Milestone 10 due Sunday, 04/26.

Part 1:

Develop the data transforms needed to implement your cross-dataset queries. The transforms should read the records from your secondary dataset's modeled tables, perform one or more transforms on the input data, and write the cleansed and normalized records back to BigQuery as new tables. The transforms can either be done as Beam pipelines or SQL scripts or a combination of the two. Decide which technology to use based on the nature of your data and your comfort level with these technologies.

If using Beam, remember to develop and test the pipelines on a small subset of the data using the `DirectRunner`. Once tested, convert the pipelines to use the `DataflowRunner` and run them over the entire input data.

General Coding Conventions:

- The code should be commented sufficiently to follow the main logic of the transforms.
- Execute the transforms from your existing `<source>_modeled.ipynb` notebook.

Beam Coding Conventions:

- A Beam pipeline should transform a single source table.
- All transforms applied to a source table should be placed in the same Beam pipeline.
- A pipeline script should be named `<table>_beam.py` if it runs the pipeline with the Direct Runner.
- A pipeline script should be named `<table>_beam_dataflow.py` if it runs the pipeline with the Dataflow Runner.
- A table should be named `<table>_Beam` if it was produced by a Direct Runner execution.
- A table should be named `<table>_Beam_DF` if it was produced by a Dataflow Runner execution.

SQL Coding Conventions:

- A table should be named `<table>_SQL_Final` if it was produced by a SQL transform and it represents the final result for this table.
- An intermediate table produced by a SQL transform should be named `<table>_SQL_<n>` where `n` represents the step number in the series. For example, if table `Location` goes through 3 transformations, name the output table from the first

transform, `Location_SQL_1`, the output table from the second transform, `Location_SQL_2`, and the third and final result table, `Location_SQL_Final`.

Part 2:

Verify that the BigQuery result tables (e.g. `<table>_Beam_DF` or `<table>_SQL_Final`) contain a valid primary key. Child tables must also have a valid foreign key. Run the appropriate SQL statements within your `<source>_modeled.ipynb` notebook to verify these constraints.

Update your ERD to reflect the schema of your transformed tables:

- Diagram should represent only the latest version of each table (e.g. `<table>_Beam_DF` or `<table>_SQL_Final`).
- Entity types should specify field names, data types, and keys for each table.
- Diagram should denote the dataset that each table belongs to (e.g. entities from `dataset1` use one background color and entities from `dataset2` use a different background color).
- Draw the relationships between the tables within `dataset2` as well as **across** the two datasets.
- Name your updated ERD file `erd-unified.pdf`.

Part 3:

1. Implement your cross-dataset queries:
   - Create a new Jupyter notebook named `cross_dataset_analysis.ipynb`
   - Write 3 cross-dataset queries in this notebook
   - Queries should use the modeled tables from `dataset1` and `dataset2`
   - Create new BQ dataset called `reporting`
   - Wrap each cross-dataset query into a view and create view in `reporting` dataset
   - Add a short comment above each SQL statement to describe the query. Comments should be in Markdown format

2. Create visualizations in Data Studio:
   - Create a data visualization from each cross-dataset query
   - Data Sources query the SQL views from the previous section.
   - Charts should visualize the data in a compelling way.
   - Add the 3 charts to your existing Data Studio report (aka dashboard).
   - Export the report as a PDF file save it as `dashboard-final.pdf`.

CS 327E Milestone 10 Rubric
Due Date: 04/26/20

Part 1 - Create a number of data transforms in Beam or SQL or combination of the two. The Beam transforms should have two Python scripts, `<table>_beam.py` and `<table>_beam_dataflow.py` for each source table. The SQL transforms should have a CTAS statement for each transform in `<source>_modeled.ipynb`. **-X** for each missing `<table>_beam.py/<table>_beam_dataflow.py` (for Beam transforms) or CTAS statements (for SQL transforms) where X is dependent on the number of transforms listed in `TRANSFORMS.txt`. If you have 2, -**20** each. 3, **-13.3** each, and so on. **-10** transform does not execute properly **-10** Beam transforms not writing to output tables `<table>_Beam`, `<table>_Beam_DF` **-10** SQL transforms not writing to output tables `<table>_SQL_n`, `<table>_SQL_Final` **-10** missing Beam pipeline run calls in `<source>_modeled.ipynb` **-10** Beam pipelines not using both DirectRunner and DataflowRunner *(points will be broken based on number of transforms)*	40
Part 2 - Verify primary key constraints on tables transformed by Beam or SQL. Verify foreign key constraints if those tables are also child tables. **-10** missing or incorrect primary key verification on final output tables in `<source>_modeled.ipynb` **-10** missing or incorrect foreign key verification on final child output tables in `<source>_modeled.ipynb` Create an updated ERD that finalizes your table schema after Beam or SQL transformations have been applied. -**10** `./erd_unified.pdf` not found in repository **-5** ERD is missing one or more entities **-5** ERD is missing one or more primary keys **-5** ERD is missing one or more foreign keys **-5** ERD is missing or incorrect relationship between entities	20
Part 3 - Create notebook `cross_dataset_analysis.ipynb` that runs your 3 cross-dataset queries. Comment each query with the function it performs. **-20** no `cross_dataset_analysis.ipynb` in repository **-5** each missing or erroneous query, up to **-15** **-5** each missing or incorrect comment, up to **-15** **-5** each query not on a transformed table, up to **-15**	40

Create three data visualizations and add them to your existing Data Studio report. The visualizations should represent the results from the three BQ views. The Data Studio report should contain a total of **6 charts**, 2 from Milestone 7, 1 from Milestone 8, and 3 from this milestone. Each chart should have a relevant title describing the dataset. **-20** `./dashboard-final.pdf` not found in repository **-10** each missing chart, up to **-20** -**10** each chart created from a BQ table instead of a BQ view, up to **-20** **-5** each missing title, up to **-15**	

`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**