

This is a major assignment. Please plan accordingly to avoid a last-minute rush.

CS 333E Project 6, due Thursday, **03/12**.

Objectives

- Create the normalized layer of the lakehouse (aka the norm layer) such that:
 - The tables resolve anomaly types 8, 9, and 10
 - The tables satisfy integrity constraints (primary and foreign keys)

In-Scope

- Anomaly types 8, 9, and 10 ([link](#))

Out-of-Scope

- Anomaly types 4, 6, and 7

Work Items

- Create your data model for the normalized layer
- Review your design with Prof. Cohen
- Populate your data model by transforming the staging tables such that you are resolving the last three anomaly types:
 - For type 8, if the list of elements may be used for joining, filtering or grouping operations in your future marts, then flatten the list of elements into a standalone child table. Otherwise, cast the list to an array type in BQ, and keep them in the same parent entity.
 - For type 9, choose a global unique identifier for your common entities and recreate the parent and child tables based on your chosen identifier.
 - For type 10, remove any fields from your child tables which describe the properties of their parent table(s) except for foreign keys. Make sure that the fields you are removing are present in the parent table(s) (if not, be sure to add them before removing them from the child table).
- Implement referential integrity constraint checks:
 - Create a primary key constraint on each table in the normalized layer.
 - Since BQ does not enforce primary key constraints, you also need to validate that the constraint holds.
 - Create a foreign key constraint on each child table
 - Since BQ does not enforce primary key constraints, you also need to validate that the constraint holds.
 - If you encounter fuzzy duplicates while performing these checks, remove implement the fuzzy dedup technique to cluster similar records (see the Airport primary key example) or at smaller scale map the elements to a canonical set (see the Business category examples)

Code Samples

See [snippets](#) repo for code samples

Implementation Details

- Create a new folder in your repo and name it `project6`. Store all of your artifacts for this project in the `project6` folder.
- Create an ERD for the normalized layer. It should capture the most important fields for every table or entity in your normalized layer, including their data type and relationships. All your entities in the ERD should be connected with solid edges. If you end up with an entity that doesn't have any connections, something went wrong with your design. For clarity, please bold the fields which have changed and the new entities that are new to this layer.
- Create a Colab notebook named `project6-norm-layer.ipynb` with your implementation work.
- Import the [provided notebook](#) into your Colab and use it as a reference. When in doubt, consult this notebook.
- Create the normalized tables in a new dataset `[your-domain]_norm`.
- **Do not** mutate the staging tables when applying your transformations. All transformations should be materialized in the temp or normalized layers.
- Implement transformation in SQL with BQ functions such as `AI.generate()`, `ML.generate_embedding()`, and `vector_search()`.
- Annotate your notebook with observations and design decisions, especially when resolving one of the three anomaly types.
- For referential integrity, use the `ALTER TABLE` commands to create the primary key and foreign key constraints. Although BQ does not enforce these constraints, they are useful for documentation and query optimization.
- To test the validity of the referential integrity constraints, create SQL statements that test for uniqueness (primary key) and orphan records (foreign key).
- Commit and publish your work to your GitHub repo. Remember that all artifacts should go into a `project6` folder. This includes your notebook and ERD. A data dictionary is not needed at this stage.
- 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: 03/12/26

[your-domain]-erd-v6.pdf accurately depicts the normalized layer schema and logical relationships between the tables -10 required design review with Prof. Cohen not completed -10 for each missing table or incorrectly described table -10 for each missing or incorrect table relationship -5 for each change in the normalized layer not marked in bold -5 ERD file named incorrectly -60 ERD file is missing	60
project6-norm-layer.ipynb is thorough and meets all requirements -20 incorrectly used SQL commands and/or AI/ML functions -30 missing or incorrect resolution of anomaly type 8 -30 missing or incorrect resolution of anomaly type 9 -30 missing or incorrect resolution of anomaly type 10 -20 missing primary key constraints and/or logical checks -20 missing foreign key constraints and/or logical checks -20 notebook lacks observations and design decisions -10 did not follow naming convention for dataset, tables or columns -10 missing <code>_data_source</code> and <code>_load_time</code> fields in normalized tables -80 empty normalized tables in BQ dataset -100 did not create normalized tables in BQ dataset -160 notebook is missing in repo	140
submission.json submitted into Canvas. Your project will not be graded without this submission. The file should have the following schema: <pre>{ "commit-id": "your most recent commit ID from Github", "project-id": "your project ID from GCP" }</pre> Example: <pre>{ "commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9", "project-id": "some-project-id" }</pre>	Required
Total Credit:	200