Milestone 4 due Friday, 10/04.

Perform the following modeling tasks to improve the quality and usability of the data in `dataset1`.

1. Create a new BQ dataset to stored your modeled tables for `dataset1`. The dataset name should follow the naming convention `<source>_modeled` where `<source>` is the source of the data. For example, fda_modeled.

2. Create new tables in your modeled dataset by applying the design principles learned in class:
   - split any staging tables that contain more than one entity into separate tables.
   - join staging tables that store different attributes belonging to the same entity.
   - union staging tables that store distinct records belonging to the same entity type.
   - identify a primary key (PK) for each modeled table.
   - check for the presence of duplicate records in each modeled table.
   - remove any duplicate records found.
   - copy the SQL you ran for this step into a file `transforms.sql`.

3. Identify relationships between the modeled table:
   - connect the tables in the diagram using the appropriate relationship type.
   - check for any referential integrity violations.
   - remove any records that violate referential integrity.
   - copy the SQL you ran for this step into the file `transforms.sql`.

4. For each field in the modeled tables, choose a primitive data type that most precisely represents its domain of values:
   - if the field is of type `STRING` and it stores `INTEGER`, `NUMERIC`, `DATE` or `TIMESTAMP` values, cast its type to the most fitting type.
   - if the field is of type `INTEGER` and it stores a `DATE` or `TIMESTAMP` value, cast its type to the most fitting type.
   - if the field is of type `TIMESTAMP` and the values it stores are of type `DATE` (i.e. the time component is not being used), cast its type to `DATE`.
   - use BQ's **`CAST`** function to convert from one type to another.
   - if the **`CAST`** function returns an error, make a note of the field and the error in the file `TRANSFORMS.txt`.
   - copy the SQL you ran for this step into the file `transforms.sql`.

5. Create a file `ERD-dataset1-modeled.pdf` that denotes:
   - current state of your modeled tables (as opposed to future state)
   - field names, data types, and keys (PK, FK) for each entity.
   - relationships between entities.

6. Verify queries:

- rewrite the join queries you developed as part of Milestone 3 to run over the modeled tables.
- Update `join-queries.sql` with your code changes.

Commit to your repo `transforms.sql`, `TRANSFORMS.txt`, and `ERD-dataset1-modeled.pdf` and push files to Github. Note that `TRANSFORMS.txt` is only required if you encountered errors during type casting.

CS 327E Milestone 4 Rubric
Due Date: 10/04/19

For `dataset1`, all tables should have an identified primary key. Values in the primary key should have no duplicates. String fields, if able to be casted to a more fitting type, should be. In addition, identify all entity types in your tables, split additional entity types into their own tables, join tables belonging to the same entity type, and union all tables that share the same fields **-10** `<source>_modeled` dataset not found in BQ project **-40** `transforms.sql` not found in repository **-20** no primary keys identified from ERD **-10** marked primary keys contain duplicates **-10** each string field containing only `INTEGER, NUMERIC, DATE, or TIMESTAMP` not cast, up to **-40** partial credit is awarded for explanations in `TRANSFORMS.txt` **-10** each non-merged entity type, table with multiple entity types, or un-unioned tables containing the same data (i.e tables representing the same data across different years).	40
For `dataset1`, all child tables should have an identified foreign key. **-30** no foreign keys identified on child tables in ERD **-20** relation is incorrect **-15** orphaned rows contained in child table	30
An ERD should be pushed that contains all detailed information for the fields in `dataset1`. Note that credit from other parts of the assignment may rely on this part. **-30** `./ERD-dataset1-modeled.pdf` not found in repository **-10** missing field types **-10** missing field names **-10** missing field keys **-5** incorrect keys marked	30
Fix all broken SQL statements from previous milestones 3-5. Make sure each statement runs properly. Save them into the same files, replacing the broken statements with their fixed counterparts. **-5** each erroneous SQL query, up to **-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:
	Required

{
"commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9",
"project-id": "some-project-id"
}	

Total Credit:	**100**