CS 378 Project 10, due Thursday, 12/04.

Objectives

- Part 2 of orchestrating the intermediate layer with dbt
- Orchestrating the mart layer with dbt

In-Scope

- Logical key constraints from `project6-int-layer.ipynb`
- Data mart CTAS statements from `project7-mrt-layer.ipynb`

Out-of-Scope

- ALTER TABLE statements

Work Items

- Create primary key and foreign key tests on the int models
- Run the tests
- Revise your int models as needed
- Convert data marts into dbt models
- Create the mart models
- Generate dbt docs and lineage graphs

Code Samples

See snippets repo for code samples

Implementation Details

- Open the int folder and create a file named `schema.yml` in it. This file is where we specify the primary key and foreign key constraints for each intermediate model. Read the official docs and follow this example to specify the constraints in the file.

- If you have a composite primary key, you'll also need to create the macro `unique_combination_of_columns.sql`. Copy the macro code from here into your macros folder and create a `package.yml` like this.

- Run the dbt test command to compile and run your tests. You should do this iteratively so that you can easily spot the source of the errors. If a test fails due to a non-syntax issue, it is likely due to a duplicate primary key or orphan foreign key.

- To handle failed tests, review the business logic and create a patch. Try to think of a heuristic that can scale when you implement the revised logic. Do not hardcode values. See for example, this and this; the former shows how to use `rank()` and `partition by` to dedup while the latter shows how to use the LLM for fuzzy entity matching.

- Create your dbt model files for the mart (mrt) layer by transforming the CTAS statements. Place the files in the `models/mrt` folder. All data marts should source from one or more int models. They can also source from a tmp model created on top of the int layer if the int model is missing a feature. Whether sourcing from int or tmp, it is important for the mart to reference its source models with the `ref()` function.

- Generate the documentation for your project and verify that the lineage graph captures all your models. Look out for disconnected nodes in the tmp, int, or mrt layers, there shouldn't be any. Take one or more screenshots of your mart layer graph and another screenshot of your tmp layer, if it has changed in this project. Take a third screenshot of your entire dbt project (stg, tmp, int, mrt). Place the screenshots in your `dbt_project_folder/lineage`.

- Make a new project10 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 10 Rubric
Due Date: 12/04/25

dbt project folder is thorough and meets all requirements **-3** for each missing or incorrectly specified primary key or foreign key test in `models/int/schema.yml` **-3** for each primary or foreign key test that failed or was not run with `dbt test` **-3** for each missing or empty data mrt in dbt mrt dataset **-3** for each mrt table not created by `dbt run` **-3** for each mrt model that substantially deviates from its previous version created from colab **-2** for each mrt model that is using 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 the dbt-generated tables and the equivalent colab-generated tables **-90** missing dbt project folder under `project10`	90
dbt project folder contains a lineage subfolder with a screenshot of the model dependency graph **-1** for each mrt model missing from lineage graph **-1** for each disconnected model in the lineage graph (indicates `ref()` was not used) **-2** model names are not legible in the screenshot or layer-specific screenshot not provided **-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**