**Instructions**

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

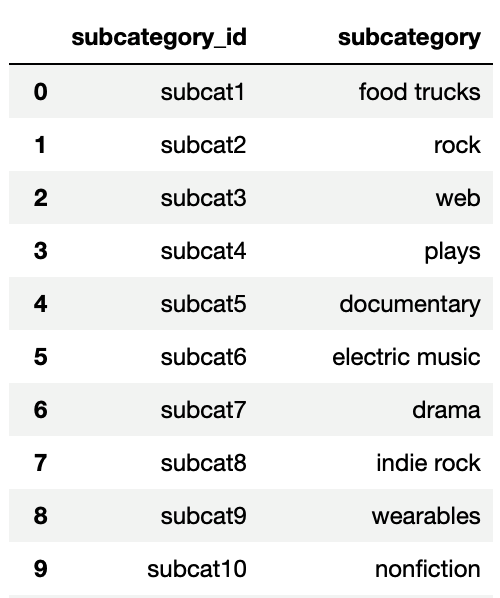
* Create the Category and Subcategory DataFrames
* Create the Campaign DataFrame
* Create the Contacts DataFrame
* Create the Crowdfunding Database

**Create the Category and Subcategory DataFrames**

1. Extract and transform the crowdfunding.xlsx Excel data to create a category DataFrame that has the following columns:
   * A "category\_id" column that has entries going sequentially from "cat1" to "cat*n*", where *n* is the number of unique categories
   * A "category" column that contains only the category titles
   * The following image shows this category DataFrame:



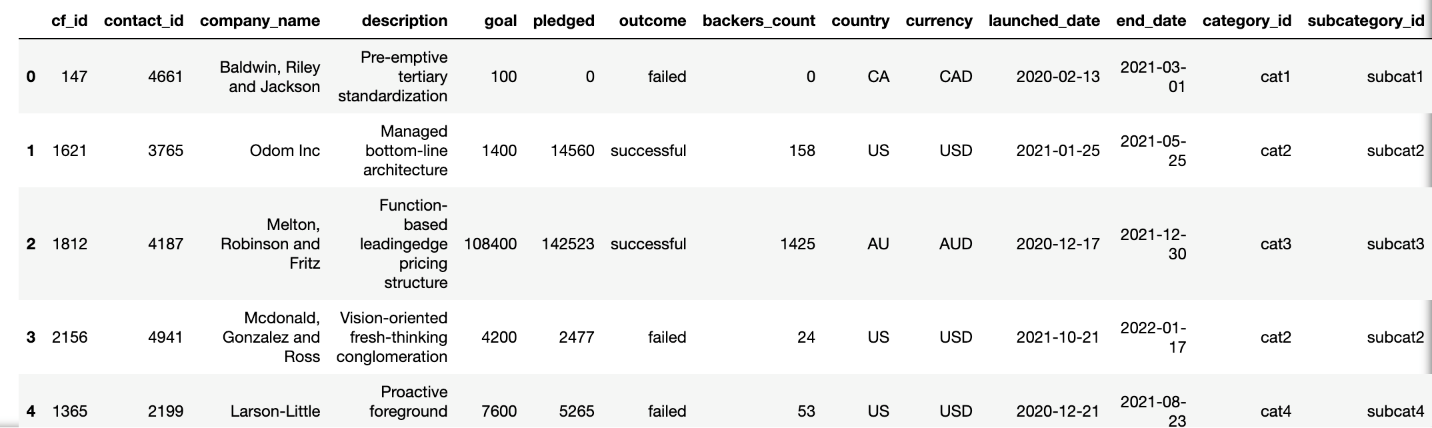
1. Export the category DataFrame as category.csv and save it to your GitHub repository.
2. Extract and transform the crowdfunding.xlsx Excel data to create a subcategory DataFrame that has the following columns:
   * A "subcategory\_id" column that has entries going sequentially from "subcat1" to "subcat*n*", where *n* is the number of unique subcategories
   * A "subcategory" column that contains only the subcategory titles
   * The following image shows this subcategory DataFrame:



1. Export the subcategory DataFrame as subcategory.csv and save it to your GitHub repository.

**Create the Campaign DataFrame**

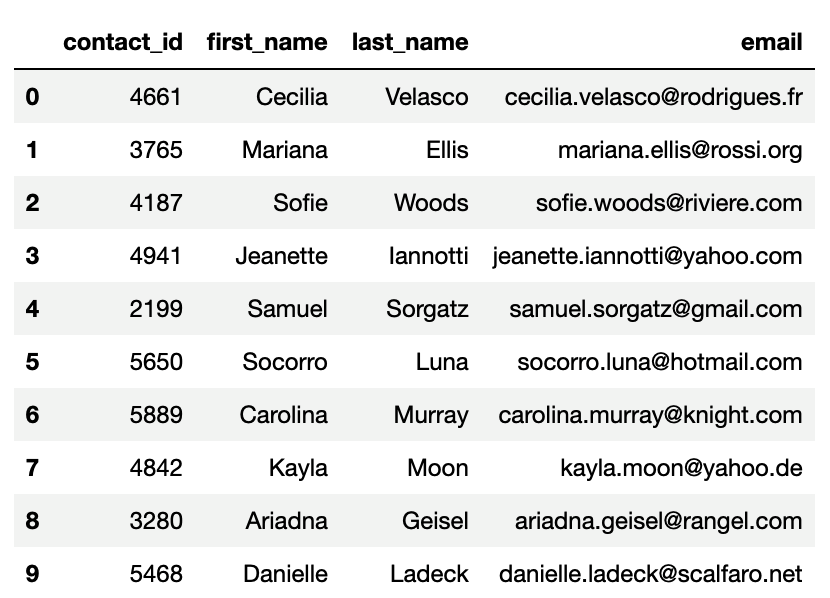
1. Extract and transform the crowdfunding.xlsx Excel data to create a campaign DataFrame has the following columns:
   * The "cf\_id" column
   * The "contact\_id" column
   * The "company\_name" column
   * The "blurb" column, renamed to "description"
   * The "goal" column, converted to the float data type
   * The "pledged" column, converted to the float data type
   * The "outcome" column
   * The "backers\_count" column
   * The "country" column
   * The "currency" column
   * The "launched\_at" column, renamed to "launch\_date" and with the UTC times converted to the datetime format
   * The "deadline" column, renamed to "end\_date" and with the UTC times converted to the datetime format
   * The "category\_id" column, with unique identification numbers matching those in the "category\_id" column of the category DataFrame
   * The "subcategory\_id" column, with the unique identification numbers matching those in the "subcategory\_id" column of the subcategory DataFrame
   * The following image shows this campaign DataFrame:



1. Export the campaign DataFrame as campaign.csv and save it to your GitHub repository.

**Create the Contacts DataFrame**

1. Choose one of the following two options for extracting and transforming the data from the contacts.xlsx Excel data:
   * **Option 1:** Use Python dictionary methods.
   * **Option 2:** Use regular expressions.
2. If you chose Option 1, complete the following steps:
   * Import the contacts.xlsx file into a DataFrame.
   * Iterate through the DataFrame, converting each row to a dictionary.
   * Iterate through each dictionary, doing the following:
     + Extract the dictionary values from the keys by using a Python list comprehension.
     + Add the values for each row to a new list.
   * Create a new DataFrame that contains the extracted data.
   * Split each "name" column value into a first and last name, and place each in a new column.
   * Clean and export the DataFrame as contacts.csv and save it to your GitHub repository.
3. If you chose Option 2, complete the following steps:
   * Import the contacts.xlsx file into a DataFrame.
   * Extract the "contact\_id", "name", and "email" columns by using regular expressions.
   * Create a new DataFrame with the extracted data.
   * Convert the "contact\_id" column to the integer type.
   * Split each "name" column value into a first and a last name, and place each in a new column.
   * Clean and then export the DataFrame as contacts.csv and save it to your GitHub repository.
4. Check that your final DataFrame resembles the one in the following image:



**Create the Crowdfunding Database**

1. Inspect the four CSV files, and then sketch an ERD of the tables by using [QuickDBDLinks to an external site.](http://www.quickdatabasediagrams.com/" \t "_blank).
2. Use the information from the ERD to create a table schema for each CSV file.

**Note:** Remember to specify the data types, primary keys, foreign keys, and other constraints.

1. Save the database schema as a Postgres file named crowdfunding\_db\_schema.sql, and save it to your GitHub repository.
2. Create a new Postgres database, named crowdfunding\_db.
3. Using the database schema, create the tables in the correct order to handle the foreign keys.
4. Verify the table creation by running a SELECT statement for each table.
5. Import each CSV file into its corresponding SQL table.
6. Verify that each table has the correct data by running a SELECT statement for each.

**Requirements**

**A Category DataFrame is Created (15 points)**

* The DataFrame contains a "category\_id" column that has entries going sequentially from "cat1" to "cat*n*", where *n* is the number of unique categories (5 points)
* The DataFrame has a "category" column that contains only the category titles (5 points)
* The category DataFrame is exported as category.csv (5 points)

**A Subcategory DataFrame is Created (15 points)**

* The DataFrame contains a "subcategory\_id" column that has entries going sequentially from "subcat1" to "subcat*n*", where *n* is the number of unique subcategories (5 points)
* The DataFrame contains a "subcategory" column that contains only the subcategory titles (5 points)
* The subcategory DataFrame is exported as subcategory.csv (5 points)

**A Campaign DataFrame is Created (30 points)**

* The DataFrame has the following columns: (25 points)
  + A "cf\_id" column
  + A "contact\_id" column
  + A "company\_name" column
  + A "description" column
  + A "goal" column that is a float data type
  + A "pledged" column that is a float data type
  + An "outcome" column
  + A "backers\_count" column
  + A "country" column
  + A "currency" column
  + A "launch\_date" with the time formatted as "YYYY-MM-DD"
  + An "end\_date" with the time formatted as "YYYY-MM-DD"
  + A "category\_id" column that contains the unique identification numbers matching those in the "category\_id" column of the category DataFrame
  + A "subcategory\_id" column that contains the unique identification numbers matching those in the "subcategory\_id" column of the subcategory DataFrame
* The campaign DataFrame is exported as campaign.csv (5 points)

**A Contacts DataFrame is Created (15 points)**

* The DataFrame has the following columns: (10 points)
  + A "contact\_id" column
  + A "first\_name" column
  + A "last\_name" column
  + An "email" column
* The contacts DataFrame is exported as contacts.csv (5 points)

**A Crowdfunding Database is Created (25 points)**

* A database schema labeled, crowdfunding\_db\_schema.sql is created (5 points)
* A crowdfunding\_db is created using the crowdfunding\_db\_schema.sql file (5 points)
* The database has the appropriate primary and foreign keys and relationships (5 points)
* Each CSV file is imported into the appropriate table without errors (5 points)
* The data from each table is displayed using a SELECT \* statement (5 points)