**Lab Exercise 9– dbt packages (dbt\_utils)**

In this lab exercise, we will use the dbt\_utils package (version 1.3.0) to create a view containing only unique records from a Snowflake table. This will involve using dbt\_utils.generate\_surrogate\_key to identify unique records based on specific columns. The final result will be saved as a view in dbt.

**Exercise Overview**

* Use dbt\_utils.generate\_surrogate\_key to identify unique records.
* Save the resulting data in a dbt model as a view.

**Prerequisites**

* A dbt project set up with a Snowflake connection.
* dbt\_utils package version 1.3.0 installed.

**Step 1: Set Up the dbt\_utils Package**

Add dbt\_utils to your packages.yml file if it isn’t already installed.

# packages.yml

packages:

- package: dbt-labs/dbt\_utils

version: 1.3.0

Then, install the package by running:

dbt deps

**Step 2: Create a Sample Table in Snowflake**

For this exercise, we’ll assume a table customer\_data with potential duplicate records. You can create and insert sample data in your Snowflake database if you have access, or use an existing table with similar structure.

**Sample SQL to Create customer\_data Table in Snowflake**

CREATE OR REPLACE TABLE customer\_data (

customer\_id INT,

first\_name VARCHAR,

last\_name VARCHAR,

signup\_date DATE

);

INSERT INTO customer\_data (customer\_id, first\_name, last\_name, signup\_date) VALUES

(1, 'John', 'Doe', '2023-01-10'),

(2, 'Jane', 'Doe', '2023-02-15'),

(1, 'John', 'Doe', '2023-01-10'), -- Duplicate

(4, 'Alice', 'Smith', '2023-03-20'),

(5, 'Bob', 'Brown', '2023-04-25');

**Step 3: Create a Model to Get Unique Records**

Next, create a model called unique\_customer\_data.sql that uses dbt\_utils.generate\_surrogate\_key to identify and select unique records based on the combination of key columns (customer\_id, first\_name, last\_name, signup\_date).

**File Location: models/unique\_customer\_data.sql**

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

WITH base AS (

SELECT

customer\_id,

first\_name,

last\_name,

signup\_date,

-- Generate a unique key based on specific columns

{{ dbt\_utils.generate\_surrogate\_key(['customer\_id', 'first\_name', 'last\_name', 'signup\_date']) }} AS unique\_key

FROM raw\_db.raw\_data. customer\_data

),

unique\_records AS (

-- Select distinct rows based on the unique key

SELECT DISTINCT \*

FROM base

)

SELECT \* FROM unique\_records

**Explanation**

* **generate\_surrogate\_key**: This macro creates a unique identifier based on the specified columns. This helps to detect and keep only unique records in the view.
* **DISTINCT**: Removes duplicate rows based on the unique columns specified in the base CTE.

**Step 5: Run the Model**

Execute the following command to create the unique\_customer\_data view in your Snowflake database:

dbt run --select unique\_customer\_data

This will create a view in your Snowflake database containing only unique records from the customer\_data table.