Fake Data Generation and Database Population Script

# Overview

This Python script generates synthetic data using the Faker library and creates corresponding CSV files for various tables in a relational database. It then uses SQLAlchemy to create an SQLite database and imports the generated CSV data into the appropriate tables. The script also includes a section to print the first row of each table in the newly created database.

# Dependencies

- pandas: A data manipulation library used for handling data in tabular form.

- Faker: A library for generating fake data such as names, addresses, and other information.

- random: A built-in Python module for generating random numbers.

- string: A built-in Python module for working with string data.

- sqlalchemy: A SQL toolkit and Object-Relational Mapping (ORM) library for Python.

# Fake Data Generation Functions

- `generate\_username`: Generates a random username with a length of 8 characters, consisting of letters and digits.

- `generate\_random\_id`: Generates a random ID with a length of 10 characters, consisting of letters and digits.

- `generate\_wait\_times`: Generates a random time interval representing wait times.

- `generate\_category`: Generates a random category from a predefined list.

- `generate\_product\_category`: Generates a random product category from a predefined list.

- `generate\_password`: Generates a random password with a length of 15 characters, consisting of letters, digits, and punctuation.

- `generate\_fake\_price`: Generates a random fake price with a specified format.

# CSV File Generation

* The script generates several CSV files, each representing a table in the database:

1. Friends.csv: Represents friendship data with columns 'FriendshipID', 'Status', 'Username', 'TargetedUsername', and 'StartTime'.
2. Post.csv: Represents post data with columns 'PostID', 'Username', 'Comments', 'Description', 'PostCreatedAt', and 'PostLikes'.
3. UserProfile.csv: Represents user profile data with columns like 'Username', 'FName', 'LName', 'Email', 'Phone\_number', 'Address', 'DOB', 'PermLevel', 'Password', 'Gender', and 'Biography'.
4. Product.csv: Represents product data with columns 'ProductID', 'ProductName', 'ProdCategory', 'PoductDescription', and 'Price'.
5. ProductReview.csv: Represents product review data with columns 'ProductReviewID', 'Username', 'Comments', 'PRDescription', 'PRCreatedAt', 'PicorVidID', and 'PRLikes'.
6. Venue.csv: Represents venue data with columns 'VenName', 'Ven\_Phone\_number', 'Location', and 'Category'.
7. VenueReview.csv: Represents venue review data with columns 'VenReviewID', 'Username', 'ReviewText', 'PRDescription', 'VRCreatedAt', 'PicorVidID', 'Likes', 'Rating', 'Ven\_Name', and 'WaitTimeReported'.
8. UserTimeLine.csv: Represents user timeline data with columns 'UserTimeLineID', 'PostID', 'VenReviewID', 'ProductReviewID', and 'Username'.

# Database Creation and Table Population

* The script uses SQLAlchemy to create an SQLite database named 'Bool.db' and defines tables for each CSV file. Table columns are defined based on the data generated for the corresponding CSV file.

# Database Population

* The script reads each CSV file and imports its data into the corresponding table in the SQLite database. If a table already exists, it replaces the existing data.

# Print First Rows of Tables

* The script connects to the SQLite database and prints the first row of each table to verify that data has been successfully imported. This is to quickly ensure the script is generating data.