Name: Mrunali Katta

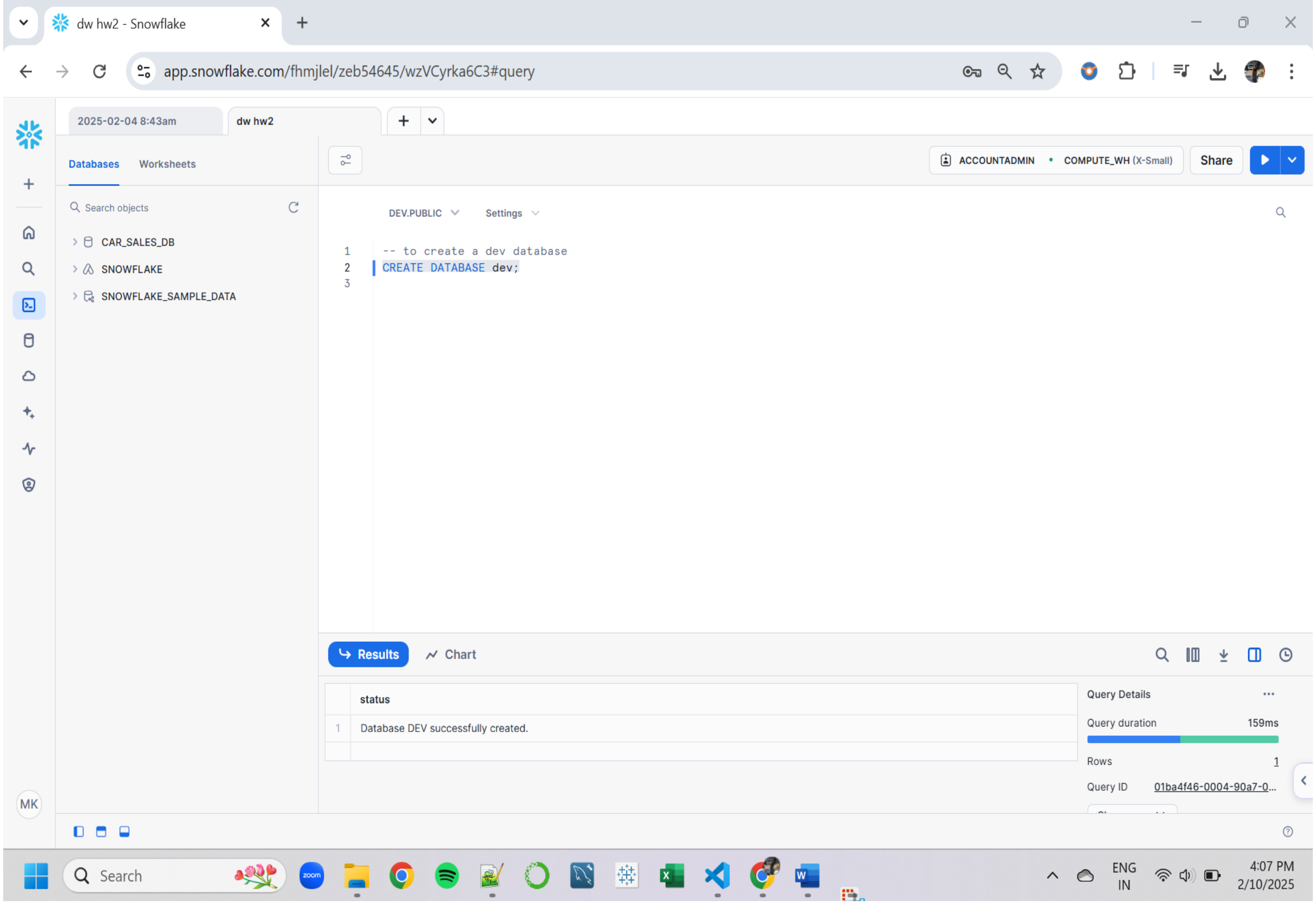
ID: 017516785

DATA 226 – Homework 2

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

Created a dev database

CREATE DATABASE dev;



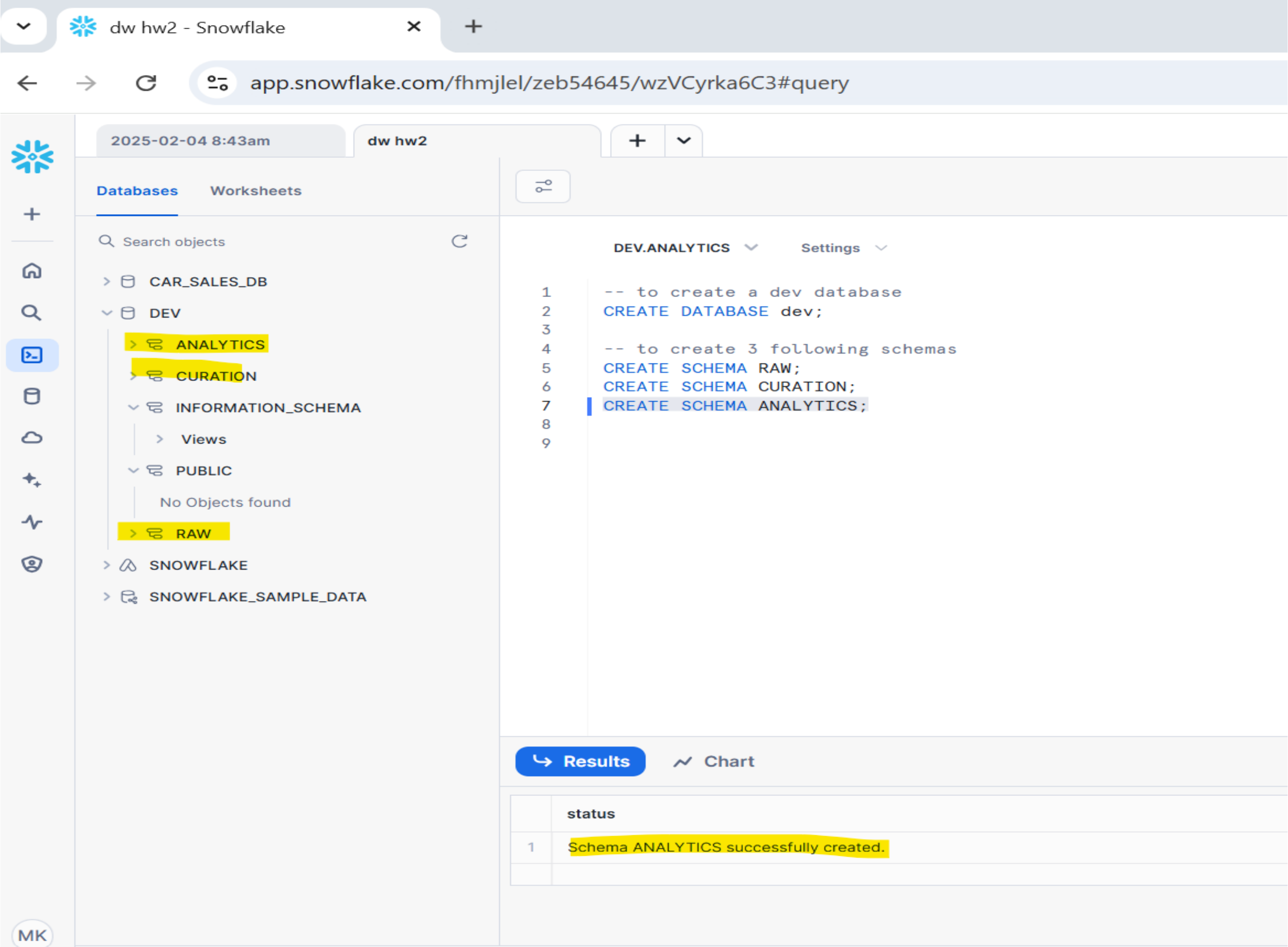
```
-- to create a dev database
CREATE DATABASE dev;
```

	status
1	Database DEV successfully created.

Query Details
Query duration 159ms
Rows 1
Query ID 01ba4f46-0004-90a7-0...

Also created the three schema's mentioned **RAW, CURATION and ANALYTICS**

CREATE SCHEMA RAW;
CREATE SCHEMA CURATION;
CREATE SCHEMA ANALYTICS;



```
-- to create a dev database
CREATE DATABASE dev;

-- to create 3 following schemas
CREATE SCHEMA RAW;
CREATE SCHEMA CURATION;
CREATE SCHEMA ANALYTICS;
```

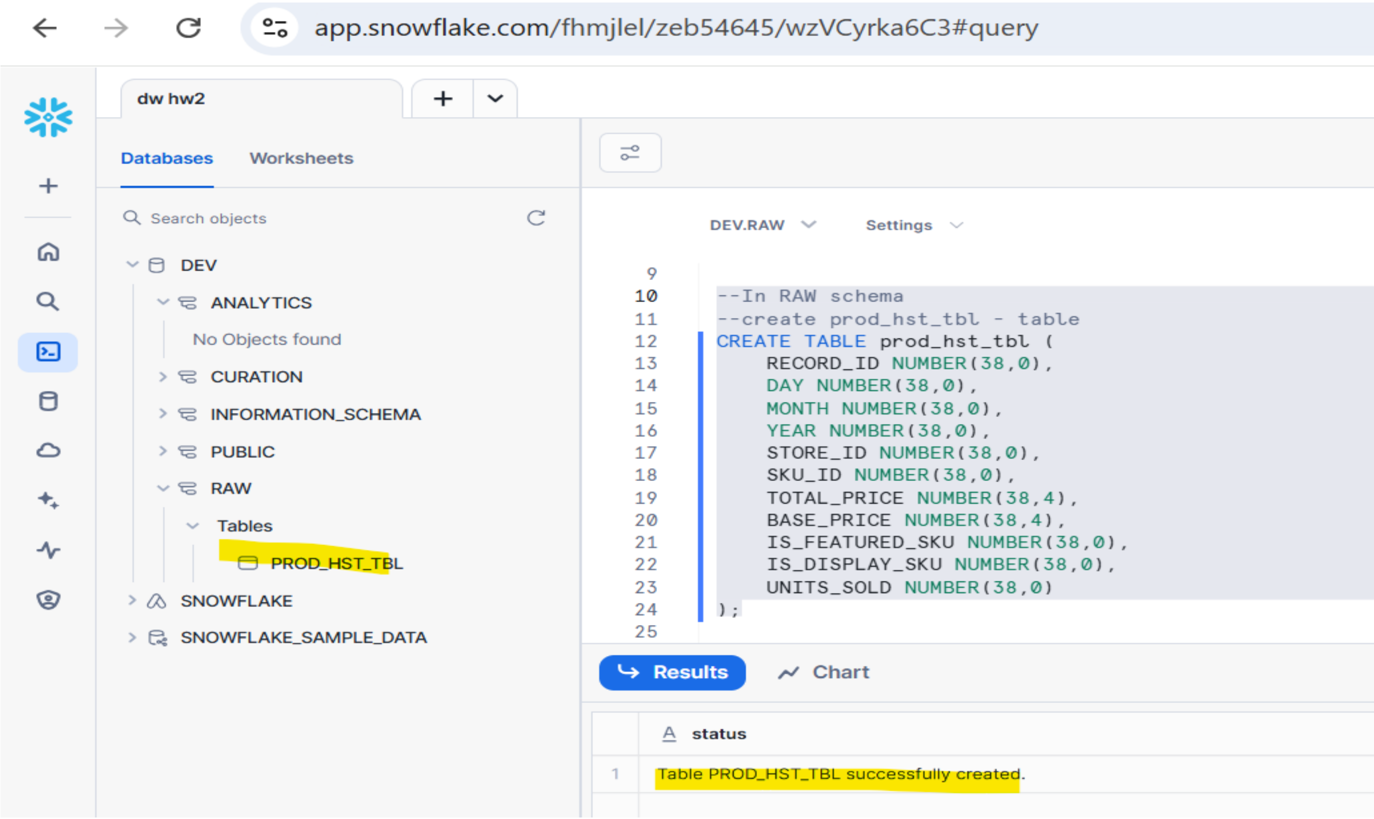
	status
1	Schema ANALYTICS successfully created.

2.(+4) In RAW schema, create

- prod_hst_tbl – table

created table **“prod_hst_tbl”** in RAW schema

```
CREATE 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)
);
```



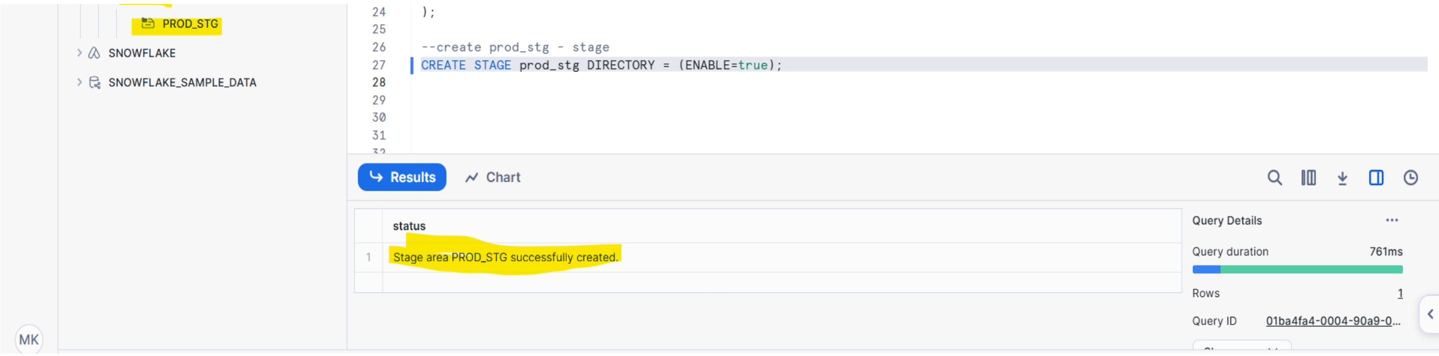
```
--In RAW schema
--create prod_hst_tbl - table
CREATE TABLE prod_hst_tbl (
    RECORD_ID NUMBER(38,0),
    DAY NUMBER(38,0),
    MONTH NUMBER(38,0),
    YEAR NUMBER(38,0),
    STORE_ID NUMBER(38,0),
    SKU_ID NUMBER(38,0),
    TOTAL_PRICE NUMBER(38,4),
    BASE_PRICE NUMBER(38,4),
    IS_FEATURED_SKU NUMBER(38,0),
    IS_DISPLAY_SKU NUMBER(38,0),
    UNITS_SOLD NUMBER(38,0)
);
```

	status
1	Table PROD_HST_TBL successfully created.

- prod_stg – stage

created stage **“prod_stg”** in RAW schema

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



```
);

--create prod_stg - stage
CREATE STAGE prod_stg DIRECTORY = (ENABLE=true);
```

	status
1	Stage area PROD_STG successfully created.

Query Details

Query duration 761ms

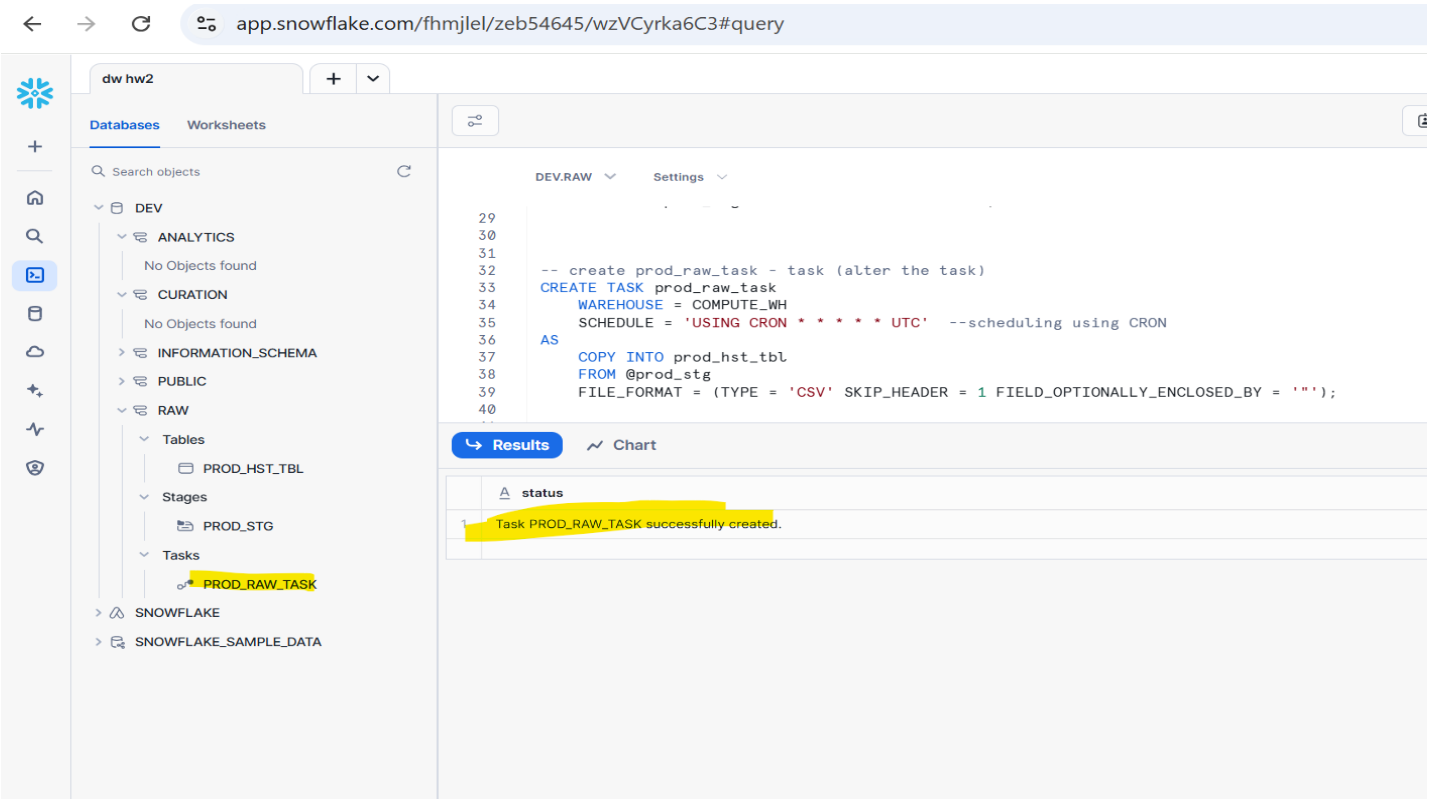
Rows 1

Query ID 01ba4fa4-0004-90a9-0...

- prod_raw_task - task (alter the task)

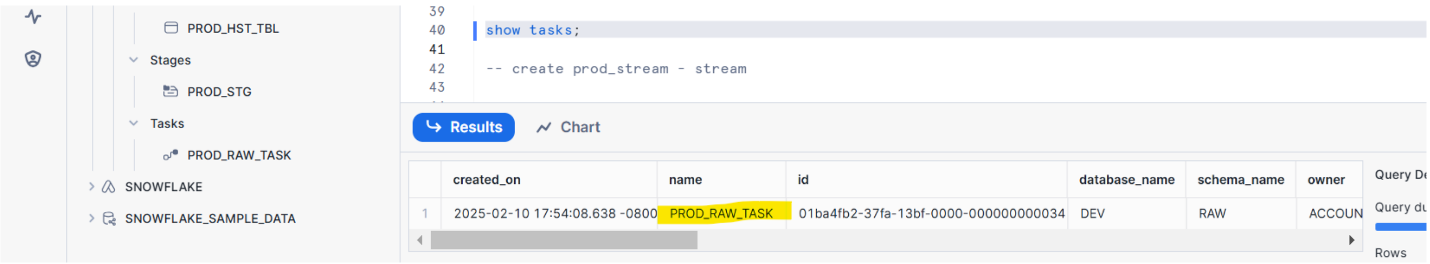
created task **“prod_raw_task”** in RAW schema

```
CREATE TASK prod_raw_task
    WAREHOUSE = COMPUTE_WH
    SCHEDULE = 'USING CRON * * * * * UTC'  --scheduling using CRON
AS
    COPY INTO prod_hst_tbl
    FROM @prod_stg
    FILE_FORMAT = (TYPE = 'CSV' SKIP_HEADER = 1 FIELD_OPTIONALLY_ENCLOSED_BY = '"');
```



```
-- create prod_raw_task - task (alter the task)
CREATE TASK prod_raw_task
    WAREHOUSE = COMPUTE_WH
    SCHEDULE = 'USING CRON * * * * * UTC'  --scheduling using CRON
AS
    COPY INTO prod_hst_tbl
    FROM @prod_stg
    FILE_FORMAT = (TYPE = 'CSV' SKIP_HEADER = 1 FIELD_OPTIONALLY_ENCLOSED_BY = '"');
```

	status
1	Task PROD_RAW_TASK successfully created.



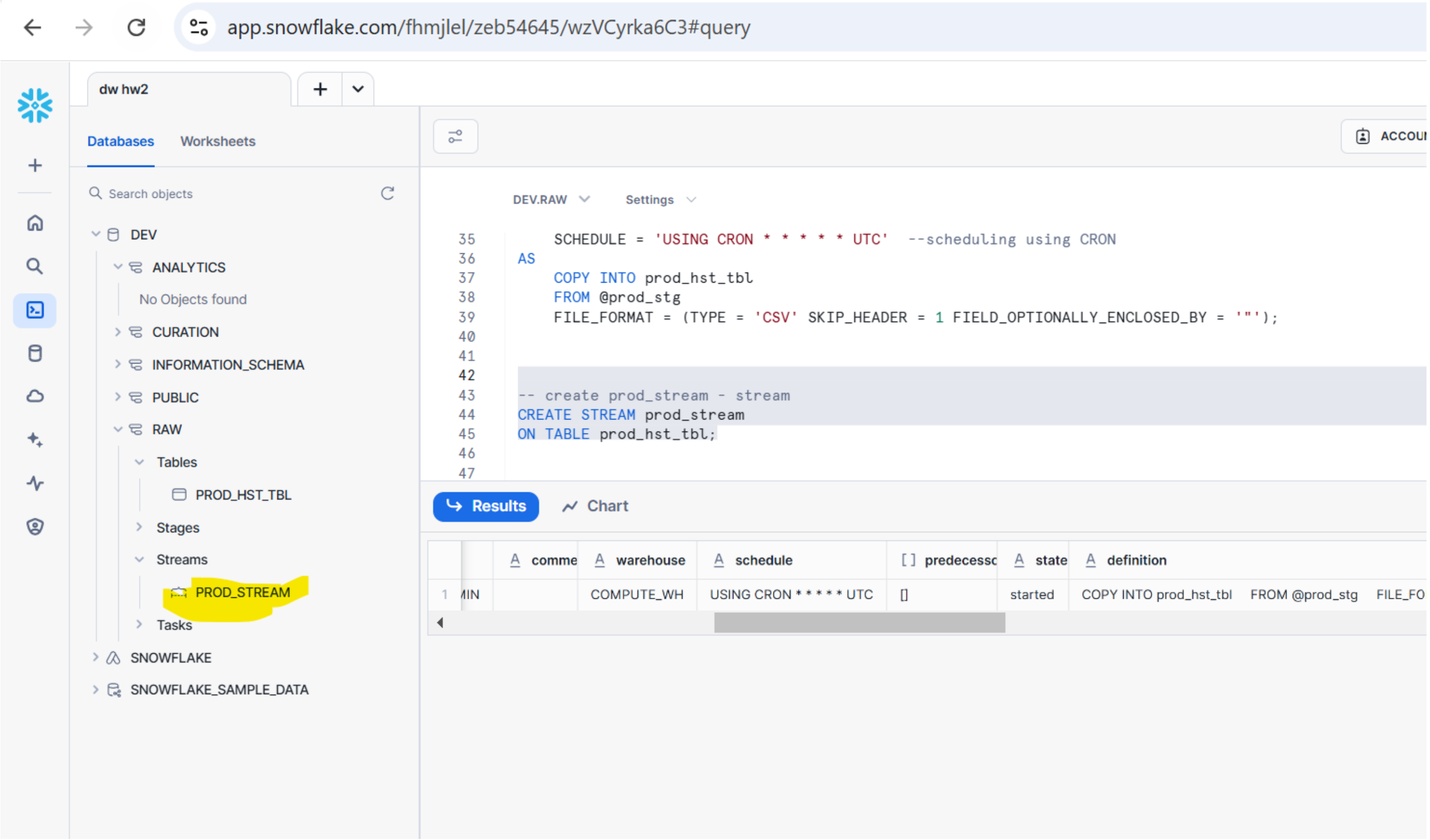
```
show tasks;

-- create prod_stream - stream
```

	created_on	name	id	database_name	schema_name	owner
1	2025-02-10 17:54:08.638 -0800	PROD_RAW_TASK	01ba4fb2-37fa-13bf-0000-000000000034	DEV	RAW	ACCOUN

- prod_stream – stream

created stream **"prod_stream"** in RAW schema

CREATE STREAM prod_stream ON TABLE prod_hst_tbl;



```
    SCHEDULE = 'USING CRON * * * * * UTC'  --scheduling using CRON
AS
    COPY INTO prod_hst_tbl
    FROM @prod_stg
    FILE_FORMAT = (TYPE = 'CSV' SKIP_HEADER = 1 FIELD_OPTIONALLY_ENCLOSED_BY = '"');


-- create prod_stream - stream
CREATE STREAM prod_stream
ON TABLE prod_hst_tbl;
```

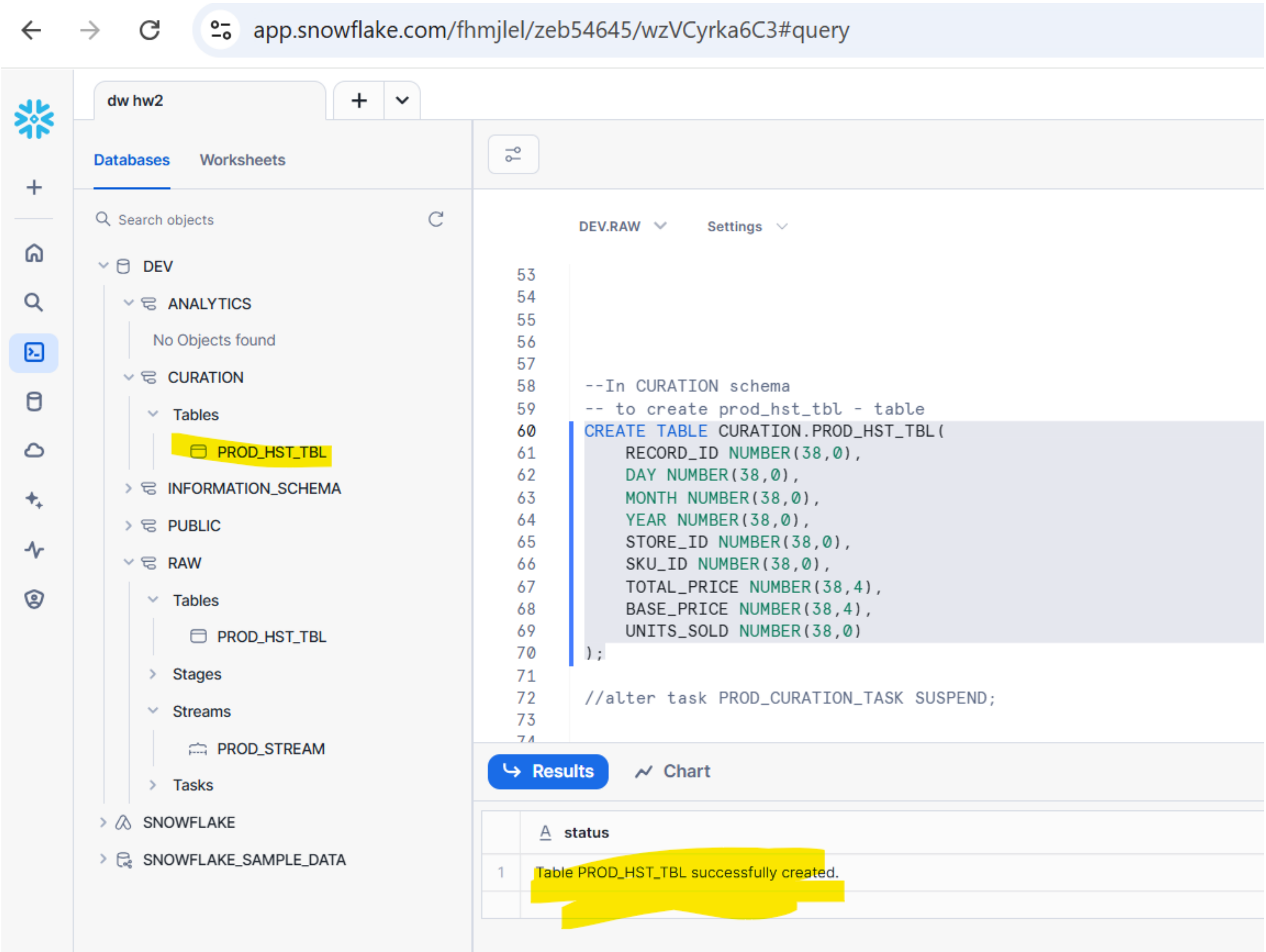
		comme	warehouse	schedule	predecessc	state	definition
1	MIN		COMPUTE_WH	USING CRON * * * * * UTC	[]	started	COPY INTO prod_hst_tbl FROM @prod_stg FILE_FO

3. In CURATION schema, create

- prod_hst_tbl – table

created stream **"prod_hst_tbl"** in CURATION schema

```
CREATE 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)
);
```



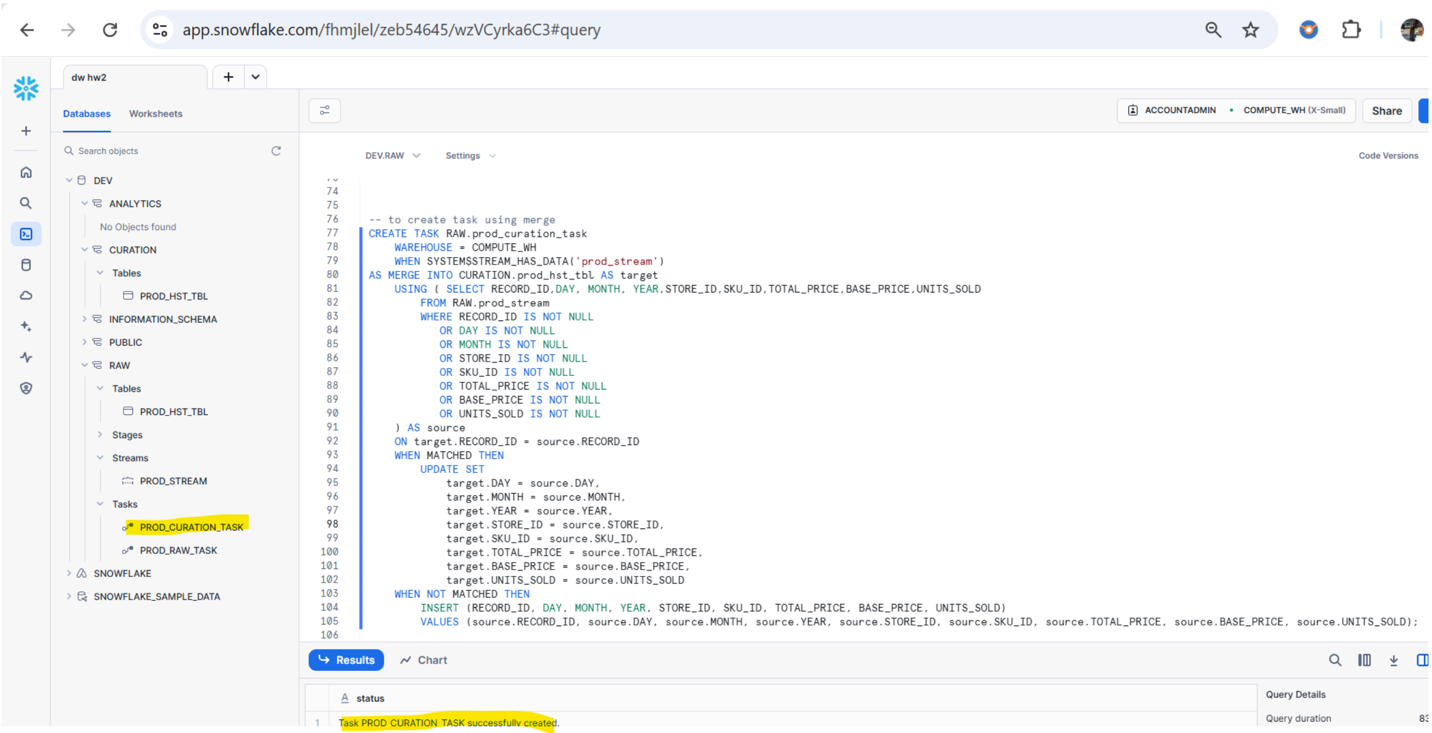
```
--In CURATION schema
-- to create prod_hst_tbl - table
CREATE TABLE CURATION.PROD_HST_TBL(
    RECORD_ID NUMBER(38,0),
    DAY NUMBER(38,0),
    MONTH NUMBER(38,0),
    YEAR NUMBER(38,0),
    STORE_ID NUMBER(38,0),
    SKU_ID NUMBER(38,0),
    TOTAL_PRICE NUMBER(38,4),
    BASE_PRICE NUMBER(38,4),
    UNITS_SOLD NUMBER(38,0)
);

//alter task PROD_CURATION_TASK SUSPEND;
```

	status
1	Table PROD_HST_TBL successfully created.

- prod_curation_task using MERGE (alter the task)

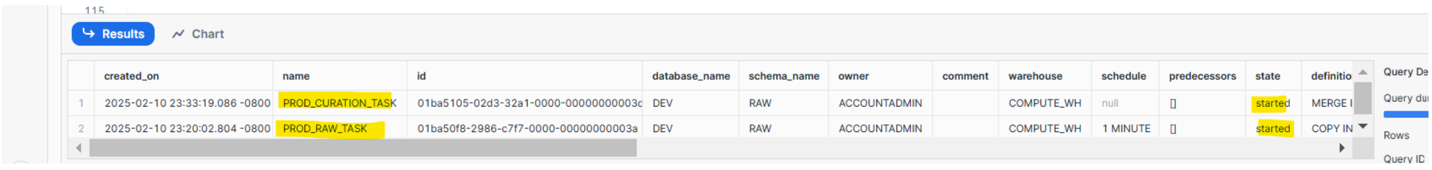
-- to create task using merge

```
CREATE 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 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);
```



```sql
-- to create task using merge
CREATE 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 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);
```

	status
1	Task PROD CURATION TASK successfully created.

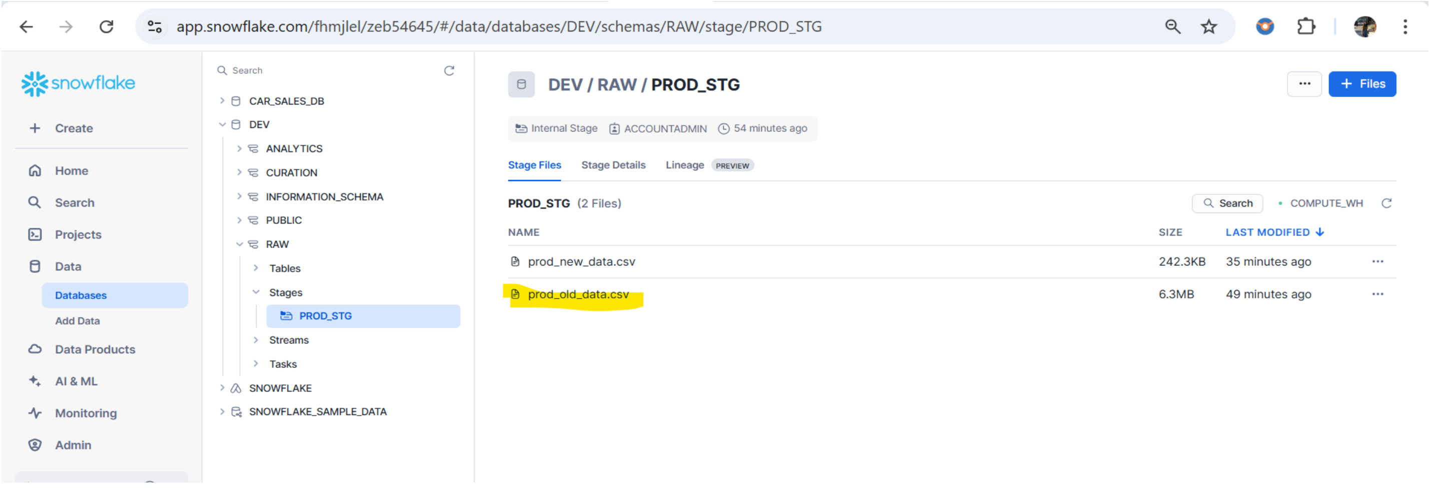


	created_on	name	id	database_name	schema_name	owner	comment	warehouse	schedule	predecessors	state	definitio
1	2025-02-10 23:33:19.086 -0800	PROD_CURATION_TASK	01ba5105-02d3-32a1-0000-00000000003c	DEV	RAW	ACCOUNTADMIN		COMPUTE_WH	null	[]	started	MERGE I
2	2025-02-10 23:20:02.804 -0800	PROD_RAW_TASK	01ba50f8-2986-c7f7-0000-00000000003a	DEV	RAW	ACCOUNTADMIN		COMPUTE_WH	1 MINUTE	[]	started	COPY IN

4.(+2) In ANALYTICS schema, create

- book_dy_tbl – Dynamic Table

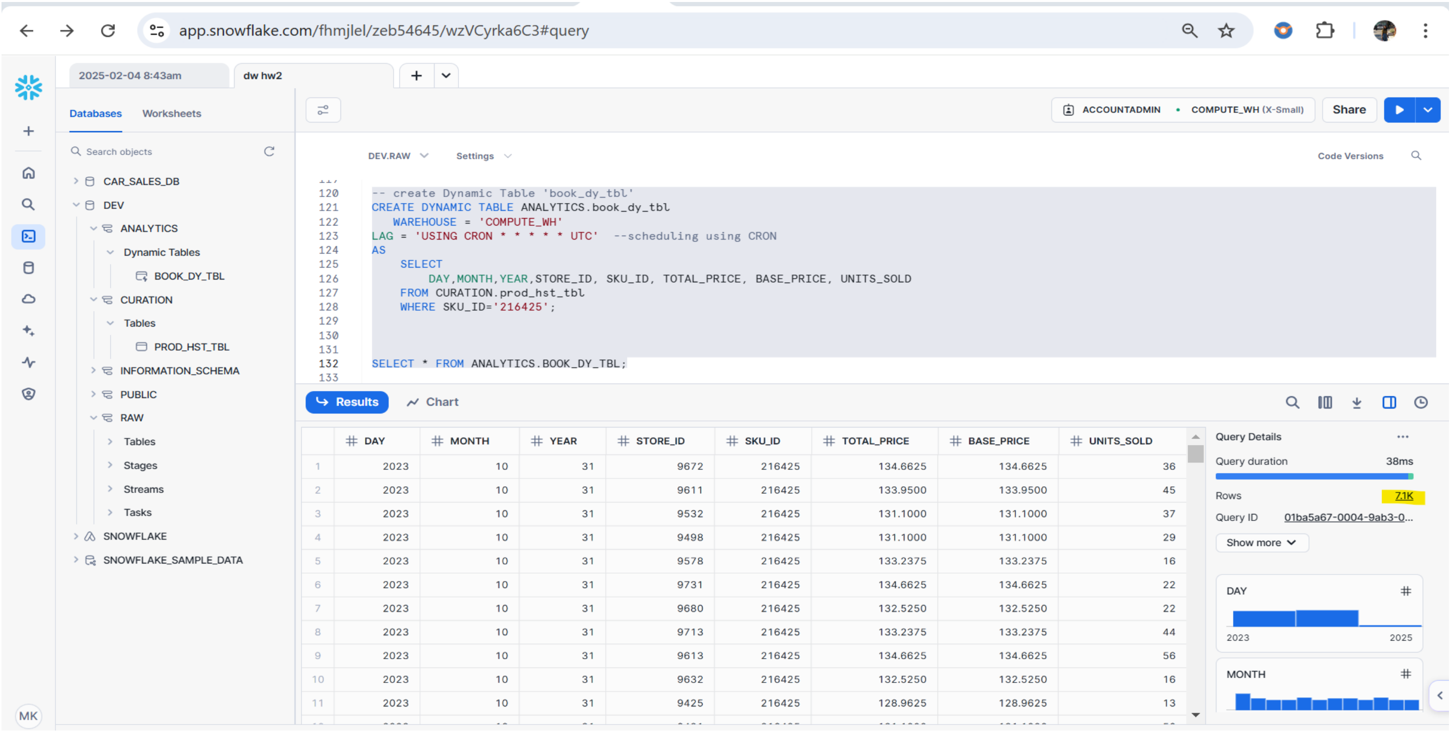
Created a table called **'book_dy_tbl'**. Later after loading file 'prod_old_data.csv' file the row count is 7.1k



DEV / RAW / PROD_STG

PROD_STG (2 Files)

NAME	SIZE	LAST MODIFIED
prod_new_data.csv	242.3KB	35 minutes ago
prod_old_data.csv	6.3MB	49 minutes ago



```sql
-- create Dynamic Table 'book_dy_tbl'
CREATE DYNAMIC TABLE ANALYTICS.book_dy_tbl
   WAREHOUSE = 'COMPUTE_WH'
LAG = 'USING CRON * * * * * UTC'  --scheduling using CRON
AS
    SELECT
        DAY,MONTH,YEAR,STORE_ID, SKU_ID, TOTAL_PRICE, BASE_PRICE, UNITS_SOLD
    FROM CURATION.prod_hst_tbl
    WHERE SKU_ID='216425';



SELECT * FROM ANALYTICS.BOOK_DY_TBL;
```

	DAY	MONTH	YEAR	STORE_ID	SKU_ID	TOTAL_PRICE	BASE_PRICE	UNITS_SOLD
1	2023	10	31	9672	216425	134.6625	134.6625	36
2	2023	10	31	9611	216425	133.9500	133.9500	45
3	2023	10	31	9532	216425	131.1000	131.1000	37
4	2023	10	31	9498	216425	131.1000	131.1000	29
5	2023	10	31	9578	216425	133.2375	133.2375	16
6	2023	10	31	9731	216425	134.6625	134.6625	22
7	2023	10	31	9680	216425	132.5250	132.5250	22
8	2023	10	31	9713	216425	133.2375	133.2375	44
9	2023	10	31	9613	216425	134.6625	134.6625	56
10	2023	10	31	9632	216425	132.5250	132.5250	16
11	2023	10	31	9425	216425	128.9625	128.9625	13

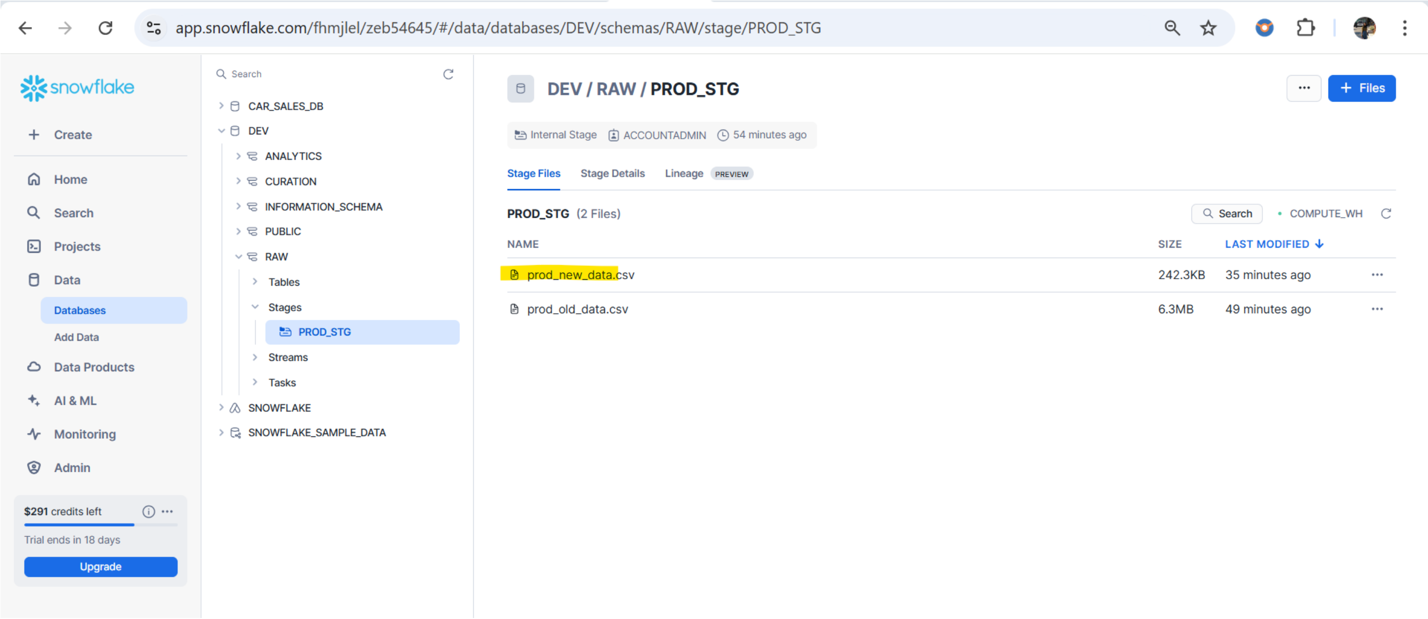
Query Details

Query duration 38ms

Rows 7.1K

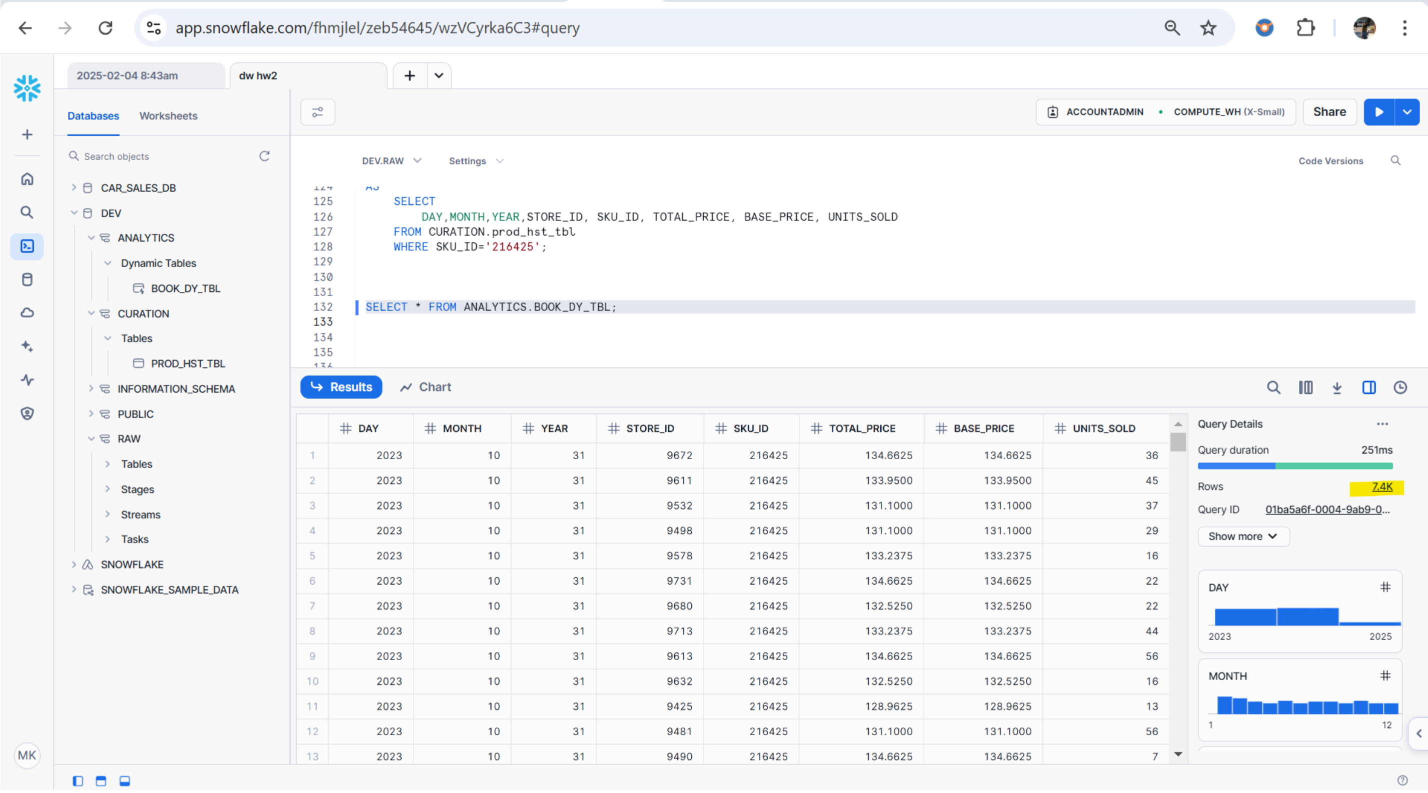
Query ID 01ba5a67-0004-9ab3-0...

After uploading prod_new_data.csv file in ‘prod_stage’ stage the rows count is 7.4k.



PROD_STG (2 Files)

NAME	SIZE	LAST MODIFIED
prod_new_data.csv	242.3KB	35 minutes ago
prod_old_data.csv	6.3MB	49 minutes ago



```
AS
    SELECT
        DAY,MONTH,YEAR,STORE_ID, SKU_ID, TOTAL_PRICE, BASE_PRICE, UNITS_SOLD
    FROM CURATION.prod_hst_tbl
    WHERE SKU_ID='216425';



SELECT * FROM ANALYTICS.BOOK_DY_TBL;
```

	DAY	MONTH	YEAR	STORE_ID	SKU_ID	TOTAL_PRICE	BASE_PRICE	UNITS_SOLD
1	2023	10	31	9672	216425	134.6625	134.6625	36
2	2023	10	31	9611	216425	133.9500	133.9500	45
3	2023	10	31	9532	216425	131.1000	131.1000	37
4	2023	10	31	9498	216425	131.1000	131.1000	29
5	2023	10	31	9578	216425	133.2375	133.2375	16
6	2023	10	31	9731	216425	134.6625	134.6625	22
7	2023	10	31	9680	216425	132.5250	132.5250	22
8	2023	10	31	9713	216425	133.2375	133.2375	44
9	2023	10	31	9613	216425	134.6625	134.6625	56
10	2023	10	31	9632	216425	132.5250	132.5250	16
11	2023	10	31	9425	216425	128.9625	128.9625	13
12	2023	10	31	9481	216425	131.1000	131.1000	56
13	2023	10	31	9490	216425	134.6625	134.6625	7

Query Details

Query duration 251ms

Rows 7.4K

Query ID 01ba5a6f-0004-9ab9-0...

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

→ Following is the end-to-end process explanation of the **Snowflake Basic Data Pipeline:**

1. **Database Setup**:
– First created a database called **“dev”.**
2. **3 Schema creation**:
– Then created three schemas as follows: **“RAW”, “CURATION”, and “ANALYTICS”.**
3. **RAW schema:**
– Then created table called **“prod_hst_tbl”** to store initial raw data.
– Also created a stage called **“prod_stg”** to stage the two .csv files which are **“prod_old_data”** and **“prod_new_data”.**
– Later created task **“prod_raw_task”** to transfer data from the stage to the table every minute using CRON.
– Also created a stream called as **“prod_stream”** to monitor any changes in **“prod_hst_tbl”**.
4. **CURATION schema**:
– Here in this schema replicated **“prod_hst_tbl”** for data processing.
– And then created a task called **“prod_curation_task”** for merging updates from **RAW** schema to **CURATION** schema, to ensure data is consistent.
5. **ANALYTICS Schema**:
– In this schema, created a table called **“book_dy_tbl”** which is a a dynamic table for aggregating the earlier curated data.
– Then using a query extracted insights such as total sales, average prices, and transaction counts by SKU and store ID.
– Overall here the data moves from initial ingestion in RAW, to refinement in CURATION, and finally to aggregation for analytics in ANALYTICS.