CS 333E Project 5, due Thursday, **02/26**.

Objectives

- Create the staging layer of the lakehouse
- Fix anomaly types 4, 6, and 7 (link)

In-Scope

- Field-level transformations

Out-of-Scope

- Entity-level transformations

Work Items

- For every table in the raw layer, apply the following transformations to all of its fields:
    - For criteria 4, review the BigQuery data types and choose the best fit type
    - For criteria 6, replace `\n`, `''`, etc. with `null`
    - For criteria 7, split the multi-valued field into atomic values
    - Rename non-descriptive fields
    - Drop fields that contain low-quality data or are not relevant to your subject area
    - Standardize low-cardinality categorical fields
- Materialize the output from these transformations into new tables in the staging layer

Code Samples

See snippets repo for code samples

Implementation Details

- Create a new folder in your repo and name it `project5`. Store all your artifacts for this project in the `project5` folder.
- Create a Colab notebook named `project5-stg-layer.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.
- Create the resulting tables into a new staging dataset `[your-domain]_stg`.
- Be sure to **not** mutate the raw tables when applying your transformations. We want the raw tables to represent the source data
- Be sure to carry forward all the raw tables into the staging layer, even those which do not need to be transformed. The resulting staging dataset should contain the complete collection of tables.
- Create an ERD for your staging layer. It should represent the most important fields in each table or entity in the staging dataset, including the data type and keys. Also, ensure that you draw the relationships between entities, using our convention of solid line edges

for relationships that hold and dotted line relationships for those which do not yet hold. You do not need to list the `_data_source` and `_load_time` fields in each table or any other fields that are less important. Think of the ERD as a bird's eye view of your data model.

- Commit and publish your work to your GitHub repo. Remember that all artifacts need to go into a `project5` folder. This includes your notebook, data dictionary, and ERD.
- 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: 02/22/26

`project5-stg-layer.ipynb` is thorough and meets all requirements **-10** incorrectly used SQL commands **-10** incorrect data type conversion (anomaly 4) **-10** incorrect replacement of empty strings or other characters (anomaly 6) **-10** incorrect splitting of concatenated fields into separate fields (anomaly 7) **-5** at least one staging table contains non-descriptive column names **-7** at least one staging table contains inconsistent categorical data which could have been standardized with a simple case statement **-5** notebook lacks Markdown annotations or is hard to follow **-10** did not follow naming convention for BQ dataset, tables or columns **-10** for each missing staging table in the BQ dataset **-60** did not create the staging tables in BQ **-80** missing file	80
`[your-domain]-erd-v5.pdf` accurately depicts the staging layer schema and logical relationships between tables **-2** for file named incorrectly **-3** for each missing staging table **-3** for each missing or incorrect relationship **-10** ERD not aligned to staging tables **-20** missing file	20

`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**