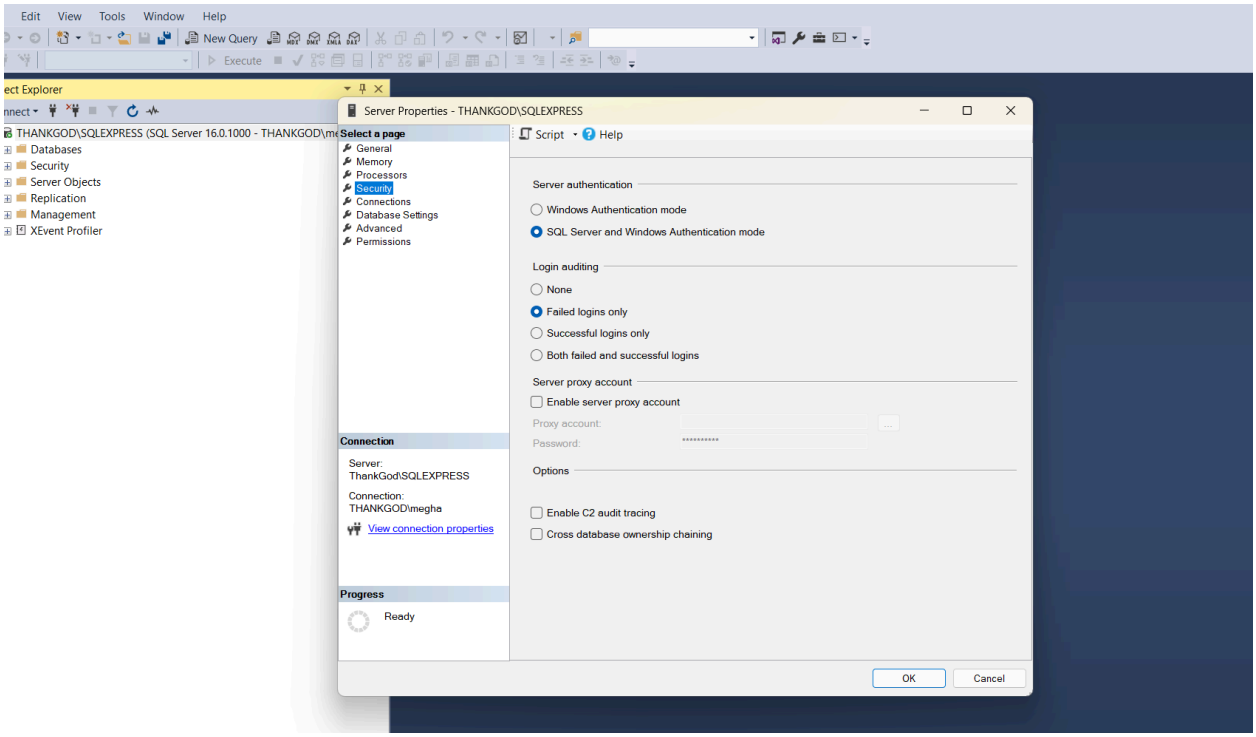
Requirements

- Works for python 3.12 and lower. It doesnt support for version 3.13 and above
- Make sure to configure the authentication in ssms as I am using Microsoft SQL



- Bypass the certificates if needed from the profiles.yml

Installation

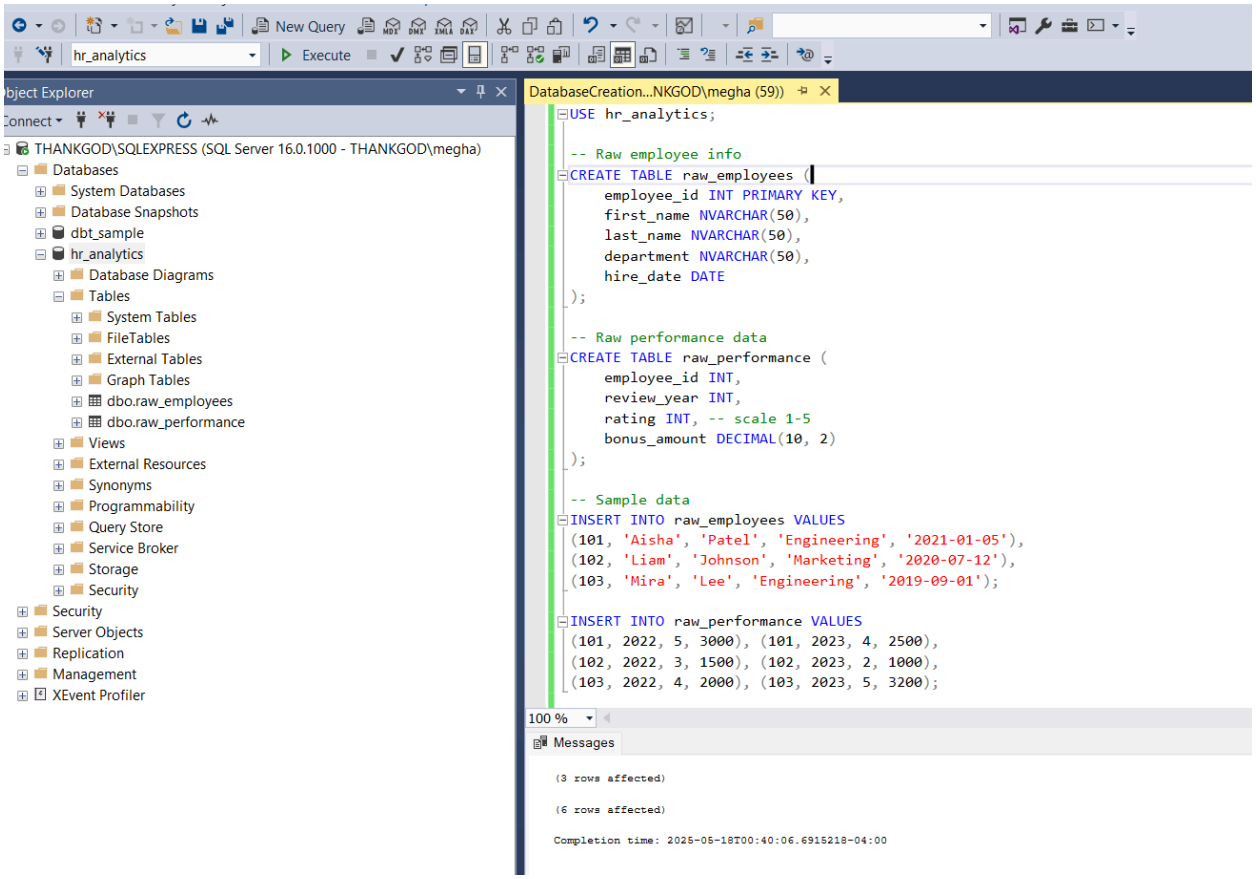
Make Sure Microsoft sql is installed and running

Please run this command in the terminal to install:

pip install dbt-core
pip install dbt-sqlserver

Step 1: Creating a database in SQL Studio

Created a database hr_analytics with two tables raw_employees and raw_performance



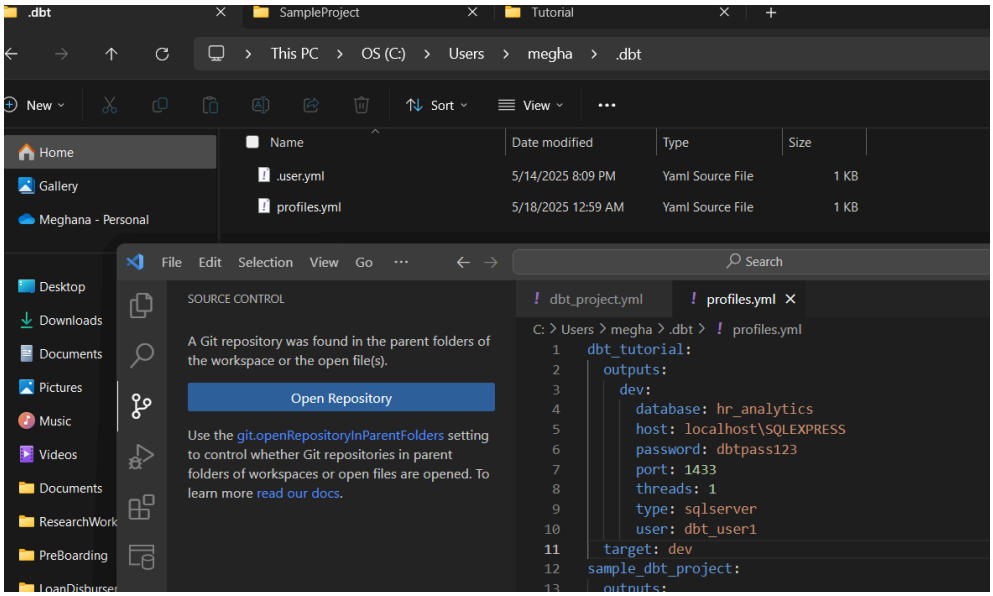
Step 2: Initialising a DBT project

Here I am using windows authentication. We can use SQL authentication as well, we have to create profiles for that

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial>dbt init dbt_tutorial
04:47:33  Running with dbt=1.8.9
04:47:33
Your new dbt project "dbt_tutorial" was created!

For more information on how to configure the profiles.yml file,
please consult the dbt documentation here:

  https://docs.getdbt.com/docs/configure-your-profile

One more thing:

Need help? Don't hesitate to reach out to us via GitHub issues or on Slack:

  https://community.getdbt.com/

Happy modeling!

04:47:33  Setting up your profile.
Which database would you like to use?
[1] fabric
[2] sqlserver
```

dbt init project_name

As I am using windows authentication, just press enter when asked about user and password details.

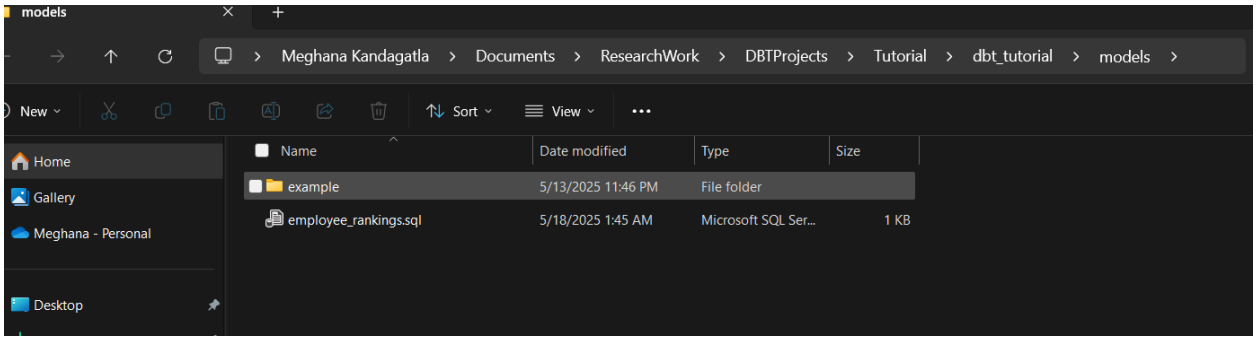


Editing the profiles.yml to windows authentication. Now navigate to the project folder and then run dbt debug to check if everything is working or not.

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial>cd dbt_tutorial

C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt debug
05:14:45  Running with dbt=1.8.9
05:14:45  dbt version: 1.8.9
05:14:45  python version: 3.12.10
05:14:45  python path: C:\Users\megha\AppData\Local\Programs\Python\Python312\python.exe
05:14:45  os info: Windows-11-10.0.26100-SP0
05:14:45  Using profiles dir at C:\Users\megha\.dbt
05:14:45  Using profiles.yml file at C:\Users\megha\.dbt\profiles.yml
05:14:45  Using dbt_project.yml file at C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial\dbt_project.yml
05:14:45  adapter type: sqlserver
05:14:45  adapter version: 1.8.7
05:14:46  Configuration:
05:14:46    profiles.yml file [OK found and valid]
05:14:46    dbt_project.yml file [OK found and valid]
05:14:46  Required dependencies:
05:14:46   - git [OK found]

05:14:46  Connection:
05:14:46    server: localhost\SQLEXPRESS
05:14:46    database: hr_analytics
05:14:46    schema: dbo
05:14:46    UID: None
05:14:46    client_id: None
05:14:46    authentication: Windows Login
05:14:46    encrypt: True
05:14:46    trust_cert: True
05:14:46    retries: 3
05:14:46    login_timeout: 0
05:14:46    query_timeout: 0
05:14:46    trace_flag: False
05:14:46    port: 1433
05:14:46  Registered adapter: sqlserver=1.8.7
05:14:46    Connection test: [OK connection ok]

05:14:46  All checks passed!

C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>
```

Step 3: Creating Models and testing them

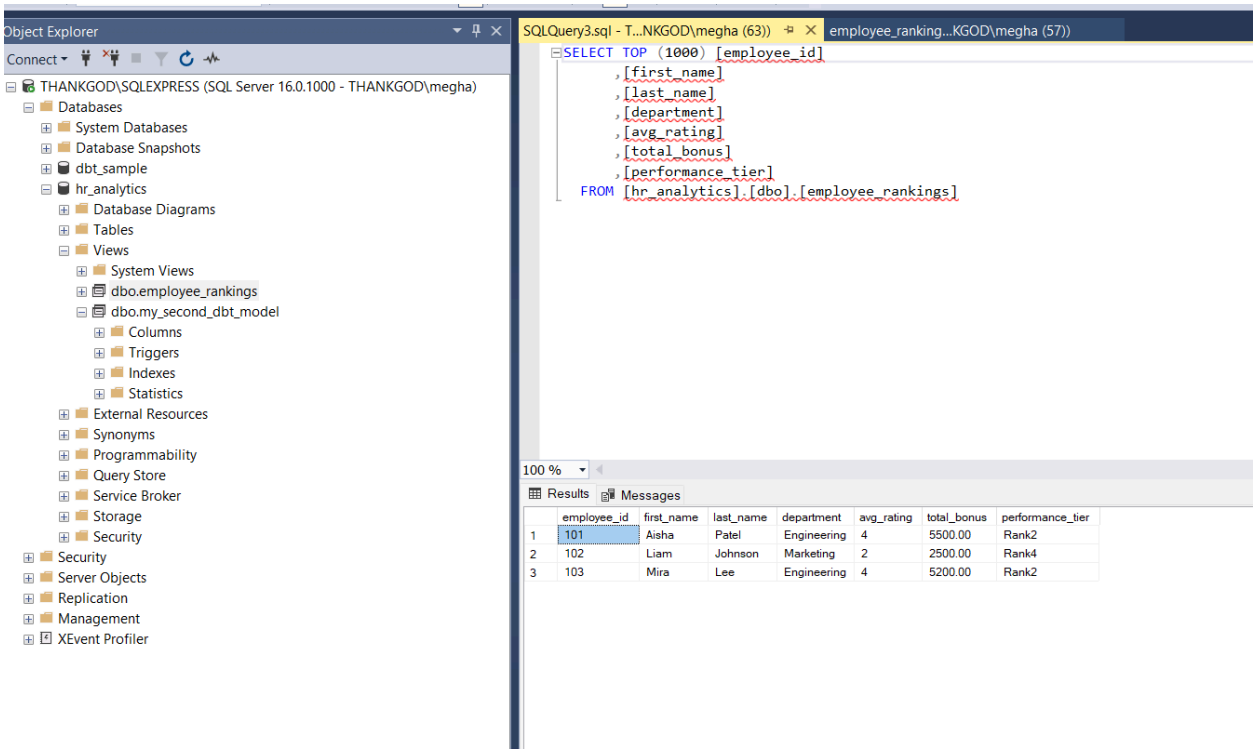
Creating a SQL script for ranking. I am placing that in the models folder. We can update the same in the schema.yml about the models.



Using dbt run. I am running all the scripts in the models folder.

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt run
05:53:07  Running with dbt=1.8.9
05:53:08  Registered adapter: sqlserver=1.8.7
05:53:08  Unable to do partial parsing because saved manifest not found. Starting full parse.
05:53:10  [WARNING]: Deprecated functionality
The `tests` config has been renamed to `data_tests`. Please see
https://docs.getdbt.com/docs/build/data-tests#new-data_tests-syntax for more
information.
05:53:10  Found 3 models, 7 data tests, 504 macros
05:53:10
05:53:11  Concurrency: 1 threads (target='dev')
05:53:11
05:53:11  1 of 3 START sql view model dbo.employee_rankings ............................. [RUN]
05:53:11  1 of 3 OK created sql view model dbo.employee_rankings ........................ [OK in 0.45s]
05:53:11  2 of 3 START sql table model dbo.my_first_dbt_model ........................... [RUN]
05:53:11  2 of 3 OK created sql table model dbo.my_first_dbt_model ...................... [OK in 0.23s]
05:53:11  3 of 3 START sql view model dbo.my_second_dbt_model ........................... [RUN]
05:53:11  3 of 3 OK created sql view model dbo.my_second_dbt_model ...................... [OK in 0.07s]
05:53:11
05:53:11  Finished running 2 view models, 1 table model in 0 hours 0 minutes and 0.97 seconds (0.97s).
05:53:12
05:53:12  Completed successfully
05:53:12
05:53:12  Done. PASS=3 WARN=0 ERROR=0 SKIP=0 TOTAL=3

C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>
```

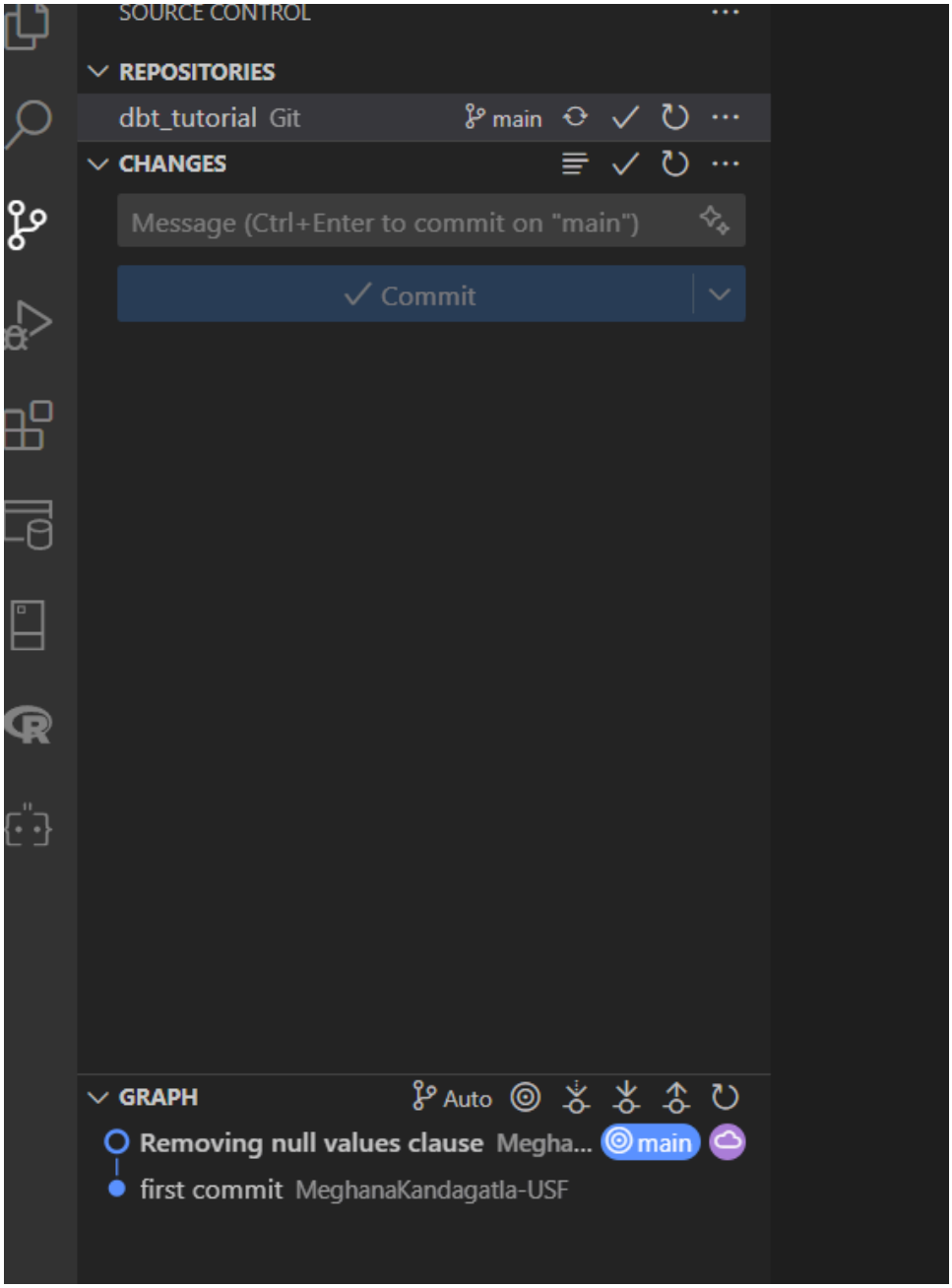
Now, I am checking ssms to check if the new table is created. It's saved as view.



Now testing the models using dbt run

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt test
05:58:43  Running with dbt=1.8.9
05:58:43  Registered adapter: sqlserver=1.8.7
05:58:44  Found 3 models, 7 data tests, 504 macros
05:58:44
05:58:44  Concurrency: 1 threads (target='dev')
05:58:44
05:58:44  1 of 7 START test accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [RUN]
05:58:44  1 of 7 PASS accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [PASS in 0.12s]
05:58:44  2 of 7 START test not_null_employee_rankings_employee_id ..................... [RUN]
05:58:44  2 of 7 PASS not_null_employee_rankings_employee_id ........................... [PASS in 0.04s]
05:58:44  3 of 7 START test not_null_my_first_dbt_model_id ............................. [RUN]
05:58:44  3 of 7 FAIL 1 not_null_my_first_dbt_model_id ................................. [FAIL 1 in 0.07s]
05:58:44  4 of 7 START test not_null_my_second_dbt_model_id ............................ [RUN]
05:58:45  4 of 7 PASS not_null_my_second_dbt_model_id .................................. [PASS in 0.03s]
05:58:45  5 of 7 START test unique_employee_rankings_employee_id ....................... [RUN]
05:58:45  5 of 7 PASS unique_employee_rankings_employee_id ............................. [PASS in 0.04s]
05:58:45  6 of 7 START test unique_my_first_dbt_model_id ............................... [RUN]
05:58:45  6 of 7 PASS unique_my_first_dbt_model_id ..................................... [PASS in 0.04s]
05:58:45  7 of 7 START test unique_my_second_dbt_model_id .............................. [RUN]
05:58:45  7 of 7 PASS unique_my_second_dbt_model_id .................................... [PASS in 0.03s]
05:58:45
05:58:45  Finished running 7 data tests in 0 hours 0 minutes and 0.57 seconds (0.57s).
05:58:45
05:58:45  Completed with 1 error and 0 warnings:
05:58:45
05:58:45  Failure in test not_null_my_first_dbt_model_id (models\example\schema.yml)
05:58:45    Got 1 result, configured to fail if != 0
05:58:45
05:58:45    compiled code at target\compiled\dbt_tutorial\models\example\schema.yml\not_null_my_first_dbt_model_id.sql
05:58:45
05:58:45  Done. PASS=6 WARN=0 ERROR=1 SKIP=0 TOTAL=7
```

As in the my_second_dbt_model, we included null as well. It’s throwing errors. Before correcting this error. I am creating a git repo.



Committed the new change I removed the null values line from the second_model. Now running the models and again using the dbt run command.

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt test
06:08:48  Running with dbt=1.8.9
06:08:49  Registered adapter: sqlserver=1.8.7
06:08:50  Found 3 models, 7 data tests, 504 macros
06:08:50
06:08:50  Concurrency: 1 threads (target='dev')
06:08:50
06:08:50  1 of 7 START test accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [RUN]
06:08:50  1 of 7 PASS accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [PASS in 0.10s]
06:08:50  2 of 7 START test not_null_employee_rankings_employee_id ..................... [RUN]
06:08:50  2 of 7 PASS not_null_employee_rankings_employee_id ........................... [PASS in 0.04s]
06:08:50  3 of 7 START test not_null_my_first_dbt_model_id ............................. [RUN]
06:08:50  3 of 7 PASS not_null_my_first_dbt_model_id ................................... [PASS in 0.02s]
06:08:50  4 of 7 START test not_null_my_second_dbt_model_id ............................ [RUN]
06:08:50  4 of 7 PASS not_null_my_second_dbt_model_id .................................. [PASS in 0.03s]
06:08:50  5 of 7 START test unique_employee_rankings_employee_id ....................... [RUN]
06:08:50  5 of 7 PASS unique_employee_rankings_employee_id ............................. [PASS in 0.04s]
06:08:50  6 of 7 START test unique_my_first_dbt_model_id ............................... [RUN]
06:08:50  6 of 7 PASS unique_my_first_dbt_model_id ..................................... [PASS in 0.04s]
06:08:50  7 of 7 START test unique_my_second_dbt_model_id ............................. [RUN]
06:08:50  7 of 7 PASS unique_my_second_dbt_model_id .................................... [PASS in 0.03s]
06:08:50
06:08:50  Finished running 7 data tests in 0 hours 0 minutes and 0.44 seconds (0.44s).
06:08:50
06:08:50  Completed successfully
06:08:50
06:08:50  Done. PASS=7 WARN=0 ERROR=0 SKIP=0 TOTAL=7

C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>
```

Step 4: Generating the documents

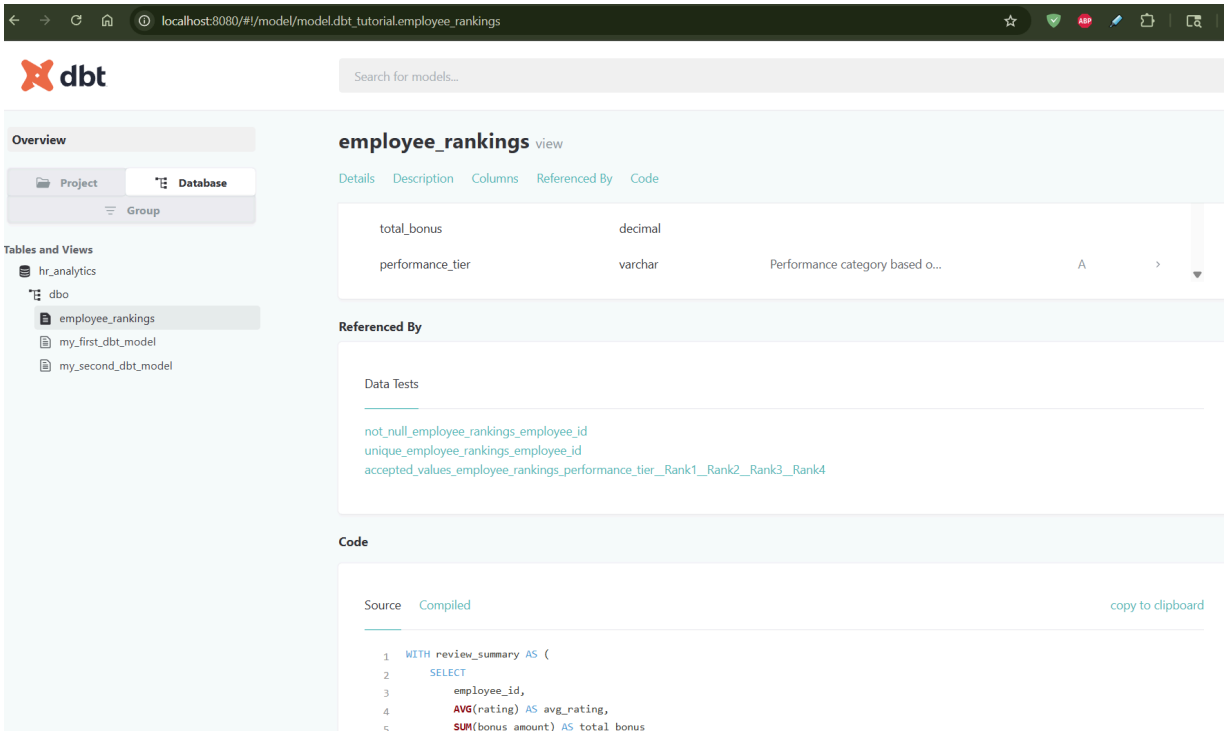
Using dbt commands we can create.

dbt docs generate
dbt docs serve

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt docs generate
06:17:11  Running with dbt=1.8.9
06:17:12  Registered adapter: sqlserver=1.8.7
06:17:12  Found 3 models, 7 data tests, 504 macros
06:17:12
06:17:13  Concurrency: 1 threads (target='dev')
06:17:13
06:17:13  Building catalog
06:17:13  Catalog written to C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial\target\catalog.json

C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt docs serve
06:17:30  Running with dbt=1.8.9
Serving docs at 8080
To access from your browser, navigate to: http://localhost:8080



Press Ctrl+C to exit.
127.0.0.1 - - [18/May/2025 02:17:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/May/2025 02:17:32] "GET /manifest.json?cb=1747549052393 HTTP/1.1" 200 -
127.0.0.1 - - [18/May/2025 02:17:32] "GET /catalog.json?cb=1747549052393 HTTP/1.1" 200 -
```

see all of the models that are used to build, or are built from, the model you're exploring.

This generated me



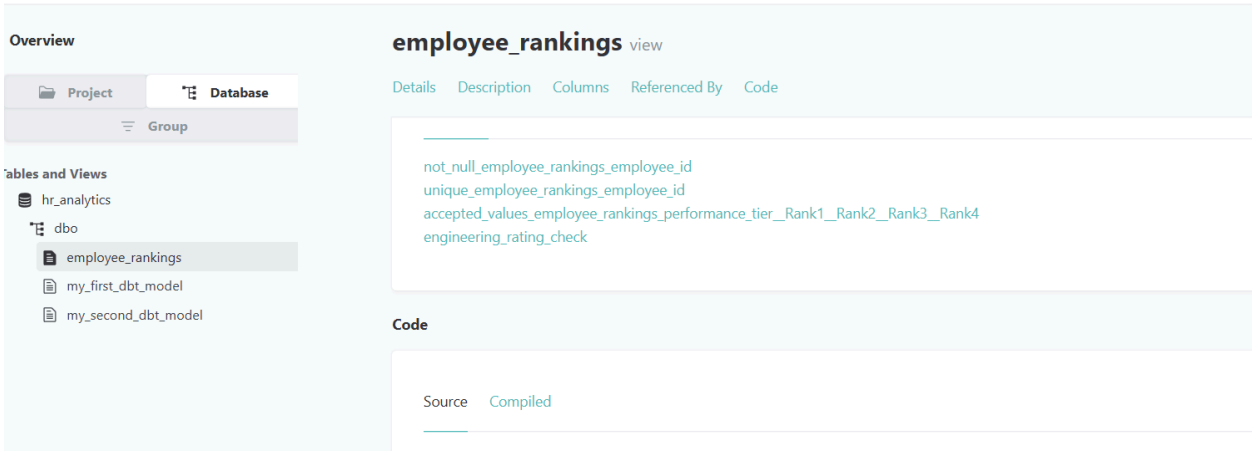
Here I can navigate through the database and the tables. All the data tests can be reused again using the commands in the terminal.

Step 5: Creating a custom test

I am creating a custom test that checks if the department is engineering or not.
Now again running dbt run and dbt test.

```
C:\Users\megha\Documents\ResearchWork\DBTProjects\Tutorial\dbt_tutorial>dbt test
06:28:22  Running with dbt=1.8.9
06:28:23  Registered adapter: sqlserver=1.8.7
06:28:23  Found 3 models, 8 data tests, 504 macros
06:28:23
06:28:24  Concurrency: 1 threads (target='dev')
06:28:24
06:28:24  1 of 8 START test accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [RUN]
06:28:24  1 of 8 PASS accepted_values_employee_rankings_performance_tier__Rank1__Rank2__Rank3__Rank4  [PASS in 0.12s]
06:28:24  2 of 8 START test engineering_rating_check ..................................... [RUN]
06:28:24  2 of 8 PASS engineering_rating_check ........................................... [PASS in 0.03s]
06:28:24  3 of 8 START test not_null_employee_rankings_employee_id ....................... [RUN]
06:28:24  3 of 8 PASS not_null_employee_rankings_employee_id ............................. [PASS in 0.04s]
```

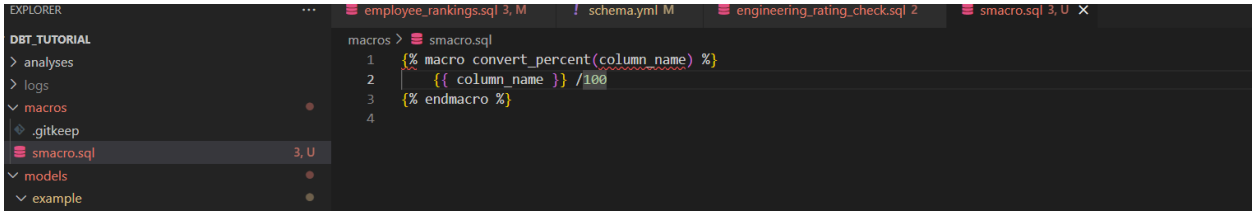
We can see engineering_rating_check being performed here.
We can see the same from the dbt docs as well. Adding the information about this in the schema.yml as well.



Step 6: Macros

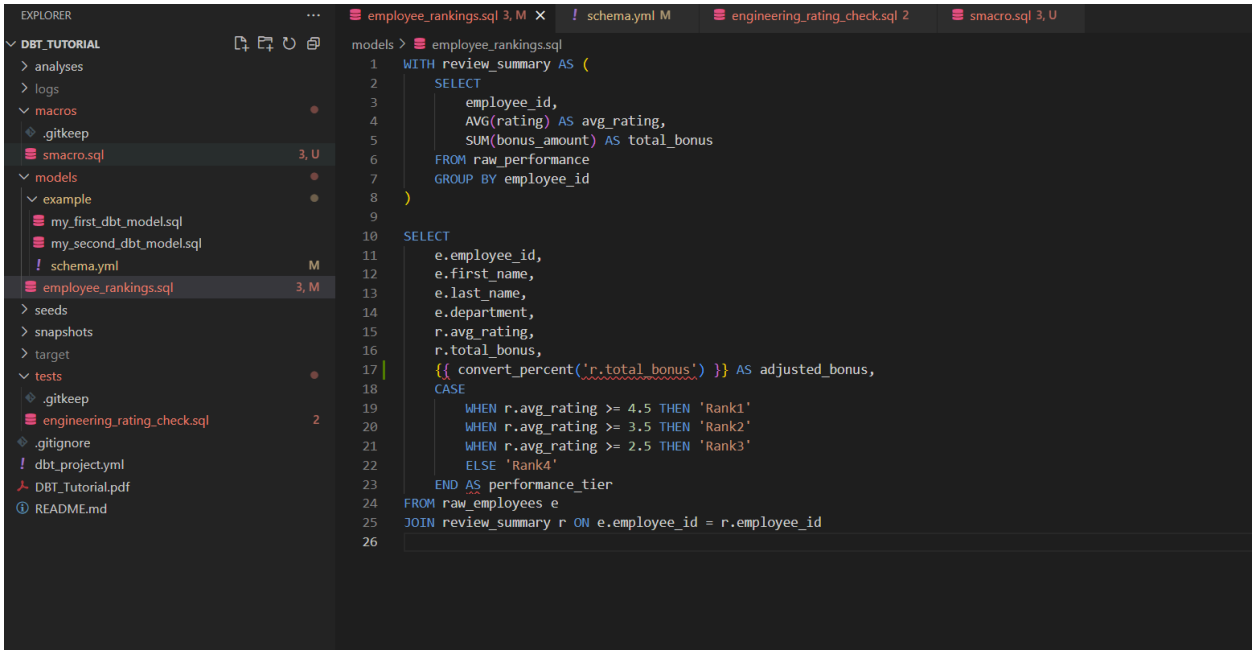
Macro is a reusable block of logic.

Here I am adding a simple macro that convert into percent to the given column.



As we can see all of these macros are kept in macros folder

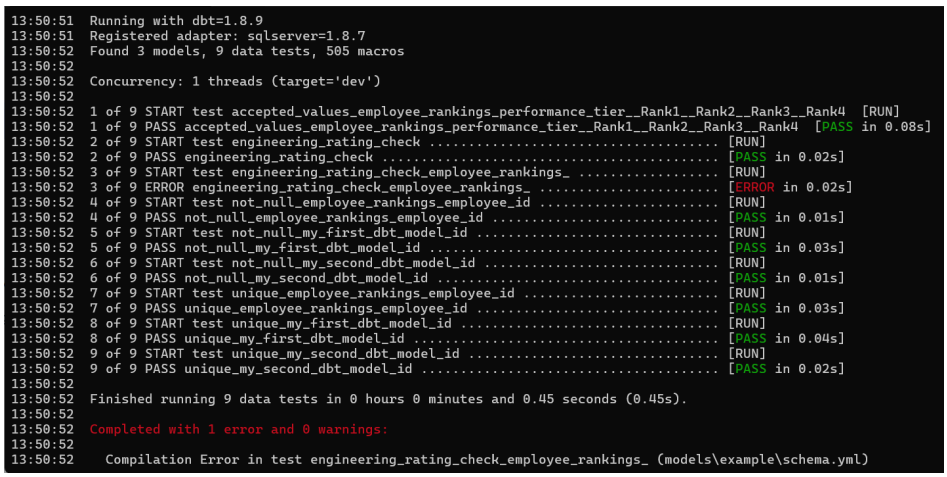
Now let’s look into how we are going to use this macro in the models.



Here we can see that we have replaced the column name as r.total_bonus in the macro and saved it as adjusted_bonus

Added the same update in the schema.yml as well

Let's run and check



There is a failed test because, custom test is failing as there is no such data that is satisfying the test. Test looks for department: engineering and avg_rating < 3.5

SQLQuery1.sql - T...NKGOD\megha (58))

```
SELECT TOP (1000) [employee_id]
      ,[first_name]
      ,[last_name]
      ,[department]
      ,[avg_rating]
      ,[total_bonus]
      ,[adjusted_bonus]
      ,[performance_tier]
  FROM [hr_analytics].[dbo].[employee_rankings]
```

100 %

Results | Messages

	employee_id	first_name	last_name	department	avg_rating	total_bonus	adjusted_bonus	performance_tier
1	101	Aisha	Patel	Engineering	4	5500.00	55.000000	Rank2
2	102	Liam	Johnson	Marketing	2	2500.00	25.000000	Rank4
3	103	Mira	Lee	Engineering	4	5200.00	52.000000	Rank2

Here we can see that adjusted_bonus is created.