CS 329E Project 3, due Thursday, 02/15.

Recall our data acceptance criteria from Project 1, copied below for convenience. In this project, we assume that your raw data conforms to criteria #5 and #7. Our focus will be twofold: remodel the entities that meet criteria #5 and #7 and create remaining tables in the staging area. By the end of this project, the staging area should be complete.

| 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<br>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**

- To remodel the tables that meet criteria #5, create a new table that merges the records from the two raw tables which represent the same entity. The new table should include the combined properties of its source tables.
- To address criteria #7, create and populate a junction table that connects the disjoint entities so that they can be queried together. The junction table should be based on some simple business logic.
- Tables that were not affected by criteria 4-7 should be copied into the staging area. Perform the usual referential integrity checks on those tables. The staging area should be complete by the end of this 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 remain untouched.

- All tables should be connected and have referential integrity. If a table in the raw layer contains some duplicate records, remove those records from the table in the staging layer.
- The business logic you used to drive the implementation of your junction table should be documented in your notebook. The business logic does not need to be 100% accurate. It is your best guess based on your knowledge of the domain and the data which you have at your disposal.
- The logic for addressing criteria #5 should be in a notebook called **merge.ipynb**.
- The logic for addressing criteria #7 should be in a notebook called **join.ipynb**.
- The logic for copying tables from raw which were unaffected by criteria 4-7, should be placed in a notebook called **catchall.ipynb**.
- Tables should be properly typed and have a **data_source** field that stores the name of the data source from which they came (e.g. BIRD, Faker, etc.).
- Table and column names should follow the naming convention adopted for the staging area.
- Update your ERD and data dictionary for the staging area to reflect the new entities you added in this project.
- Publish to your repo: **merge.ipynb, join.ipynb, catchall.ipynb, erd-stg-v2.pdf**, and **data-dict-stg-v2.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 3 Rubric
**Due Date: 02/15/24**

| | |
|---|---|
| `merge.ipynb` is thorough and meets all requirements<br>**-5** did not update data source column in merged table<br>**-5** did not drop staging table at the end<br>**-5** did not set primary keys<br>**-5** did not set foreign keys<br>**-10** did not use left join or did not attempt to get all records, including nulls<br>**-10** did not verify contents of merged table<br>**-15** lack of create table statements<br>**-25** missing file | 25 |
| `join.ipynb` is thorough and meets all requirements<br>**-5** did not update global variables in python code, such as project_id, stg_name, etc (if applicable)<br>**-10** did not set primary and/or foreign keys<br>**-10** did not verify contents of new table<br>**-20** incorrect logic or code in table creation<br>**-25** missing file | 25 |
| `catchall.ipynb` is thorough and meets all requirements<br>**-5** did not drop staging table at the end (unless indicated there was no need)<br>**-5** did not add data_source field to the tables<br>**-10** did not verify table counts<br>**-10** did not add primary keys<br>**-15** did not properly remove duplicate entries in tables<br>**-25** missing file | 25 |
| ERD diagram accurately depicts relations between the staging tables<br>**-5** for each missing important link<br>**-5** for each missing staging table<br>**-10** ERD not aligned with data dictionary columns<br>**-15** missing file | 15 |
| Data dictionary has all important information about staging tables<br>**-2** for each missing column of staging table<br>**-5** missing description column<br>**-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:<br>`{` | Required |

```
    "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"
}
```

| | |
|---|---:|
| **Total Credit:** | **100** |