**Methodology for Crowdfunding Database Project**

This project involves designing a database for managing crowdfunding campaigns and their associated data. The methodology outlines the steps taken to extract, transform, and load (ETL) the data from various Excel files into a relational database. It also includes the creation of dataframes, cleaning, and structuring the data into CSV files, which will then be used to create SQL tables. Here's the methodology in detail:

**1. Data Extraction and Transformation**

**1.1. Category DataFrame:**

* **Objective:** Extract and transform the crowdfunding Excel data to create a category DataFrame.
* **Process:**
  + Identify the unique categories present in the crowdfunding dataset.
  + Create a category\_id column, which contains sequential identifiers such as "cat1", "cat2", etc.
  + The category column will store the actual names of the categories.
* **Export:** The resulting DataFrame is exported to a CSV file (category.csv) for further processing.

**1.2. Subcategory DataFrame:**

* **Objective:** Extract and transform the data to create a subcategory DataFrame.
* **Process:**
  + Identify the unique subcategories in the dataset.
  + Create a subcategory\_id column with sequential identifiers (e.g., "subcat1", "subcat2", etc.).
  + The subcategory column will store the names of the subcategories.
* **Export:** The transformed data is exported to a CSV file (subcategory.csv).

**1.3. Campaign DataFrame:**

* **Objective:** Extract and transform the crowdfunding dataset to create a campaign DataFrame.
* **Process:**
  + Extract relevant columns from the original dataset such as cf\_id, contact\_id, company\_name, blurb, goal, pledged, outcome, backers\_count, country, currency, launched\_at, deadline, and others.
  + Rename certain columns to make them more readable (e.g., rename blurb to description, launched\_at to launch\_date).
  + Convert goal and pledged columns to the float data type.
  + Convert the launched\_at and deadline columns to a datetime format.
  + Add category\_id and subcategory\_id columns, linking them to the previously created category and subcategory DataFrames.
* **Export:** The transformed campaign data is saved as campaign.csv.

**2. Contacts DataFrame Creation and Processing**

**2.1. Import and Process Contact Data:**

* **Objective:** Import the contacts.xlsx file and transform it into a usable DataFrame.
* **Process:**
  + Iterate through the rows of the contact DataFrame and convert each row into a dictionary for easier processing.
  + Extract the dictionary values by using Python list comprehensions.
  + Create a new DataFrame containing the extracted data.
  + Split the name column into two new columns: first\_name and last\_name.
* **Export:** The cleaned and transformed data is saved as contacts.csv.

**3. Database Design and Schema Creation**

**3.1. Entity-Relationship Diagram (ERD):**

* **Objective:** Create an ERD to visually represent the relationships between the different tables.
* **Process:** Use tools like QuickDBD to sketch an ERD that represents the relationships among the four primary tables:
  + category
  + subcategory
  + campaign
  + contacts
* **Relationships:**
  + Each campaign is associated with one category and one subcategory.
  + Each campaign can have many contacts (backers).

**3.2. Schema Definition:**

* **Objective:** Define the schema for the tables based on the ERD.
* **Process:**
  + Specify data types for each column (e.g., VARCHAR, INTEGER, FLOAT, DATE, etc.).
  + Define primary keys and foreign keys.
  + Add constraints to ensure data integrity (e.g., NOT NULL).
* **Export:** The SQL schema is saved in a file named crowdfunding\_db\_schema.sql.

**4. Database Creation and Data Import**

**4.1. Create PostgreSQL Database:**

* **Objective:** Set up a PostgreSQL database to store the crowdfunding data.
* **Process:**
  + Create a new database named crowdfunding\_db using PostgreSQL.

**4.2. Create Tables:**

* **Objective:** Use the SQL schema to create the necessary tables.
* **Process:**
  + Execute the crowdfunding\_db\_schema.sql script to create tables in the correct order.
  + The table creation script ensures that foreign keys are handled appropriately, respecting referential integrity.

**4.3. Import CSV Files:**

* **Objective:** Load the CSV files into the corresponding SQL tables.
* **Process:**
  + Import the category.csv, subcategory.csv, campaign.csv, and contacts.csv files into their respective tables.
  + Use PostgreSQL's COPY command or a similar method for bulk data import.

**4.4. Verify Data Integrity:**

* **Objective:** Confirm that the tables have been correctly populated.
* **Process:**
  + Execute SELECT statements for each table to ensure that data has been correctly imported.
  + Review sample rows from each table to verify the data accuracy and consistency.

**5. Final Steps and Repository Update**

**5.1. Documentation:**

* **Objective:** Create a detailed README file documenting the methodology and steps taken throughout the project.
* **Process:** Describe the objectives, steps, and tools used in this project. Include explanations of the structure of each table, the relationships between them, and how the data was processed.

**5.2. GitHub Repository:**

* **Objective:** Commit all relevant files to a GitHub repository.
* **Process:**
  + Save the following files to GitHub:
    - category.csv
    - subcategory.csv
    - campaign.csv
    - contacts.csv
    - crowdfunding\_db\_schema.sql
    - README.md

This methodology ensures a structured approach to the design, transformation, and population of the crowdfunding database while maintaining data integrity and making the database easy to use and expand upon in future applications.