SMU-Project2-JAbney-IReese-EPresley

Introduction
The mini collaborative ETL (extract, transform, load) 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.

Data Engineering
The project was divided into the following subsections, thus creating four tables to gain understanding from clean, up to date and accurate data:

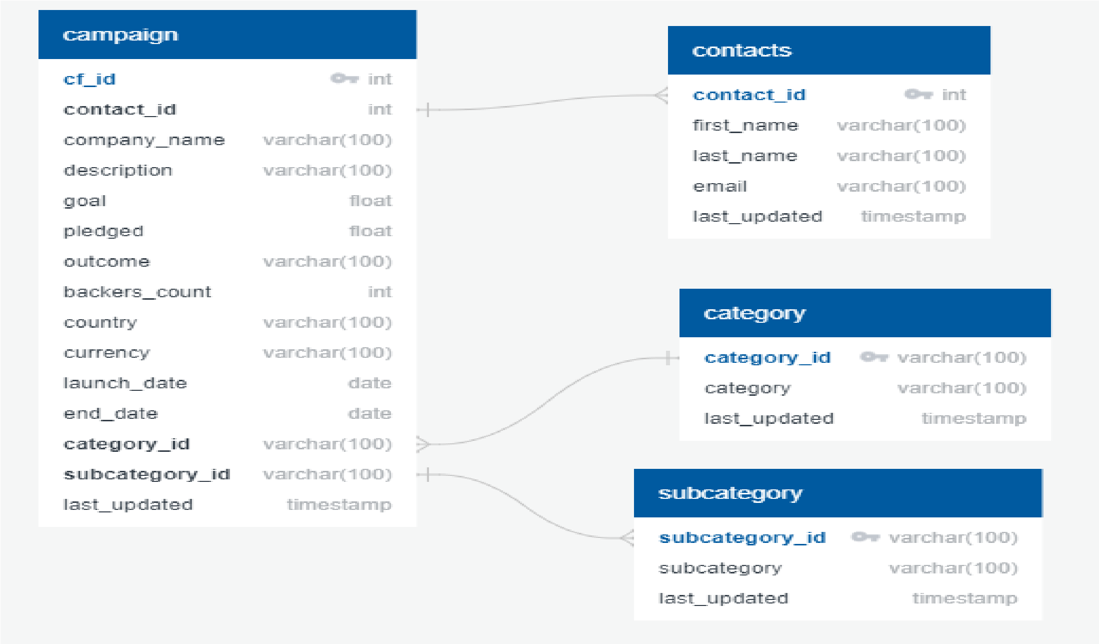
- Create the Category and Subcategory DataFrames
- Create the Campaign DataFrame
- Create the Contacts DataFrame
- Create the Crowdfunding Database

First, we extracted and transformed the crowdfunding Excel data into category, subcategory, and campaign DataFrames. Then, we exported and saved the three DataFrames as category, subcategory, and campaign CSV files. Next, we had two options to create a fourth 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.

To 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.

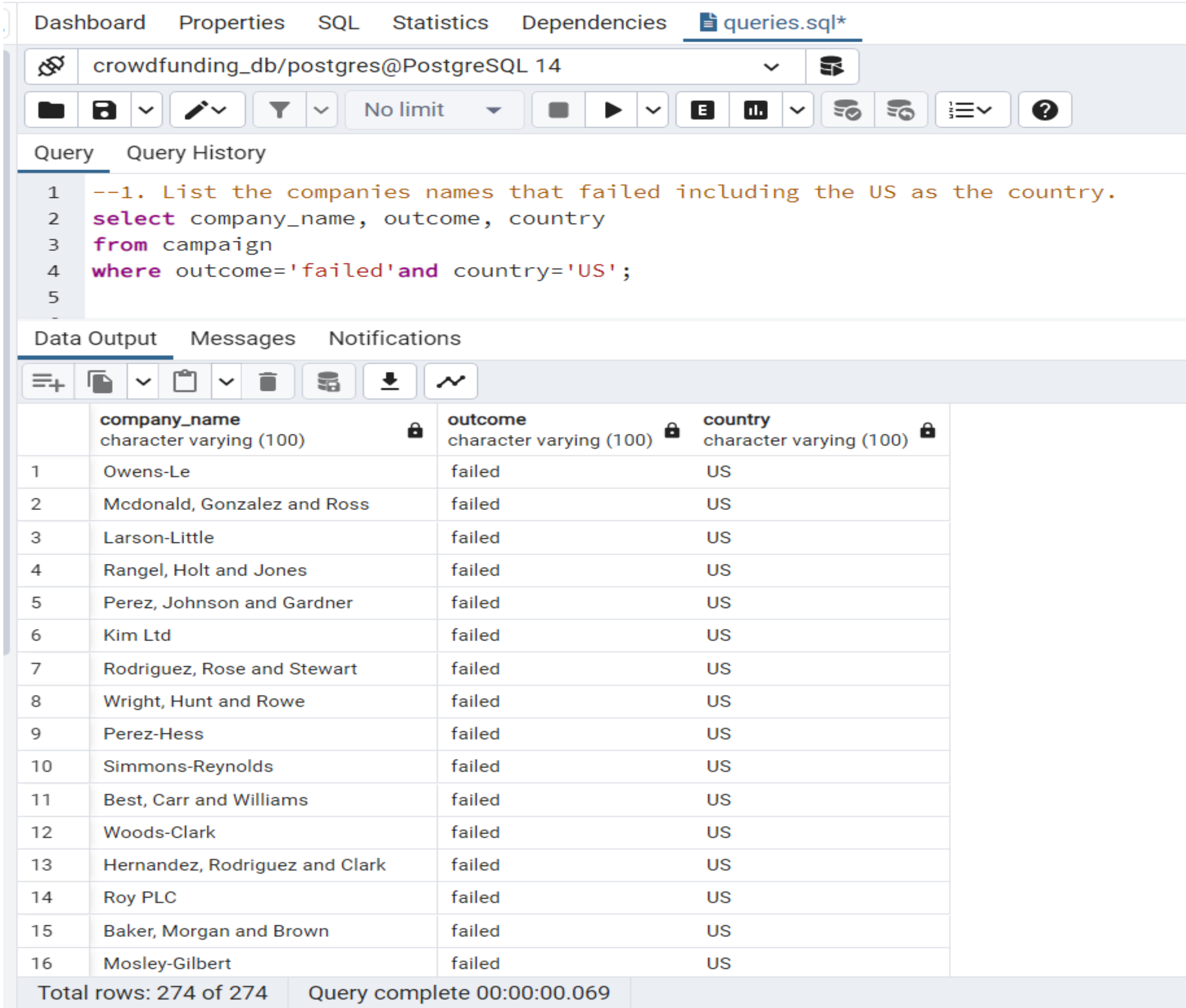


Using the information from the ERD, we created a table schema for each CSV file and saved it as a Postgres file named crowdfunding_db_schema.sql. Next, we loaded the SQL file into a PostgreSQL database named crowdfunding_db using PGAdmin4. Using the database schema, we created the four tables and verified 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

Below are the results of the three queries that explain the relational database functionality.

1. List the companies names that failed including the US as the country.

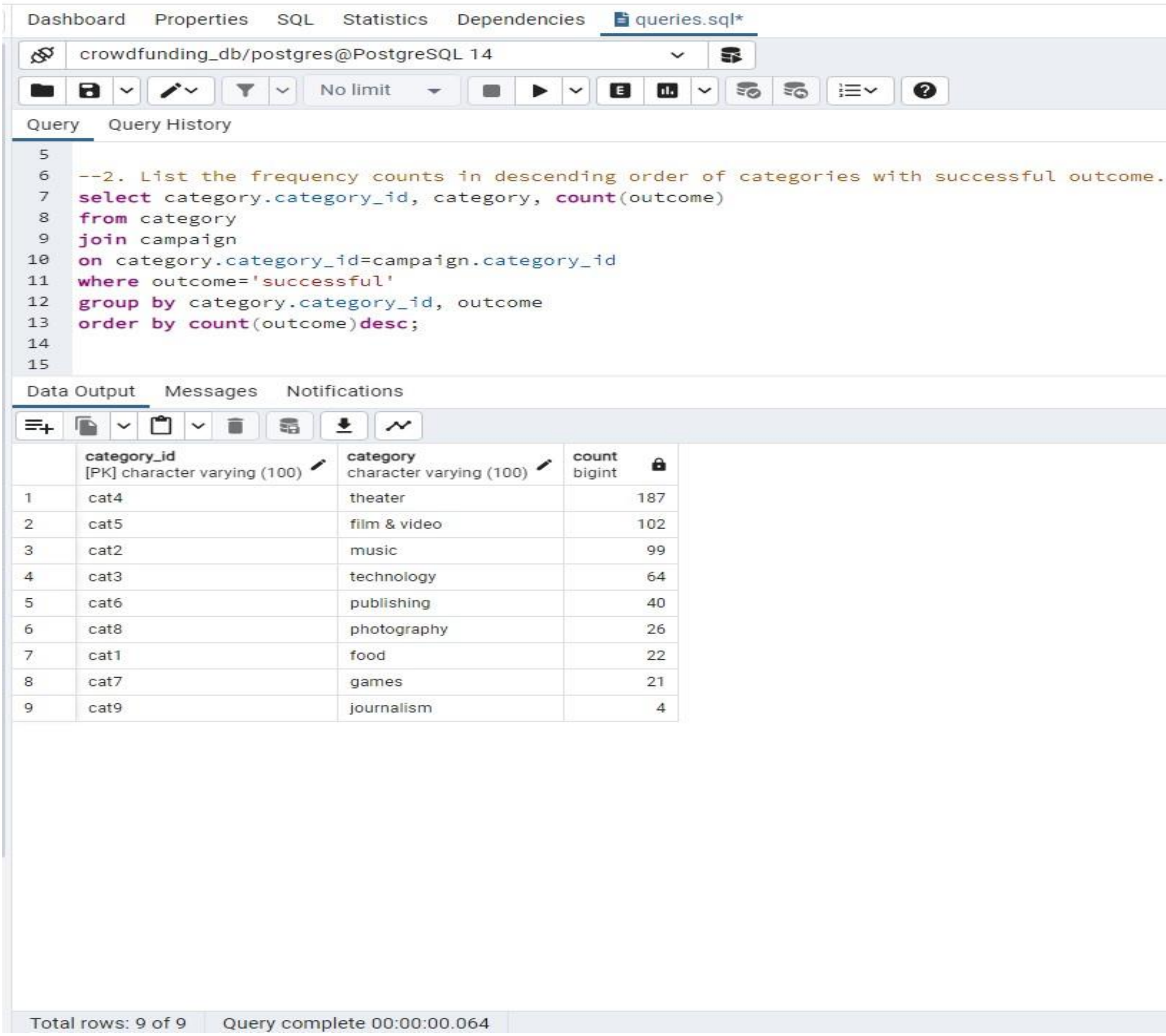



```sql
--1. List the companies names that failed including the US as the country.
select company_name, outcome, country
from campaign
where outcome='failed'and country='US';
```

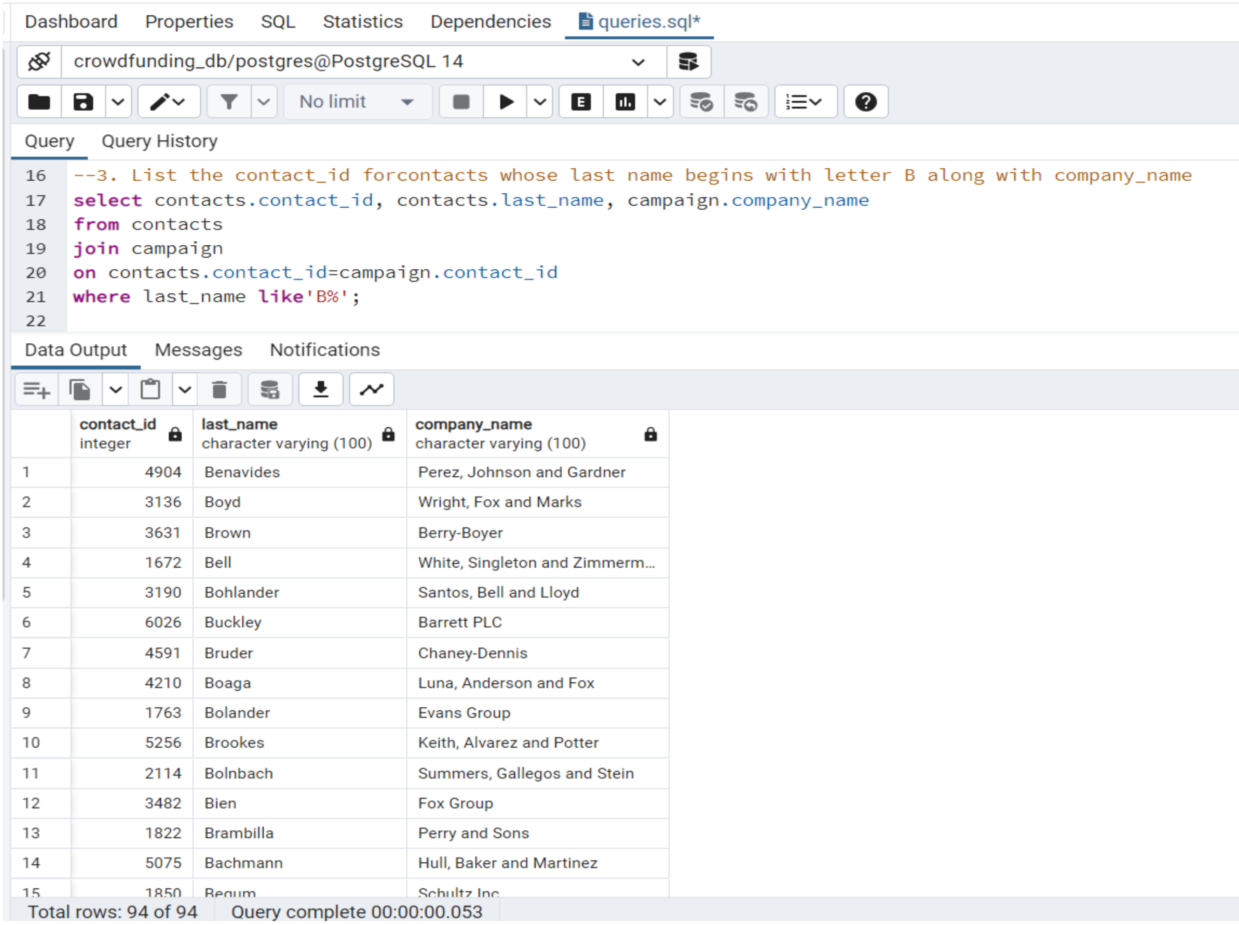
	company_name character varying (100)	outcome character varying (100)	country character varying (100)
1	Owens-Le	failed	US
2	Mcdonald, Gonzalez and Ross	failed	US
3	Larson-Little	failed	US
4	Rangel, Holt and Jones	failed	US
5	Perez, Johnson and Gardner	failed	US
6	Kim Ltd	failed	US
7	Rodriguez, Rose and Stewart	failed	US
8	Wright, Hunt and Rowe	failed	US
9	Perez-Hess	failed	US
10	Simmons-Reynolds	failed	US
11	Best, Carr and Williams	failed	US
12	Woods-Clark	failed	US
13	Hernandez, Rodriguez and Clark	failed	US
14	Roy PLC	failed	US
15	Baker, Morgan and Brown	failed	US
16	Mosley-Gilbert	failed	US

Total rows: 274 of 274 Query complete 00:00:00.069

2. List the frequency counts in descending order of categories with successful outcome.



3. List the contact_id for contacts whose last name begins with letter B along with company_name.



Findings

The United States had the highest number of failed companies. Out of 364 failed companies, 274 were from the US.

The theater category had the highest success rate out of 9, while journalism had the lowest.

Out of 1000 last-name contacts, there were 94 whose last names begin with the letter B.

Conclusion

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, ensuring data quality.

Future Work

Because of the constant evolution of data, a new understanding of ETL may be required. We welcome collaboration, feedback, and further contributions to enhance the depth of our basic ETL pipeline.

Bonus Work

In the notebook title “ETL_Project_Bonus.ipynb” we added data to our tables using SQLite and also added a visualization to one of our queries. We found that importing the data via SQLite was much more efficient than manually importing in PGAdmin4. The visualization below was created using pandas and matplotlib: