CS 333E Project 2, due Thursday, **01/29**.

Objective

- Build the first half of the raw layer of the lakehouse

In-scope

- The structured data sources (csv, tsv, json)

Out-of-Scope

- The unstructured data sources (.txt, .pdf, .jpg, etc.)

Work Items

- Convert data files to parquet format
- Create the Iceberg tables in BQ
- Load the tables with the parquet files
- Extend the tables with `_data_source` and `_load_time` fields
- Populate the `_data_source` and `_load_time`
- Check your work against our validation criteria

Code Samples

See snippets repo for code samples

Implementation Details

- Create a new folder in your repo and name it `project2`. Store all your artifacts for this project in the `project2` folder.
- Go to Colab Enterprise in the console and enable the required APIs
- Create a Colab notebook named `project2-data-ingestion.ipynb`.
- Annotate your notebook with section headers and short Markdown comments throughout your work.
- Import the provided notebook into your Colab and follow the same steps.
- When creating the top-level `[your-domain]-lakehouse` folder in GCS, be sure to place it in the root of your bucket.
- Make sure that the data for each data source is in its own subfolder under the `lakehouse` folder.
- When creating the dataset in BQ, follow the naming: `[your-domain]_raw` (e.g. air_travel_raw, etc.).
- After loading your tables, run some basic sanity checks by retrieving a few rows from them and ensuring that the data looks ok.
- Update your data dictionary and ERD to reflect the tables in the raw layer. The dictionary should include all the fields for every table that you have loaded into the lakehouse. The ERD should contain the most important fields for every table or entity in your lakehouse.

This data types and relationships. You do not need to list the `_data_source` and `_load_time` fields in the ERD or any other field that is unimportant. Think of the ERD as a bird's eye view of the raw layer and the data dictionary as a granular view of the raw layer.

- Review the validation criteria and cross-reference them with your tables. Determine which ones you have satisfied to-date and which ones are missing. Create a Markdown file called `criteria-analysis.md` with the results from your analysis. Please include a specific example for each criteria you are able to satisfy. If you are missing a criteria that is required for project 2 (as indicated by the Required By field), try to resolve it ASAP. If you are not able to do so on your own, make note of it in your analysis and discuss it with the Prof. To be clear, you are expected to meet the first 4 validation criteria, the ones that are marked as required by all projects and project 2.
- Commit and publish your work to your GitHub repo. Remember that all artifacts need to go into a `project2` folder. This includes your notebook, data dictionary, ERD, and criteria analysis.
- Create a submission.json file and upload it to Canvas by the deadline. Only one person per group needs to do this step.

Grading Rubric

Due Date: 01/29/26

`project2-data-ingestion.ipynb` correctly written and indicates successful execution **-10** for each missing or empty table in BQ **-7** at least one data file (parquet) and Iceberg file (metadata) are not located in the right path **-7** at least one table missing or has empty data_source or load_time fields **-10** ERD and/or data dictionary missing or out-of-date with tables in the raw layer **-90** missing notebook **-80** didn't run the code	90
`criteria-analysis.md` is thorough and has all information. All 10 criteria, including the 6 anomaly types should be included in the analysis. Include placeholders for any criteria that you are not able to satisfy. **-5** at least one of the required criteria for this project is not met (criteria 1-4) **-2** for each required criteria whose explanation is not thorough **-2** for crucial misunderstanding of a required criteria **-10** missing file	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**