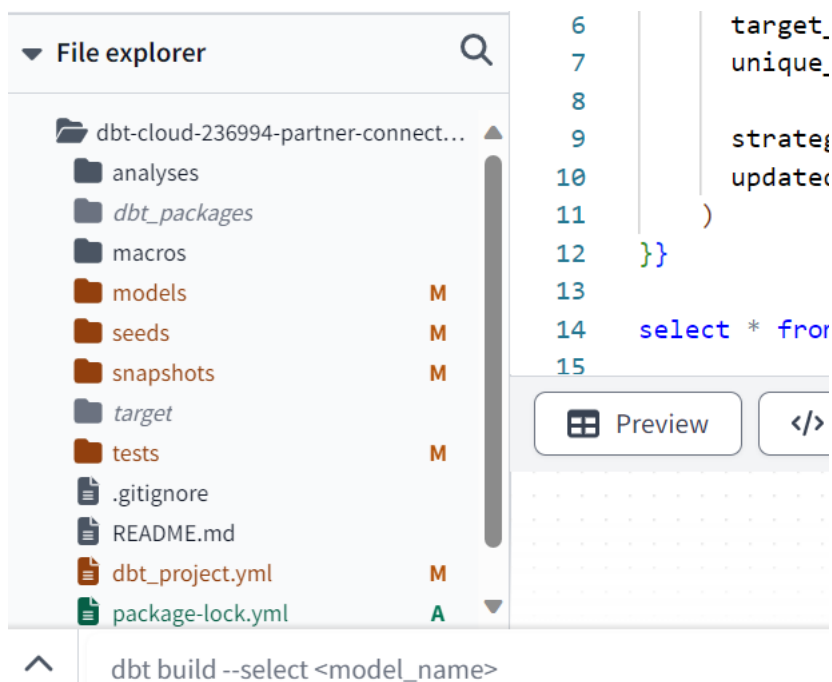1. Use case Overview
2. dbt-File Explorer
3. run First Model in dbt and Validate Output in Snowflake
4. Build dbt Staging Models- 1. Orders 2. Line Items with Transformations
5. Test the Output in Snowflake for Models- 1. Orders 2. Line Items

Use Case Overview:

- Raw Data: Source Data Set
- Transformations:
- Enriched
- TEST
- Documentation

```
 6         target_
 7         unique_
 8
 9         strateg
10         updated
11      )
12  }}
13
14  select * from
15
```

Preview   </>

dbt build --select <model_name>
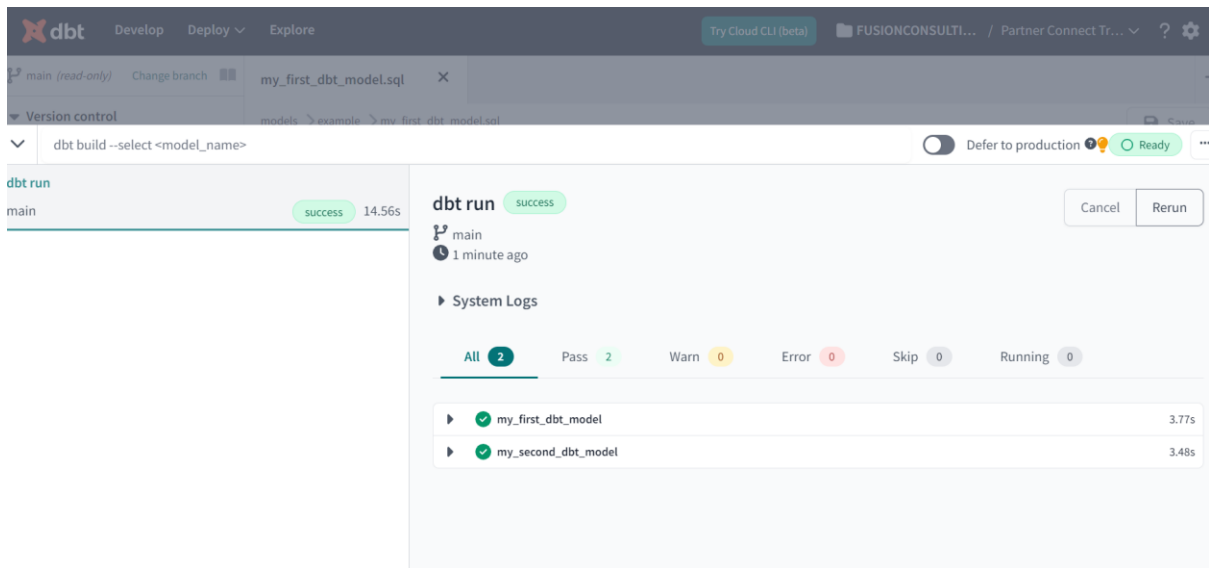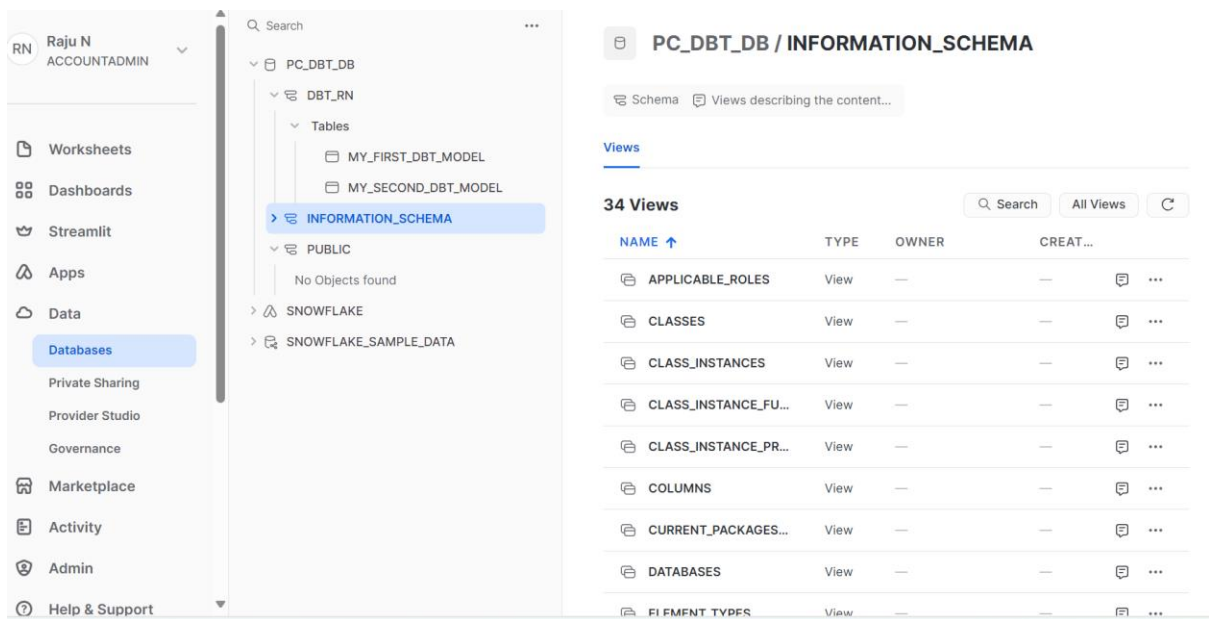
dbt- RUN First Model

- models > example > my_first_dbt_model.sql

- models > example > my_second_dbt_model.sql

dbt commands- to process the Two models

Type in `dbt run` into the command line at the bottom of the screen and hit `Enter` on your keyboard

USE CASE Development

dbt steps:

Step1: Configure dbt_project.yml

File name: dbt_project.yml

```yaml
name: 'snowflake_workshop'
version: '1.0.0'
config-version: 2

profile: 'default'

source-paths: ["models"]
analysis-paths: ["analysis"]
test-paths: ["tests"]
seed-paths: ["seeds"]
macro-paths: ["macros"]
snapshot-paths: ["snapshots"]

target-path: "target"
clean-targets:
    - "target"
    - "dbt_modules"

models:
  snowflake_workshop:
    staging:
      materialized: view
      snowflake_warehouse: pc_dbt_wh
    marts:
      materialized: table
      snowflake_warehouse: pc_dbt_wh_large
```
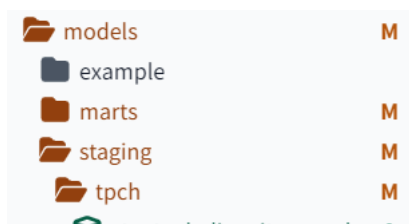
save the file using the save button in the upper right hand corner of the screen.

Create Folder

click the three dots that appear to the right of the folder name, then click `Create Folder`. We're going to add two new folders to the file path, typing `staging/tpch` into the file path. Make sure you're not including additional folder names and click `Create`.
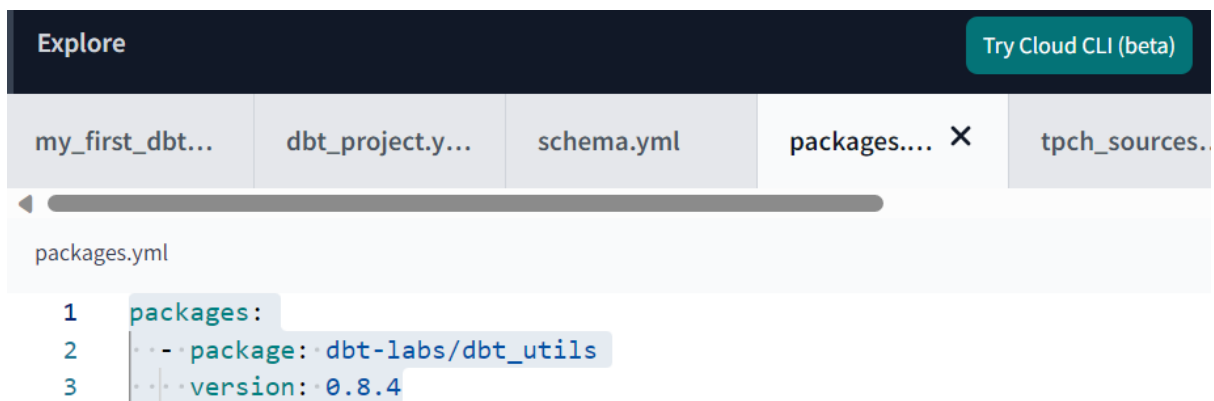
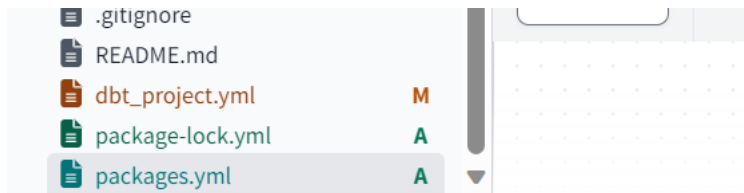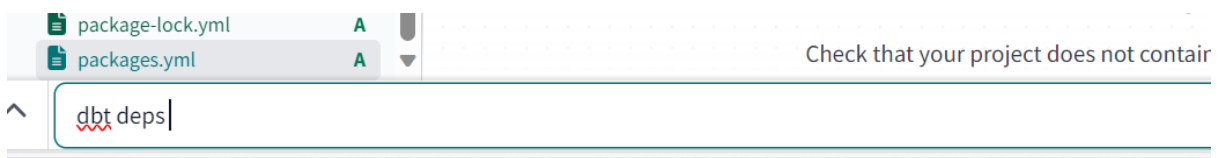## Create Package:

`packages.yml`.

```
packages:
  - package: dbt-labs/dbt_utils
    version: 0.8.4
```
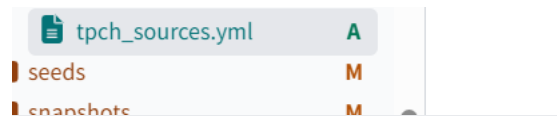

Dbt Commands:

Type in `dbt deps` to the command line, click `Enter`

you should see a success message there when it completes.

Create Sources

defined the database where the data is coming from (snowflake_sample_data), the schema (tpch_sf1)

dbt Source File - Logic

```yaml
version: 2
sources:
  - name: tpch
    description: source tpch data
    database: snowflake_sample_data
    schema: tpch_sf1
    tables:
      - name: orders
        description: main order tracking table

        columns:
          - name: o_orderkey
            description: SF*1,500,000 are sparsely populated
            tests:
              - unique
              - not_null

      - name: lineitem
        description: main lineitem table
        columns:
          - name: l_orderkey
            description: Foreign Key to O_ORDERKEY
            tests:
              - relationships:
                  to: source('tpch', 'orders')
                  field: o_orderkey
```

Create Staging Models-orders table

dbt Transformations - Logic for Orders Table

```
with source as (

    select * from {{ source('tpch', 'orders') }}

),

renamed as (

    select

        o_orderkey as order_key,
        o_custkey as customer_key,
        o_orderstatus as status_code,
        o_totalprice as total_price,
        o_orderdate as order_date,
        o_orderpriority as priority_code,
        o_clerk as clerk_name,
        o_shippriority as ship_priority,
        o_comment as comment

    from source

)

select * from renamed
```

Create Staging Models-Line Items table

models/staging/tpch/stg_tpch_line_items.sql

Transformations Logic Code-for Line Items

```
with source as (

    select * from {{ source('tpch', 'lineitem') }}

),

renamed as (

    select

        {{ dbt_utils.surrogate_key(
            ['l_orderkey',
            'l_linenumber']) }}
                as order_item_key,
        l_orderkey as order_key,
        l_partkey as part_key,
        l_suppkey as supplier_key,
        l_linenumber as line_number,
        l_quantity as quantity,
        l_extendedprice as extended_price,
        l_discount as discount_percentage,
        l_tax as tax_rate,
        l_returnflag as return_flag,
        l_linestatus as status_code,
        l_shipdate as ship_date,
        l_commitdate as commit_date,
        l_receiptdate as receipt_date,
        l_shipinstruct as ship_instructions,
        l_shipmode as ship_mode,
        l_comment as comment

    from source

)

select * from renamed
```

pass the dbt run command at the command line to run all of the models in our project,

staging models - dbt run

validate Transformation & Tables in Snowflake

PC_DBT_DB / DBT_RN / STG_TPCH_LINE_ITEMS

View | PC_DBT_ROLE | 1 hour ago

View Details | Columns | **Data Preview**

• COMPUTE_WH | Updated 3 minutes ago

	ORDER_ITEM_KEY	ORDER_KEY	PART_KEY	SUPPLIER_KEY	LINE_NUMBER
1	2cf9834dd78feb2271adebe399e5010e	5400001	30674	5681	1
2	c99f78c1f5e73a84b9459bdb020c9ab8	5400001	135464	5465	2
3	cf46dc560b07e86a125170081831b827	5400001	139144	6684	3
4	40841afe62d077e6c4223e768f770ceb	5400001	190382	5421	4
5	9c84ad642b125fa0741e092306421d8a	5400001	138279	793	5
6	36d3dd3ca266e6d78ab20c97b24949fd	5400001	49366	4375	6
7	7cf7e829b2fffb62c78f580a32cc6dff	5400001	94338	1866	7
8	fc8e070831767137103a7e9ffd79d483	5400002	132026	9566	1
9	68773191e7657be029bfa29bf68bf2c8	5400002	147891	2920	2
10	75805dde594544bd81e9b7ea95c0b3c7	5400002	190848	8406	3

TWO staging models built and we're able to start transforming our data to meet our modeling needs

Lab2- Final Output:

#1: StagingModel-Orders with Transformations

PC_DBT_DB / DBT_RN / STG_TPCH_ORDERS

View | PC_DBT_ROLE | 1 hour ago

View Details | Columns | **Data Preview**

● COMPUTE_WH Updated 1 minute ago

	ORDER_KEY	CUSTOMER_KEY	STATUS_CODE	TOTAL_PRICE	ORDER_DATE	PRIORITY_CODE
1	3600001	106660	O	130445.43	1995-10-25	3-MEDIUM
2	3600002	106873	O	226263.36	1997-06-26	1-URGENT
3	3600003	112288	F	110840.45	1993-06-19	3-MEDIUM
4	3600004	59149	O	45849.03	1997-12-24	5-LOW
5	3600005	42071	O	124317.01	1996-10-04	5-LOW
6	3600006	133325	F	188542.47	1993-06-21	2-HIGH
7	3600007	86974	O	42802.28	1997-01-30	2-HIGH
8	3600032	3928	F	242467.01	1995-02-17	1-URGENT
9	3600033	141910	O	237938.61	1998-03-09	2-HIGH
10	3600034	81127	O	205675.85	1997-04-11	1-URGENT

#2: Staging Model -Line Items with Transformations

PC_DBT_DB / DBT_RN / STG_TPCH_LINE_ITEMS

View | PC_DBT_ROLE | 1 hour ago

View Details | Columns | **Data Preview**

● COMPUTE_WH Updated 3 minutes ago

	ORDER_ITEM_KEY	ORDER_KEY	PART_KEY	SUPPLIER_KEY	LINE_NUMBER
1	2cf9834dd78feb2271adebe399e5010e	5400001	30674	5681	1
2	c99f78c1f5e73a84b9459bdb020c9ab8	5400001	135464	5465	2
3	cf46dc560b07e86a125170081831b827	5400001	139144	6684	3
4	40841afe62d077e6c4223e768f770ceb	5400001	190382	5421	4
5	9c84ad642b125fa0741e092306421d8a	5400001	138279	793	5
6	36d3dd3ca266e6d78ab20c97b24949fd	5400001	49366	4375	6
7	7cf7e829b2fffb62c78f580a32cc6dff	5400001	94338	1866	7
8	fc8e070831767137103a7e9ffd79d483	5400002	132026	9566	1
9	68773191e7657be029bfa29bf68bf2c8	5400002	147891	2920	2
10	75805dde594544bd81e9b7ea95c0b3c7	5400002	190848	8406	3