2022 Challenge #3 Anonymize Data

Approach

The task involves anonymizing sensitive data from an SQLite database before passing it to a third-party provider for statistical analysis. We designed a Python script to connect to the SQLite database, extract the required data from three related tables, anonymize the data, and write the output to a CSV file. The following steps outline the approach:

1. Database Connection

The SQLite database is accessed via Python's sqlite3 module.

A function (db_connect) is written to connect to the database file, which is passed as an argument to the script from the command line.

The function checks if the connection is successful and returns a database cursor to execute queries.

2. SQL Query for Data Extraction

A SQL query was written to extract the required columns from the three related tables (t_customer, t_location, t_customer_status).

The query performs joins between these tables using foreign keys to pull the gender, lastname, birthdate, zip, and status.

Data anonymization is achieved at the SQL query level using string functions like SUBSTR to mask parts of the lastname, truncate the birthdate to only the year, and generalize the zip code precision.

3. Anonymization

Gender: Directly retrieved without modification.

Lastname: Masked by keeping the first two characters and replacing the rest with eight hyphens (-).

Birthyear: Extracted from the birthdate by selecting only the year.

ZIP Code: Precision reduced by keeping only the first two digits, replacing the last two with zeros.

Status: Retrieved as-is from the related table.

4. Writing Output to CSV

The write_to_csv function takes the query result and the column headers (extracted from the cursor's description) and writes it to a CSV file.

The result is written row by row after the header is inserted.

5. Tools and Methods Used

Python with sqlite3: To connect to the SQLite database and execute the SQL query.

SQL: For extracting and anonymizing data from the tables.

CSV Module: To write the anonymized data into a CSV file.

6. Commands and Scripts Used

Python was used to run the script that connected to the database:

```
python AnonymazeData.py database.db
```

The script reads the database, processes the query, and outputs the anonymized data in anonymized.csv.

The Python code and logic are stored in the script file AnonymazeData.py provided, which contains the core methods for connecting to the database, querying, and saving the output.

This approach ensures that sensitive data (like last names, birthdates, and zip codes) is anonymized (before being stored in a variable) according to the provided specifications before exporting the results to a CSV file.