CS 329E Project 1, due Thursday 01/23.

Part 1: Code Repo

1. Find a student in the class who wants to partner with you on the projects for this course. If you need help finding a partner, please create a post on Ed and/or speak to the instructors.

2. Once you have a partner, you and your partner should choose a name for your joint code repository. Every group will be given their own repo in our GitHub organization (link). Your repo will be private to your group and the instructors.

3. Request a GitHub repo by following these steps:
   - Email your name, EID, and GitHub account for you and your partner
   - The name of your requested repo
   - Address the email to the Prof. and TA
   - Copy your partner on the email
   - Use the email subject line: [CS 329E] Spring25 Repo Request

   Note: You cannot create your own GitHub repo. You must request a private repo under our GitHub organization by following the steps mentioned above.

4. Once the TAs have created your repo, your group will receive an email invitation to join it. Accept the invitation and set up your local git repository.

5. Create a README file in your repo with you and your partner's full names, EIDs, utexas email accounts, and gmail accounts.

   Example:

   ```
   William Chia, wc1234, chiaw@utexas.edu, chiaw@gmail.com
   Prithvi Chowhan, pc1234, chowp@utexas.edu, chowp@gmail.com
   ```

Part 2: GCP Project

1. Create a shared GCP project for you and your partner to use for all your projects. Follow our project guide to create your shared GCP project and be sure to go through all the sections of the guide. Only one person per group needs to create and set up the project. **You should end up with one GCP project per group.**

Part 3: Code Samples

For subsequent parts of this assignment, please consult the technical artifacts provided in the snippets repo. In general, each project will come with a set of artifacts which you will find in the repo. The artifacts will be organized by folder, named after each project. For example, for this project, you will find the artifacts in the project1 folder.

Part 4: Datasets

The overarching project for this course requires you to have a number of datasets to work with. A major component of the current assignment (Project 1) is to look for and find the datasets that you will use to populate your warehouse with.

The datasets should be logically related and come from different sources. The datasets should contain both structured data (tabular and json) and unstructured data (pdf, images).

Think of a domain that you and your partner are interested in exploring (e.g. gaming, education, cooking, sports, etc.) and start looking for data in that domain. Feel free to use the following resources to find your datasets:

- Data is Plural (link)
- BIRD training set (link)
- Google dataset search (link)

Note: The BIRD data comes in SQLite files, which consume ~40G of disk. Please see Appendix A if you need guidance on how to work with SQLite.

You should collect 4-6 datasets that come from different sources and are logically related. At least one of those datasets should be comprised of unstructured data.

Please note that certain datasets may come with a single file while others may include multiple related files. You are not required to find a dataset that has multiple files, but if your collection of datasets is made up exclusively of single files, you need to collect a minimum of 8 datasets instead of 4-6 datasets.

The goal is to end up with 8-15 raw tables in BQ where each table is sourced from a single file. The only exception to this rule is if you are collecting time-series data where the same type of data is split up by time period and each period is in its own file. In that case, the time series files would all get loaded into the same table in BQ.

If you have found some good data that is not downloadable in bulk, you should see if you can pull it from a public API. If that's not an option, try to scrape it from the site. Please refer to the code samples in the snippets repo for some working examples.

If you end up writing scripts to pull or scrape your data, be sure to include those to your repo.

All your artifacts for this project should go into a project1 folder in your repo (with the expectation of the README).

Part 5: Google Cloud Storage

Once you have collected your datasets, copy the files into a bucket on Google Cloud Storage.

1. Create a GCS bucket in the us-central1 region and name it something descriptive (e.g. air-travel).

   Note that bucket names must be globally unique and you may not get your first choice in name. That's okay, add a suffix to the name to make it globally unique or come up with an alternate name that's available.

2. Create a folder inside your bucket called **initial-loads**.

3. Create individual subfolders under the raw folder for each one of your datasets.

4. Upload the data files into their respective folders. You should have at least one folder containing unstructured data (pdfs or images).

Part 6: Data Dictionary

Create a data dictionary to describe your datasets. The dictionary represents the entities and attributes of each dataset in your collection. For example, the source of the data, its location on GCS, the file type, etc. Please consult the sample data dictionary for more details.

Name your dictionary `[your-domain]-data-dict-v1.xlsx`. Add it to your repo, inside the project1 folder.

Part 7: Entity Relationship Diagram

Sign up for a Lucidchart Education account and use Lucidchart to create an initial ERD.

The ERD should show the main entities present in the raw data that you've collected. For each entity, you should include its attribute names, data types, and keys. Because we will be using BigQuery as our data warehouse system, you want to use BigQuery's data types.

The ERD should also include the relationships between the entities. Use a **solid** line to indicate that a relationship exists between two entities in the raw data. This means that the entities can be joined on their common attributes. Use a **dashed** line to indicate that the entities share one or more common attributes, but that more analysis is needed to determine if a join is possible.

Please consult the sample ERD in our snippets repo for more details.

To improve the diagram's readability, you should also color-code your entities. It doesn't matter which background color you choose to color them with so long as each data source's entities are assigned a different color. For example, my open flights dataset comes with airports, countries, airlines, flight routes, and aircrafts data. All five entities would be drawn with the same background color because they originated from the same source.

Once you have created your ERD, export it as a pdf. Name the file `[your-domain]-erd-v1.pdf` and add it to your repo, inside the project1 folder.

Part 8: Submission

To submit your work, push your commits to your remote repo on GitHub. You should have a README file in the root directory as well as a project1 folder with your data dictionary, diagram, and any scripts that you wrote to pull data from an API or scrape it from a site.

Create a submission.json file that looks like this:

```
{
     "commit-id":"[PASTE COMMIT ID HERE]",
     "project-id":"[PASTE GCP PROJECT ID HERE]"
}
```

Upload your submission.json to Canvas by the assignment deadline.

Note: only one person per group needs to create and upload this file.

Grading Rubric

Due Date: 01/23/25

Create a `README.md` file in your team's private repository under our cs329e-spring2025 GitHub organization **-5** no private repository under the cs329e-spring2025 GitHub organization **-5** no README.md file, file incorrectly named **-5** `README.md` is missing group member's full names, EIDs or emails	10
GCP Project is named properly and shared with your partner and instructors **-5** incorrect project name **-5** project not shared	10
Datasets are uploaded to GCS bucket and follow the folder structure specified **-20** missing `initial-load` folder and datasets **-10** missing one or more subfolders and datasets **-5** missing data in at least one subfolder	20
Data dictionary is accurate and captures 4-8 related datasets (you'll need at least 8 if you have only single file datasets) **-5** for each dataset not captured in dictionary or whose entry is incomplete **-5** for each entry that does not follow our format	30
ERD is accurate and demonstrates relationships between the entities in the raw data **-30** missing diagram **-10** missing entities **-10** missing links between entities **-10** missing important attributes and their data types	30

`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**

Appendix A: Exploring the BIRD data

This section is optional. It is intended to help you work with the BIRD data should you choose to explore it.

Download the BIRD training set to your laptop and uncompress its contents (you can read up on BIRD here). The training set uses about 40GB of space. If you don't have enough free space on your laptop, spin up a VM on GCP and download the file to the VM. You can also create a JupyterLab environment from the Vertex AI Workbench page. Use the `wget` command to download the file from the provided link.

The BIRD data is in SQLite files (*.sqlite). There is one SQLite database per dataset or domain. Download the SQLite binaries and start exploring the data.

To open a SQLite file, navigate to the directory where the file is located and run:
`sqlite3 FILE_NAME`

For example, to open the file named `address.sqlite`:
```
cd train/train_databases/address
sqlite3 address.sqlite
```

This puts you in the sqlite prompt and then you can type `.help` to see a list of the available commands:
```
sqlite> .help
```

Explore the datasets available and try to find one that you want to work with. If you choose more than one, make sure that they are logically related. You can choose any of the datasets except for the airline one.

Once you have chosen your dataset, export the tables in your dataset to CSV files. Here's an example of how to open airlines.sqlite, inspect the database schema, and export its Airlines table, to CSV:

```
sqlite3 airline.sqlite
sqlite> .schema
sqlite> .headers on
sqlite> .mode csv
sqlite> .output Airlines.csv
sqlite> select * from Airlines;
sqlite> .quit
```