SMU-Project2-JAbney-IReese-EPresley

Welcome to the SMU Project2-Crowdfunding ETL repository!

Our team has diligently collaborated to uncover valuable insights into the benefits and challenges of the ETL process through creating DataFrames and a database.

**Project Overview**

The mini collaborative ETL project consisted of running a Jupyter Notebook to extract data from the crowdfunding and contacts' Excel sheets, transforming and cleaning the data into four CSV files, and then loading the transformed data into a relational database.

**Create the Category, Subcategory, and Campaign DataFrames**

We extracted and transformed the crowdfunding Excel data and created the category, subcategory, and campaign DataFrames. Then, we exported and saved the three DataFrames as category, subcategory, and campaign CSV files.

**Create the Contacts DataFrame**

We had two options to create a DataFrame from the contacts' Excel sheet data: the Python dictionary method or regular expressions (Regex).

**Option 1**: We converted each row to a dictionary, then used a Python list comprehension to extract the dictionary values from the keys. We then added the values to a new list and created the DataFrame. We cleaned the DataFrame, then exported and saved it as a contacts CSV file.

**Option 2**: Using regular expressions to create the contacts DataFrame, we extracted and manipulated the string data using wildcards' special characters and created and used capture groups. Transformed, cleaned, exported, and saved the data as contacts CSV file.

**Create the Crowdfunding Database**

We inspected the data of the four CSV files, and using QuickDBD, we created the entity relationship diagram (ERD) of the four tables. Using the information from the ERD, we created a table schema for each CSV file and saved it as a Postgres file named crowdfunding\_db\_schema.SQL. Next, we loaded the SQL file into a PostgreSQL database named crowdfunding\_db using PGAdmin4. Using the database schema, we created the four tables and verified the creation by running a Select statement for each table. Next, we imported each CSV file into the corresponding SQL table. We then verified that each table had the correct data by running a Select statement for each.

**Results**

We worked on three queries in which the results explained the relational database functionality. Our Queries are shown at the bottom of this file

**Implications**

We have learned that ETL tools can handle large amounts of data, making them suitable for big data projects. Also, there can be multiple ways to perform an ETL into either a CSV or a database. The process improves the data; however, it can be challenging as it requires employees to stay up to date to adapt to the changing or increasing data volumes and latest developments. We welcome collaboration, feedback, and further contributions to enhance the depth of our basic ETL pipeline.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated