Project: Crowdfunding ETL
June 2023
Project 2 Group 2

Project Members:
Abi Chambers
John Banowsky

Project Goal:
Build an ETL pipeline to extract and transform data. Create CSV files and use the CSV file data to create an ERD and a table schema. Upload the CSV file data into a Postgres database. Use the Postgres database to run queries and create a data visualization.

Starting Files:

- crowdfunding.xlsx
- contacts.xlsx

Part 1 Jupyter Notebook Transformations:
Notebook File: ETL_Mini_Project_Starter_Code_AChambers_JBanowsky.ipynb

Resource Files:

- campaign.csv
- category.csv
- contacts.csv
- subcategory.csv

For part 1, we used the starter code note book and the given xlsx files to transform and create 4 csv files.

We started by reading crowdfunding.xlsx, splitting the "category & sub-category" into a category and a subcategory column, followed by creating 2 csv files of category and subcategory with table ID's.

We then used the crowdfunding.xlsx to create campaign.csv by renaming columns, changing column data types to floats and date times, and then created a the csv using only specific columns including category_id and subcategory_id.

Our final step in the first notebook was to use json to convert the contacts.xlsx file into a readable dictionary that could be used to create a data frame. We then split the name column into first and last names and created a final contacts.csv file.

Part 2 Jupyter Notebook Reading CSVs into a Database
Notebook File: ETL_Mini_Project_Read_In_CSV_AChambers_JBanowsky.ipynb

ERD Files:

- ERD.txt
- crowdfunding_db_schema.sql
- crowdfunding_db_ERD.png

For part 2, we started by using https://www.quickdatabasediagrams.com/ to create an ERD (crowdfunding_db_ERD.png) and a SQL table schema (crowdfunding_db_schema.sql). We then imported the table schema into Postgres to create the crowdfunding_db database. We wrote a notebook using sqlalchemy to upload the csv files into the database. We then wrote 3 Queries using the data.

- Query 1: crowdfunding_query1.sql

  We joined contacts and campaign to create a table that showed the campaigns with their contacts in order of percent of the pledge they earned. They thought behind this was to find the relevant information of the most successful campaigns.
- Query 2: crowdfunding_query2.sql

  For the second query, we joined campaign and category to identify the count of each outcome per category and the percent success. The percent success required use to use a nested query because we used aggregated functions to create the columns. We viewed this as a good query to identify the success of campaigns by category.
- Query 3: crowdfunding_query3.sql

  For the last query, we joined campaign, category and subcategory to single out the music subcategories to identify which subcategory tends to earn the highest total pledge amount. We chose Music based on the 2nd query showing that music was the most successful category that had subcategories.

Part 3 Jupyter Notebook for Data Visualization
Notebook File: ETL_Mini_Project_Visualization_AChambers_JBanowsky.ipynb

Part 3 required us to use sqlalchemy to import the database and created a data visualization the showed a side by side bar graph showing total campaigns per category with particular outcomes. We used a seaborn count plot for the visualization. Our conclusions are that Film & Video and Music are popular categories that perform similarly. Theater has greater success but also failures meaning that more theatre campaigns are attempted than others but it doesn’t have the best success rate. Journalism has a 100% success rate and games is the only category that is more likely to fail than succeed.