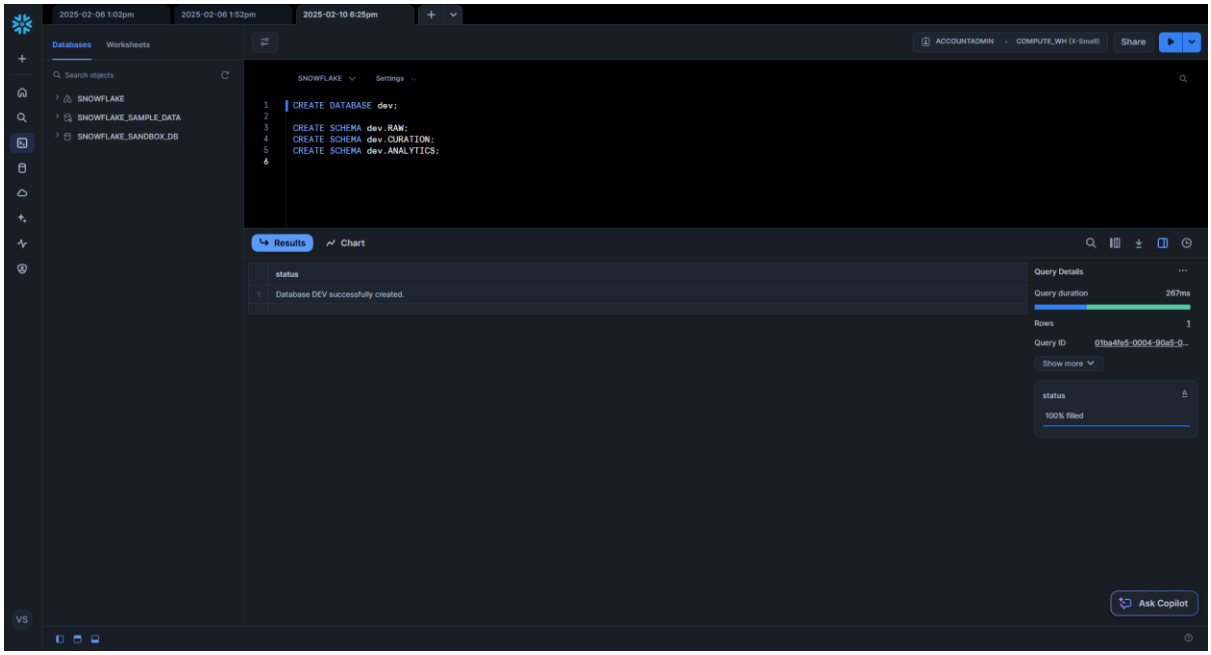
DATA 226- DATAWAREHOUSE

Homework 2

Name : Vimalanandhan Sivanandham

SJSU ID: 017596436

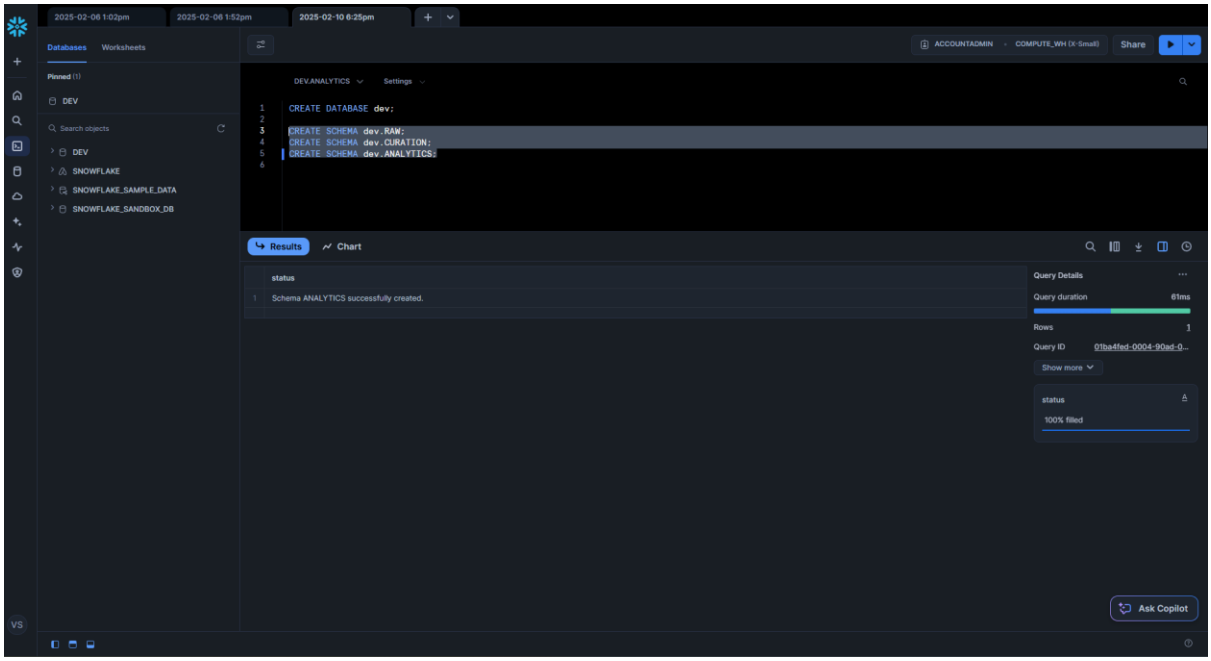
1.(+1) Create database dev and schemas RAW, CURATION and ANALYTICS



CODE:

--- DROP DATABASE DEV

CREATE DATABASE dev;



CODE:

--- CREATE SCHEMA

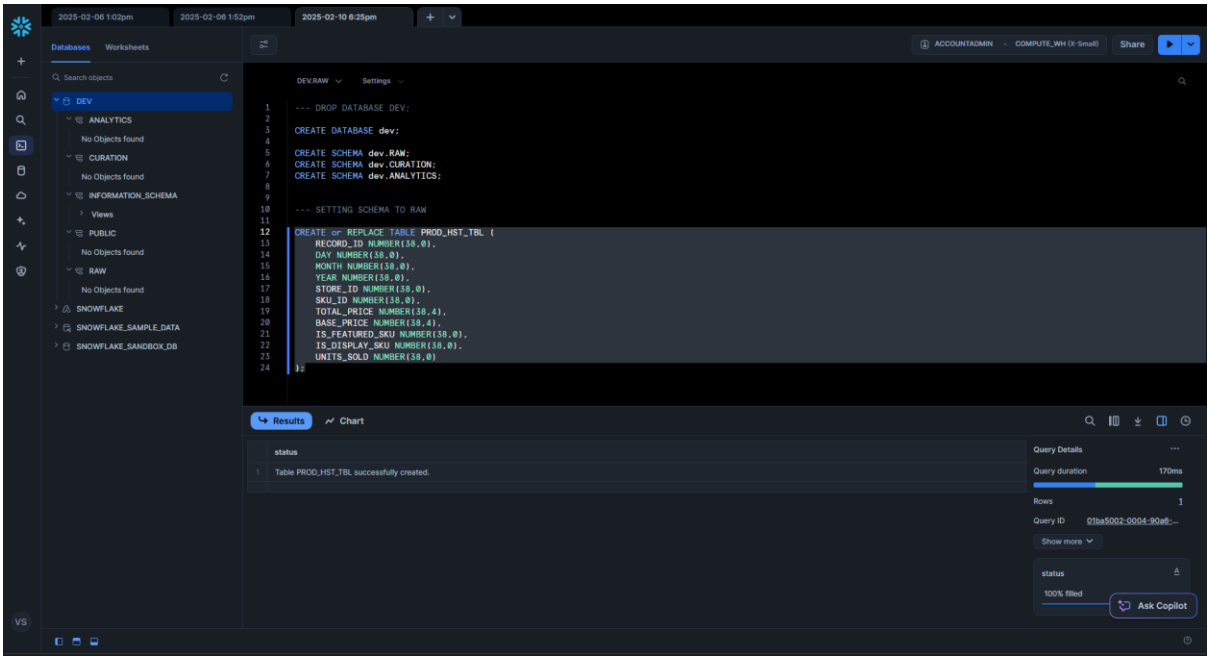
CREATE SCHEMA dev.RAW;

CREATE SCHEMA dev.CURATION;

CREATE SCHEMA dev.ANALYTICS;

2.(+4) In RAW schema, create

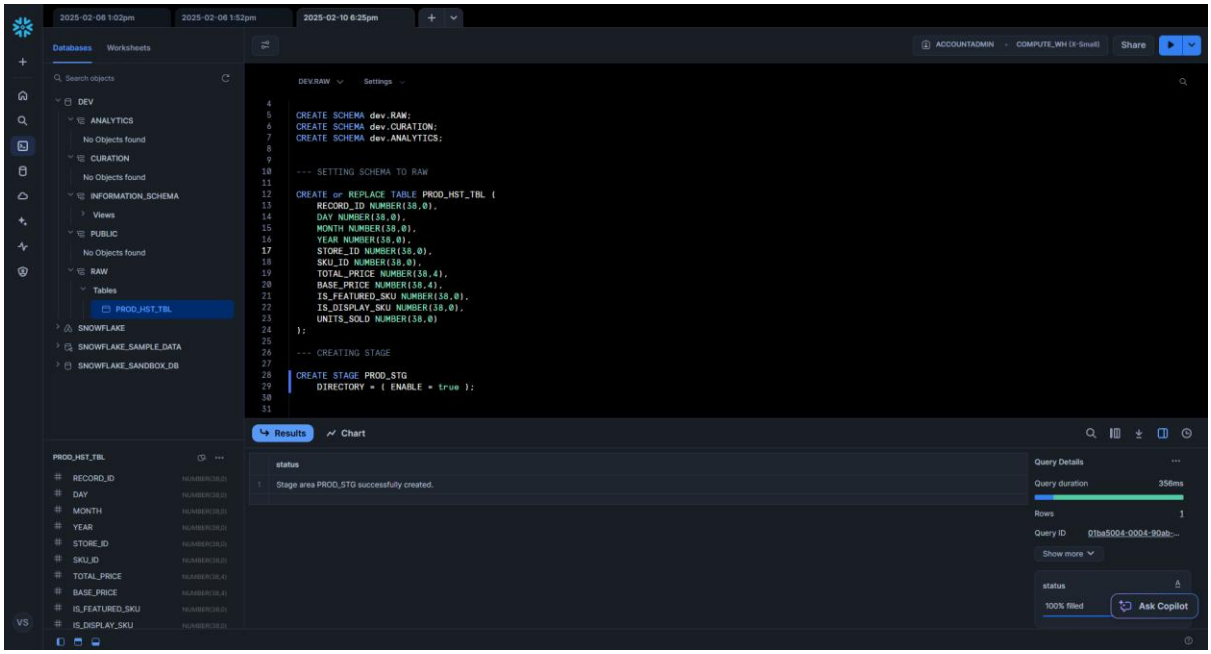
- **prod_hst_tbl - table**
- **prod_stg - stage**
- **prod_raw_task - task (alter the task)**
- **prod_stream – stream**



CODE:

--- SETTING SCHEMA TO RAW

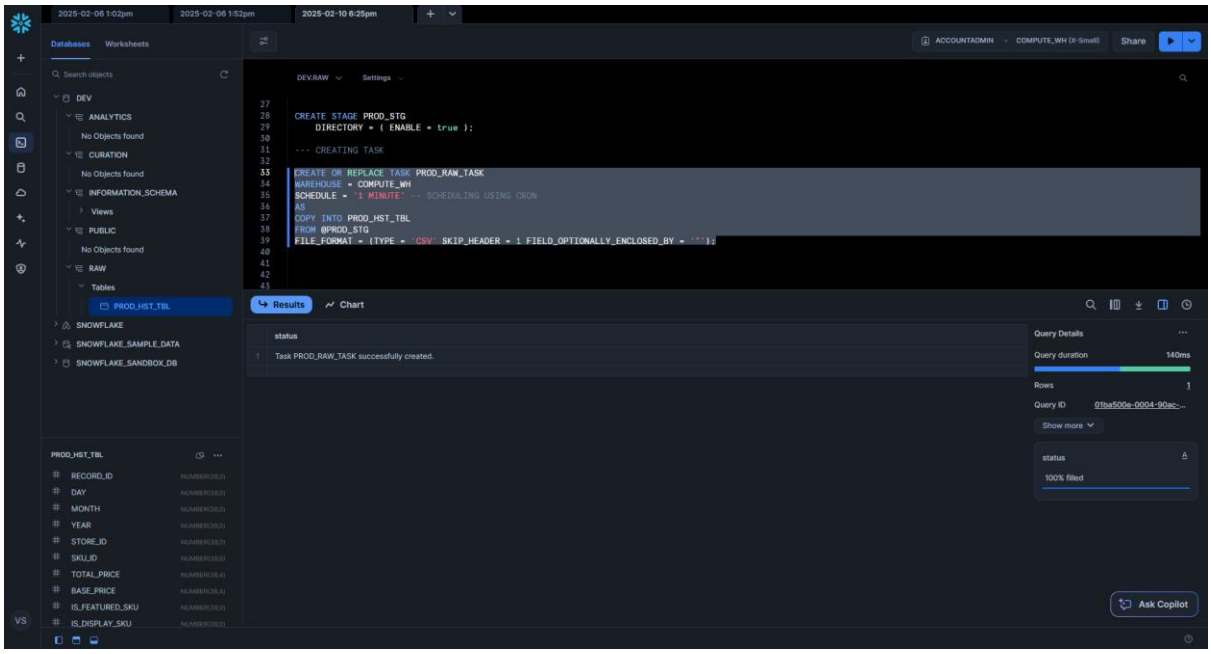
```
CREATE or REPLACE TABLE PROD_HST_TBL (
   RECORD_ID NUMBER(38,0),
   DAY NUMBER(38,0),
   MONTH NUMBER(38,0),
   YEAR NUMBER(38,0),
   STORE_ID NUMBER(38,0),
   SKU_ID NUMBER(38,0),
   TOTAL_PRICE NUMBER(38,4),
   BASE_PRICE NUMBER(38,4),
   IS_FEATURED_SKU NUMBER(38,0),
   IS_DISPLAY_SKU NUMBER(38,0),
   UNITS_SOLD NUMBER(38,0)
);
```



CODE:

--- CREATING STAGE

```
CREATE STAGE PROD_STG
   DIRECTORY = ( ENABLE = true );
```



CODE:

--- CREATING TASK

```
CREATE OR REPLACE TASK PROD_RAW_TASK
```

WAREHOUSE = COMPUTE_WH

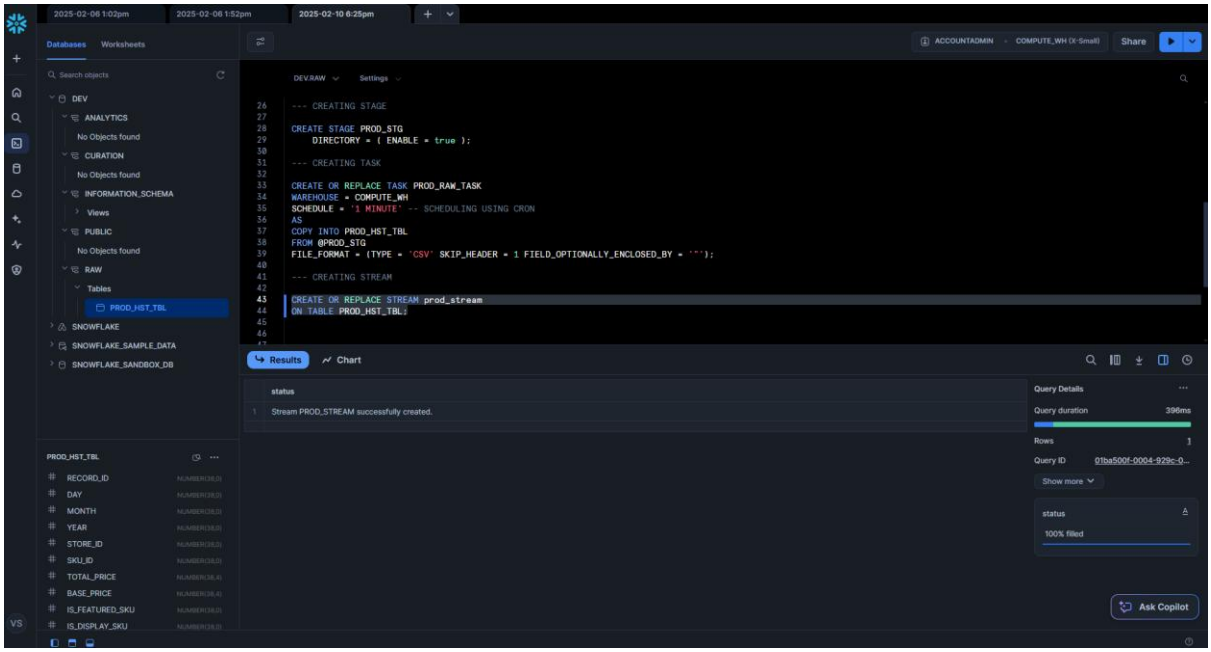
SCHEDULE = '1 MINUTE' -- SCHEDULING USING CRON

AS

COPY INTO PROD_HST_TBL

FROM @PROD_STG

FILE_FORMAT = (TYPE = 'CSV' SKIP_HEADER = 1 FIELD_OPTIONALLY_ENCLOSED_BY = '"');



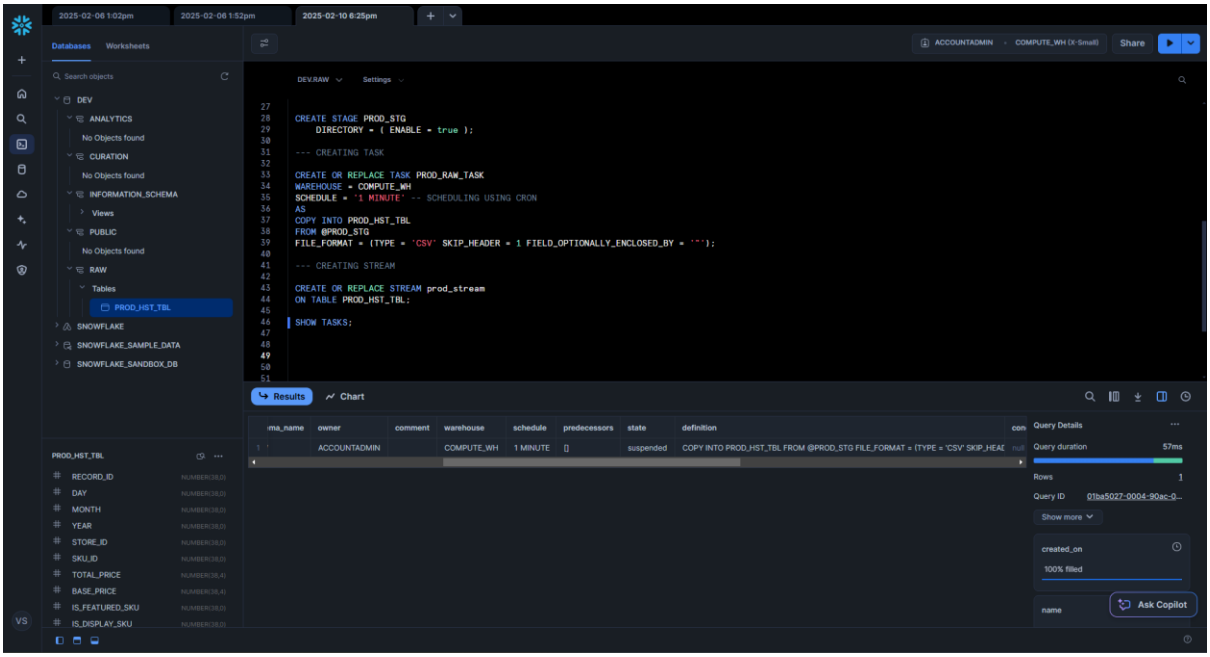
CODE:

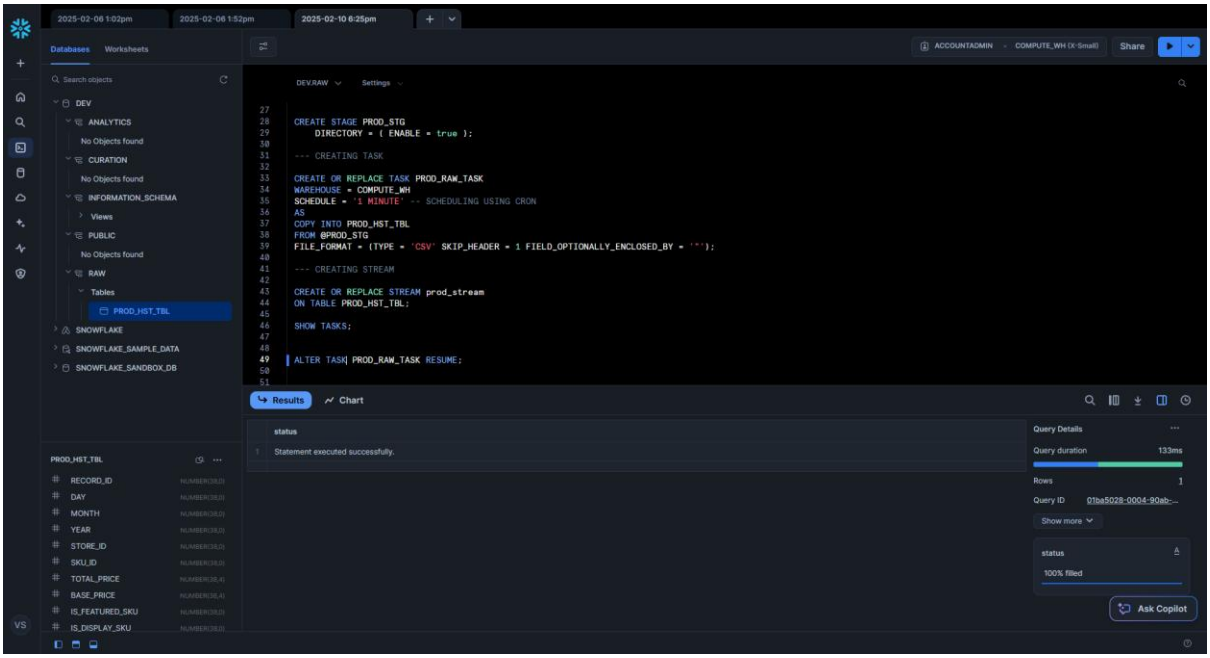
--- CREATING STREAM

CREATE OR REPLACE STREAM prod_stream

ON TABLE PROD_HST_TBL;

SHOW TASKS;

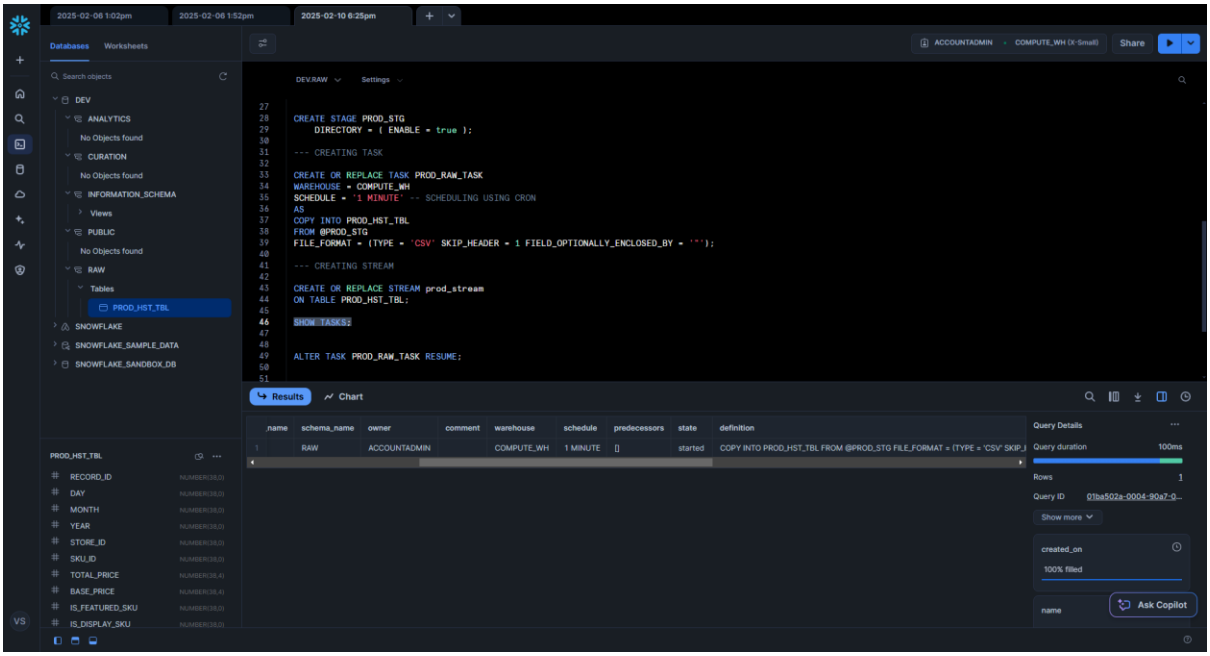


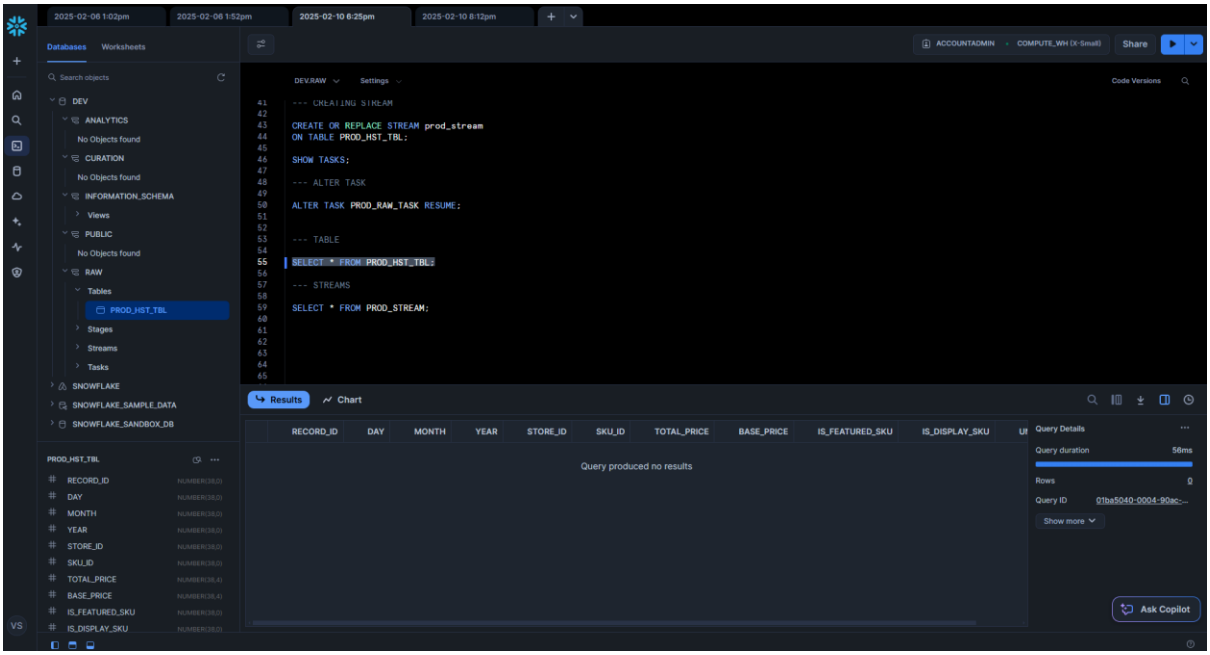


CODE:

--- ALTER TASK

ALTER TASK PROD_RAW_TASK RESUME;

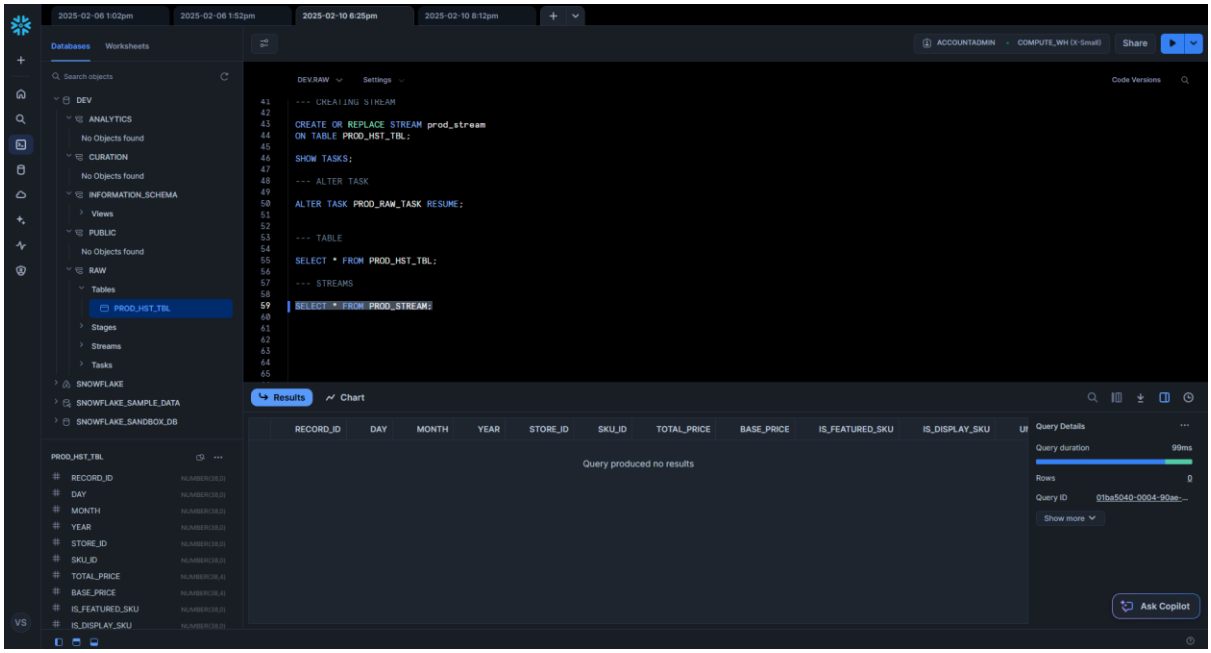




CODE:

--- TABLE

SELECT * FROM PROD_HST_TBL;



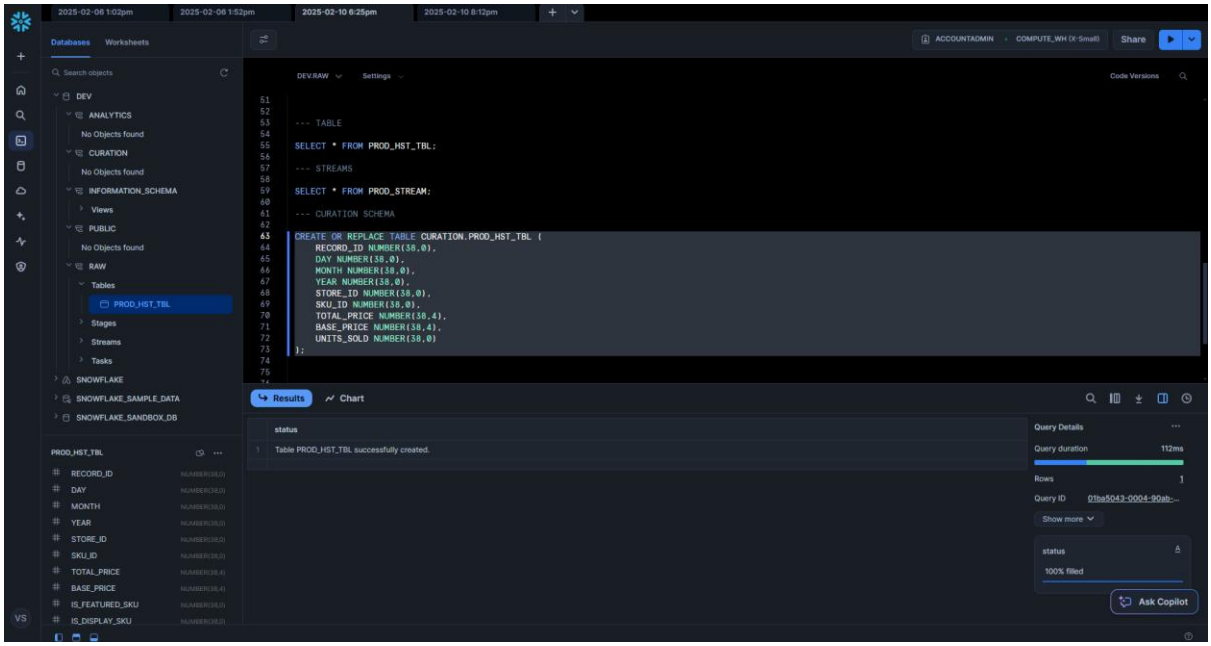
CODE:

--- STREAMS

SELECT * FROM PROD_STREAM;

3.(+3) In CURATION schema, create

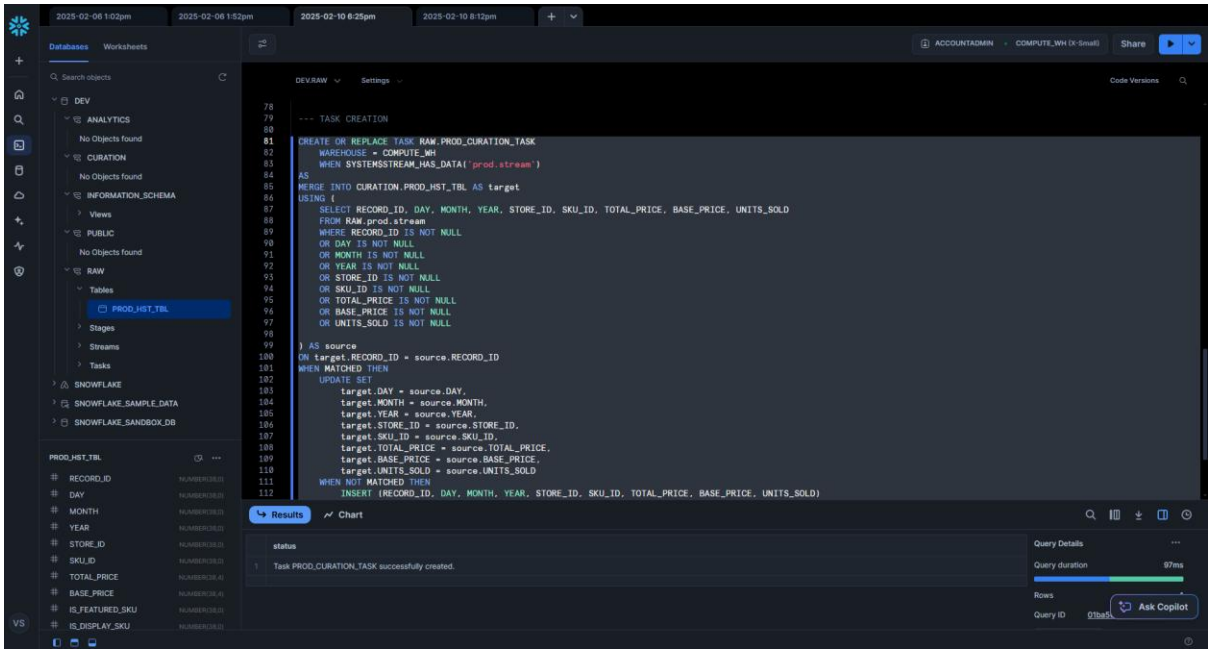
- **prod_hst_tbl - table**
- **prod_curation_task using MERGE (alter the task)**

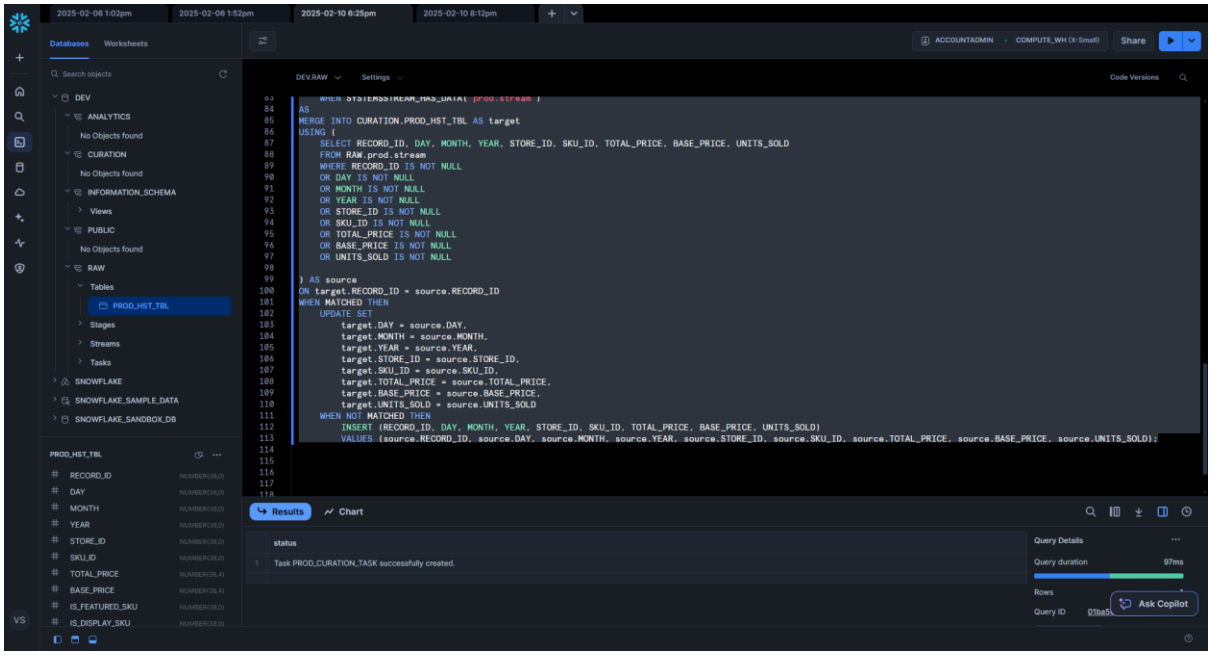


CODE:

--- CURATION SCHEMA

```
CREATE OR REPLACE TABLE CURATION.PROD_HST_TBL (
   RECORD_ID NUMBER(38,0),
   DAY NUMBER(38,0),
   MONTH NUMBER(38,0),
   YEAR NUMBER(38,0),
   STORE_ID NUMBER(38,0),
   SKU_ID NUMBER(38,0),
   TOTAL_PRICE NUMBER(38,4),
   BASE_PRICE NUMBER(38,4),
   UNITS_SOLD NUMBER(38,0)
);
```





CODE:

--- TASK CREATION

CREATE OR REPLACE TASK RAW.PROD_CURATION_TASK

WAREHOUSE = COMPUTE_WH

WHEN SYSTEM$STREAM_HAS_DATA('prod.stream')

AS

MERGE INTO CURATION.PROD_HST_TBL AS target

USING (

SELECT RECORD_ID, DAY, MONTH, YEAR, STORE_ID, SKU_ID, TOTAL_PRICE, BASE_PRICE, UNITS_SOLD

FROM RAW.prod.stream

WHERE RECORD_ID IS NOT NULL

OR DAY IS NOT NULL

OR MONTH IS NOT NULL

OR YEAR IS NOT NULL

OR STORE_ID IS NOT NULL

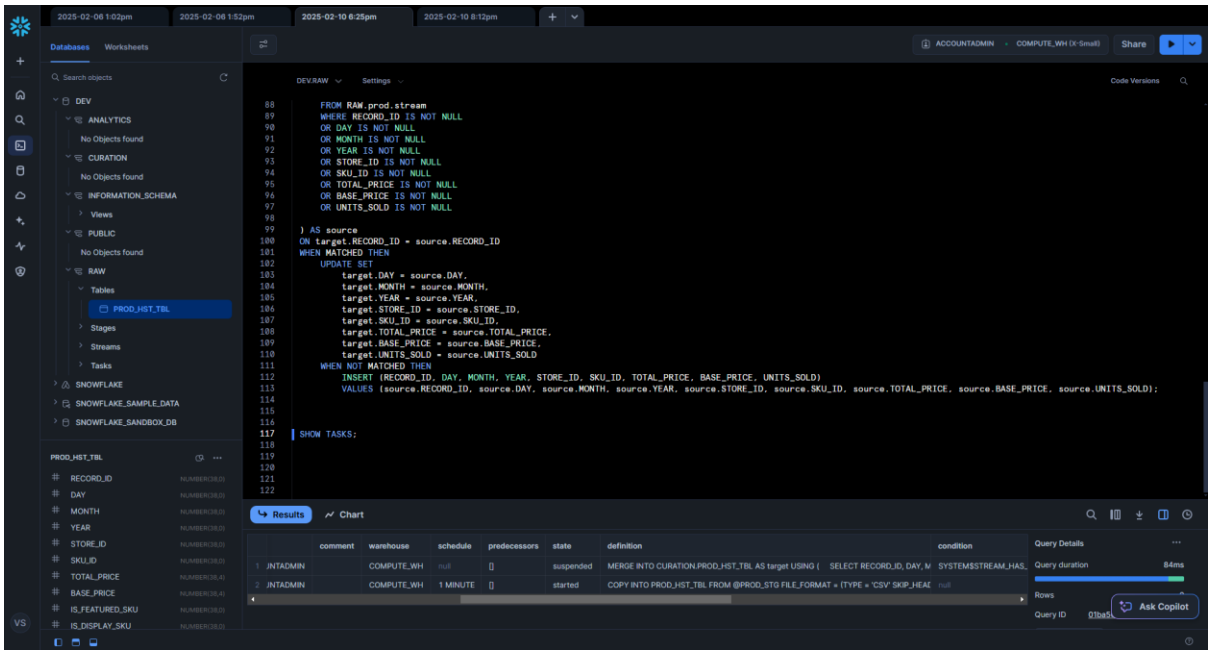
OR SKU_ID IS NOT NULL

OR TOTAL_PRICE IS NOT NULL

OR BASE_PRICE IS NOT NULL

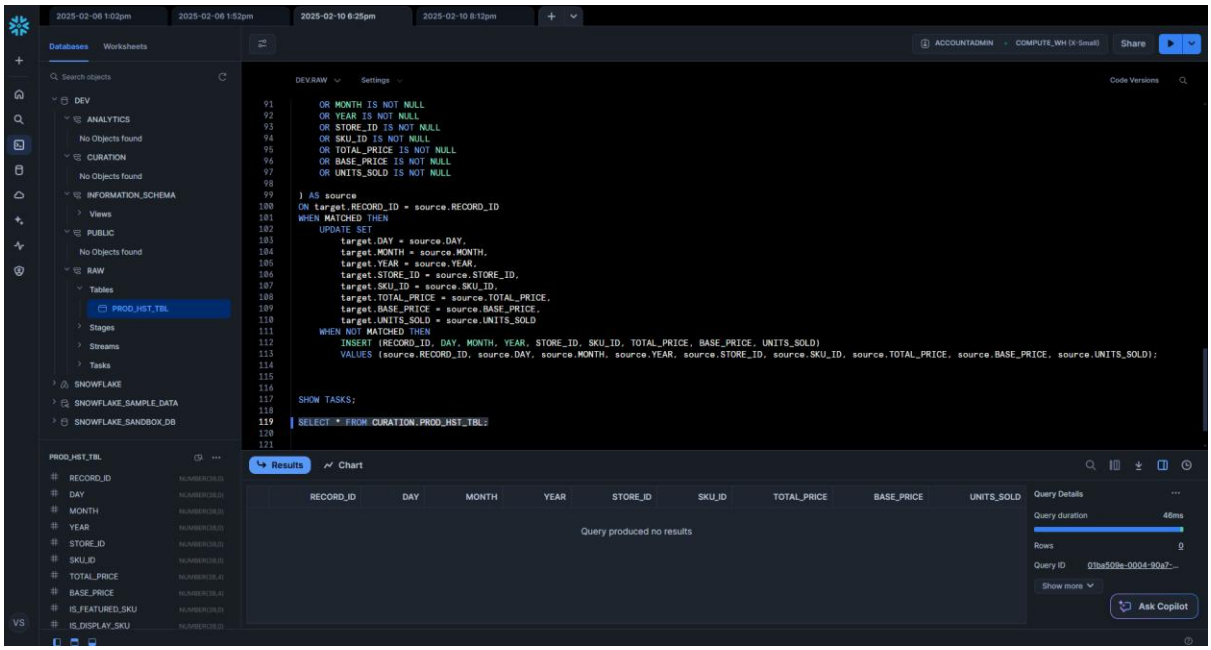
```
   OR UNITS_SOLD IS NOT NULL


) AS source
ON target.RECORD_ID = source.RECORD_ID
WHEN MATCHED THEN
   UPDATE SET
      target.DAY = source.DAY,
      target.MONTH = source.MONTH,
      target.YEAR = source.YEAR,
      target.STORE_ID = source.STORE_ID,
      target.SKU_ID = source.SKU_ID,
      target.TOTAL_PRICE = source.TOTAL_PRICE,
      target.BASE_PRICE = source.BASE_PRICE,
      target.UNITS_SOLD = source.UNITS_SOLD
   WHEN NOT MATCHED THEN
      INSERT (RECORD_ID, DAY, MONTH, YEAR, STORE_ID, SKU_ID, TOTAL_PRICE,
BASE_PRICE, UNITS_SOLD)
      VALUES (source.RECORD_ID, source.DAY, source.MONTH, source.YEAR,
source.STORE_ID, source.SKU_ID, source.TOTAL_PRICE, source.BASE_PRICE,
source.UNITS_SOLD);
```



CODE:

SHOW TASKS;

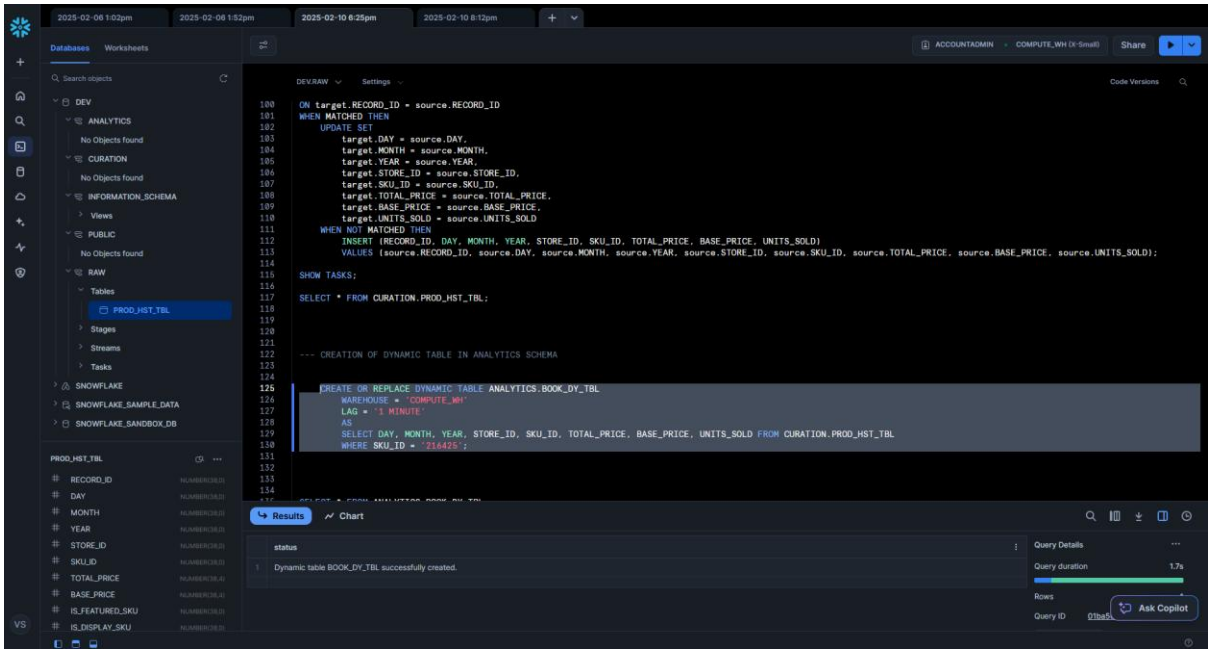


CODE:

SELECT * FROM CURATION.PROD_HST_TBL;

4.(+2) In ANALYTICS schema, create

- **book_dy_tbl – Dynamic Table**

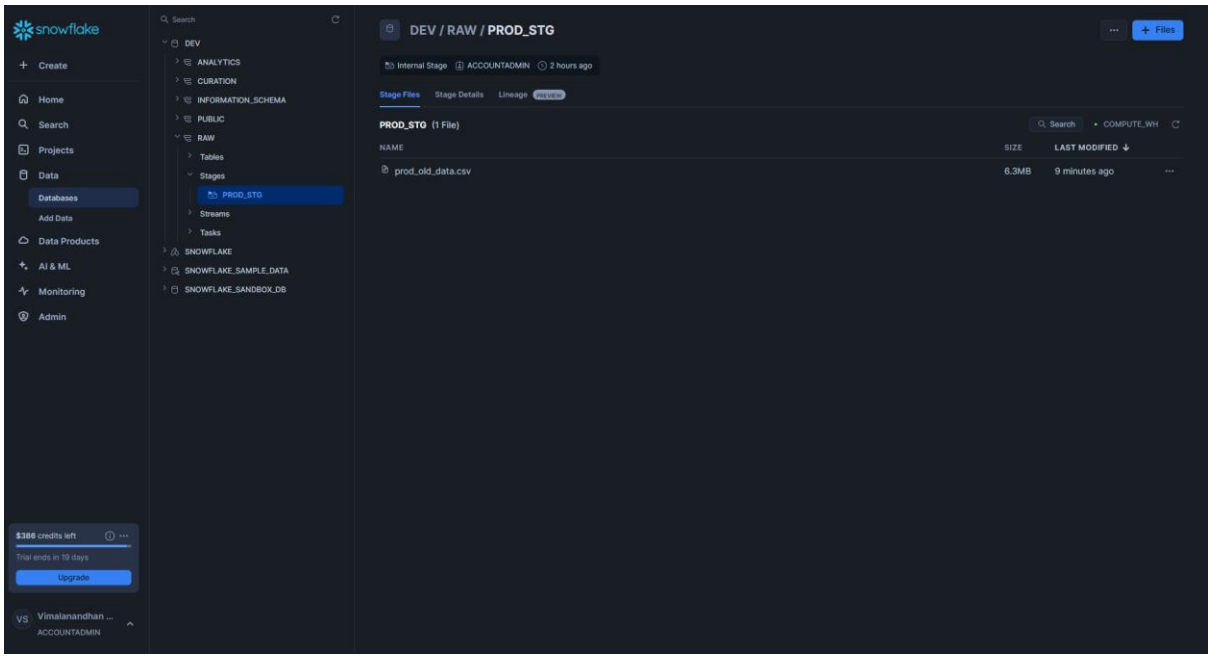


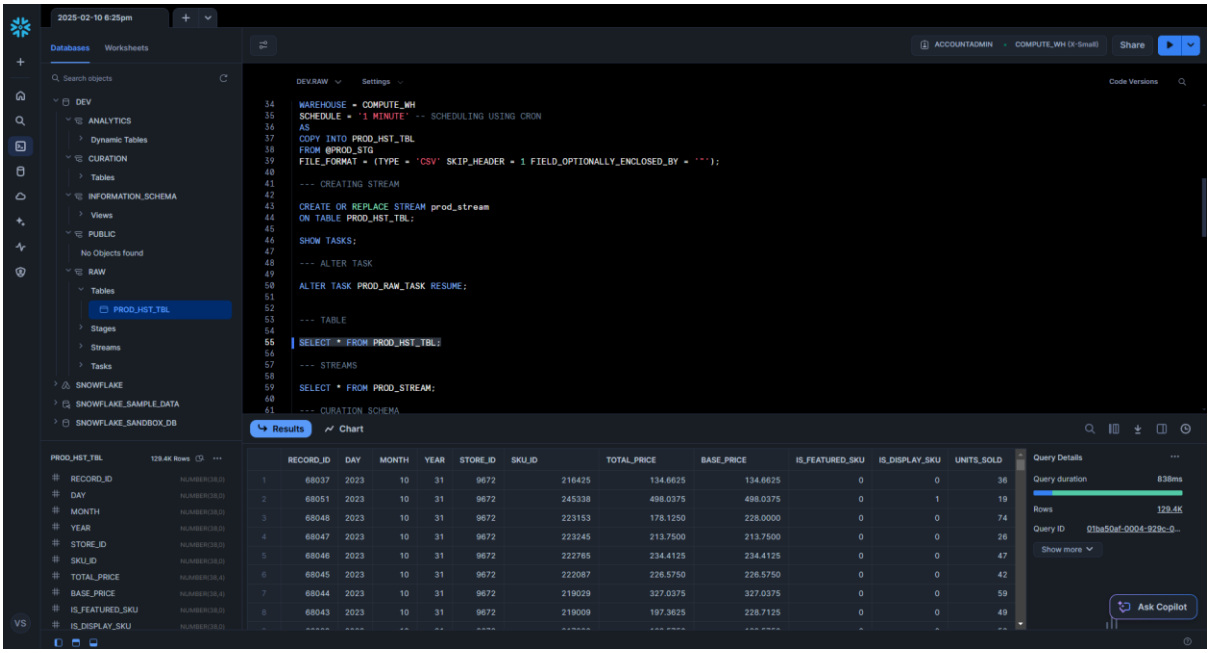
CODE:

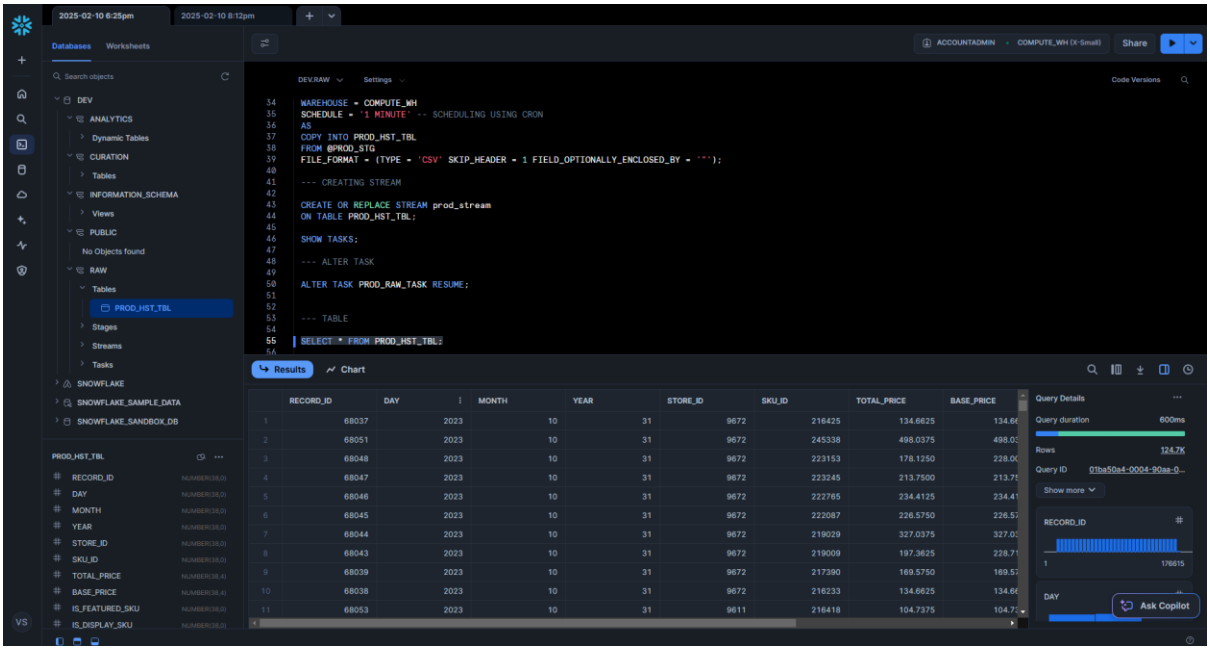
--- CREATION OF DYNAMIC TABLE IN ANALYTICS SCHEMA

```
CREATE OR REPLACE DYNAMIC TABLE ANALYTICS.BOOK_DY_TBL
    WAREHOUSE = 'COMPUTE_WH'
    LAG = '1 MINUTE'
    AS
    SELECT DAY, MONTH, YEAR, STORE_ID, SKU_ID, TOTAL_PRICE, BASE_PRICE,
UNITS_SOLD FROM CURATION.PROD_HST_TBL
    WHERE SKU_ID = '216425';
```

Uploading the old_csv_file:





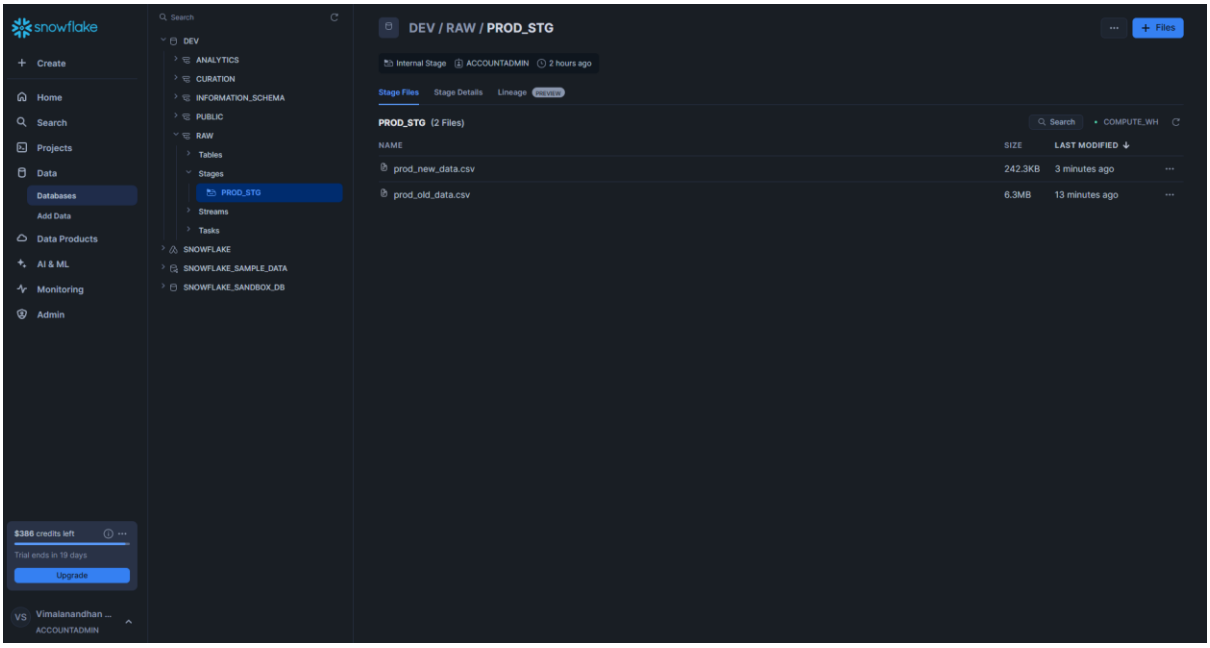


--- TASK CREATION

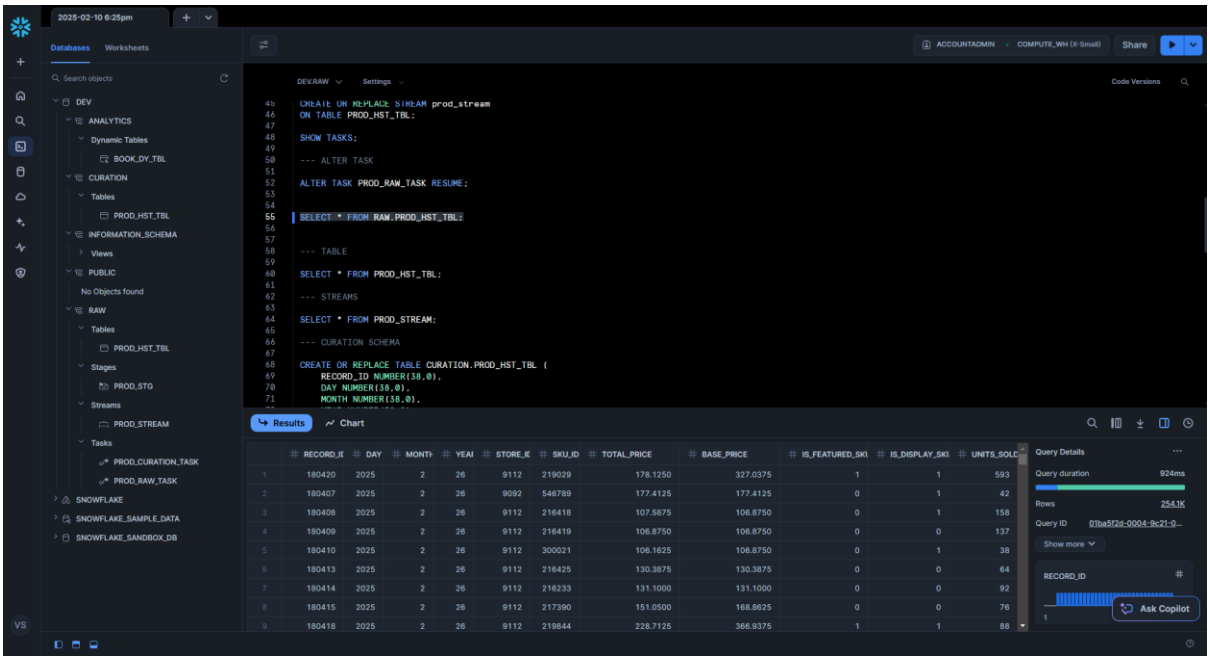
```
CREATE OR REPLACE TASK RAW.PROD_CURATION_TASK
   WAREHOUSE = COMPUTE_WH
   WHEN SYSTEM$STREAM_HAS_DATA('RAW.PROD_STREAM')
AS
MERGE INTO CURATION.PROD_HST_TBL AS target
USING (
```

```
  SELECT RECORD_ID, DAY, MONTH, YEAR, STORE_ID, SKU_ID, TOTAL_PRICE,
BASE_PRICE, UNITS_SOLD
  FROM RAW.prod.stream
  WHERE RECORD_ID IS NOT NULL
  OR DAY IS NOT NULL
  OR MONTH IS NOT NULL
  OR YEAR IS NOT NULL
  OR STORE_ID IS NOT NULL
  OR SKU_ID IS NOT NULL
  OR TOTAL_PRICE IS NOT NULL
  OR BASE_PRICE IS NOT NULL
  OR UNITS_SOLD IS NOT NULL

) AS source
ON target.RECORD_ID = source.RECORD_ID
WHEN MATCHED THEN
  UPDATE SET
    target.DAY = source.DAY,
    target.MONTH = source.MONTH,
    target.YEAR = source.YEAR,
    target.STORE_ID = source.STORE_ID,
    target.SKU_ID = source.SKU_ID,
    target.TOTAL_PRICE = source.TOTAL_PRICE,
    target.BASE_PRICE = source.BASE_PRICE,
    target.UNITS_SOLD = source.UNITS_SOLD
  WHEN NOT MATCHED THEN
    INSERT (RECORD_ID, DAY, MONTH, YEAR, STORE_ID, SKU_ID, TOTAL_PRICE,
BASE_PRICE, UNITS_SOLD)
    VALUES (source.RECORD_ID, source.DAY, source.MONTH, source.YEAR,
source.STORE_ID, source.SKU_ID, source.TOTAL_PRICE, source.BASE_PRICE,
source.UNITS_SOLD);
```



Uploaded the prod_new_data.csv



5.(+4) Explain end-to-end process based on your understanding.

In order for raw data to be converted into analytics-ready format, Snowflake data pipeline has a documented three-step process: RAW → CURATION → ANALYTICS. Data is first imported from Stages like internal and external Amazon S3 (External Stage) in RAW schema and staged in prod_stg. Data in the stage is loaded in RAW automatically via a Task (prod_raw_task). A stream (prod_stream) on the other hand streams newly added and updated records. All changes to the raw data are efficiently logged and available for further processing due to this setup. New and updated records are added to CURATION by utilizing the prod_stream in the CURATION schema. PROD_HST_TBL ensures data consistency and integrity using a MERGE process. Last but not least, the data that has been curated is updated periodically by a Dynamic Table (book_dy_tbl) in the ANALYTICS schema, resulting in a formatted, real-time data set that can be used for reporting and analysis.