1. Setup of the project

* Created a transform role and a user with that role in Snowflake for dbt to use
* Passwords to connections are stored in ~/.dbt/profiles.yml so that they don’t get pushed to git
* When setting up target schema in dbt we are indicating a place where all the output is created
* dbt power user – extension for VS code to work better with dbt

1. Data flow

* Raw layer – raw data that we have in Snowflake
* Staging Layer – basic checks, renaming/aliasing columns
* Core Layer – where big transformation/cleansing of data happens

1. Snowflake Pro Tips

* When clicking on a table we can copy it’s reference to paste in Worksheet SQL. “Place Name in SQL”

1. Terms

* model – an sql file that contains sql statements that perform some transformations
* materializations:
* view – select statement used for lightweight representation, don’t use it when you read from a view model several times (it’s a select statement so it needs to execute – sloooow)
* table – actual physical table that needs to be recreated every time model is used, it’s useful when you read repeatedly from the model that generates a table
* incremental (table appends) – records are appended, as data comes in it is appended to the table. It wouldn’t be useful if you wanted to edit historical records
* ephermal (CTE) – intermediate step model, for example when you want to simply add alias to colums. You just create a CTE model that renames the models and destroys materialization at the end
* refering to model – we use dbt template when refering to other models, eg.

SELECT \* FROM {{ ref('src\_listings') }}