Crowdfunding ETL Mini Project

By: Zahra Dunia Wieser

26 June 2023

Project Overview

Purpose:

- The ETL Mini Project aims to demonstrate the process of extracting, transforming, and loading data for effective analysis.

Goals:

- The project focuses on creating an efficient ETL pipeline to handle crowdfunding and contacts data.

Tools and Languages:

- Python, Pandas, and Postgres are the key tools and languages used for data processing and database management.

Data Extraction and Transformation

Extracting and transforming data from crowdfunding.xlsx and contacts.xlsx files.

Ensure data consistency and suitability for analysis.

Python and Pandas enable efficient data extraction and transformation.

Python dictionary methods or regular expressions facilitate data organization.

Creation of separate DataFrames enhances data management and analysis.

Category DataFrames

category_id	category
cat1	food
cat2	music
cat3	technology
cat4	theater
cat5	film & video
cat6	publishing
cat7	games
cat8	photography
cat9	journalism

* The Category DataFrame plays a vital role in organizing and categorizing data within the ETL Mini Project.
* The purpose of the Category DataFrame is to provide a structured representation of categories for efficient data management and analysis.

DataFrame Structure:

* The Category DataFrame consists of two columns: **category_id** and **category.**
* The category_id column is sequentially numbered to uniquely identify each category.
* The category column contains the titles or names of the categories.

Sequential Numbering

* Each category is assigned a unique identifier for easy reference and analysis.
* Sequential numbering facilitates easy referencing, sorting, and analysis of the categories.

Category Titles

* Category titles are added to provide meaningful and descriptive labels to each category.
* Category titles enhance the understanding and interpretation of the data and enable effective data analysis.

Subcategory DataFrame

* The Subcategory DataFrame is an essential component of the ETL Mini Project, contributing to efficient data organization and categorization.
* The purpose of the Subcategory DataFrame is to provide a structured representation of subcategories for enhanced data management and analysis.

DataFrame Structure:

* The Category DataFrame consists of two columns: **subcategory_id** and **subcategory.**
* The subcategory_id column utilizes sequential numbering to assign a unique identifier to each subcategory.
* The category column contains the titles or names of the subcategories.

Sequential Numbering

* The subcategory_id column employs sequential numbering to ensure distinct and consistent identification of each subcategory..
* Sequential numbering facilitates easy referencing, sorting, and analysis of the subcategories.

Subcategory Titles

* The subcategory column holds the titles or names of the subcategories, providing descriptive labels to each subcategory
* Subcategory titles enhance the understanding and categorization of the data, enabling effective data analysis.

subcategory_id	subcategory
subcat1	food trucks
subcat2	rock
subcat3	web
subcat4	plays
subcat5	documentary
subcat6	electric music
subcat7	drama
subcat8	indie rock
subcat9	wearables
subcat10	nonfiction
subcat11	animation
subcat12	video games
subcat13	shorts
subcat14	fiction
subcat15	photography books
subcat16	radio & podcasts
subcat17	metal
subcat18	jazz
subcat19	translations
subcat20	television
subcat21	mobile games
subcat22	world music
subcat23	science fiction
subcat24	audio

Campaign DataFrame

company_name	campaign_blurb
Baldwin, Riley and Jackson	Pre-emptive tertiary standardization
Odom Inc	Managed bottom-line architecture
Melton, Robinson and Fritz	Function-based leadingedge pricing structure
Mcdonald, Gonzalez and Ross	Vision-oriented fresh-thinking conglomeration
Larson-Little	Proactive foreground core
Harris Group	Open-source optimizing database
Ortiz, Coleman and Mitchell	Operative upward-trending algorithm
Carter-Guzman	Centralized cohesive challenge
Nunez-Richards	Exclusive attitude-oriented intranet
Rangel, Holt and Jones	Open-source fresh-thinking model
Green Ltd	Monitored empowering installation
Perez, Johnson and Gardner	Grass-roots zero administration system engine
Kim Ltd	Assimilated hybrid intranet
Walker, Taylor and Coleman	Multi-tiered directional open architecture
Rodriguez, Rose and Stewart	Cloned directional synergy
Wright, Hunt and Rowe	Extended eco-centric pricing structure
Hines Inc	Cross-platform systemic adapter
Cochran-Nguyen	Seamless 4thgeneration methodology
Johnson-Gould	Exclusive needs-based adapter
Perez-Hess	Down-sized cohesive archive
Reeves, Thompson and Richardson	Proactive composite alliance
Simmons-Reynolds	Re-engineered intangible definition
Collier Inc	Enhanced dynamic definition
Gray-Jenkins	Devolved next generation adapter
Scott, Wilson and Martin	Cross-platform intermediate frame
Caldwell, Velazquez and Wilson	Monitored impactful analyzer
Spencer-Bates	Optional responsive customer loyalty
Best, Carr and Williams	Diverse transitional migration

* The Campaign DataFrame is a key component of the ETL Mini Project, designed to store and analyze data related to crowdfunding campaigns.
* The purpose of the Campaign DataFrame is to capture and organize important campaign details for further analysis and insights.

Data Type Conversions and DateTime Formatting:

Data type conversions and datetime formatting are applied to specific columns:

* Specific columns undergo data type conversions and datetime formatting:
  * Numeric data types (integers or floats) are used for goal, pledged, and backers_count columns to facilitate calculations and analysis.
  * DateTime data type is applied to launch_date and end_date columns to store and manipulate date values accurately.
* Data type conversions ensure appropriate storage and handling of the campaign data.
* The campaign data undergoes necessary transformations to ensure consistent and accurate data representation.
* Proper data types and datetime formatting enable effective analysis and calculations on the campaign DataFrame.

campaign_launched_at	campaign_deadline
13/02/2020 06:00	01/03/2021 06:00
25/01/2021 06:00	25/05/2021 05:00
17/12/2020 06:00	30/12/2021 06:00
21/10/2021 05:00	17/01/2022 06:00
21/12/2020 06:00	23/08/2021 05:00
11/12/2020 06:00	29/08/2021 05:00
31/07/2020 05:00	11/05/2021 05:00
22/12/2020 06:00	21/09/2021 05:00
08/04/2020 05:00	10/03/2021 06:00
13/08/2021 05:00	31/08/2021 05:00
11/07/2020 05:00	02/08/2021 05:00
11/08/2020 05:00	26/06/2021 05:00
14/11/2020 06:00	09/04/2021 05:00
11/11/2020 06:00	06/11/2021 05:00
14/11/2021 06:00	02/01/2022 06:00
25/10/2021 05:00	16/12/2021 06:00
20/08/2021 05:00	22/12/2021 06:00
12/04/2020 05:00	24/03/2021 05:00
24/03/2021 05:00	27/01/2022 06:00
12/10/2021 05:00	17/11/2021 06:00
12/04/2020 05:00	09/04/2021 05:00
14/03/2021 06:00	29/04/2021 05:00
06/07/2020 05:00	04/08/2021 05:00
23/07/2021 05:00	12/09/2021 05:00
01/05/2021 05:00	01/10/2021 05:00
09/10/2020 05:00	23/05/2021 05:00
20/11/2020 06:00	27/07/2021 05:00
08/01/2021 06:00	15/10/2021 05:00

Contacts DataFrame

* The Contacts DataFrame is an essential component of the ETL Mini Project, designed to store and manage contact information.
* The purpose of the Contacts DataFrame is to organize and maintain contact details for efficient data processing and analysis.

DataFrame Structure:

Columns:

* **contact_id:** Unique identifier for each contact
* **first_name:** First name of the contact
* **last_name:** Last name of the contact
* **email:** Email address of the contact

Python Dictionary Methods or Regular Expressions:

* The Contacts DataFrame is created using either Python dictionary methods or regular expressions.
* Python dictionary methods can be employed to directly convert contact data into a DataFrame.
* Alternatively, regular expressions can be utilized to extract and structure contact information from raw data.

contact_id	first_name	last_name	email
4661	Cecilia	Velasco	cecilia.velasco@rodrigues.fr
3765	Mariana	Ellis	mariana.ellis@rossi.org
4187	Sofie	Woods	sofie.woods@riviere.com
4941	Jeanette	Iannotti	jeanette.iannotti@yahoo.com
2199	Samuel	Sorgatz	samuel.sorgatz@gmail.com
5650	Socorro	Luna	socorro.luna@hotmail.com
5889	Carolina	Murray	carolina.murray@knight.com
4842	Kayla	Moon	kayla.moon@yahoo.de
3280	Ariadna	Geisel	ariadna.geisel@rangel.com
5468	Danielle	Ladeck	danielle.ladeck@scalfaro.net
3064	Tatiana	Thompson	tatiana.thompson@hunt.net
4904	Caleb	Benavides	caleb.benavides@rubio.com
1299	Sandra	Hardy	sandra.hardy@web.de
5602	Lotti	Morris	lotti.morris@yahoo.co.uk
5753	Reinhilde	White	reinhilde.white@voila.fr
4495	Kerry	Patel	kerry.patel@hutchinson.com
4269	Sophie	Antoine	sophie.antoine@andersen.com
2226	Martha	Girard	martha.girard@web.de

Contacts DataFrame

Python Dictionary Methods and Regular Expressions

* Python dictionary methods and regular expressions were utilized to create the Contacts DataFrame.

Python Dictionary Methods

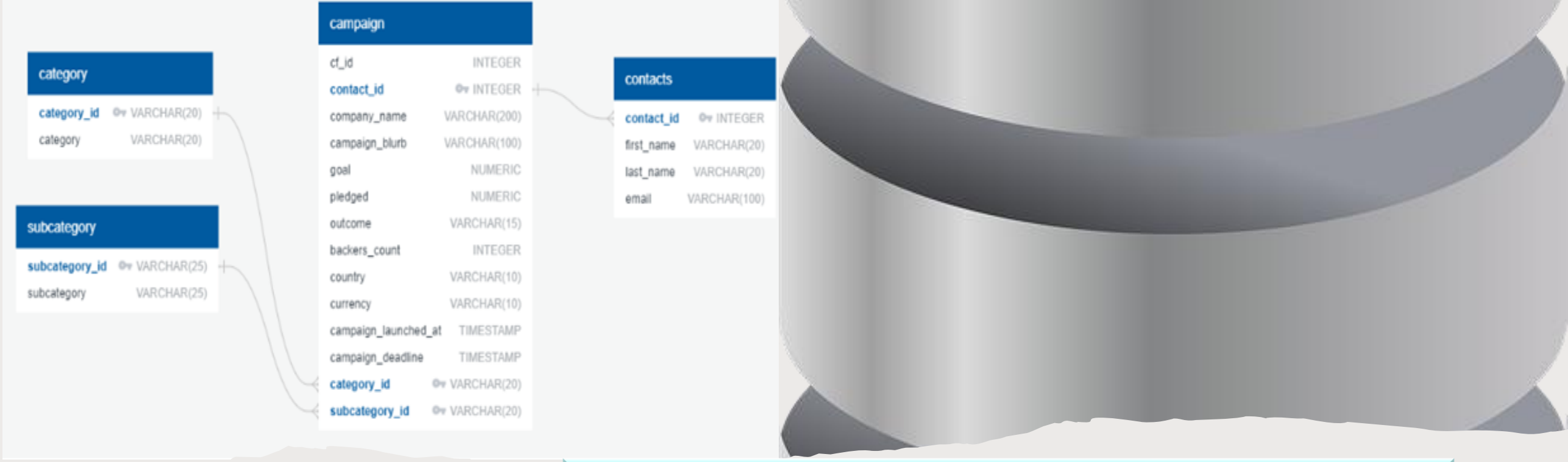
* Straightforward and intuitive approach for mapping contact attributes to their values.
* Simplifies the process of creating and populating the DataFrame.
* Provides simplicity and ease of use in organizing contact information.

Regular Expressions

* Powerful and flexible method for extracting and organizing contact details from unstructured data.
* Allows for defining patterns and searching for specific patterns within text data.
* Useful for handling complex or varied contact information formats.

Advantages

* **Simplicity:** Python dictionary methods offer a straightforward syntax for creating the DataFrame, while regular expressions simplify the extraction of specific patterns.
* **Efficiency:** Both methods are efficient in terms of processing time and computational resources.
* **Flexibility:** Python dictionary methods and regular expressions accommodate various data formats and structures.



Crowdfunding Database (ERD)

Entity-Relationship Diagram (ERD):

- The ERD -display represents the relationships between the tables in the crowdfunding database.
- The ERD provides a visual overview of how the tables are connected and their associations.

Table Schema Creation Process:

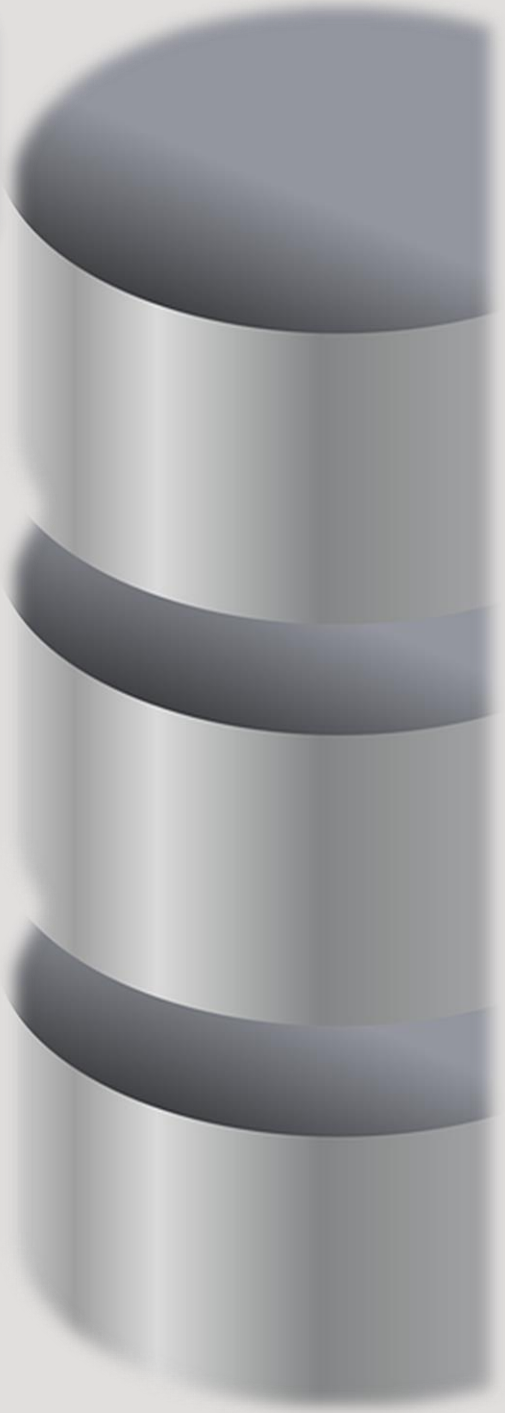
- **Steps involved in creating the table schema based on the ERD:**
  - Identify the entities and their attributes from the ERD.
  - Determine the data types and constraints for each attribute.
  - Define the primary keys, foreign keys, and relationships between the tables.
  - Establish indexes for efficient data retrieval.
  - Ensure data integrity through the use of constraints.

Crowdfunding Database PostgreSQL

```sql
-- category---
CREATE TABLE category (
    category_id VARCHAR(20) NOT NULL,
                category VARCHAR(20) NOT NULL,
                PRIMARY KEY (category_id)
);

-- subcategory---
CREATE TABLE subcategory (
    subcategory_id VARCHAR(25) NOT NULL,
                subcategory VARCHAR(25) NOT NULL,
                PRIMARY KEY (subcategory_id)
);

-- campaign---
CREATE TABLE campaign (
    cf_id INTEGER NOT NULL,
    contact_id INTEGER NOT NULL,
    company_name VARCHAR(200) NOT NULL,
    campaign_blurb VARCHAR(100) NOT NULL,
    goal NUMERIC NOT NULL,
    pledged NUMERIC NOT NULL,
    outcome VARCHAR(15) NOT NULL,
    backers_count INTEGER NOT NULL,
    country VARCHAR(10) NOT NULL,
    currency VARCHAR(10) NOT NULL,
    campaign_launched_at TIMESTAMP NOT NULL,
    campaign_deadline TIMESTAMP NOT NULL,
    category_id VARCHAR(20) NOT NULL,
    subcategory_id VARCHAR(20) NOT NULL,
                PRIMARY KEY(contact_id, category_id, subcategory_id),
                FOREIGN KEY (category_id) REFERENCES category(category_id),
                FOREIGN KEY (subcategory_id) REFERENCES subcategory(subcategory_id),
                CONSTRAINT campaign_contact_id_unique UNIQUE (contact_id)
);

-- contacts--
CREATE TABLE contacts (
    contact_id INTEGER NOT NULL,
    first_name VARCHAR(20) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    email VARCHAR(100) NOT NULL,
                PRIMARY KEY(contact_id),
                FOREIGN KEY (contact_id) REFERENCES campaign(contact_id)
);

----- View Tables------
SELECT * FROM category;
SELECT * FROM subcategory;
SELECT * FROM campaign;
SELECT * FROM contacts;
```

- **Steps involved in creating the table schema based on the PostgreSQL:**

  - The crowdfunding_db_schema.sql file is executed in PostgreSQL to create the database schema.
  - The table creation statements from the schema file are executed, creating the corresponding tables.
  - Data import processes are performed to populate the tables with data from the extracted and transformed DataFrames.
  - Verification steps ensure the successful creation of the database, including data integrity checks and querying the tables to validate the imported data.



Key Achievements:


Q&A


NumPy


PostgreSQL


jupyter


QUICK
DBD


pandas


GitHub

References


[REGEX]*
Software Services- Learn To Create