Project 2: the ETL mini project

In this project you will be building an ETL pipeline using Python, Pandas, and either Python dictionary methods or regular expressions to extract and transform the data. After you transform the data, you'll create four CSV files and use the CSV file data to create an ERD and a table schema. Finally, you’ll upload the CSV file data into a Postgres database.

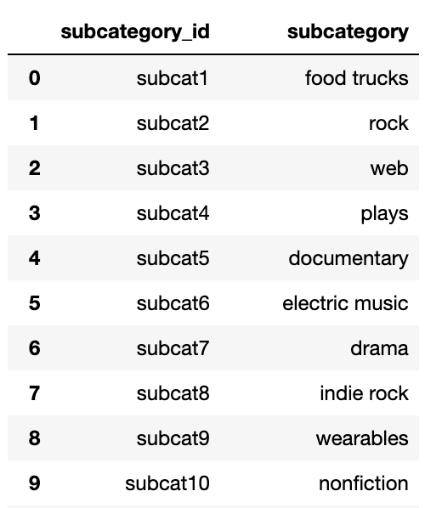
Instructions:

The instructions for this mini project are divided into the following subsections:

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

**Part-1:** Create the Category and Subcategory DataFrames

1. We read the crowdfunding excel file in pandas by using pd.read\_excel.
2. Split the column name “ category & sub-category “ to two column and then create these two new column by using one code for both job : ( crowdfunding\_info\_df[['category', 'sub-category']] = crowdfunding\_info\_df['category & sub-category'].str.split("/", expand = True) ).
3. Create a arrays for each new column math its length.
4. Using the list to add the word “ cat “ to the category array and the word “subcat” to the subcategory array.
5. Create a data frame for each new list as category\_ ids and subcategory\_id look like



1. Last save the two list as csv file.

#### Part-2: Create the Campaign DataFrame

1. Create a copy of the crowdfunding\_info\_df DataFrame name campaign\_df.
2. Rename the blurb, launched\_at, and deadline columns.
3. Convert the goal and pledged columns to a `float` data type.
4. Format the launched\_date and end\_date columns to datetime format using pd.to\_datetime code.
5. Merge the campaign\_df with the category\_df on the "category" column and

the subcategory\_df on the "subcategory" column.

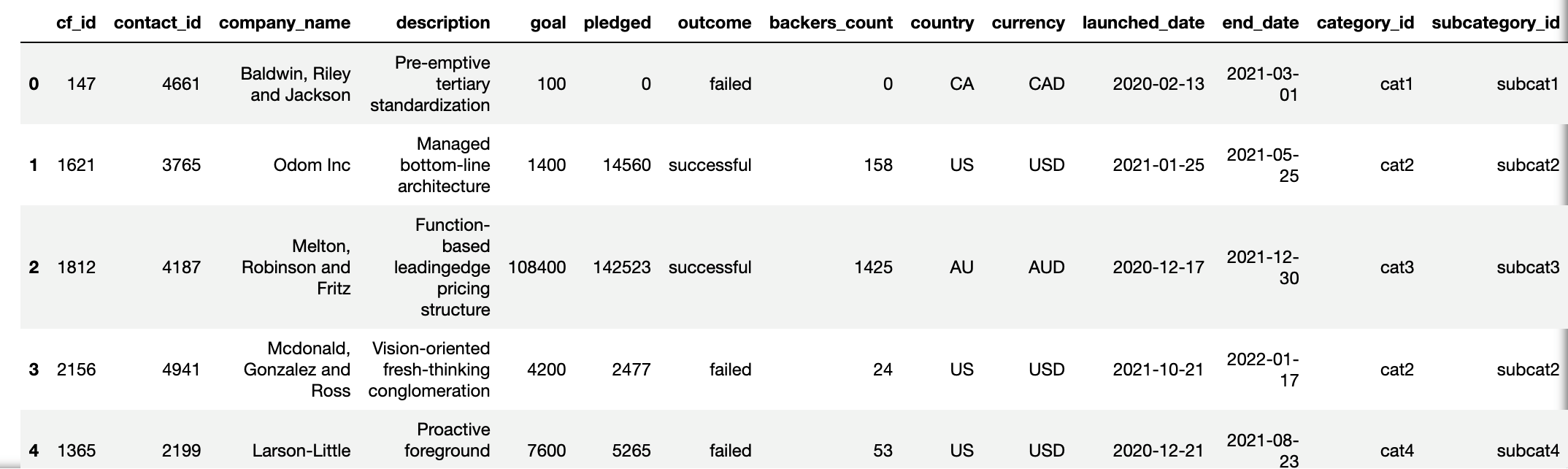
To get the merge we had to change the column names on both category and subcategory dataframe to match the names in compaign data frame, the we merge by two steps, first merge the category table with the compaign table, than merge the subcategory table with the new table, we used this code : ( cat\_merged\_df= campaign\_df.merge(category\_df, how='outer', on='category')

campaign\_merged\_df= cat\_merged\_df.merge(subcategory\_df, how='outer', on='sub-category')

campaign\_merged\_df.tail(10) )

1. Drop the category & subcategotry column
2. Export the data Frame to a csv file and save it in our repository.

Here is our result:



#### Part-3: Create the Contacts DataFrame

We Choose option 2 : Use regular expressions;

1. Read the excel file. We had to change the number of the header from 2 to 3 to make sure we are going to have column name as “ contact\_ info”
2. Extract the four-digit contact ID number using the code:

contact\_info\_df\_copy['contact\_id'] = contact\_info\_df\_copy['contact\_info'].str.extract(r'contact\_id": (\d{4})')

1. Convert the "contact\_id" column to an int64 data type using the code:

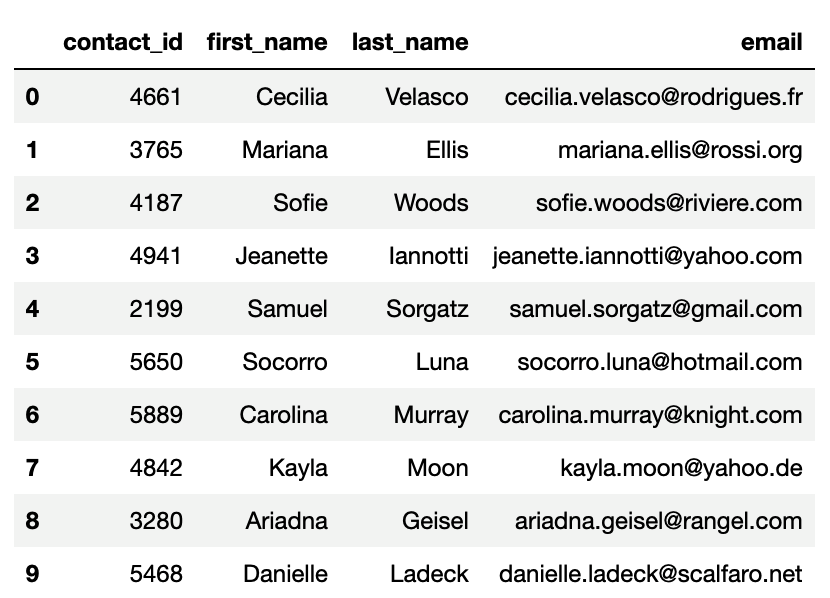
contact\_info\_df\_copy = contact\_info\_df\_copy.astype({'contact\_id': 'int64'})

1. Extract the name of the contact and add it to a new column using the code:

contact\_info\_df\_copy['contact\_name'] = contact\_info\_df\_copy['contact\_info'].str.extract(r'"name": "([^"]\*)"')

1. Extract the email from the contacts and add the values to a new column using the code:

contact\_info\_df\_copy['email'] = contact\_info\_df\_copy['contact\_info'].str.extract(r'"email": "([^"]\*)"')

1. Create a copy of the contact\_info\_df with the 'contact\_id', 'name', 'email' columns.
2. We had to split the name column to two column as first name and last name.
3. Drop the contact\_name column
4. Export the DataFrame as a CSV file.

Here is our result :

#### Create the Crowdfunding Database

1. After we inspect all the csv files, we could create all the tables with the relation between them and type of the data. Here is our result with the tables details :

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create a new Postgres database, named crowdfunding\_db.
2. To create the database and be able to import the csv file, we had to respect a order while we are creating the table.

The table order was:

* First was the contact table.
* Second was the category table.
* Third was the subcategory table.
* Last was the comping table.

1. We import all csv file with no error since our order was correct by respecting the foreign keys.
2. Using select statement to verify the results

Here is our result by showing the result of the comping table:

A screenshot of a computer

Description automatically generated