***Створення зрізів (VIEWs). Оператор LOOP. Імпорт та експорт даних.***

***Мета роботи*:** *опрацювати додавання даних, оновлення даних та видалення даних з таблиць засобами мови SQL; розібрати створення циклів засобами мови PL/pgSQL; розібрати поняття зрізу бази даних (VIEW), створення зрізів засобами мови SQL; опрацювати імпорт даних з csv-файлів, експорт даних в csv- та json-формати.*

1. ***VIEWs.*** На основі запитів 6a – 6c частини 1 створити три зрізи бази даних та надати їм інтуїтивно зрозумілі імена. Використовуючи створені зрізи, побудувати графіки з частини 1. Оформити даний код як файл **main.py**.
2. ***Цикли.*** Використовуючи оператор LOOP, написати PL/pgSQL код, що заносить тестові дані в одну з таблиць, що були спроектовані у частині 1. Оформити даний код як файл **loop.sql**.
3. ***Імпорт даних.*** Написати код, що імпортує дані з csv-файлу, взятого з *kaggle* для частини 1, у таблиці PostgreSQL. НЕ обов’язково заповнювати всі поля таблиць, достатньо декілька. Розмір csv-файлу не повинен перевищувати 50 Мб. Оформити даний код як файл **kaggle\_import.py**.
4. Після імпорту даних перебудувати графіки з пункту 1. Зберегти одержані графіки як файл у форматі png.
5. ***Експорт даних в csv-файли.*** Написати код, що експортує усі дані з кожної таблиці у відповідні csv-файли. Оформити даний код як файл **export\_csv.py**.
6. ***Експорт даних в json-файл.*** Написати код, що експортує усі дані з усіх таблиць в єдиний json-файл. Оформити даний код як файл **export\_json.py**.
7. Створити проект на Github, в якому розташувати файли:

* *create.sql*
* *populate.sql при необхідності (для тих полів, які заповнюються власними даними, а не даними з csv.файлу )*
* *main.py*
* *loop.sql*
* *kaggle\_import.py*
* *new\_graphs.png*
* *export\_csv.py*
* *export\_json.py*