**Lab Exercise 1- Simple dbt model with Snowflake**

**Objective:**

Here's a step-by-step **lab exercise** to help you get started with **dbt Cloud** and **Snowflake**. This exercise includes setting up a dbt project, creating a raw data table in Snowflake, building a simple dbt model to transform the data, running the model, and validating results.

**Prerequisites**

* Access to **dbt Cloud** and **Snowflake**.
* The **ACCOUNTADMIN** role (or equivalent permissions) in Snowflake to create and modify tables, databases, and schemas.

**Objective**

1. Set up a simple dbt Cloud project connected to Snowflake.
2. Create a raw data table in Snowflake.
3. Write a dbt model to transform the data.
4. Run and test the transformation model in dbt Cloud.
5. Verify the results in Snowflake.

**Step 1: Set Up the dbt Cloud Project**

1. **Log into dbt Cloud** and create a new project.
2. Go to **Settings** > **Environments** and set up a **Development** environment.
3. Configure the **Snowflake** connection with the following fields:
   * **Account**: Snowflake account identifier (e.g., abc12345.us-east-1).
   * **User**: Snowflake username.
   * **Password**: Snowflake password.
   * **Role**: ACCOUNTADMIN or a role with similar permissions.
   * **Warehouse**: Name of a Snowflake warehouse (e.g., COMPUTE\_WH).
   * **Database**: Name of your Snowflake database (e.g., DBT\_LAB).
   * **Schema**: Schema for dbt to create models (e.g., DBT\_SCHEMA).
4. Test and save the connection.

**Step 2: Create a Raw Data Table in Snowflake**

In Snowflake, create a database, schema, and a raw data table that dbt will transform.

1. **Run the following SQL commands** in Snowflake to set up the raw data table:

-- Create a database and schema for the lab

CREATE DATABASE IF NOT EXISTS DBT\_LAB;

CREATE SCHEMA IF NOT EXISTS DBT\_LAB.RAW\_DATA;

-- Create a raw data table for orders

CREATE OR REPLACE TABLE DBT\_LAB.RAW\_DATA.orders (

order\_id INT,

customer\_id INT,

order\_date DATE,

order\_amount DECIMAL(10, 2)

);

-- Insert sample data into the orders table

INSERT INTO DBT\_LAB.RAW\_DATA.orders (order\_id, customer\_id, order\_date, order\_amount) VALUES

(1, 101, '2023-10-01', 250.50),

(2, 102, '2023-10-02', 300.00),

(3, 101, '2023-10-05', 450.75),

(4, 103, '2023-10-07', 125.00),

(5, 102, '2023-10-10', 175.50);

This creates a table called orders with sample order data.

**Step 3: Define a Source in dbt**

1. In dbt Cloud, go to the **Develop** tab and open the IDE.
2. In the models folder, create a new YAML file called schema.yml.
3. Define the orders table as a **source** in schema.yml:

version: 2

sources:

- name: raw\_data # Schema where the raw data resides

database: DBT\_LAB

tables:

- name: orders # Table name in Snowflake

1. Save the file.

**Step 4: Create a dbt Model to Transform Data**

1. In the models folder, create a new SQL file called orders\_summary.sql.
2. Write a transformation model to summarize order data:

***-- models/orders\_summary.sql***

SELECT

customer\_id,

COUNT(order\_id) AS order\_count,

SUM(order\_amount) AS total\_spent

FROM

{{ source('raw\_data', 'orders') }}

GROUP BY

customer\_id

1. Save the file.

**Step 6: Run the Model in dbt Cloud**

1. In the dbt Cloud IDE, click on the **Play** button to run the model (or type dbt run in the command line).
2. dbt Cloud will execute the transformation and create a view (or table) called orders\_summary in the DBT\_SCHEMA schema in Snowflake.

**Step 7: Verify the Results in Snowflake**

1. Log into Snowflake and query the orders\_summary view (or table) to see the transformed data:

SELECT \* FROM DBT\_LAB.DBT\_SCHEMA.orders\_summary;

You should see a summary of the orders, with the customer\_id, order\_count, and total\_spent columns.