**Lab Exercise 4 -Jinja Template in dbt and Snowflake**

If you want to create a simple dbt model using **Jinja templating** without defining variables in dbt\_project.yml, you can define the variables directly in the model file. This approach is especially useful for quick experiments or simple labs where you don’t need to centralize variables across multiple models.

**Exercise Overview**

In this exercise, you’ll:

1. Set a hardcoded variable directly within the model file.
2. Use a Jinja if statement to conditionally add a column.

**Step 1: Create the Model File with Jinja Templating**

Let’s create a model file called ***filtered\_customers.sql*** in the models directory. In this file:

1. Define the variable signup\_year directly in the file.
2. Use the signup\_year variable to filter customers.
3. Add an if statement to conditionally include a signup\_status column based on the value of signup\_year.

-- models/filtered\_customers.sql

{{ config(

materialized = 'table'

) }}

{% set signup\_year = 2024 %} -- Define the variable directly within the model

WITH base AS (

SELECT

customer\_id,

first\_name,

last\_name,

email,

created\_at,

EXTRACT(YEAR FROM created\_at) AS signup\_year

FROM raw\_db.raw\_data.customers

)

SELECT

customer\_id,

first\_name,

last\_name,

email,

created\_at,

signup\_year

-- Conditionally add the signup\_status column if signup\_year is 2024 or later

{% if signup\_year >= 2024 %}

, 'New Signup' AS signup\_status

{% endif %}

FROM base

**Explanation of the Code**

* {% set signup\_year = 2024 %}: This line defines a signup\_year variable within the model file itself.
* WHERE signup\_year = {{ signup\_year }}: The signup\_year variable is used to filter records based on the specified year.
* The {% if signup\_year >= 2024 %} Jinja if statement conditionally adds a signup\_status column to the query output if signup\_year is 2024 or greater.

**Step 2: Run and Test the Model in dbt Cloud**

1. **Run the Model**: In dbt Cloud, go to **Deploy** -> **Run** or use the command line if working with dbt locally:

dbt run --select filtered\_customers

1. **Check Results**: Look at the filtered\_customers table in Snowflake to verify:
   * The table includes only records where signup\_year is equal to 2024.
   * The signup\_status column is added if signup\_year is 2024 or later.

**Summary**

This exercise demonstrates how to:

* Use Jinja to define a variable directly within the model file.
* Apply conditional logic without external configurations or a dbt\_project.yml variable definition.

This approach keeps the setup minimal while showing how Jinja templating can make your dbt SQL dynamic.