Assignment 1 Part C - SE 02 Big Data & Analytics B

Presented by: Kabuniang Buhawi Monjardin Ezra Runnath

Fatih Selim Salihoglu

Shukurolloh Abdulboqiyev

To:

Instructor Ali Vaisifard

Prerequisites:

The assignment is contained in a GitHub repository with the following link: https://github.com/Dowiw/UE library.git

This allows for more thorough team collaboration and version control of the assignment. It also allowed access throughout systems as an open repository.

To run:

- One must have **Git** to clone the repository.
- Also have **postgres** and **python3** to have the python script installed into your server and database.
- One must consider postgres user, postgres password and postgres initial database before running the python code.
- Recommended but not required: one must try to have **pgAdmin 4** to have a **better visualization of the data** and **querying into it.**

Part 1 - Creation Script

The creation script code is housed in the builder python file. This has initial variables that are defined by the user to access the currently available database and postgres user.

```
builder.py > ...
    import csv
    import psycopg2
    from psycopg2 import sql

# Configuration
    HOST = "localhost"
    USER = "postgres"
    PASSWORD = "somerandompassword"
    PORT = "5432"
    initial_db = "postgres" # Initial Database
    new_db_name = "library" # Database to create

# Connect to initial database first ('postgres')
print("Connecting to initial database...")
connection = psycopg2.connect(
    initial_db,
    initial_database...")

    initial_db,
    i
```

The user must input the **correct host**, **user**, **password**, **port** and **initial_db** to ensure that no errors will occur.

Next, the script creates a new database called "library" from new_db_name.

Next, tables are created using the execution method within the sql import which executes a query based on the string passed.

```
# CREATE ALL TABLES

print("Creating tables...:")

# "ID Type table

print("Creating id_type_code table...")

cursor.execute("""

CREATE TABLE id_type_code (

CREATE TABLE id_type_code (

TABLE TABLE TEXT NOT NULL

TABLE TEXT NOT NULL

TABLE TEXT NOT NULL

TABLE TABLE TABLE TEXT NOT NULL

TABLE T
```

Then, a query to fill up the tables is executed.

These are initial values that are tables with no foreign keys.

The code above uses open to read through the csv files and execute them as strings.

Assignment 1 Part C - SE 02 Big Data & Analytics B

```
314 | ....| ....| ....)
315
316 print("Done filling tables!")
317 print("Builder script done! Use the library database for queries!")
```

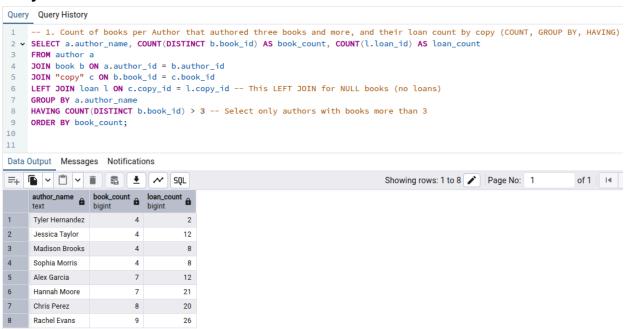
Finally, the finishing statement is printed out showing successful running.

Part 2 - Sample Data

The sample data is clonable within the GitHub repository inside the directory "csv_data". https://github.com/Dowiw/UE_library.git

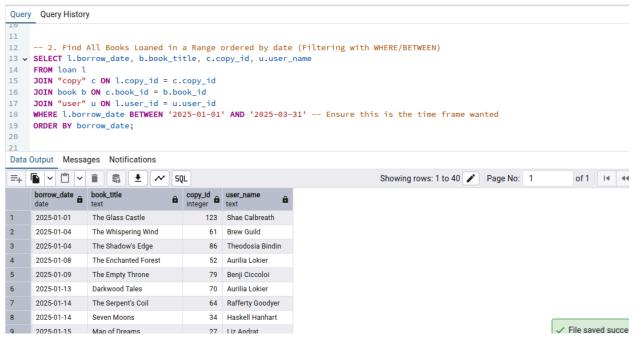
Part 3 - SQL Queries and Results

Query 1:



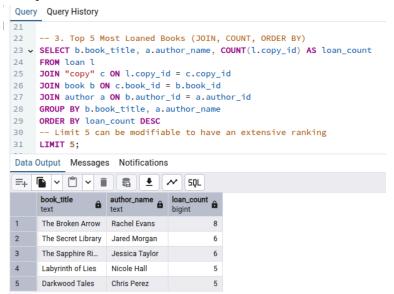
The query shows the number of books by an author along with the number of loans of their copies which is cut down to those with 3 or more books in the library. This is important in our business context because **we want to be aware of the most constant contributors.** If an author is active and has books that are frequent within the loaning timeframe (January 2024 to May 2025), we may give **more focus to that author's works.**

Query 2:



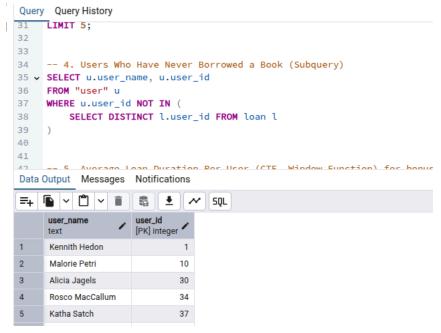
This query shows the loans that occurred within a timeframe. This is very important for our business context because it is showing a traceback of the data ensuring that when something occurs (i.e. book is lost in library but returned in data) an investigation can occur.

Query 3:



This query shows the top 5 most loaned books along with author name. This is relevant to our business context because we want to know which books are the most popular throughout the business. Ensuring that these books are served to customers now and beyond.

Query 4:



This query is simple but powerful, it shows the users that have never borrowed a book at all. The data is **very important for the business** to assure that **data processing (which is quite costly) is reduced and a reach-out to inactive users** is implemented.

Query 5:

```
avg_loan_days
-- 5. Average Loan Duration Per User (CTE, Window Function) for bonus :- )
                                                                                              user_name
                                                                                              text
WITH loan_duration AS (
                                                                                         1
                                                                                              Adi Sacher
                                                                                                                        22.00
  SELECT
                                                                                        2
                                                                                              Madeleine Giacomi
                                                                                                                        21.00
    l.user_id,
    (r.return_date - l.borrow_date) AS duration
                                                                                        3
                                                                                              Anetta Tuvey
                                                                                                                        16.00
  FROM loan l
                                                                                        4
                                                                                              Fzra Sheber
                                                                                                                        15.50
  JOIN "return" r ON l.loan_id = r.loan_id
                                                                                         5
                                                                                              Marti Ablott
                                                                                                                        15.50
  WHERE r.return_date IS NOT NULL
                                                                                         6
                                                                                              Theodosia Bindin
                                                                                                                        15.00
                                                                                              Omar Songist
                                                                                                                        14.75
SELECT
 u.user_name,
                                                                                              Amos Nenci
                                                                                                                        14.67
 ROUND(AVG(l.duration), 2) AS avg_loan_days
                                                                                         9
                                                                                              Brigg Surtees
                                                                                                                        14.00
FROM "user" u
                                                                                         10
                                                                                              Gaynor McCamish
                                                                                                                        14.00
JOIN loan_duration l ON u.user_id = l.user_id
                                                                                                                        14.00
                                                                                         11
                                                                                              Dwight Delong
GROUP BY u.user_name
                                                                                              Jaquith Garbert
                                                                                                                        14.00
ORDER BY avg_loan_days DESC;
```

This query shows the average loan duration of a user.

This is important to our business context so that we can determine which users have the tendency to return books very late (either on time or not). This is also helpful to determine which users hog a book for a long time on average.

Assignment 1 Part C - SE 02 Big Data & Analytics B

Part 4 - Challenges

The most challenging part was **generating relevant data**. The **developer (KB)** had to specify terms within mockaroo in Rust (a language that he never learned), other than that, data schemas for the csv files were also checked with an **LLM** to diversify results which included:

- Having users that have not borrowed or had overdue books
- Matching staff shift time to loan time and return time
- Having books currently borrowed (recent time and labelled as borrowed in copies table)
- Having a distribution of random durations between loan date and return date for data diversification

And all the little details to ensure that the gueries produce results that are different.

Another challenge was **querying**. So many syntax errors had to occur before releasing a final product.

The more "easier" part of the assignment was the python builder which the developer just referred to slides given in the teams. The logic for creating, filling and ensuring correctness is following this because **the developer is not a python coder.** However, the results were smooth and the builder is functioning.

To sum it up, the assignment was a fun one. This just goes to show how much effort is required to create mock data to test databases, thinking hard about working queries and ensuring that these are relevant to a certain topic.