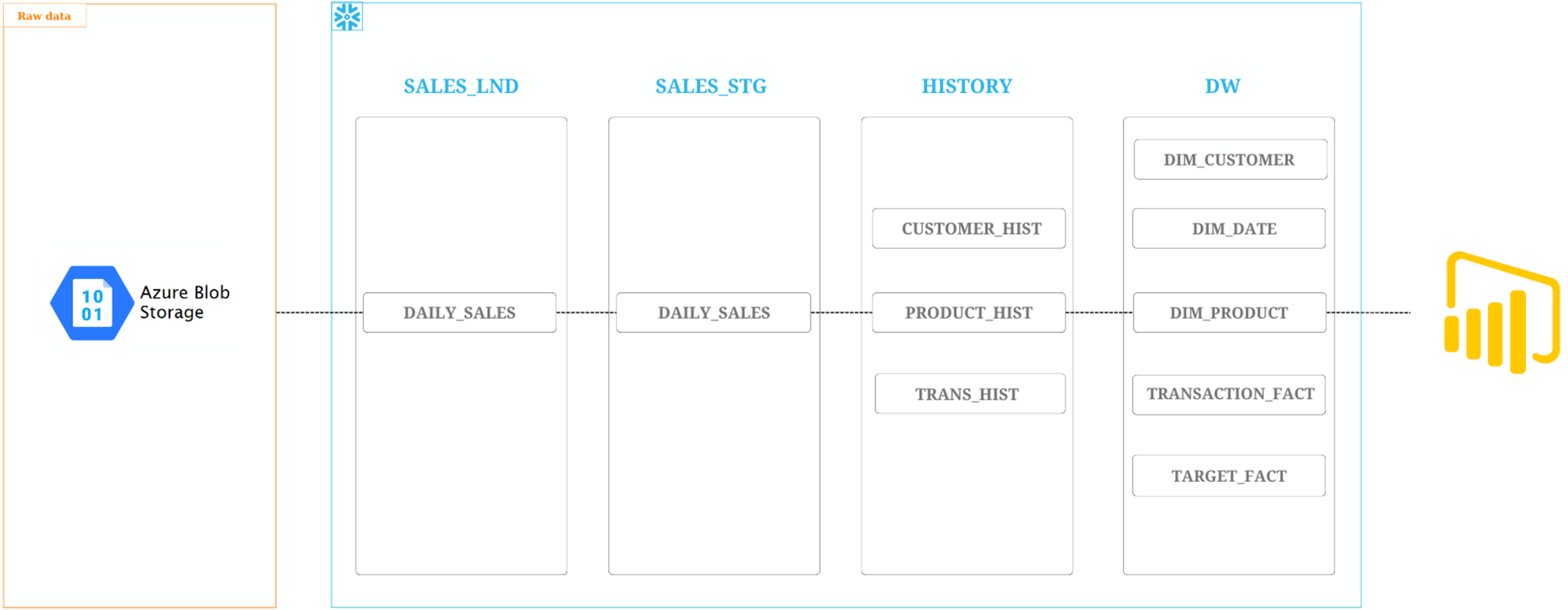
# Data Layer


Raw data
Azure Blob Storage
SALES_LND
DAILY_SALES
SALES_STG
DAILY_SALES
HISTORY
CUSTOMER_HIST
PRODUCT_HIST
TRANS_HIST
DW
DIM_CUSTOMER
DIM_DATE
DIM_PRODUCT
TRANSACTION_FACT
TARGET_FACT

# Processing data

# Input data

Upload | Add Directory | Refresh | Rename | Delete | Change tier | Acquire lease | Break lease | Give feedback

**Authentication method:** Access key (Switch to Microsoft Entra user account)
**Location:** snowflakecsv

Search blobs by prefix (case-sensitive) | Show deleted objects

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| dailysales_02102021.csv | 12/9/2024, 4:37:22 PM | Hot (Inferred) | | Block blob | 10.79 KiB | Available |
| dailysales_03102021.csv | 12/12/2024, 4:45:42 ... | Hot (Inferred) | | Block blob | 3.52 KiB | Available |
| dailysales_04102021.csv | 12/14/2024, 4:05:12 ... | Hot (Inferred) | | Block blob | 4.13 KiB | Available |

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ,Customer_Code,customer_name,customer_type_name,area_name,region_name,Invoice_Number,Invoice_Date,ZP_Item_Code,Price,Quantity | | | | | | | | | | | | | |
| 2 | 0,30190521,CN Quay CPDP Ben Tai- NT So 02 Hello,Beauty,Ho Chi Minh,HCMC,1225690651,2021-10-02,21085258,204286,1 | | | | | | | | | | | | | |
| 3 | 1,30191419,Cong Ty TMDP Huynh Kim Hello,Wholesaler,Ho Chi Minh,HCMC,1225691640,2021-10-02,21193950,179600,3 | | | | | | | | | | | | | |
| 4 | 2,30191419,Cong Ty TMDP Huynh Kim Hello,Wholesaler,Ho Chi Minh,HCMC,1225691630,2021-10-02,21193950,179600,31 | | | | | | | | | | | | | |
| 5 | 3,30173102,Cong Ty Co Phan NThuong Da KThanh Thanhn My Da Lat Hello,Beauty Dep,Lam Dong,North,1212577120,2021-10-02,21193955,547627,5 | | | | | | | | | | | | | |
| 6 | 4,30173737,Cong Ty Co Phan TMDV Thang Long Hello,Beauty Dep,Khanh Hoa,North,1212576969,2021-10-02,21193949,204286,10 | | | | | | | | | | | | | |
| 7 | 5,30191729,Cong Ty CPDP Gia Dinh- NT So 21 Hello,Beauty,Ho Chi Minh,HCMC,1225690758,2021-10-02,21085258,204286,100 | | | | | | | | | | | | | |
| 8 | 6,30191729,Cong Ty CPDP Gia Dinh- NT So 21 Hello,Beauty,Ho Chi Minh,HCMC,1225690661,2021-10-02,21193946,203743,2 | | | | | | | | | | | | | |
| 9 | 7,30189144,Chau Thanhng Sinh Hello,Beauty,Nghe An,Mekong,1225691005,2021-10-02,21193949,204286,10 | | | | | | | | | | | | | |
| 10 | 8,30185015,Dai Ly Ban Cho Quay CPDP TV.Faom (To ) Hello,Beauty,Thai Nguyen,Mekong,1225691687,2021-10-02,21193966,169886,2 | | | | | | | | | | | | | |

# Ingest data

```
CREATE OR REPLACE PROCEDURE DEMO_DB.SALES_LND.SP_LOAD_RAW_TO_LND()
RETURNS TABLE()
LANGUAGE SQL
EXECUTE AS OWNER
AS
DECLARE
    res RESULTSET;
    select_statement VARCHAR;
BEGIN
    create or replace stream DEMO_DB.SALES_LND.stream_lnd on table DEMO_DB.SALES_LND.DAILY_SALES APPEND_ONLY=TRUE;

    COPY INTO DAILY_SALES
    FROM @stage_azure
    FILE_FORMAT = (format_name = azure_fileformat)
    PATTERN = '.*dailysales.*';
    --RETURN 'Successful load';
    select_statement:= 'select * from table(result_scan(last_query_id()))';
    res := (EXECUTE IMMEDIATE :select_statement);
    RETURN TABLE(res);
END;
```

# LOAD TO STAGING

```
INSERT INTO DEMO_DB.SALES_STG.DAILY_SALES
SELECT TRIM(s.id) as id,
    TRIM(s.customer_code) as customer_code,
    TRIM(s.customer_name) as customer_name,
    PARSE_JSON(s.customer_info):name::STRING as contact_person,
    PARSE_JSON(s.customer_info):email::STRING as contact_email,
    TRIM(s.customer_type_name) as customer_type_name,
    TRIM(s.area_name) as area_name,
    TRIM(s.region_name) as region_name,
    TRIM(s.invoice_number) as invoice_code,
    TRIM(s.item_code) as item_code,
    TRIM(s.item_name) as item_name,
    s.price,
    s.quantity,
    TO_DATE(s.invoice_date, 'YYYY-MM-DD') as invoice_date,
    CURRENT_TIMESTAMP() as created_at
FROM DEMO_DB.SALES_LND.stream_lnd as s
WHERE METADATA$ACTION = 'INSERT';
```

# SCD TYPE 2

# Records in initial load

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 30172683 | NT Thanhng Hello | gregorybooth@example.org | 2024-12-12 01:43:26.383 | null | 1000-01-01 | 9999-12-31 | Y |

# Capture the changes with SCD 2

| | CUSTOMER_CODE | CUSTOMER_NAME | CONTACT_EMAIL | INSERT_AT | UPDATE_AT | START_DATE | END_DATE | CURRENT_FLAG |
|---|---|---|---|---|---|---|---|---|
| 1 | 30172406 | KThanh  NThuong Tinh Khanh Thanh Hello | kinganna@example.org | 2024-12-12 01:43:26.383 | 2024-12-14 01:05:30.511 | 1000-01-01 | 2024-12-13 | N |
| 2 | 30172406 | KThanh  NThuong Tinh Khanh Thanh | cory81@example.org | 2024-12-14 01:05:30.511 | null | 2024-12-14 | 9999-12-31 | Y |
| 3 | 30174268 | Quay  Doanh Nghiep Becamex So 131 Hello | sosborne@example.net | 2024-12-12 01:43:26.383 | 2024-12-12 01:46:28.120 | 1000-01-01 | 2024-12-11 | N |
| 4 | 30174268 | Quay  Doanh Nghiep Becamex So 132 | sydneysmith@example.com | 2024-12-12 01:46:28.120 | 2024-12-14 01:05:30.511 | 2024-12-12 | 2024-12-13 | N |
| 5 | 30174268 | Quay  Doanh Nghiep Becamex So 131 Hello | sydneysmith@example.com | 2024-12-14 01:05:30.511 | null | 2024-12-14 | 9999-12-31 | Y |

## Insert data

```
INSERT_DATA AS (
    SELECT
        c.*,
        CASE
            WHEN d.primary_checksum is null THEN TO_DATE('1000-01-01')
            ELSE current_date()
        END as start_date,
        TO_DATE('9999-12-31') as end_date,
        'I' as insert_or_update
    FROM
        LATEST_CUSTOMER_INFO as c
        LEFT JOIN DEMO_DB.SALES_HISTORY.CUSTOMER_HISTORY as d ON c.primary_checksum = d.primary_checksum
    WHERE
        d.primary_checksum IS NULL
        OR (
            c.record_checksum != d.record_checksum
            AND d.current_flag = 'Y'
        )
```

## Update data

```
UPDATE_DATA AS (
    SELECT
        c.*,
        TO_DATE('1000-01-01') as start_date,
        TO_DATE('9999-12-31') as end_date,
        'U' as insert_or_update
    FROM
        LATEST_CUSTOMER_INFO as c
        INNER JOIN DEMO_DB.SALES_HISTORY.CUSTOMER_HISTORY as d ON c.primary_checksum = d.primary_checksum
    WHERE
        c.record_checksum != d.record_checksum
        AND d.current_flag = 'Y'
),
FINAL_DATA AS (
    (
        SELECT * FROM  INSERT_DATA
    )
    UNION ALL
    (
        SELECT * FROM UPDATE_DATA
    )
```

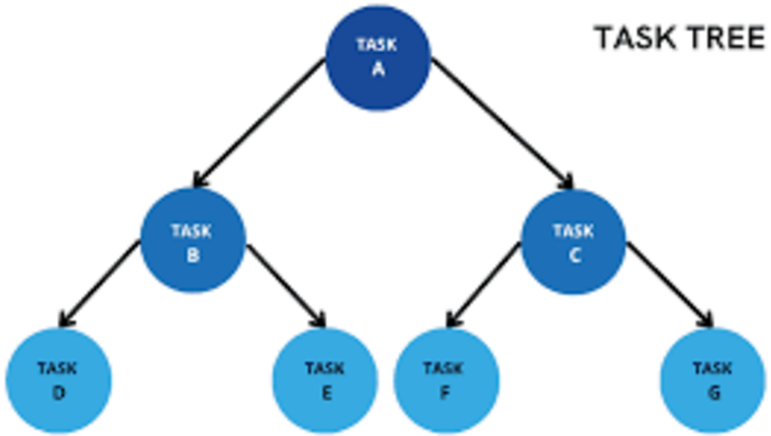
## create checksum values

```
md5(concat_ws('||', customer_code)) as primary_checksum,
md5(
    concat_ws(
        '||',
        coalesce(customer_name, ''),
        coalesce(contact_person, ''),
        coalesce(contact_email, ''),
        coalesce(area_name, ''),
        coalesce(region_name, '')
    )
) as record_checksum,
```

## check if a record is updated or inserted

```
            FINAL_DATA
    ) as source ON target.primary_checksum = source.primary_checksum
    AND source.insert_or_update = 'U'
    AND target.current_flag = 'Y'
    WHEN MATCHED THEN
    UPDATE
    SET
        target.update_at = current_timestamp(),
        target.end_date = DATEADD(day, -1, current_date()),
        target.current_flag = 'N'
    WHEN NOT MATCHED THEN
    INSERT
```

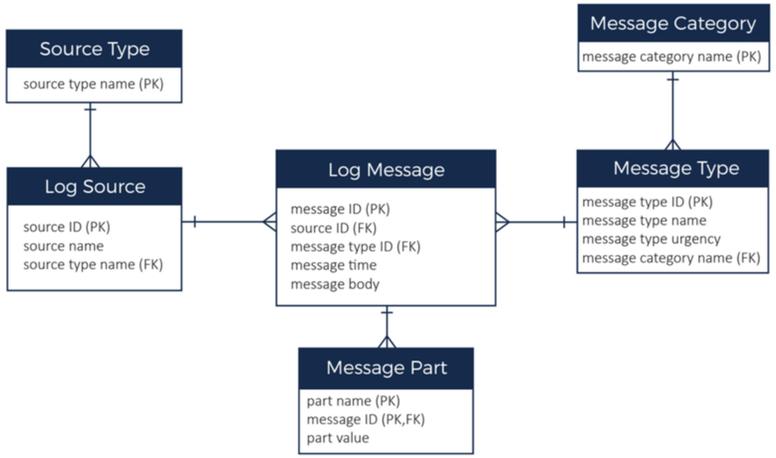
# TASK SCHEDULE

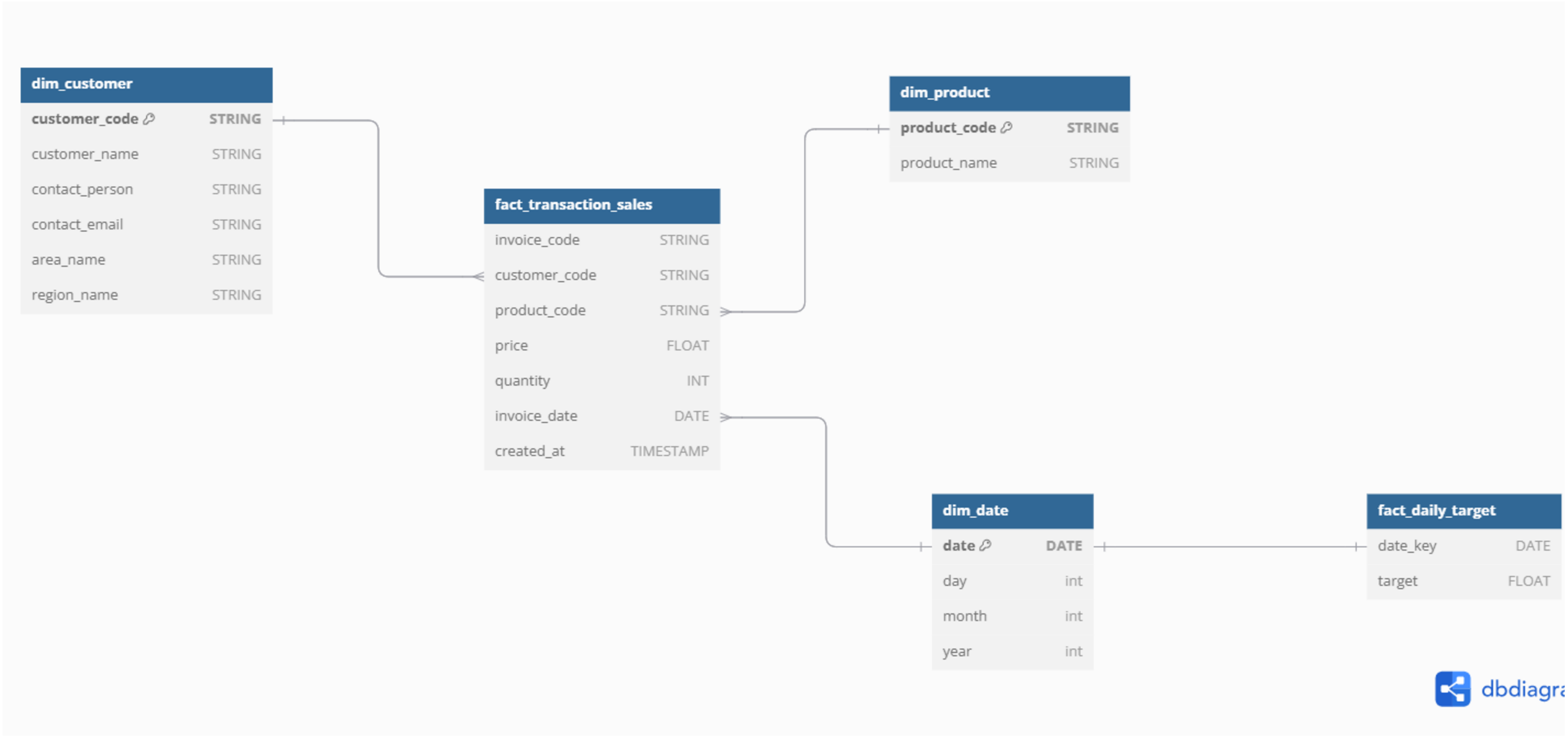


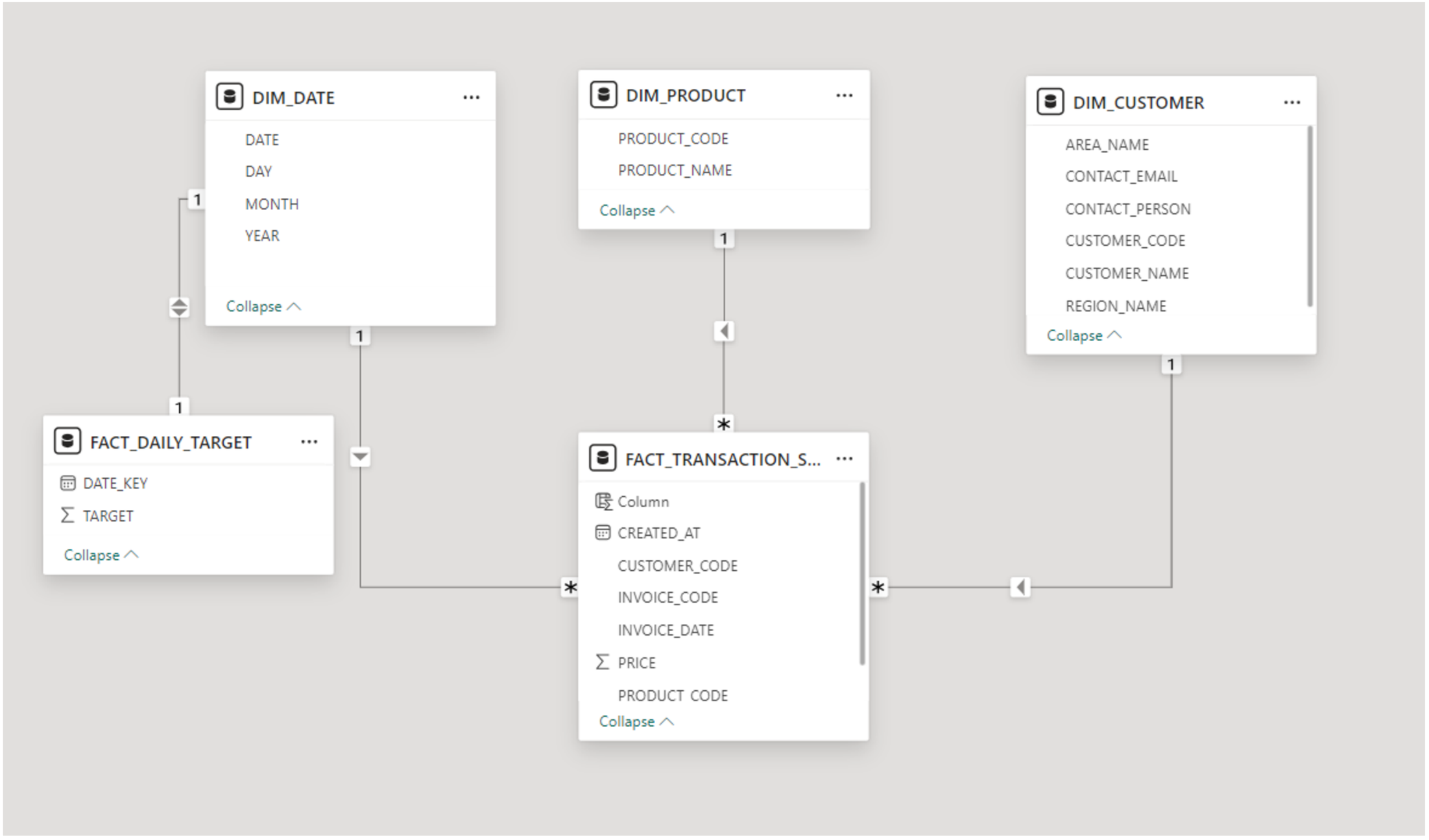
| | |
|---|---|
| TS_INGEST_TO_LND | SP_LOAD_RAW_TO_LND() |
| TS_LOAD_LND_TO_STG | SP_LOAD_LND_TO_STG() |
| TS_LOAD_STG_TO_PRODUCT_HIST | SP_LOAD_TO_PRODUCT_HIST() |
| TS_LOAD_STG_TO_CUSTOMER_HIST | SP_LOAD_TO_CUSTOMER_HIST() |
| TS_LOAD_STG_TO_TRANSACTION_HIST | SP_LOAD_TO_TRANSACTION_HIST() |
| TS_LOAD_HIST_TO_DIM_CUSTOMER | SP_LOAD_TO_DIM_CUSTOMER() |
| TS_LOAD_HIST_TO_DIM_PRODUCT | SP_LOAD_TO_DIM_PRODUCT() |
| TS_LOAD_HIST_TO_FACT_TRANSACTION | SP_LOAD_TO_FACT_TRANSACTION() |


TS_INGEST_TO_LND
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_LND_TO_STG
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_STG_TO_CUST...
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_HIST_TO_DIM_...
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_STG_TO_PROD...
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_HIST_TO_DIM_...
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_STG_TO_TRAN...
DEMO_DB.TASK_FLOW
Succeeded
TS_LOAD_HIST_TO_FACT_...
DEMO_DB.TASK_FLOW
Succeeded

# DATA MODELING




dim_customer
customer_code STRING
customer_name STRING
contact_person STRING
contact_email STRING
area_name STRING
region_name STRING
fact_transaction_sales
invoice_code STRING
customer_code STRING
product_code STRING
price FLOAT
quantity INT
invoice_date DATE
created_at TIMESTAMP
dim_product
product_code STRING
product_name STRING
dim_date
date DATE
day int
month int
year int
fact_daily_target
date_key DATE
target FLOAT
dbdiagra


DIM_DATE
DATE
DAY
MONTH
YEAR
Collapse
DIM_PRODUCT
PRODUCT_CODE
PRODUCT_NAME
Collapse
DIM_CUSTOMER
AREA_NAME
CONTACT_EMAIL
CONTACT_PERSON
CUSTOMER_CODE
CUSTOMER_NAME
REGION_NAME
Collapse
FACT_DAILY_TARGET
DATE_KEY
TARGET
Collapse
FACT_TRANSACTION_S...
Column
CREATED_AT
CUSTOMER_CODE
INVOICE_CODE
INVOICE_DATE
PRICE
PRODUCT_CODE
Collapse
1
1
1
1
1
*
*
*

# Power BI

# DASHBOARD

Centre | Hanoi | HCMC | Mekong | North

Updated to
04-10-2021

**Total sales**
20.34bn

**Previous day sales**
20.34bn

**Sales gap**
0%

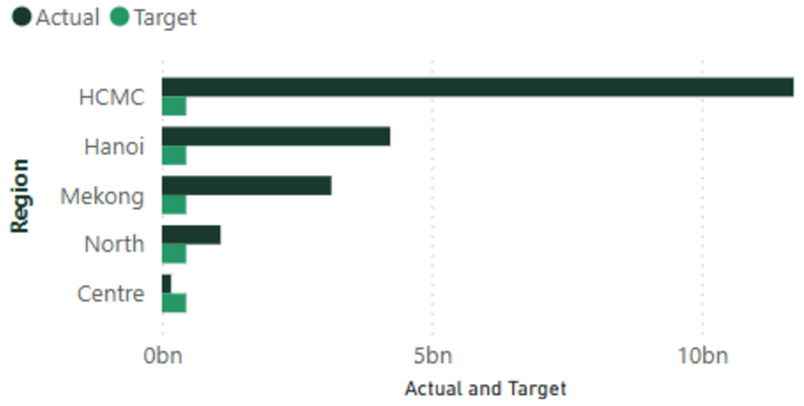

**% Achieve**
No data

**Date**
All

**Product**
All

**Area**
All

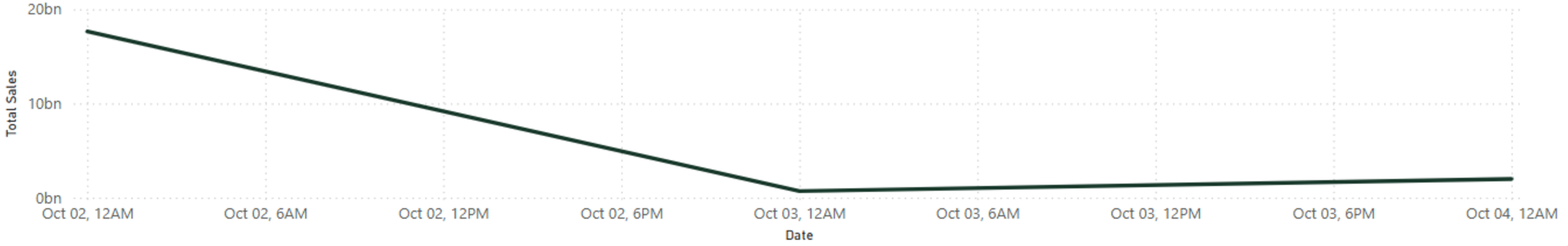
## Actual vs Target




Customer | Product

| Customer | Product |
|---|---|
| Quay Tuan Huyen Hello | QUAT MAY SENKO |
| Quay So 1011-Ds Chu Manh Cuong Hello | TU LANH SAMSUNG |
| Quay So 0-Quay CPDP Trang Anh Hello | MAY GIAT ELECTROLU |
| Quay So 0-Quay CPDP Trang Anh Hello | MAY LANH SONY |
| Quay Nguyen Tai Hello | LAPTOP LENOVO |
| Quay Nguyen Tai Hello | MAY GIAT ELECTROLU |
| Quay Nguyen Tai Hello | MAY LOC KHONG KHI |
| **Total** | |

## Sales by time



# DETAIL

**Quay Tuan Huyen Hello**

| Invoice code | Date | Customer code | Customer name | Region | Product | Unit Price | Quantity | Total Sales |
|---|---|---|---|---|---|---|---|---|
| 1225691726 | Sunday, October 03, 2021 | 30184794 | Quay Tuan Huyen Hello | Mekong | QUAT MAY SENKO | 206,914.00 | 10 | 2,069,140 |
| **Total** | | | | | | **206,914.00** | **10** | **2,069,140** |

# Data Build Tool

## 1. What’s DBT

Data Build Tool follows the concept of ELT (Extract, Load and Transform). Traditionally, data processes are familiar to ETL, that the data transformation take place on third-party applications (like SSIS or ADF)

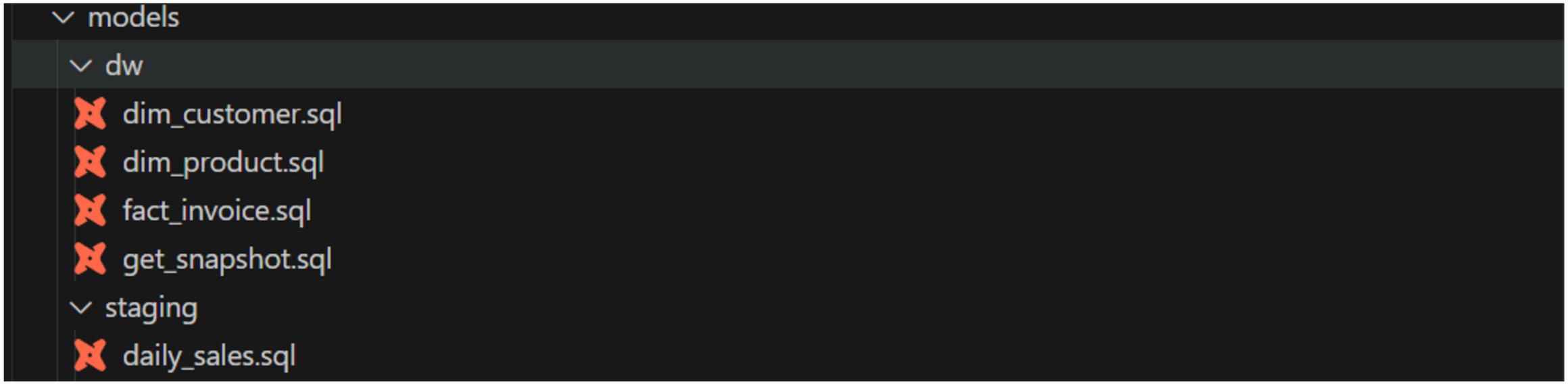
Now, to modern warehouse as  Snowflake, Redshift or Bigquery provide warehouse computing resources that empower massive transformation along with databases

**There are some advantages of ELT:**

- **Cost-saving due to using the same computing warehouse with databases**
- **The data transformation takes place right in Snowflake, help the process slow down**

## 2. Model

Model refers to a SQL file that defines transformation applied to the raw data in a datawarehouse (also the types of load). A model **(/models/schema_name/model_name.sql)** is named after its destination



# 3. Model Configuration

File dbt_project.yml defines which table with specified schema relates to a model

```
33  models:
34    dbtlearn:
35      dw:
36        dim_customer:
37          +materialized: table
38          +schema: dw
39        dim_product:
40          +materialized: table
41          +schema: dw
42        fact_invoice:
43          +materialized: table
44          +schema: dw
45      staging:
46        +materialized: table
47        +schema: stg
```

```yaml
dbtlearn:
  outputs:
    dev:
      account: bb59126.southeast-asia.azure
      database: demo_db
      password: *****
      role: accountadmin
      schema: dbt
```

**profiles.yml** stores the information of connection to specified storage. If any changes occur, just update the data in this file, in window this file's placed in *C:\User\Admin\.dbt* (run **dbt init** will get this file)

The schema in **.dbt/profiles.yml** defines the prefix of schema. If you specify the schema of **dim_customer.sql** in **dbt_project.yml** as **dw**, you actually get the table **DEMO_DB.DBT_DW.DIM_CUSTOMER** in Snowflake database

# 3. Incremental load

```
{{
    config(
        materialized='incremental'
    )
}}
with DAILY_SALES as
(
    SELECT s.id as id,
        TRIM(s.customer_code) as customer_code,
        TRIM(s.customer_name) as customer_name,
        PARSE_JSON(s.customer_info):name::STRING as contact_person,
        PARSE_JSON(s.customer_info):email::STRING as contact_email,
        TRIM(s.customer_type_name) as customer_type_name,
        TRIM(s.area_name) as area_name,
        TRIM(s.region_name) as region_name,
        TRIM(s.invoice_number) as invoice_code,
        TRIM(s.item_code) as item_code,
        TRIM(s.item_name) as item_name,
        s.price,
        s.quantity,
        TO_DATE(s.invoice_date, 'YYYY-MM-DD') as invoice_date,
        CURRENT_TIMESTAMP() as created_at
    FROM DEMO_DB.SALES_LND.DAILY_SALES as s
)
select * from DAILY_SALES as d
{% if is_incremental() %}
  where d.invoice_date > (select
    CASE WHEN (select count(*) from {{ this }})=0 THEN TO_DATE('1000-01-01')
    ELSE
    MAX(invoice_date) END from DEMO_DB.DBT_STG.DAILY_SALES)
{% endif %}
```

# 4. SCD 2

```
INSERT_DATA AS (
    SELECT
        c.customer_code,
        c.customer_name,
        c.contact_person,
        c.contact_email,
        c.area_name,
        c.region_name,
        current_timestamp() as insert_at,
        NULL as update_at,
        (CASE
            WHEN (d.primary_checksum IS NULL) THEN TO_DATE('1000-01-01')
            ELSE current_date()
        END) as start_date,
        TO_DATE('9999-12-31') as end_date,
        'Y' as current_flag,
        c.primary_checksum,
        c.record_checksum
        FROM
            LATEST_CUSTOMER_INFO as c
            LEFT JOIN DEMO_DB.DBT_DW.DIM_CUSTOMER as d ON c.primary_checksum = d.primary_checksum
        WHERE
            d.primary_checksum IS NULL
            OR (
                c.record_checksum != d.record_checksum
                AND d.current_flag = 'Y'
            )
```

# create update_data and keep_date

```
UPDATE_DATA AS (
        SELECT
            d.customer_code as customer_code,
            d.customer_name as customer_name,
            d.contact_person as contact_person,
            d.contact_email as contact_email,
            d.area_name as area_name,
            d.region_name as region_name,
            d.insert_at as insert_at,
            current_timestamp() as update_at,
            d.start_date as start_date,
            DATEADD(day,-1,current_date()) as end_date,
            'N' as current_flag,
            d.primary_checksum,
            d.record_checksum
        FROM
            LATEST_CUSTOMER_INFO as c
            INNER JOIN DEMO_DB.DBT_DW.DIM_CUSTOMER as d ON c.primary_checksum = d.primary_checksum
        WHERE
            c.record_checksum != d.record_checksum
            AND d.current_flag = 'Y'
    ),
    KEEP_DATA AS
    (
        select d.* from DEMO_DB.DBT_DW.DIM_CUSTOMER as d LEFT JOIN LATEST_CUSTOMER_INFO as c ON d.primary_check
        where d.current_flag = 'N' OR c.primary_checksum IS NULL
        OR (d.current_flag='Y' and d.record_checksum=c.record_checksum)
    ),
```

## the final data

```
        FINAL_DATA AS
        (
            (select * from UPDATE_DATA)
            UNION ALL
            (select * from INSERT_DATA)
            UNION ALL
            (select * from KEEP_DATA)
        )
select * from FINAL_DATA
```

Because **MERGE** operation doesn't work in dbt model, I create 3 flows: **INSERT_DATA, UPDATE_DATE and KEEP_DATA**. Finally, union all of them and full load to the demo_db.dbt_dw.dim_customer

This is my initial approach for SCD type 2. It works, but not in effective way compared to dbt snapshot

## 5. SCD 2 with snapshot

| id | status | updated_at | dbt_valid_from | dbt_valid_to | dbt_is_deleted |
|---|---|---|---|---|---|
| 1 | pending | 2024-01-01 10:47 | 2024-01-01 10:47 | 2024-01-01 11:05 | False |
| 1 | shipped | 2024-01-01 11:05 | 2024-01-01 11:05 | 2024-01-01 11:20 | False |
| 1 | deleted | 2024-01-01 11:20 | 2024-01-01 11:20 | 2024-01-01 12:00 | True |
| 1 | restored | 2024-01-01 12:00 | 2024-01-01 12:00 | | False |

Same as model, snapshot is defined in **/snapshot/customer_snapshot.sql**. The changes of customer extracted from **{{ ref("daily_sales") }}** are tracked and stored in the snapshot. You just specify which columns are used for the unique_key and which ones are used for determining the changes

```
{% snapshot customer_snapshot %}
{{
    config(
        target_schema = 'snapshot',
        unique_key = 'customer_code',
        strategy = 'check',
        check_cols = ["customer_name", "contact_person", "contact_email", "area_name","region_name"],
        data_valid_to_current = "dateadd(day, -1, current_date())"
    )
}}
select customer_code,
customer_name,
contact_person,
contact_email,
region_name,
area_name
from
(select *, row_number() over(partition by customer_code order by invoice_date DESC) as row_id from {{ {{ source('st
where row_id = 1
{% endsnapshot %}
```

The snapshot with the name of **customer_snapshot** is created in **customer_snapshot,** you can get this data in **demo_db.snapshot.customer_snapshot** or **{{ ref('customer_snapshot') }}**

| | EMAIL | REGION_NAME | AREA_NAME | DBT_SCD_ID | DBT_UPDATED_AT | DBT_VALID_FROM | DBT_VALID_TO |
|---|---|---|---|---|---|---|---|
| 1 | ey@example.com | North | Ba Ria | 560d6f76f848c2d119e8ab29b44b61a9 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 2 | @example.com | Mekong | Nghe An | 0a1ba90c239c1b9e382f2075d621ec95 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 3 | kayla@example.com | HCMC | Ho Chi Minh | dd557520429571810fc4e31477094543 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 4 | h@example.com | North | Khanh Hoa | ee52def09a221308f9f71be00b7da932 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 5 | ela@example.com | HCMC | Ho Chi Minh | a71fc37f68b6e6ff0fcbbbc7f0f8e510 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 6 | xample.net | HCMC | Ho Chi Minh | 1299bcedfb1a81a766bc057e517c05d4 | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 7 | example.com | Mekong | Khanh Hoa | 7476752040b449504d419a128306e11c | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 8 | ample.com | Mekong | Hai Phong | 910bd99258c9983e3609e3970e4f55dd | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 9 | n@example.net | Mekong | Nghe An | 2b970a508645974657ca63cc8df35dcb | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |
| 10 | ample.com | Mekong | Thanh Hoa | fcb5f9d8af4fe1d2240da2c9ae6eb94a | 2024-12-23 04:13:50.302 | 2024-12-23 04:13:50.302 | null |