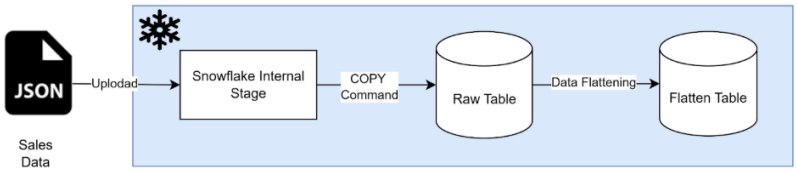
SNOWFLAKE - Mini Project 3



---

1. Create a Database & Schema
    1. Database : SALES_DATA
        1. Schema : RAW_DATA
        2. Schema : FLATTEN_DATA

---

2. Create Raw Table in the RAW_DATA schema
    1. Table :
        1. Database: SALES_DATA
        2. Schema: RAW_DATA
        3. Name: SALES_RAW
    2. Table structure

        JSON_DATA VARIANT

---

3. Create flatten table in the FLATTEN_DATA schema
    1. Table Name :
        1. Database: SALES_DATA
        2. Schema: FLATTEN_DATA
        3. Name: SALES_FLATTEN
    2. Table structure

        COMPANIES STRING
        SALES_PERIOD STRING
        TOTAL_REVENUE FLOAT
        TOTAL_UNITS_SOLD FLOAT
        REGIONS STRING
        TOTAL_SALES FLOAT
        PRODUCTS STRING
        UNITS_SOLD FLOAT
        REVENUE FLOAT

---

4. Upload the JSON File to an Internal Stage
    1. Create an Internal Stage in the RAW_DATA schema
        1. Database: SALES_DATA
        2. Schema: RAW_DATA
        3. Name: SALES_STAGE
    2. Upload the sales json data file in the internal stage
    3. List the file in the internal stage

5. Load Data from the Stage into the Raw Table
    1. To load the raw sales data into the SALES_RAW table, use the COPY INTO command

6. Perform JSON flattening and insert data into the flatten Table
    1. Extract the required columns from the raw table and load into the flatten table by flattening the JSON data. Here we need to use the LATERAL FLATTEN functionality of snowflake

7. Data Analysis on the flatten data
    1. Calculate the revenue for each company.
    2. Determine the top 3 companies with the highest revenue.
    3. Identify the 2 regions with the lowest number of units sold.
    4. Identify the product with the highest revenue generated from sales.
    5. Find the product with the fewest units sold.
    6. Determine the region with the highest number of laptop units sold.
    7. Identify the bottom 3 companies and regions with the lowest revenue generated from smartphone sales.

Solution:

```
-- ############################################################
-- ####### (1) Create a Database & Schema
-- ############################################################

-- SET ROLE AND WAREHOUSE:
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE COMPUTE_WH;

-- CREATE A NEW DATABASE:
CREATE OR REPLACE DATABASE SALES_DATA;
USE DATABASE SALES_DATA;

-- CREATE TWO NEW SCHEMAS:
CREATE OR REPLACE SCHEMA RAW_DATA;
CREATE OR REPLACE SCHEMA FLATTEN_DATA;
```

```
-- ############################################################
-- ####### (2) Create Raw Table in the RAW_DATA schema
-- ############################################################

-- Create Raw Table in the RAW_DATA schema:
USE SCHEMA SALES_DATA.RAW_DATA;

CREATE OR REPLACE TABLE SALES_RAW
    (JSON_DATA VARIANT);

SELECT * FROM SALES_RAW;
```

```
-- ############################################################
-- ####### (3) Create flattened table in the FLATTEN_DATA schema
-- ############################################################

-- Create flattened table in the FLATTEN_DATA schema:
USE SCHEMA SALES_DATA.FLATTEN_DATA;

CREATE OR REPLACE TABLE SALES_FLATTEN (
    COMPANIES STRING,
    SALES_PERIOD STRING,
    TOTAL_REVENUE FLOAT,
    TOTAL_UNITS_SOLD FLOAT,
    REGIONS STRING,
    TOTAL_SALES FLOAT,
    PRODUCTS STRING,
    UNITS_SOLD FLOAT,
    REVENUE FLOAT);

SELECT * FROM SALES_FLATTEN;
```

```
-- ############################################################
-- ####### (4) Upload the JSON File to an Internal Stage
-- ############################################################

-- Create an Internal Stage in the RAW_DATA schema
USE SCHEMA SALES_DATA.RAW_DATA;

CREATE OR REPLACE STAGE SALES_STAGE;

-- List the file in the internal stage before uploading the file
LIST @SALES_STAGE;

-------------------------------------------------------------------------------
------- UPLOAD MANUALLY THE SALES JSON DATA INTO THE INTERNAL CUSTOMER_STAGE -------
-------------------------------------------------------------------------------

-- List the file in the internal stage after uploading the file
LIST @SALES_STAGE;
```

```
-- ###########################################################################
-- ####### (5) Load Data from the Stage into the Raw Table using COPY INTO command
-- ###########################################################################

-- create the JSON file format:
USE SCHEMA SALES_DATA.RAW_DATA;

CREATE OR REPLACE FILE FORMAT JSON_FORMAT
TYPE = 'JSON';
```

```
-- load uploaded JSON file from the internal CUSTOMER_STAGE into table CUSTOMER_RAW:
SELECT * FROM SALES_RAW;

COPY INTO SALES_RAW
FROM @SALES_STAGE
FILE_FORMAT = (FORMAT_NAME = JSON_FORMAT);

SELECT * FROM SALES_RAW;
```

```
-- ###############################################################################
-- ####### (6) Perform JSON flattening and insert data into the flatten Table
-- ###############################################################################

INSERT INTO SALES_DATA.FLATTEN_DATA.SALES_FLATTEN
SELECT
    COMPANY.KEY::STRING AS COMPANIES,
    COMPANY.VALUE:sales_period::STRING AS SALES_PERIOD,
    COMPANY.VALUE:total_revenue::FLOAT AS TOTAL_REVENUE,
    COMPANY.VALUE:total_units_sold::FLOAT AS TOTAL_UNITS_SOLD,
    REGION.KEY::STRING AS REGIONS,
    REGION.VALUE:total_sales::FLOAT AS TOTAL_SALES,
    PRODUCT.KEY::STRING AS PRODUCTS,
    PRODUCT.VALUE:units_sold::FLOAT AS UNITS_SOLD,
    PRODUCT.VALUE:revenue::FLOAT AS REVENUE
FROM SALES_DATA.RAW_DATA.SALES_RAW,
    LATERAL FLATTEN (INPUT => JSON_DATA:companies) AS COMPANY,
    LATERAL FLATTEN (INPUT => COMPANY.VALUE:regions) AS REGION,
    LATERAL FLATTEN (INPUT => REGION.VALUE:products) AS PRODUCT;

SELECT * FROM SALES_DATA.FLATTEN_DATA.SALES_FLATTEN;
```

```
-- ###############################################################################
-- ####### (7) Data Analysis on the flattened data
-- ###############################################################################

USE SCHEMA SALES_DATA.FLATTEN_DATA;

-----------------------------------------------------------------
-- 1. Calculate the revenue for each company.
-----------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_1 AS
SELECT
    COMPANIES,
    SUM(REVENUE) AS COMPANY_REVENUE
FROM SALES_FLATTEN
GROUP BY COMPANIES;

SELECT * FROM VIEW_1;
```

```sql
-----------------------------------------------------------------
-- 2. Determine the top 3 companies with the highest revenue.
-----------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_2 AS
SELECT
    COMPANIES,
    COMPANY_REVENUE AS HIGHEST_REVENUE
FROM VIEW_1
ORDER BY HIGHEST_REVENUE DESC
LIMIT 3;

SELECT * FROM VIEW_2;

-----------------------------------------------------------------
-- 3. Identify the 2 regions with the lowest number of units sold.
-----------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_3 AS
SELECT
    REGIONS,
    SUM(UNITS_SOLD) AS REGIONS_UNITS_SOLD
FROM SALES_FLATTEN
GROUP BY REGIONS
ORDER BY REGIONS_UNITS_SOLD ASC
LIMIT 2;

SELECT * FROM VIEW_3;

-----------------------------------------------------------------------
-- 4. Identify the product with the highest revenue generated from sales.
-----------------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_4 AS
SELECT
    PRODUCTS,
    SUM(REVENUE) AS PRODUCT_REVENUE
FROM SALES_FLATTEN
GROUP BY PRODUCTS
ORDER BY PRODUCT_REVENUE DESC
LIMIT 1;

SELECT * FROM VIEW_4;

-----------------------------------------------------------------
-- 5. Find the product with the fewest units sold.
-----------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_5 AS
SELECT
    PRODUCTS,
    SUM(UNITS_SOLD) AS PRODUCT_UNITS_SOLD
FROM SALES_FLATTEN
GROUP BY PRODUCTS
ORDER BY PRODUCT_UNITS_SOLD ASC LIMIT 1;

SELECT * FROM VIEW_5;
```

```sql
-----------------------------------------------------------------------
-- 6. Determine the region with the highest number of laptop units sold.
-----------------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_6 AS
SELECT
    REGIONS,
    SUM(UNITS_SOLD) AS REGIONS_UNITS_SOLD
FROM SALES_FLATTEN
GROUP BY REGIONS, PRODUCTS
HAVING PRODUCTS = 'laptop'
ORDER BY REGIONS_UNITS_SOLD DESC
LIMIT 1;

SELECT * FROM VIEW_6;

-----------------------------------------------------------------------------------
-- 7. Identify the bottom 3 companies and regions with the lowest revenue generated
from smartphone sales.
-----------------------------------------------------------------------------------
CREATE OR REPLACE VIEW VIEW_7 AS
SELECT
    COMPANIES,
    REGIONS,
    SUM(REVENUE) AS COMPANIES_REGIONS_REVENUE
FROM SALES_FLATTEN
GROUP BY COMPANIES, REGIONS, PRODUCTS
HAVING PRODUCTS = 'smartphone'
ORDER BY COMPANIES_REGIONS_REVENUE ASC
LIMIT 3;

SELECT * FROM VIEW_7;
```