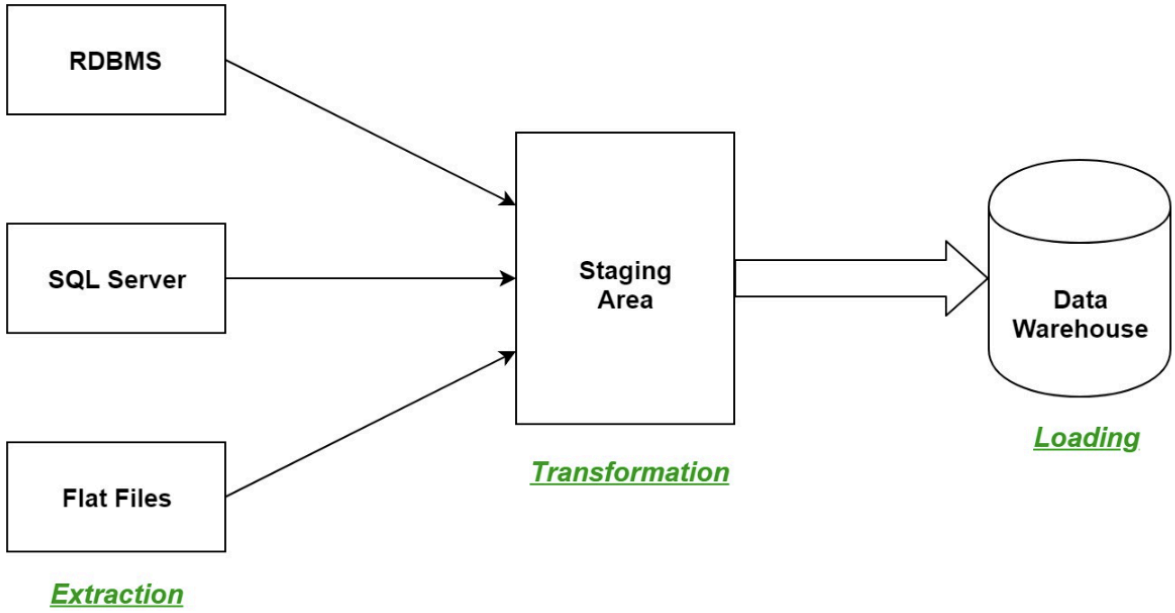
Crowdfunding Extract/Transform/Load (ETL) Project

Authors: Mrunmai Gadbail, Drew Garza, Spencer Garrett

Extract/Transform/Load (ETL) Overview

The diagram below provides a general overview of the ETL flow which begins with extracting data from a source, then transforming the data in a staging area, which enables the data to be loaded into a data store. This is a pipeline that can be setup to run as needed which enables the data to be up to date. A couple other advantages include the ability to integrate data from multiple sources and increased automation. Some disadvantages include potential complexities in setup, along with limited flexibility.



"ETL Process in Data Warehouse." GeeksforGeeks, GeeksforGeeks, 2 Feb. 2023, www.geeksforgeeks.org/etl-process-in-data-warehouse/.

Introduction

The Crowdfunding ETL project aims to design and implement an Extract, Transform, Load (ETL) pipeline for a crowdfunding dataset. This dataset contains information about various crowdfunding campaigns, including details about the projects like their creators, the pledged amounts, and the country that their campaign is based in. The primary goal of this project is to create a relational database to store this information and facilitate data retrieval and analysis. By designing an ETL pipeline, we ensure that data from two excel files is extracted, transformed to match our database schema, and loaded into the database seamlessly. Once it is available in the database we can then explore any questions we have about the data by analyzing it.

Database Design Considerations

The database was designed to accurately reflect the structure of the data we intended to use, which includes defining the correct primary keys, foreign keys, constraints, and datatypes. The primary keys ensure that there is a unique identifier for each record in a given table, while the foreign keys define the relationships between tables where a primary key is used by another table. The constraints, along with datatype definitions, guarantee that values are as expected for certain fields.

Extract/Transform/Load (ETL) Code Overview

The ETL pipeline is structured into three main stages:

1. **Extract**:
    - **Source Identification & Extraction**: Identify the data source to use and extract the data into a more usable format. In this case the sources were excel files that we read in using pandas to convert them into dataframes that are easier to manipulate which is crucial for performing effective transformations.

Extract data into a pandas dataframe after identifying the source

```
# Read the data into a Pandas DataFrame
crowdfunding_info_df = pd.read_excel('Resources/crowdfunding.xlsx')
```

2. **Transform**:
    - **Data Transformation**: Convert the extracted data to match the database schema, which includes altering data types, creating new columns and formatting them correctly, or dropping unnecessary columns. Once the transformations are complete then the data frames can be written to CSV files that can be loaded into the database to populate the tables that we wanted to match in the first place based on our database design.

Create new columns

```
# Assign the category and subcategory values to category and subcategory columns.
crowdfunding_df[['category', 'subcategory']] = crowdfunding_df['category & sub-category'].str.split('/', expand=True)
```

Create new dataframes with necessary columns

```
# Create a category DataFrame with the category_id array as the category_id and categories list as the category name.
category_df = pd.DataFrame({
    'category_id': cat_ids,
    'category': categories
})

# Create a category DataFrame with the subcategory_id array as the subcategory_id and subcategories list as the subcategory name.
subcategory_df = pd.DataFrame({
    'subcategory_id': scat_ids,
    'subcategory': subcategories
})
```

Write dataframes to CSV files

```
# Export categories_df and subcategories_df as CSV files.
category_df.to_csv("Resources/category.csv", index=False)

subcategory_df.to_csv("Resources/subcategory.csv", index=False)
```

3. **Load**:
    - **Database Connection**: Establish a connection to the relational database using SQLAlchemy that lets us communicate with the database. For this project we connect to the postgres database that we created based on our design considerations.

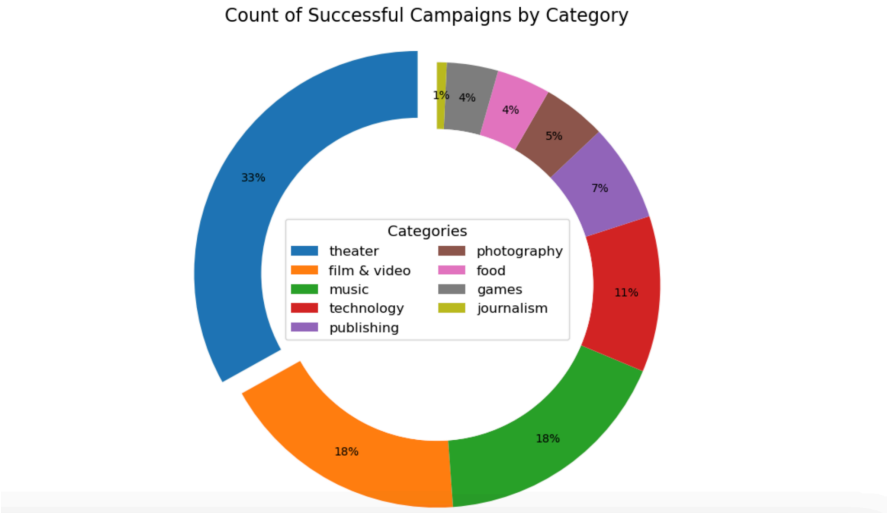
```
# create connection using pre-defined variables
connection_string = f"postgresql+psycopg2://{SQL_USERNAME}:{SQL_PASSWORD}@{SQL_IP}:{PORT}/{DATABASE}"
engine = create_engine(connection_string)
```

    - **Data Insertion**: Write scripts using SQLAlchemy and Pandas to insert the transformed data into the appropriate database tables. Inserting the data into the target tables completes the ETL process by making this data available for analysis from this table. This pipeline also enables us to keep the tables up to date by running these steps whenever data is added to excel files. This creates more reliable and widely available data for more people to use.

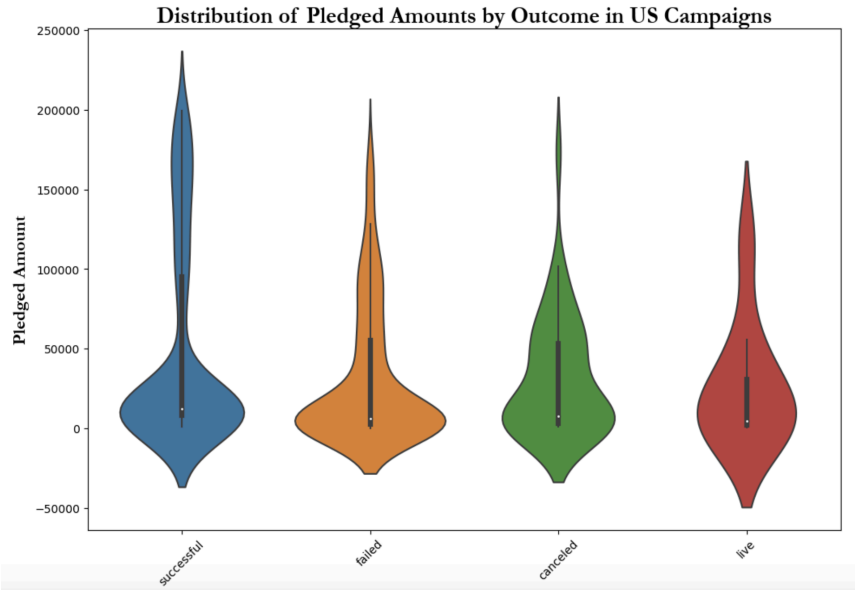
```
# Use pandas and sqlalchemy to insert category dataframe into crowdfunding_db - category table
category_df.to_sql("category", schema="public", con=engine, index=False, if_exists="append", method="multi")
```

Analysis

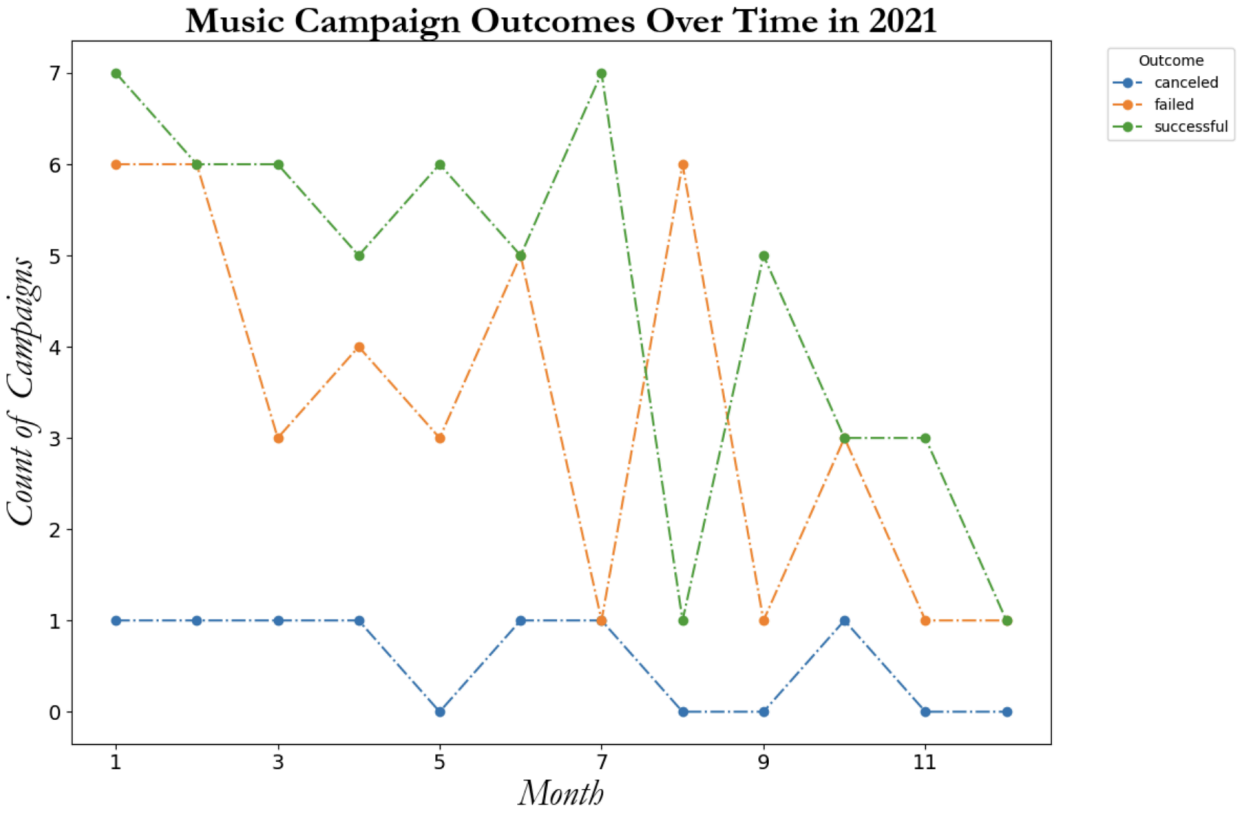
We queried the database to analyze the success rate of crowdfunding campaigns by category. Using SQLAlchemy, we aggregated the number of successful campaigns for each category. This helped identify that theater campaigns have the most successful campaigns, while journalism has had the fewest successful campaigns. However, this does not capture the full picture, on a closer look journalism has only had four campaigns which have all been successful.



Using raw SQL to explore the distributions of pledge amounts by campaign outcome in the United States. The result made sense to us with successful campaigns having more higher pledges than the other outcomes, which is evident from the thicker tail. All of the distributions are skewed with tails to higher pledge amounts, and averages closer to zero.



Finally, we dove deeper into how music campaigns performed over time by examining the count of music campaign outcomes over the month. Based on this the best time to launch music campaigns is January and July since both have the highest count of successful campaigns. The worst months to launch a campaign are January and August. Since January has the most successful and failed campaigns that would still be a better time to launch one than in August which has the fewest successful campaigns.



Limitations

1. **Scalability**: The ETL pipeline may need modifications to handle large volumes of data efficiently. It might also need to be modified as new data is included in the source files. Additional modifications would need to be made in order to account for other data sources.
2. **Timeliness**: Depending on the frequency of data updates, the ETL pipeline may need to be scheduled to run at specific intervals to ensure the database remains up-to-date.

Conclusions

This project demonstrates the importance of a well-designed ETL pipeline in managing and utilizing large datasets. By designing a robust relational database and implementing an efficient ETL process, we ensure that data is accurately and efficiently extracted, transformed, and loaded into the database. The insights gained from this project highlight the need for careful planning in database design and ETL implementation to ensure data integrity and performance. Future improvements could focus on optimizing the ETL process for scalability and incorporating more advanced data validation techniques to handle more data sources and formats.