Things needed :

Anaconda to create env
Snowflake DW

Concepts :

Models
How to override the default materialized command
Type of materialization

Note - keep everthing in lower case

Command :

dbt run -> executes all the commands
dbt run -m dim_customer -> executes only dim_customer.sql model

Note : loaded_at_field should be added before tables

ref function is used to build the dependencies between the models
source function is used to build the dependencie of a models to its source

Why do we need source file and why cant we refer the raw data table directly

Source Freshness - How Fresh your data is? (Updated data or older data)

Command :

dbt source freshness

TEST function - Test function is assestion you make on your moedel

Two type of test

1 - Generic test
2 - Singular test

Different type of Generic test in dbt
1 - unique - It say that I want all the values in the particular column to be unique
2 - not_null -> Every valie in this column is not null
3 - accepted_values -> Each value in this column is a value from a given list
4 - relationship -> Every value in this column exists in the column of another table

- The Singular Test are written as select statement
- They run again your materialization

Generic test - Written in YML file / check common data quality issues and returns the number of records that do not meet your assestions. / They are designed to be reusable across different models and columns

Singular test - They are written in .sql file. These tests run against the entire model and allow for complex validation logic that cannot be captured by generic tests. They are run on entire model

singular test are written in test folder as sql file

Command :

dbt test -> Run tests
dbt test --select specific_model -> will run the tests defined for the specified model in your dbt project. It will run both generic and singular tests defined for the specified model
dbt build -> Runs all the models as well as all type of test
dbt test --select test
dbt test --select test_type : generic -> Run all the generic test
dbt test --select test_type : Singular -> Run all the Singular test

"dbt test" command will run both generic and singular test

dbt documentations:

Command :

dbt docs generate -> Generate json file
dbt docs serve --port 8000 -> opens web UI on specific port

Analysis :

Analysis Models files are seperate from the actual models in mart and staging folder
The Models that are written in the analysis are not built as table or view inside our DW

1. Analyses are .sql files that live in the analyses folder.
2. Analyses will not be run with dbt run like models. However, you can still compile these from Jinja-SQL to pure SQL using dbt compile. These will compile to the target folder.
3. Analyses are useful for training queries, one-off queries, and audits.

Command :

dbt show --select Model_name -> Runs the analysis models

Seeds :

Seeds are .csv files that are in seeds folder
These csv files are raw data files (usually in CSV or JSON format) that represent static data, such as lookup tables or reference data.
we put these files manually in the seeds folder
seeds should be used for data that does not change frequently (For ex : country code and pincodes)
These files are built as table in your DW
Seeds should not be the process for uploading data than changes frequently

Command :

dbt seed -> It will seed the data

Jinja :

it is used in dbt to write functional sql

{% ------ %} -> is used for statements (for ex - setting some variable, if statement)

{{ ------- }} -> used for expressions

{# ------ # } -> used for comments

Note :
Which command will only materialize dim_customers.sqland its downstream models?
dbt run --select dim_customers+

If any of the tests on sources fails, how will dbt Cloud handle the rest of the commands?
-- dbt will not execute any further commands

Staging :
Source file - source tables and you can run tests on the columns of the source data tables