**Lab Exercise 2- ref and DAG in dbt Cloud with Snowflake**

In this lab exercise, we’ll use dbt to transform data in a **Data Warehouse** setup with Snowflake, following a **raw-to-analytics** pipeline. The objective is to create a staged transformation layer in dbt, connecting and processing data from the raw schema, and outputting transformed data in the analytics schema.

**Objective**

* Use dbt to transform data from raw to analytics schemas in Snowflake.
* Understand how dbt builds a transformation pipeline by using the ref function to create a Directed Acyclic Graph (DAG).

**Prerequisites**

* A **dbt Cloud** project connected to a Snowflake account.
* Ensure the following schemas and tables exist in Snowflake:
  + **Database:** raw
  + **Schemas:** jaffle\_shop and stripe in the raw database
  + **Target Database:** analytics

Note: If tables are not present, you can use example data, as outlined below.

**Step-by-Step Lab Exercise**

**Step 1: Create the raw and analytics Databases in Snowflake (if not already set up)**

Run the following SQL commands in Snowflake:

CREATE DATABASE IF NOT EXISTS raw;

CREATE DATABASE IF NOT EXISTS analytics;

CREATE SCHEMA IF NOT EXISTS raw.jaffle\_shop;

CREATE SCHEMA IF NOT EXISTS raw.stripe;

CREATE SCHEMA IF NOT EXISTS analytics;

**Step 2: Create Example Tables in the raw Schema**

For this lab, let’s assume the following tables are available:

1. **raw.jaffle\_shop.orders** – Contains order details.
2. **raw.jaffle\_shop.customers** – Contains customer information.
3. **raw.stripe.payments** – Contains payment information.

Create tables with sample data if these tables aren’t already set up.

CREATE OR REPLACE TABLE raw.jaffle\_shop.orders (

order\_id INT,

customer\_id INT,

order\_date DATE,

amount DECIMAL(10, 2)

);

INSERT INTO raw.jaffle\_shop.orders VALUES

(1, 101, '2024-01-01', 50.00),

(2, 102, '2024-01-05', 75.00),

(3, 101, '2024-01-10', 100.00);

CREATE OR REPLACE TABLE raw.jaffle\_shop.customers (

customer\_id INT,

customer\_name STRING,

signup\_date DATE

);

INSERT INTO raw.jaffle\_shop.customers VALUES

(101, 'John Doe', '2023-12-15'),

(102, 'Jane Smith', '2023-12-18');

**Step 3: Set Up Your dbt Project Structure**

In your dbt Cloud project:

1. Create folders staging/ and analytics/ under models.
2. In staging/, you’ll create models that pull data from the raw schema.
3. In analytics/, you’ll create final transformed models based on the staging models.

**Step 4: Create Staging Models**

1. In the models/staging/ folder, create a file named stg\_orders.sql:

{{ config(materialized='view') }}

SELECT

order\_id,

customer\_id,

order\_date,

amount

FROM {{ source('jaffle\_shop', 'orders') }}

1. In the same folder, create a file named stg\_customers.sql:

{{ config(materialized='view') }}

SELECT

customer\_id,

customer\_name,

signup\_date

FROM {{ source('jaffle\_shop', 'customers') }}

**Step 4: Create the Analytics Model Using ref**

In the models/analytics/ folder, create a file named customer\_order\_summary.sql:

{{ config(materialized='table') }}

SELECT

c.customer\_id,

c.customer\_name,

COUNT(o.order\_id) AS total\_orders,

SUM(o.amount) AS total\_order\_amount,

MIN(o.order\_date) AS first\_order\_date,

MAX(o.order\_date) AS last\_order\_date

FROM {{ ref('stg\_customers') }} AS c

LEFT JOIN {{ ref('stg\_orders') }} AS o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id, c.customer\_name

This query builds an aggregated customer summary based on staging models. Here, the ref function creates dependencies:

* customer\_order\_summary depends on stg\_customers, stg\_orders, and stg\_payments.
* dbt uses these dependencies to build the **DAG**.