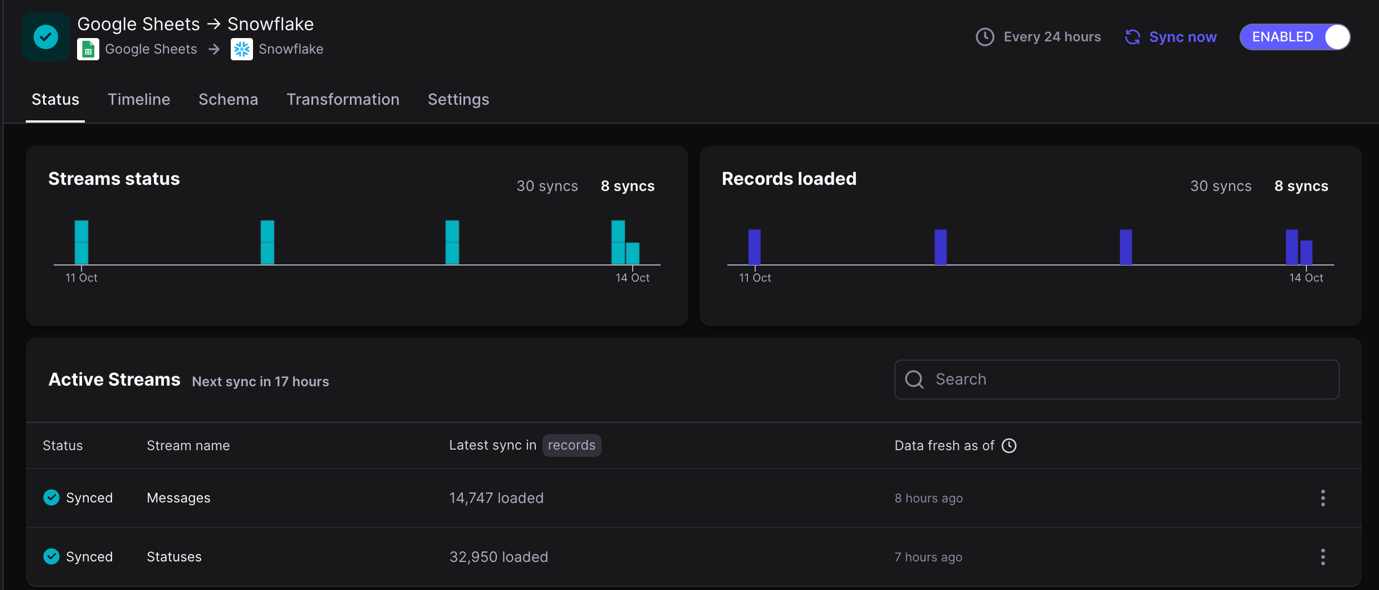
# Data Engineering Assignment Documentation

#### By Shubhangi Sanjay

## Data Extraction and Loading

I have used Airbyte to extract data from Google Sheets and loaded it into Snowflake

Video Link to setup pipeline on AIrbyte:  
https://www.loom.com/share/f49526c4366e42d2b6eb8478ed141353?sid=3dd5bac5-d7b8-46ac-ace5-fe3fcafeb073



Data Transformation is done on Snowflake and all the queries are provided below in Data Transformation section.

## Data Transformation

### Row count validation to check ETL pipeline has successfully run.

Select id from messages

group by id

having count(id) > 1;

select id from statuses

group by id

having count(id) > 1;

*Verify the duplicate rows with the source data using Ctrl+F*

### Query to create table in Snowflake to store the result of merging tables – messages and statuses and store the data into the new table ‘combined\_message\_status’

CREATE OR REPLACE TABLE combined\_message\_status (

-- Message table columns

message\_id STRING,

message\_content STRING,

message\_message\_type STRING,

message\_masked\_addressee STRING,

message\_masked\_from\_addr STRING,

message\_direction STRING,

message\_external\_id STRING,

message\_external\_timestamp TIMESTAMP,

message\_is\_deleted BOOLEAN,

message\_last\_status STRING,

message\_last\_status\_timestamp TIMESTAMP,

message\_rendered\_content STRING,

message\_uuid STRING,

message\_inserted\_at TIMESTAMP,

message\_updated\_at TIMESTAMP,

-- Status table columns

status\_id STRING,

status\_status STRING,

status\_timestamp TIMESTAMP,

status\_uuid STRING,

status\_message\_uuid STRING,

status\_number\_id STRING,

status\_inserted\_at TIMESTAMP,

status\_updated\_at TIMESTAMP

);

*Here we have stored all columns from messages and statuses table with prefix message\_ and status\_ respectively. This is done keeping in mind future prospects.*

INSERT INTO combined\_message\_status (

message\_id, message\_content, message\_message\_type, message\_masked\_addressee,

message\_masked\_from\_addr, message\_direction, message\_external\_id,

message\_external\_timestamp, message\_is\_deleted, message\_last\_status,

message\_last\_status\_timestamp, message\_rendered\_content, message\_uuid,

message\_inserted\_at, message\_updated\_at,

status\_id, status\_status, status\_timestamp, status\_uuid, status\_message\_uuid,

status\_number\_id, status\_inserted\_at, status\_updated\_at

)

SELECT

-- Message table columns

distinct(m.id) AS message\_id,

m.content AS message\_content,

m.message\_type AS message\_message\_type,

m.masked\_addressees AS message\_masked\_addressee,

m.masked\_from\_addr AS message\_masked\_from\_addr,

m.direction AS message\_direction,

m.external\_id AS message\_external\_id,

m.external\_timestamp AS message\_external\_timestamp,

m.is\_deleted AS message\_is\_deleted,

m.last\_status AS message\_last\_status,

m.last\_status\_timestamp AS message\_last\_status\_timestamp,

m.rendered\_content AS message\_rendered\_content,

m.uuid AS message\_uuid,

m.inserted\_at AS message\_inserted\_at,

m.updated\_at AS message\_updated\_at,

-- Status table columns

s.id AS status\_id,

s.status AS status\_status,

s.timestamp AS status\_timestamp,

s.uuid AS status\_uuid,

s.message\_uuid AS status\_message\_uuid,

s.number\_id AS status\_number\_id,

s.inserted\_at AS status\_inserted\_at,

s.updated\_at AS status\_updated\_at

FROM messages m

LEFT JOIN (

SELECT

s.\*

,ROW\_NUMBER() OVER (PARTITION BY s.message\_uuid ORDER BY s.timestamp DESC) AS rn

FROM statuses s

) s

ON m.uuid = s.message\_uuid

WHERE s.rn = 1;

Here we are selecting rows from status that are of status timestamp per message using message\_uuid as it is uniquely identifying each message. We are doing this because we want one row per message in our new table.

### FLAGGING DUPLICATE VALUES

ALTER TABLE combined\_message\_status

ADD COLUMN record\_status STRING;

*Adding an extra column record\_status that will have value – Unique or Duplicate*

UPDATE combined\_message\_status

SET record\_status = CASE

WHEN duplicate\_check.rank > 1 THEN ‘Duplicate’

ELSE ‘Unique’

END

FROM (

SELECT

message\_uuid,message\_content,message\_inserted\_at,

DENSE\_RANK() OVER (PARTITION BY message\_content,

DATE\_TRUNC(‘minute’, message\_inserted\_at)

ORDER BY message\_inserted\_at) AS rank

FROM combined\_message\_status

) AS duplicate\_check

WHERE combined\_message\_status.message\_uuid = duplicate\_check.message\_uuid;

*We are flagging values as duplicate when dense\_rank function has rank more than 1 for a particular row, where content and inserted\_at timestamp are the same.*

### VALIDATION CHECK

SELECT

COUNT(\*) AS total\_records,

SUM(CASE WHEN message\_content IS NULL THEN 1 ELSE 0 END) AS null\_content,

SUM(CASE WHEN message\_message\_type IS NULL THEN 1 ELSE 0 END) AS null\_message\_type,

SUM(CASE WHEN message\_direction IS NULL THEN 1 ELSE 0 END) AS null\_direction

FROM combined\_message\_status;

*Here we are checking for null values in content, null message type, direction.*

SELECT

COUNT(\*) AS total\_records,

SUM(CASE WHEN MESSAGE\_UPDATED\_AT < MESSAGE\_INSERTED\_AT THEN 1 ELSE 0 END) AS inconsistent\_timestamps

FROM combined\_message\_status;

*Here we are checking for Consistency in Timestamps: Ensuring that the updated\_at is always greater than or equal to inserted\_at.*

### Data Integrity Check:

*Verifying that combined table has same records as source tables by checking count of tables from messages table, combined\_message\_status table. Here we are not checking statuses table because we have done a left join.*

Select count(\*)

from messages;

--14,747

select count(\*)

from combined\_message\_status;

--12848

select count(\*) from

messages

where uuid not in

(select message\_uuid from statuses);

--1899

*Here few records seem to be missing from messages table because their message\_uuid were not found in statuses.*

## Data Visualisation

Data visualization is done on Google Collab. Here is the link with all necessary comments:

<https://colab.research.google.com/drive/1_kYA_wVPx88UqvjYY5sRTxOlQOYVeWyN?usp=sharing>