CS 329E Project 2, due Thursday, 02/08.

Recall our data acceptance criteria from Project 1, copied below for convenience. In this project, we assume that your raw data conforms to criteria #4 and #6. Our focus will be to construct a subset of the staging tables by splitting fields into their individual components and splitting tables into their logical entities.

Criteria	Description	Min No.	Examples
1	Dataset must have multiple unique entities. These are logical entities as opposed to how the raw tables are layed out.	5	Air Carrier, Airport, Flight, Flight History, Snack, and Meal
2	Dataset must come from multiple sources of data. You are free to come up with your own sources, you do not need to use the same ones I did.	4	BIRD, Faker, Open Food Facts, and The Meal DB.
3	Functional dependencies must hold on all tables, which means that the values are consistent across each record. For example, if a record has (city, state, country), we want the values of city and state to determine the value of country.	Applies to all tables	meals.meal_name -> meals.cat_name bird_airports.code -> bird_airports.description
4	There exists a column among the raw tables that stores more than one property in a given cell. Description and comment columns are usually a good place to look for such embeddings.	2	bird_airports.description contains these components (city, state, airport name)
5	There exists two raw tables coming from two different sources that represent the same entity. However, the entity may have slightly different properties in one table from another.	1	bird_airport and faker_airport both represent an Airport entity
6	There exists a raw table that represents more than one entity. You can usually spot those tables by looking for repeated values among their records.	1	The airlines table represents two different entities: Flight and Flight History.
7	There exists at least two disjoint entities coming from different sources that could be connected through a third entity. The third entity is not present in the dataset.	1	Flights, Snacks, and Meals can be joined through an In Flight Shopping or an In Flight Meal Service entity. Neither one is present in the raw area.

Objectives

- Create a staging area in BigQuery and populate it with the resulting tables from the transformations performed as part of this project.
- For each column that meets criteria #4, split its values into its individual properties and store the resulting table in the staging area.
- For each table that meets criteria #6, decompose its records into two or more logical entities and store the resulting tables in the staging area.

Note: you do not need to create all the staging tables as part of this project. Please create only those which are impacted by criteria #4 and #6. The remaining staging tables will be created in the next project.

Implementation Guidelines

The following guidelines apply only to the tables and columns in the staging area. They do not apply to the raw area. The raw tables are immutable.

- Tables should be stored in their own dataset in BigQuery. The name of the dataset should follow the convention **[domain]_stg** where [domain] is the name of your data domain and **stg** is a short name for staging. For example, airline_stg.
- Tables should be properly typed. If you used string types in the raw tables to represent numeric or date fields, convert them to their proper types while creating the staging tables. Consult the BigQuery built-in string functions and cast functions for more details.
- Tables should have a **data_source** field that stores the name of the data source from which they came (e.g. BIRD, Faker, etc.). This is an additional column that should be added to the create table statement.
- Tables should inherit all the attributes from their origin table in the raw layer except for the transformed attributes. For example, if a raw table contains a description field that gets split into three fields, the staging table should not inherit the original description field, but only the three new fields.
- Table names should uppercase the first letter of a word and use an underscore between two words. For example, Meal_Service.
- Column names should remain in lowercase across the board.
- Both table and column names should be descriptive. If the original name is not descriptive, rename it to a more descriptive name while conforming to our naming convention.
- Create an ERD and data dictionary for the staging area. This is only a first draft to capture the entities that you have created as part of this project. You will continue to add to the diagram and dictionary in the next project.
- Publish to your repo: **field-decomp.ipynb, entity-decomp.ipyng, erd-stg-v1.pdf**, and **data-dict-stg-v1.xlsx**.
- Create a submission.json file and upload it to Canvas by the deadline. Only one person per group needs to do this step.

CS 329E Project 2 Rubric
Due Date: 02/08/24

`field-decomp.ipynb` is thorough and meets all requirements **-10** incorrectly used BigQuery commands and/or Python code **-10** did not check for primary key conformation **-10** did not verify creation of staging table in staging dataset **-15** incorrect field splitting for each staging table **-20** did not create staging tables **-35** missing file	35
`entity-decomp.ipynb` is thorough and meets all requirements **-5** for lack of adding primary key for each newly created table **-5** did not delete intermediate table **-10** no indication of choosing primary key **-10** missing foreign keys at all (acceptable if a single table doesn't have FK) **-10** for lack of verifying each staging table created (for each entity) **-15** lack of create table statements for each entity missing **-35** missing file	35
ERD diagram accurately depicts relations between the staging tables **-5** for each missing important link **-5** for each missing staging table **-10** ERD not aligned with data dictionary columns **-20** missing file	20
Data dictionary has all important information about staging tables **-2** for each missing column of staging table **-5** missing description column **-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

-10 Wrong commit id	
Total Credit:	**100**