CS 329E Project 8, due Thursday, 11/13.

Objectives

- Orchestrate the staging layer of the lakehouse with dbt

In-Scope

- CTAS statements from `project5-stg-layer.ipynb`

Out-of-Scope

- SELECT statements from `project5-stg-layer.ipynb`

Work Items

- Create a new dbt project
- Create the configuration files
- Create the folder structure
- Create the macros
- Create and run the models
- Create and run the data validation tests
- Generate the docs and lineage graph

Code Samples

See snippets repo for code samples

Implementation Details

- Go to the dbt folder that you created during the setup and run the `dbt init` command to create a new dbt project. Name your dbt project `[your_domain]` and name your dbt dataset `dbt_[your_domain]_stg`.

- Open your profile file `profiles.yml` (located in your home directory) and change the region of the dataset from us to us-central1.

- Open the `dbt_profile.yml` file and change the model section of the file to look like this:

```
models:
 air_travel:
   # Config indicated by + and applies to all files under models/example/
   stg:
     +materialized: table
     +schema: dbt_air_travel_stg
   int:
```

```
    +materialized: table
    +schema: dbt_air_travel_int
  mrt:
    +materialized: table
    +schema: dbt_air_travel_mrt
```

For the schema values, make sure you replace `air_travel` with your domain.

- Open the models folder and delete the `examples` subfolder. Create the following three folders in place of the `examples` folder:
  - `stg`
  - `int`
  - `mrt`

- Create a file `generate_schema_name.sql` in the macros folder of your dbt project and copy the contents from here into it. This macro is needed to overwrite the default naming of datasets which would yield `dbt_air_travel_stg_dbt_air_travel_stg` for the dataset name.

- Open the stg folder and create a file named `schema.yml` in it. This file is where we specify the source for each staging table. Follow this example to specify the `sources` in the file. What this will do is register the raw tables with dbt and allow us to refer to the raw tables with dbt's `source()` function. Note: we are not going to create the raw dataset with dbt, we will use the existing raw dataset. You do not need to add the `models` section yet.

- Create your dbt model files for the staging layer. Place the files in the stg folder and make sure you source the raw table with the `source()` function.

- Run the dbt run command to compile your models. You should do this iteratively so that you can easily identify the source of the errors.

- Create a second macro in the dbt macros folder to validate the row count of the staging models. Name the file `check_row_count_equals.sql` and copy its contents from here. This function compares the row counts of the staging and raw tables and raises an error if they don't match.

- Open the `schema.yml` file that you previously created. Add a data validation test for each model. The test should call the `check_row_count_equals` macro to compare the row counts. Follow this example to see how to format your tests.

- Run the dbt test command to run the tests. Make sure that all of your tests pass.

- Generate the documentation for your project and verify that the lineage graph captures all your staging models. Refer to our setup guide if you don't remember how to do this step. Take a screenshot of your lineage graph and place it in your dbt project folder under a new subfolder named lineage. Name the screenshot file `staging-layer.png`.

- Make a new project8 folder in your local git repo. Copy your entire dbt project folder into it (e.g. `[your_domain]`). Edit the `.gitignore` file in your dbt project folder to not exclude the `target/` and `log/` folders from git. Commit and push your changes to GitHub.

- Once you have pushed your changes, create the usual submission.json file and upload it to Canvas by the deadline. Only one person per group needs to do this step.

CS 329E Project 8 Rubric
Due Date: 11/13/25

dbt project folder is thorough and meets all requirements **-3** for each table not found in the dbt staging dataset **-3** for each table found in the dbt staging dataset, but not generated with `dbt run` **-3** for each empty table in the dbt stg dataset **-2** for each staging model not written with `source()` **-2** for each missing dbt test in `schema.yml` **-2** for each failed dbt test **-3** for not following our dataset naming convention **-3** for one or both macro files missing **-5** for `target/` or `logs/` folders not found in repo **-90** missing dbt project folder under `project8`	90
dbt project folder contains a lineage subfolder with a screenshot of the model dependency graph **-1** for each staging model or dependency missing from graph **-2** model names are not legible in the screenshot **-10** lineage subfolder not found under dbt project folder	10

`submission.json` submitted into Canvas. Your project **will not** be graded without this submission. The file should have the following schema:
 	
`{`
`  "commit-id": "your most recent commit ID from Github",`
`  "project-id": "your project ID from GCP"`
`}`
 	
Example:
 	
`{`
`  "commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9",`
`  "project-id": "some-project-id"`
`}`
	Required

Total Credit:	**100**