Snowflake Data Vault 2.0

A logo of a cloud with a stack of coins

AI-generated content may be incorrect.

Nuwan Keshara Galappaththi

Data Engineer

[LinkedIn](https://www.linkedin.com/in/nuwan-keshara/)

Contents

[Project Overview 3](#_Toc194364123)

[1. Project Description 3](#_Toc194364124)

[2. Project Purpose 3](#_Toc194364125)

[3. Reason for the project selection 3](#_Toc194364126)

[4. Project Objectives 4](#_Toc194364127)

[5. Project Architecture 5](#_Toc194364128)

[Technology Stack: 6](#_Toc194364129)

[Project Flow: 7](#_Toc194364130)

[Technical Documentation 8](#_Toc194364131)

[1. Setting up the Environment 8](#_Toc194364132)

[2. Staging Area 12](#_Toc194364133)

[3. Raw Data Vault 17](#_Toc194364134)

[4. Business Data Vault 28](#_Toc194364135)

[5. Information Delivery 32](#_Toc194364136)

[References 35](#_Toc194364137)

# Project Overview

## Project Description

* This project focused on the redesign and implementation of a robust and scalable data warehouse solution utilizing the Data Vault 2.0 methodology within the Snowflake cloud data platform. By using the publicly available ‘snowflake\_sample\_data’ database, specifically the ‘tpch\_sf10’ schema, this project demonstrated the practical application of Data Vault 2.0 principles to real world datasets. The project covered the Staging area, Raw Data Vault, Business Data Vault and Information Delivery stages, incorporating data sources such as the Customer, Orders, Region and Nation tables, which contain 1.5 million, 15 million, 5 and 25 records respectively. Furthermore the Information Delivery stage provided a dimensional data model using a star schema to facilitate analytical querying for business intelligence and reporting.

## Project Purpose

* Develop hands on experience with Snowflake cloud data platform and building data vault 2.0 architecture.
* Improve knowledge of Data Vault concepts and its architecture.
* Keen interest in learning new technologies and tools

## Reason for the project selection

* My background research on Brightly particularly related to open Data Engineer and Data Architect positions revealed that the organization as a data project development company, focusing on Snowflake and Data Vault practices. To align with this and to gain practical experience with these technologies, I chose this specific tech stack for this project.

## Project Objectives

* **Data Acquisition**: Extract data from source systems and make it accessible. Set up the environment to load the data.
* **Staging Area**: Load the data into the staging tables and create Snowpipe and Stream to ingest data near real-time into the Raw Data Vault and capture change data in the tables, respectively.
* **Raw Data Vault**: Implement the Data Vault 2.0 model on the dataset.
* **Business Data Vault**: Data is aggregated and transformed into tables and views for better analytics
* **Information Delivery**: Develop a dimensional model on the data to facilitate analytics and business reports

## Project Architecture

A diagram of a company

AI-generated content may be incorrect.

A close-up of a document

AI-generated content may be incorrect.

### Technology Stack:

* Snowflake cloud data platform
* Data Vault 2.0
* Dimensional Model (Star Schema)



### Project Flow:

* Data Sources
  + snowflake\_sample\_data.tpch\_sf10
    - Customer – 1.5 M records
    - Orders – 15 M records
    - Region – 5 records
    - Nation – 25 records
* Data Pipeline
  + Streams
  + Snowpipe
  + Tasks
* Data Vault
  + HUB\_CUSTOMER
  + HUB\_ORDER
  + SAT\_CUSTOMER
  + SAT\_ORDER
  + LNK\_CUSTOMER\_ORDER
  + REF\_REGION
  + REF\_NATION
* Information Delivery
  + DIM\_CUSTOMER
  + DIM\_ORDER
  + FCT\_CUSTOMER\_ORDER

# Technical Documentation

## Setting up the Environment

* For this project, I used the 'accountadmin' role for the purpose of the Data Vault implementation.

Ex: *USE ROLE accountadmin;*

* Create a database and use it.

Ex: *CREATE OR REPLACE DATABASE dv\_lab;*

*USE DATABASE dv\_lab;*

* Create two virtual data warehouses for generic warehouse work and to run Data Vault object pipelines

Ex:

* *CREATE OR REPLACE WAREHOUSE dv\_lab\_wh WITH WAREHOUSE\_SIZE = 'XSMALL' MIN\_CLUSTER\_COUNT = 1 MAX\_CLUSTER\_COUNT = 1 AUTO\_SUSPEND = 60 COMMENT = 'Generic WH';*
* *CREATE OR REPLACE WAREHOUSE dv\_rdv\_wh WITH WAREHOUSE\_SIZE = 'XSMALL' MIN\_CLUSTER\_COUNT = 1 MAX\_CLUSTER\_COUNT = 1 AUTO\_SUSPEND = 60 COMMENT = 'WH for Raw Data Vault object pipelines';*
* Create schemas for the staging area, Raw Data Vault, Business Data Vault and Information Delivery.

Ex:

* *USE WAREHOUSE dv\_lab\_wh;*
* *CREATE OR REPLACE SCHEMA l00\_stg COMMENT = 'Schema for Staging Area objects';*
* *CREATE OR REPLACE SCHEMA l10\_rdv COMMENT = 'Schema for Raw Data Vault objects';*
* *CREATE OR REPLACE SCHEMA l20\_bdv COMMENT = 'Schema for Business Data Vault objects';*
* *CREATE OR REPLACE SCHEMA l30\_id COMMENT = 'Schema for Information Delivery objects';*

A close-up of a text

AI-generated content may be incorrect.

* Create stg\_customer and stg\_orders tables in the staging area.

Ex:

* *USE SCHEMA l00\_stg;*
* *CREATE OR REPLACE TABLE stg\_customer*

*(*

*raw\_json VARIANT*

*, filename STRING NOT NULL*

*, file\_row\_seq NUMBER NOT NULL*

*, ldts STRING NOT NULL*

*, rscr STRING NOT NULL*

*);*

* *CREATE OR REPLACE TABLE stg\_orders*

*(*

*o\_orderkey NUMBER*

*, o\_custkey NUMBER*

*, o\_orderstatus STRING*

*, o\_totalprice NUMBER*

*, o\_orderdate DATE*

*, o\_orderpriority STRING*

*, o\_clerk STRING*

*, o\_shippriority NUMBER*

*, o\_comment STRING*

*, filename STRING NOT NULL*

*, file\_row\_seq NUMBER NOT NULL*

*, ldts STRING NOT NULL*

*, rscr STRING NOT NULL*

*);*

* To demonstrate the purpose of Snowpipe, I created a stage to store customer data (JSON) and order data (CSV) files from a subset of the customer and orders source table data.

Ex:

* *CREATE OR REPLACE STAGE customer\_data FILE\_FORMAT = (TYPE = JSON);*
* *CREATE OR REPLACE STAGE orders\_data FILE\_FORMAT = (TYPE = CSV) ;*
* Before loading customer and orders data into the stg\_customer and stg\_orders tables in the staging area, create a data Stream on those tables to capture change data.

Ex:

* *CREATE OR REPLACE STREAM stg\_customer\_strm ON TABLE stg\_customer;*
* *CREATE OR REPLACE STREAM stg\_orders\_strm ON TABLE stg\_orders;*
* Finally, before loading data into the stage, create a Snowpipe from the stage to the stg\_customer and stg\_orders tables in the staging area to facilitate near real-time stream data.

Ex:

* *CREATE OR REPLACE PIPE stg\_orders\_pp*

*AS*

*COPY INTO stg\_orders*

*FROM*

*(*

*SELECT $1,$2,$3,$4,$5,$6,$7,$8,$9*

*, metadata$filename*

*, metadata$file\_row\_number*

*, CURRENT\_TIMESTAMP()*

*, 'Orders System'*

*FROM @orders\_data);*

* *CREATE OR REPLACE PIPE stg\_customer\_pp*

*AS*

*COPY INTO stg\_customer*

*FROM*

*(*

*SELECT $1*

*, metadata$filename*

*, metadata$file\_row\_number*

*, CURRENT\_TIMESTAMP()*

*, 'Customers System'*

*FROM @customer\_data*

*);*

* Start Snowpipe

Ex:

* *ALTER PIPE stg\_customer\_pp REFRESH;*
* *ALTER PIPE stg\_orders\_pp REFRESH;*

## Staging Area

A white sheet of paper with black text

AI-generated content may be incorrect.

* Now load the data into the staging tables

Ex: Region and Nation data are loaded into the staging area's stg\_region and stg\_nation tables from the source data.

* *CREATE OR REPLACE TABLE stg\_nation*

*AS*

*SELECT src.\**

*, CURRENT\_TIMESTAMP() ldts*

*, 'Static Reference Data' rscr*

*FROM snowflake\_sample\_data.tpch\_sf10.nation src;*

* *CREATE OR REPLACE TABLE stg\_region*

*AS*

*SELECT src.\**

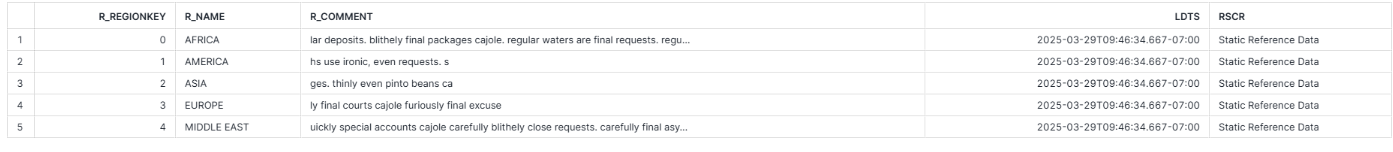
*, CURRENT\_TIMESTAMP() ldts*

*, 'Static Reference Data' rscr*

*FROM snowflake\_sample\_data.tpch\_sf10.region src;*

A screenshot of a computer

AI-generated content may be incorrect.Ex: stg\_nation table

Ex: stg\_region table

* Send data from customer table (10 records) and orders table(1000 records)

Ex:

* *COPY INTO @customer\_data*

*FROM*

*(SELECT object\_construct(\*)*

*FROM snowflake\_sample\_data.tpch\_sf10.customer limit 10*

*)*

*INCLUDE\_QUERY\_ID=TRUE;*

* *COPY INTO @orders\_data*

*FROM*

*(SELECT \**

*FROM snowflake\_sample\_data.tpch\_sf10.orders limit 1000*

*)*

*INCLUDE\_QUERY\_ID=TRUE;*

A white paper with blue lines

AI-generated content may be incorrect.A white background with black lines

AI-generated content may be incorrect.

* Now using Snowpipe, the data will automatically be sent into the stg\_customer and stg\_orders tables in the staging area

A close up of a text

AI-generated content may be incorrect.Ex: stg\_customer table

A white paper with black text

AI-generated content may be incorrect.Ex: stg\_orders table

* Using the Stream data captured from change data in the stg\_customer and stg\_orders tables, create views for later use.

Ex:

* *CREATE OR REPLACE VIEW stg\_customer\_strm\_outbound AS*

*SELECT src.\**

*, raw\_json:C\_CUSTKEY::NUMBER c\_custkey*

*, raw\_json:C\_NAME::STRING c\_name*

*, raw\_json:C\_ADDRESS::STRING c\_address*

*, raw\_json:C\_NATIONKEY::NUMBER C\_nationcode*

*, raw\_json:C\_PHONE::STRING c\_phone*

*, raw\_json:C\_ACCTBAL::NUMBER c\_acctbal*

*, raw\_json:C\_MKTSEGMENT::STRING c\_mktsegment*

*, raw\_json:C\_COMMENT::STRING c\_comment*

*, SHA1\_BINARY(UPPER(TRIM(c\_custkey))) sha1\_hub\_customer*

*, SHA1\_BINARY(UPPER(ARRAY\_TO\_STRING(ARRAY\_CONSTRUCT(*

*NVL(TRIM(c\_name) ,'-1')*

*, NVL(TRIM(c\_address) ,'-1')*

*, NVL(TRIM(c\_nationcode) ,'-1')*

*, NVL(TRIM(c\_phone) ,'-1')*

*, NVL(TRIM(c\_acctbal) ,'-1')*

*, NVL(TRIM(c\_mktsegment) ,'-1')*

*, NVL(TRIM(c\_comment) ,'-1')*

*), '^'))) AS customer\_hash\_diff*

*FROM stg\_customer\_strm src;*

* *CREATE OR REPLACE VIEW stg\_order\_strm\_outbound AS*

*SELECT src.\**

*, SHA1\_BINARY(UPPER(TRIM(o\_orderkey))) sha1\_hub\_order*

*, SHA1\_BINARY(UPPER(TRIM(o\_custkey))) sha1\_hub\_customer*

*, SHA1\_BINARY(UPPER(ARRAY\_TO\_STRING(ARRAY\_CONSTRUCT( NVL(TRIM(o\_orderkey) ,'-1')*

*, NVL(TRIM(o\_custkey) ,'-1')*

*), '^'))) AS sha1\_lnk\_customer\_order*

*, SHA1\_BINARY(UPPER(ARRAY\_TO\_STRING(ARRAY\_CONSTRUCT( NVL(TRIM(o\_orderstatus) , '-1')*

*, NVL(TRIM(o\_totalprice) , '-1')*

*, NVL(TRIM(o\_orderdate) , '-1')*

*, NVL(TRIM(o\_orderpriority) , '-1')*

*, NVL(TRIM(o\_clerk) , '-1')*

*, NVL(TRIM(o\_shippriority) , '-1')*

*, NVL(TRIM(o\_comment) , '-1')*

*), '^'))) AS order\_hash\_diff*

*FROM stg\_orders\_strm src;*

## Raw Data Vault

A diagram of a company

AI-generated content may be incorrect.

* Create data vault tables
* *USE SCHEMA l10\_rdv;*

Ex: hub\_customer

* *CREATE OR REPLACE TABLE hub\_customer*

*(*

*sha1\_hub\_customer BINARY NOT NULL*

*, c\_custkey NUMBER NOT NULL*

*, ldts TIMESTAMP NOT NULL*

*, rscr STRING NOT NULL*

*, CONSTRAINT pk\_hub\_customer PRIMARY KEY(sha1\_hub\_customer)*

*);*

Ex: hub\_order

* *CREATE OR REPLACE TABLE hub\_order*

*(*

*sha1\_hub\_order BINARY NOT NULL*

*, o\_orderkey NUMBER NOT NULL*

*, ldts TIMESTAMP NOT NULL*

*, rscr STRING NOT NULL*

*, CONSTRAINT pk\_hub\_order PRIMARY KEY(sha1\_hub\_order)*

*);*

Ex: sat\_customer

* *CREATE OR REPLACE TABLE sat\_customer*

*(*

*sha1\_hub\_customer BINARY NOT NULL*

*, ldts TIMESTAMP NOT NULL*

*, c\_name STRING*

*, c\_address STRING*

*, c\_phone STRING*

*, c\_acctbal NUMBER*

*, c\_mktsegment STRING*

*, c\_comment STRING*

*, nationcode NUMBER*

*, hash\_diff BINARY NOT NULL*

*, rscr STRING NOT NULL*

*, CONSTRAINT pk\_sat\_customer PRIMARY KEY(sha1\_hub\_customer, ldts)*

*, CONSTRAINT fk\_sat\_customer FOREIGN KEY(sha1\_hub\_customer) REFERENCES hub\_customer*

*);*

Ex: sat\_order

* *CREATE OR REPLACE TABLE sat\_order*

*(*

*sha1\_hub\_order BINARY NOT NULL*

*, ldts TIMESTAMP NOT NULL*

*, o\_orderstatus STRING*

*, o\_totalprice NUMBER*

*, o\_orderdate DATE*

*, o\_orderpriority STRING*

*, o\_clerk STRING*

*, o\_shippriority NUMBER*

*, o\_comment STRING*

*, hash\_diff BINARY NOT NULL*

*, rscr STRING NOT NULL*

*, CONSTRAINT pk\_sat\_order PRIMARY KEY(sha1\_hub\_order, ldts)*

*, CONSTRAINT fk\_sat\_order FOREIGN KEY(sha1\_hub\_order) REFERENCES hub\_order*

*);*

Ex: lnk\_customer\_order

* *CREATE OR REPLACE TABLE lnk\_customer\_order*

*(*

*sha1\_lnk\_customer\_order BINARY NOT NULL*

*, sha1\_hub\_customer BINARY*

*, sha1\_hub\_order BINARY*

*, ldts TIMESTAMP NOT NULL*

*, rscr STRING NOT NULL*

*, CONSTRAINT pk\_lnk\_customer\_order PRIMARY KEY(sha1\_lnk\_customer\_order)*

*, CONSTRAINT fk1\_lnk\_customer\_order FOREIGN KEY(sha1\_hub\_customer) REFERENCES hub\_customer*

*, CONSTRAINT fk2\_lnk\_customer\_order FOREIGN KEY(sha1\_hub\_order) REFERENCES hub\_order*

*);*

* Create and load region and nation data into ref\_region and ref\_nation

Ex: ref\_region

* *CREATE OR REPLACE TABLE ref\_region*

*(*

*regioncode NUMBER*

*, ldts TIMESTAMP*

*, rscr STRING NOT NULL*

*, r\_name STRING*

*, r\_comment STRING*

*, CONSTRAINT PK\_REF\_REGION PRIMARY KEY (REGIONCODE)*

*)*

*AS*

*SELECT r\_regionkey*

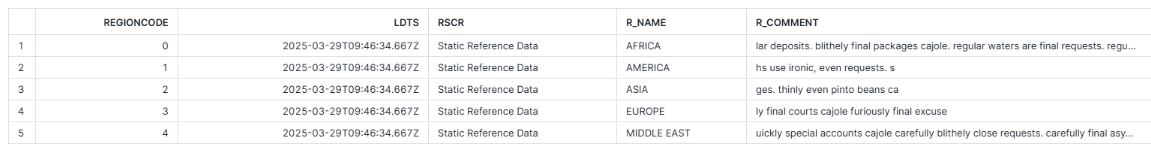
*, ldts*

*, rscr*

*, r\_name*

*, r\_comment*

*FROM l00\_stg.stg\_region;*



Ex: ref\_nation

* *CREATE OR REPLACE TABLE ref\_nation*

*(*

*nationcode NUMBER*

*, regioncode NUMBER*

*, ldts TIMESTAMP*

*, rscr STRING NOT NULL*

*, n\_name STRING*

*, n\_comment STRING*

*, CONSTRAINT pk\_ref\_nation PRIMARY KEY (nationcode)*

*, CONSTRAINT fk\_ref\_region FOREIGN KEY (regioncode) REFERENCES ref\_region(regioncode)*

*)*

*AS*

*SELECT n\_nationkey*

*, n\_regionkey*

*, ldts*

*, rscr*

*, n\_name*

*, n\_comment*

*FROM l00\_stg.stg\_nation;*

*A screenshot of a computer

AI-generated content may be incorrect.*

* In the previous staging area, if new or changed data has arrived, then use a Task to execute and load the data into hub\_customer and sat\_customer.
* First, Task checks if there is changed data captured in stg\_customer\_strm. If there is, the task proceeds and inserts the new data into the Raw Data Vault customer tables. This checking occurs at 1-minute intervals.

Ex:

* *CREATE OR REPLACE TASK customer\_strm\_tsk*

*WAREHOUSE = dv\_rdv\_wh*

*SCHEDULE = '1 minute'*

*WHEN*

*SYSTEM$STREAM\_HAS\_DATA('L00\_STG.STG\_CUSTOMER\_STRM')*

*AS*

*INSERT ALL*

*WHEN (SELECT COUNT(1) FROM hub\_customer tgt WHERE tgt.sha1\_hub\_customer = src\_sha1\_hub\_customer) = 0*

*THEN INTO hub\_customer*

*( sha1\_hub\_customer*

*, c\_custkey*

*, ldts*

*, rscr*

*)*

*VALUES*

*( src\_sha1\_hub\_customer*

*, src\_c\_custkey*

*, src\_ldts*

*, src\_rscr*

*)*

*WHEN (SELECT COUNT(1) FROM sat\_customer tgt WHERE tgt.sha1\_hub\_customer = src\_sha1\_hub\_customer AND tgt.hash\_diff = src\_customer\_hash\_diff) = 0*

*THEN INTO sat\_customer*

*(*

*sha1\_hub\_customer*

*, ldts*

*, c\_name*

*, c\_address*

*, c\_phone*

*, c\_acctbal*

*, c\_mktsegment*

*, c\_comment*

*, nationcode*

*, hash\_diff*

*, rscr*

*)*

*VALUES*

*(*

*src\_sha1\_hub\_customer*

*, src\_ldts*

*, src\_c\_name*

*, src\_c\_address*

*, src\_c\_phone*

*, src\_c\_acctbal*

*, src\_c\_mktsegment*

*, src\_c\_comment*

*, src\_nationcode*

*, src\_customer\_hash\_diff*

*, src\_rscr*

*)*

*SELECT sha1\_hub\_customer src\_sha1\_hub\_customer*

*, c\_custkey src\_c\_custkey*

*, c\_name src\_c\_name*

*, c\_address src\_c\_address*

*, c\_nationcode src\_nationcode*

*, c\_phone src\_c\_phone*

*, c\_acctbal src\_c\_acctbal*

*, c\_mktsegment src\_c\_mktsegment*

*, c\_comment src\_c\_comment*

*, customer\_hash\_diff src\_customer\_hash\_diff*

*, ldts src\_ldts*

*, rscr src\_rscr*

*FROM l00\_stg.stg\_customer\_strm\_outbound src*

*;*

A white paper with black lines

AI-generated content may be incorrect.

A white sheet with black text

AI-generated content may be incorrect.

* Similar to the hub\_customer and sat\_customer tables, hub\_order, sat\_order, and lnk\_customer\_order first check the stg\_order\_str stream, and if there is changed data, those tables will be populated

Ex:

* *CREATE OR REPLACE TASK order\_strm\_tsk*

*WAREHOUSE = dv\_rdv\_wh*

*SCHEDULE = '1 minute'*

*WHEN*

*SYSTEM$STREAM\_HAS\_DATA('L00\_STG.STG\_ORDERS\_STRM')*

*AS*

*INSERT ALL*

*WHEN (SELECT COUNT(1) FROM hub\_order tgt WHERE tgt.sha1\_hub\_order = src\_sha1\_hub\_order) = 0*

*THEN INTO hub\_order*

*( sha1\_hub\_order*

*, o\_orderkey*

*, ldts*

*, rscr*

*)*

*VALUES*

*( src\_sha1\_hub\_order*

*, src\_o\_orderkey*

*, src\_ldts*

*, src\_rscr*

*)*

*WHEN (SELECT COUNT(1) FROM sat\_order tgt WHERE tgt.sha1\_hub\_order = src\_sha1\_hub\_order AND tgt.hash\_diff = src\_order\_hash\_diff) = 0*

*THEN INTO sat\_order*

*(*

*sha1\_hub\_order*

*, ldts*

*, o\_orderstatus*

*, o\_totalprice*

*, o\_orderdate*

*, o\_orderpriority*

*, o\_clerk*

*, o\_shippriority*

*, o\_comment*

*, hash\_diff*

*, rscr*

*)*

*VALUES*

*(*

*src\_sha1\_hub\_order*

*, src\_ldts*

*, src\_o\_orderstatus*

*, src\_o\_totalprice*

*, src\_o\_orderdate*

*, src\_o\_orderpriority*

*, src\_o\_clerk*

*, src\_o\_shippriority*

*, src\_o\_comment*

*, src\_order\_hash\_diff*

*, src\_rscr*

*)*

*WHEN (SELECT COUNT(1) FROM lnk\_customer\_order tgt WHERE tgt.sha1\_lnk\_customer\_order = src\_sha1\_lnk\_customer\_order) = 0*

*THEN INTO lnk\_customer\_order*

*(*

*sha1\_lnk\_customer\_order*

*, sha1\_hub\_customer*

*, sha1\_hub\_order*

*, ldts*

*, rscr*

*)*

*VALUES*

*(*

*src\_sha1\_lnk\_customer\_order*

*, src\_sha1\_hub\_customer*

*, src\_sha1\_hub\_order*

*, src\_ldts*

*, src\_rscr*

*)*

*SELECT sha1\_hub\_order src\_sha1\_hub\_order*

*, sha1\_lnk\_customer\_order src\_sha1\_lnk\_customer\_order*

*, sha1\_hub\_customer src\_sha1\_hub\_customer*

*, o\_orderkey src\_o\_orderkey*

*, o\_orderstatus src\_o\_orderstatus*

*, o\_totalprice src\_o\_totalprice*

*, o\_orderdate src\_o\_orderdate*

*, o\_orderpriority src\_o\_orderpriority*

*, o\_clerk src\_o\_clerk*

*, o\_shippriority src\_o\_shippriority*

*, o\_comment src\_o\_comment*

*, order\_hash\_diff src\_order\_hash\_diff*

*, ldts src\_ldts*

*, rscr src\_rscr*

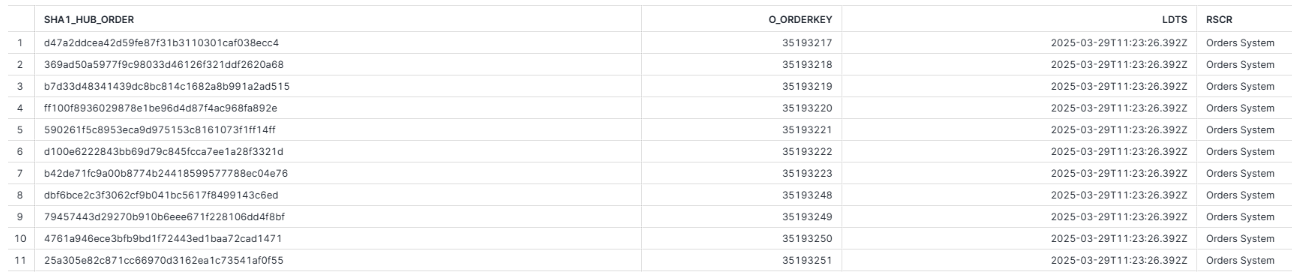
*FROM l00\_stg.stg\_order\_strm\_outbound src;*

* Start the Tasks

Ex:

*ALTER TASK customer\_strm\_tsk RESUME;*

*ALTER TASK order\_strm\_tsk RESUME;*



A white background with black text

AI-generated content may be incorrect.

A white sheet with black text

AI-generated content may be incorrect.

* Create two views for later use in the Information Delivery stage.

Ex: sat\_customer\_curr\_vw

* *CREATE VIEW sat\_customer\_curr\_vw*

*AS*

*SELECT \**

*FROM sat\_customer*

*QUALIFY LEAD(ldts) OVER (PARTITION BY sha1\_hub\_customer ORDER BY ldts) IS NULL;*

Ex: sat\_order\_curr\_vw

* *CREATE OR REPLACE VIEW sat\_order\_curr\_vw*

*AS*

*SELECT \**

*FROM sat\_order*

*QUALIFY LEAD(ldts) OVER (PARTITION BY sha1\_hub\_order ORDER BY ldts) IS NULL;*

## Business Data Vault

A close-up of a document

AI-generated content may be incorrect.

* Create a view

Ex: sat\_customer\_bv

* *CREATE OR REPLACE VIEW sat\_customer\_bv*

*AS*

*SELECT rsc.sha1\_hub\_customer*

*, rsc.ldts*

*, rsc.c\_name*

*, rsc.c\_address*

*, rsc.c\_phone*

*, rsc.c\_acctbal*

*, rsc.c\_mktsegment*

*, rsc.c\_comment*

*, rsc.nationcode*

*, rsc.rscr*

*, rrn.n\_name nation\_name*

*, rrr.r\_name region\_name*

*FROM l10\_rdv.sat\_customer rsc*

*LEFT OUTER JOIN l10\_rdv.ref\_nation rrn*

*ON (rsc.nationcode = rrn.nationcode)*

*LEFT OUTER JOIN l10\_rdv.ref\_region rrr*

*ON (rrn.regioncode = rrr.regioncode)*

*;*

* Create a table to populate data

Ex: sat\_order\_bv

* *CREATE OR REPLACE TABLE sat\_order\_bv*

*(*

*sha1\_hub\_order BINARY NOT NULL*

*, ldts TIMESTAMP NOT NULL*

*, o\_orderstatus STRING*

*, o\_totalprice NUMBER*

*, o\_orderdate DATE*

*, o\_orderpriority STRING*

*, o\_clerk STRING*

*, o\_shippriority NUMBER*

*, o\_comment STRING*

*, hash\_diff BINARY NOT NULL*

*, rscr STRING NOT NULL*

*-- additional attributes*

*, order\_priority\_bucket STRING*

*, CONSTRAINT pk\_sat\_order PRIMARY KEY(sha1\_hub\_order, ldts)*

*, CONSTRAINT fk\_sat\_order FOREIGN KEY(sha1\_hub\_order) REFERENCES l10\_rdv.hub\_order*

*)*

*AS*

*SELECT sha1\_hub\_order*

*, ldts*

*, o\_orderstatus*

*, o\_totalprice*

*, o\_orderdate*

*, o\_orderpriority*

*, o\_clerk*

*, o\_shippriority*

*, o\_comment*

*, hash\_diff*

*, rscr*

*-- derived additional attributes*

*, CASE WHEN o\_orderpriority IN ('2-HIGH', '1-URGENT') AND o\_totalprice >= 200000 THEN 'Tier-1'*

*WHEN o\_orderpriority IN ('3-MEDIUM', '2-HIGH', '1-URGENT') AND o\_totalprice BETWEEN 150000 AND 200000 THEN 'Tier-2'*

*ELSE 'Tier-3'*

*END order\_priority\_bucket*

*FROM l10\_rdv.sat\_order;*

* Create a new task to populate the sat\_order\_bv table after the order\_strm\_task (the task to populate hub\_order, sat\_order, and lnk\_customer\_order from stream data in the staging area) is executed.

Ex:

* *CREATE OR REPLACE TASK l10\_rdv.hub\_order\_strm\_sat\_order\_bv\_tsk*

*WAREHOUSE = dv\_rdv\_wh*

*AFTER l10\_rdv.order\_strm\_tsk*

*AS*

*INSERT INTO l20\_bdv.sat\_order\_bv*

*SELECT*

*sha1\_hub\_order*

*, ldts*

*, o\_orderstatus*

*, o\_totalprice*

*, o\_orderdate*

*, o\_orderpriority*

*, o\_clerk*

*, o\_shippriority*

*, o\_comment*

*, hash\_diff*

*, rscr*

*-- derived additional attributes*

*, CASE WHEN o\_orderpriority IN ('2-HIGH', '1-URGENT') AND o\_totalprice >= 200000 THEN 'Tier-1'*

*WHEN o\_orderpriority IN ('3-MEDIUM', '2-HIGH', '1-URGENT') AND o\_totalprice BETWEEN 150000 AND 200000 THEN 'Tier-2'*

*ELSE 'Tier-3'*

*END order\_priority\_bucket*

*FROM sat\_order\_strm;*

*ALTER TASK l10\_rdv.hub\_order\_strm\_sat\_order\_bv\_tsk RESUME;*

*ALTER TASK l10\_rdv.order\_strm\_tsk RESUME;*

* Create 2 views for later use in Information Delivery stage

Ex: sat\_order\_bv\_curr\_vw

* *CREATE VIEW sat\_order\_bv\_curr\_vw*

*AS*

*SELECT \**

*FROM sat\_order\_bv*

*QUALIFY LEAD(ldts) OVER (PARTITION BY sha1\_hub\_order ORDER BY ldts) IS NULL;*

Ex: sat\_customer\_bv\_curr\_vw

* *CREATE VIEW sat\_customer\_bv\_curr\_vw*

*AS*

*SELECT \**

*FROM sat\_customer\_bv*

*QUALIFY LEAD(ldts) OVER (PARTITION BY sha1\_hub\_customer ORDER BY ldts) IS NULL;*

## Information Delivery

A close-up of a document

AI-generated content may be incorrect.

* Implement dimensional modelling views for fast analytics queries and reporting.

Ex: dim\_customer

* *CREATE OR REPLACE VIEW dim1\_customer*

*AS*

*SELECT hub.sha1\_hub\_customer AS dim\_customer\_key*

*, sat.ldts AS effective\_dts*

*, hub.c\_custkey AS customer\_id*

*, sat.rscr AS record\_source*

*, sat.\**

*FROM l10\_rdv.hub\_customer hub*

*, l20\_bdv.sat\_customer\_bv\_curr\_vw sat*

*WHERE hub.sha1\_hub\_customer = sat.sha1\_hub\_customer;*

Ex: dim\_order

* *CREATE OR REPLACE VIEW dim1\_order*

*AS*

*SELECT hub.sha1\_hub\_order AS dim\_order\_key*

*, sat.ldts AS effective\_dts*

*, hub.o\_orderkey AS order\_id*

*, sat.rscr AS record\_source*

*, sat.\**

*FROM l10\_rdv.hub\_order hub*

*, l20\_bdv.sat\_order\_bv\_curr\_vw sat*

*WHERE hub.sha1\_hub\_order = sat.sha1\_hub\_order;*

Ex: fct\_customer\_order

* *CREATE OR REPLACE VIEW fct\_customer\_order*

*AS*

*SELECT lnk.ldts AS effective\_dts*

*, lnk.rscr AS record\_source*

*, lnk.sha1\_hub\_customer AS dim\_customer\_key*

*, lnk.sha1\_hub\_order AS dim\_order\_key*

*-- this is a factless fact, but here you can add any measures, calculated or derived*

*FROM l10\_rdv.lnk\_customer\_order lnk;*

* Create charts from Snowsight

Ex:

* *SELECT dc.nation\_name*

*, dc.region\_name*

*, do.order\_priority\_bucket*

*, COUNT(1) cnt\_orders*

*FROM fct\_customer\_order fct*

*, dim1\_customer dc*

*, dim1\_order do*

*WHERE fct.dim\_customer\_key = dc.dim\_customer\_key*

*AND fct.dim\_order\_key = do.dim\_order\_key*

*GROUP BY 1,2,3;*

A colorful lines on a white background

AI-generated content may be incorrect.

## References

* [<https://quickstarts.snowflake.com/guide/vhol_data_vault/index.html?index=..%2F..index#0>] – Snowflake
* [[https://www.snowflake.com/en/emea/]](https://www.snowflake.com/en/emea/) - Snowflake