CS 378 Project 9, due Thursday, 11/20.

Objectives

- Part 1 of orchestrating the intermediate layer with dbt

In-Scope

- CTAS and DML statements from `project6-int-layer.ipynb`

Out-of-Scope

- Data validation tests for the int layer. This will be done in Part 2.

Work Items

- Translate your CTAS and DML statements to dbt models
- Run all the models to create the int tables
- Generate the docs and lineage graphs

Code Samples

See snippets repo for code samples

Implementation Details

- Profile your CTAS and DML statements from `project6-int-layer.ipynb`. If you have any final tables which were derived by multiple statements, you should configure a tmp area in dbt to store intermediate results. This will allow us to avoid cluttering the int area. To set up the tmp space, make the following changes to your `dbt_profile.yml` file:

```
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
   tmp:
     +materialized: table
     +schema: dbt_air_travel_tmp
```

For the schema value, make sure you replace `air_travel` with your domain. If you don't have any multi-step transformation, you don't need to add a tmp space.

- If you are adding a tmp space, you also need to open the models folder and create a `tmp` subfolder in it. You should have the following subfolders under models:
  - `stg`
  - `int`
  - `mrt`
  - `tmp`

- Create your dbt model files for the int layer. Place the files in the `int` or `tmp` folders, depending on their function. Be sure to use the `ref()` function in the from clause. You may only hardcode the path if you are referencing the text embedding model (text-embedding-005).

- 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.

- To avoid timing out during long-running builds, you should increase the `job_execution_timeout_seconds` parameter in `profiles.yml`. I've increased mine to 1 hour (3600).

- To avoid recompiling long-running models, you should make use of tags. Here is an example of a tag:

  ```
  {{ config(tags=["llm"]) }}
  ```

  I have added this tag to every model that calls the LLM. Then, once I've compiled the model successfully, I only recompile the models that don't have this tag:

  ```
  dbt run --exclude tag:llm
  ```

  This is to avoid wasting time and resources on the long-running models.

- 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 one screenshot of your int layer graph and another screenshot of your tmp layer, if applicable. Take a third screenshot of your entire dbt project (stg, tmp, int). Place the screenshots in your `dbt_project_folder/lineage`.

- Make a new project9 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 378 Project 9 Rubric
Due Date: 11/20/25

dbt project folder is thorough and meets all requirements **-3** for each final table in the int layer not found in the dbt int dataset **-3** for each final table in the int layer not generated with `dbt run` **-3** for each empty int table **-2** for each int or tmp model with hardcoded paths, (i.e. not using the `ref()` function) **-3** for not following our dataset naming convention **-5** for `target/` or `logs/` folders not found in repo **-5** for substantial deviations found between dbt generated tables and equivalent colab ones **-90** missing dbt project folder under `project9`	90
dbt project folder contains a lineage subfolder with a screenshot of the model dependency graph **-1** for each int model missing from the graph **-1** for each tmp model missing from the graph **-2** model names are not legible in the screenshots **-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**