Лабораторна робота №3

**Створити REST сервіс для роботи з базою даних**

**(виконання CRUD операцій)**

Хід роботи

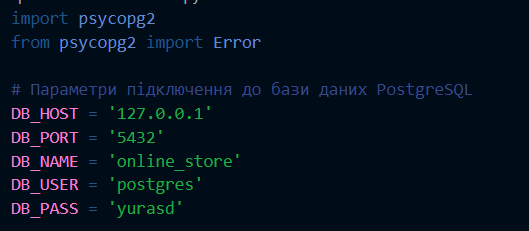
1. Завдання

* Створити REST сервіс для роботи з базою даних (виконання CRUD операцій), розробленої у попередніх лабораторних роботах.
* Використати технології (СУБД: PostgreSQL, мова програмування на вибір: Java | Python | C#). Для виконання CRUD операцій використовувати SQL (забороняється використовувати технологію ORM та подібні).
* Забезпечити використання процедур, функцій, тригерів та транзакцій.
* Надати інструкцію з використання сервісу в окремому файлі.
* Викласти результати виконання лабораторної роботи (файли з кодом та документацію)
* у репозиторій GitHub у папку lab3.

2. Виконання

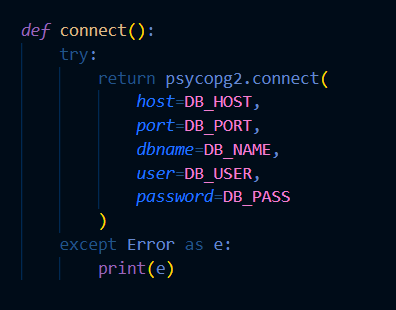
Створюємо REST сервіс для підключення до БД.

Налаштовуємо підключення:

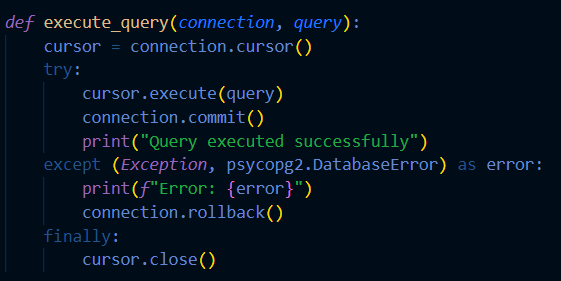


Тут я імпортував необхідні бібліотеки і зберіг параметри підключення до БД у змінних.

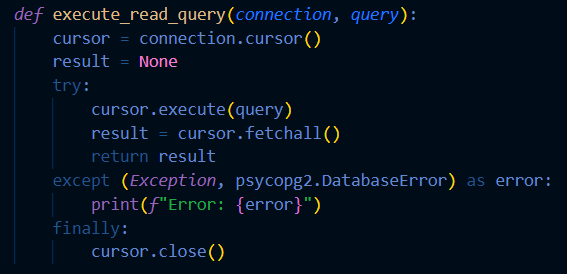
Підключаємося до БД з нашими даними:



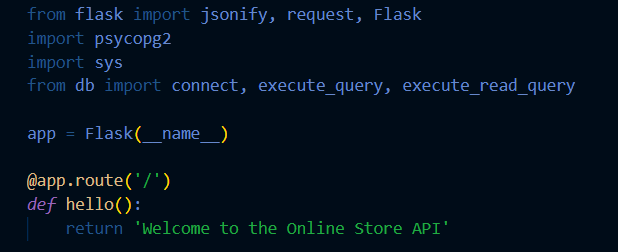
Функція для виконання запитів до БД:



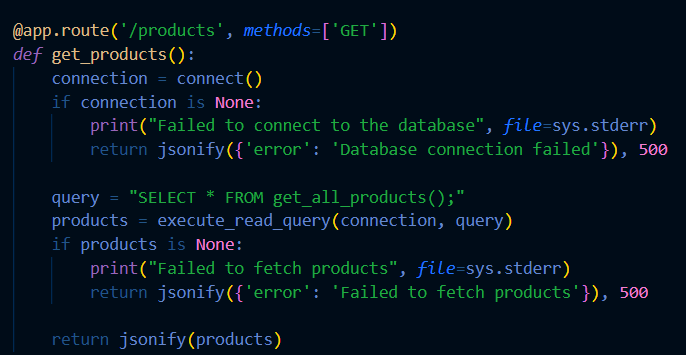
Функція для виконання зчитування:



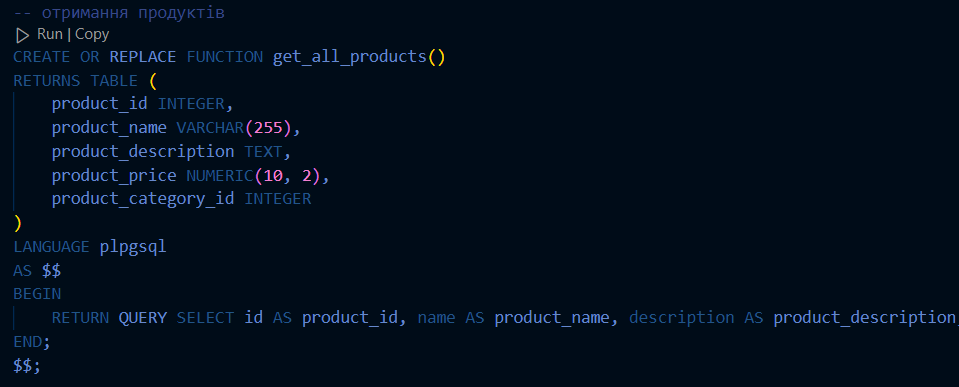
Далі файл routes.py

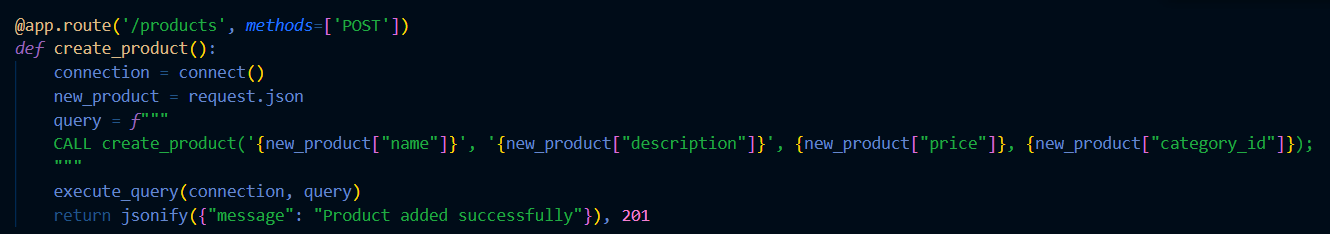


Import Flask для створення веб-додатка.

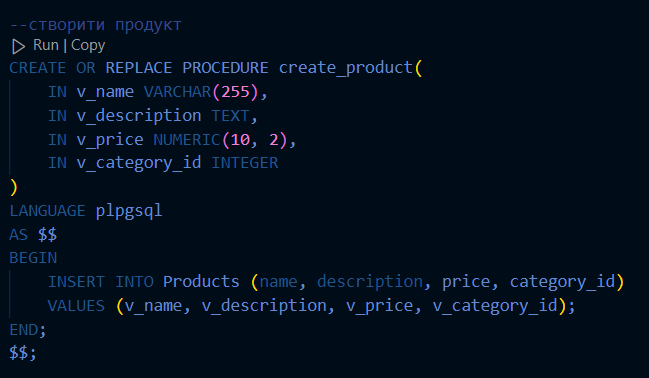


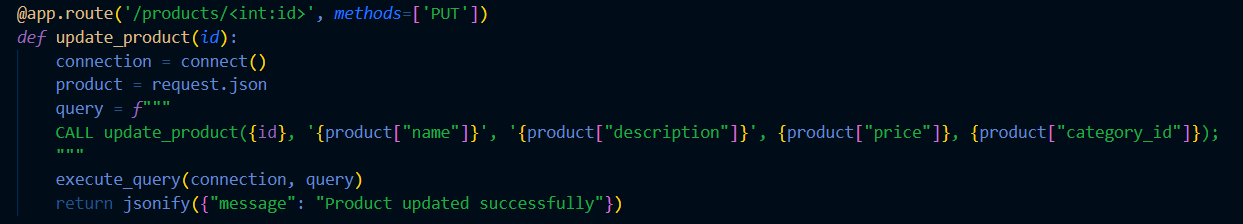
Налаштування роута для отримання всіх запитів з таблиці Products. Забезпечено логування у випадку помилок. Викликаємо функцію get\_all\_products():



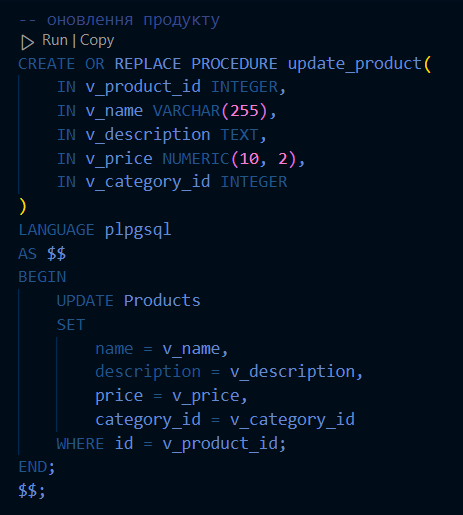


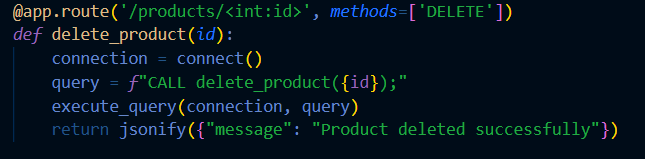
Виконання POST запиту для створення нового запису. Викликається процедура create\_product():

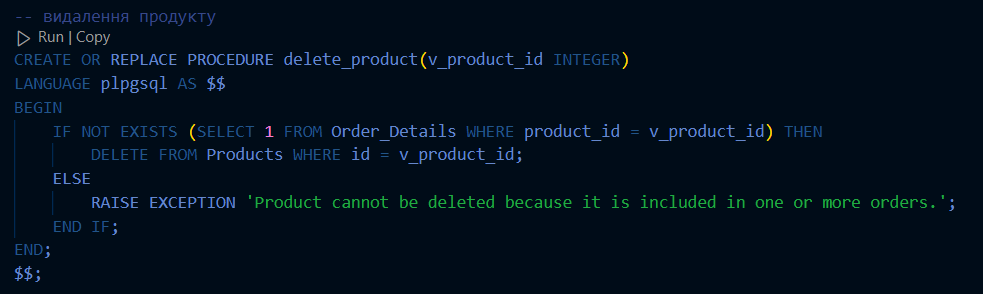




Виконання PUT запиту для оновлення у записі даних. Викликається процедура update\_product():





Роут для виконання DELETE запиту. Викликається процедура delete\_product()

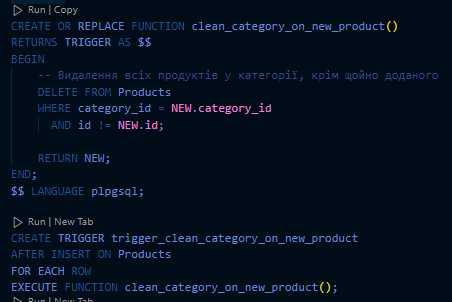


Для виконання транзакції. На SQL:

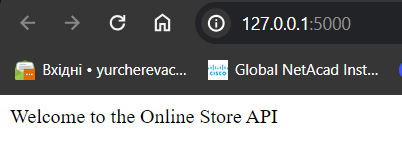


Тобто коли транзакція виконується, додається новий продукт та оновлюється запис в таблиці де id = 4

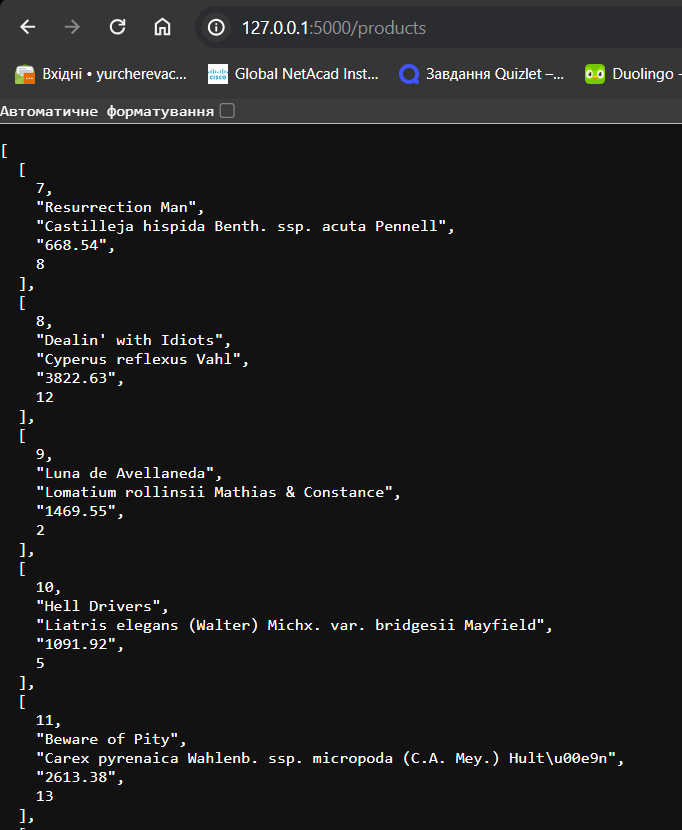
Тригер:



Повинен видаляти всі записи з продуктів, де категорія співпадає з додаваною категорією, окрім неї. Запускаю flask додаток і роблю запити:

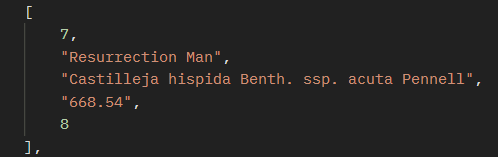


GET запит до таблиці Products:



Виводить. Далі Postman:

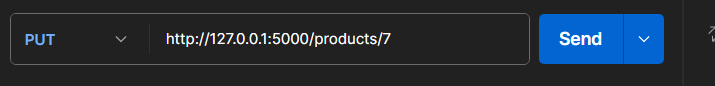
PUT запит. Оновимо дані тут:

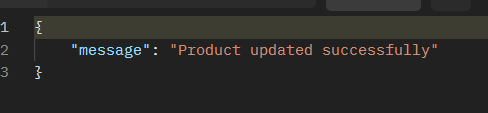


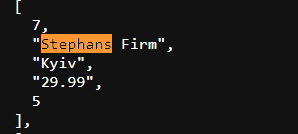
Ось так:



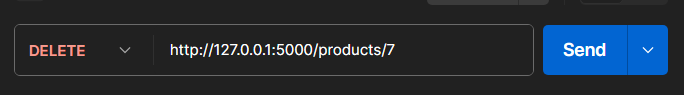
PUT-запит:





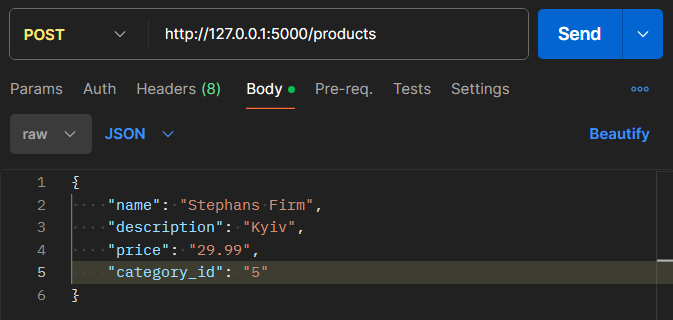


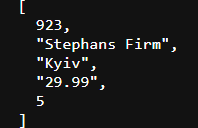
Запис оновлено. Видалимо:



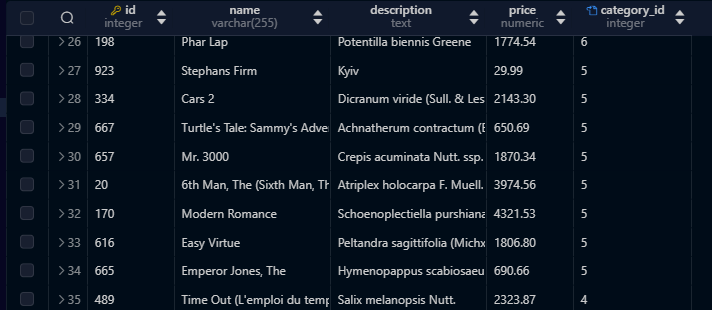
Видалено.

Останньою буде запит POST, додачі. Тому що має спрацювати тригер.

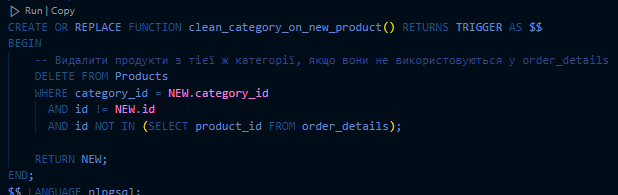




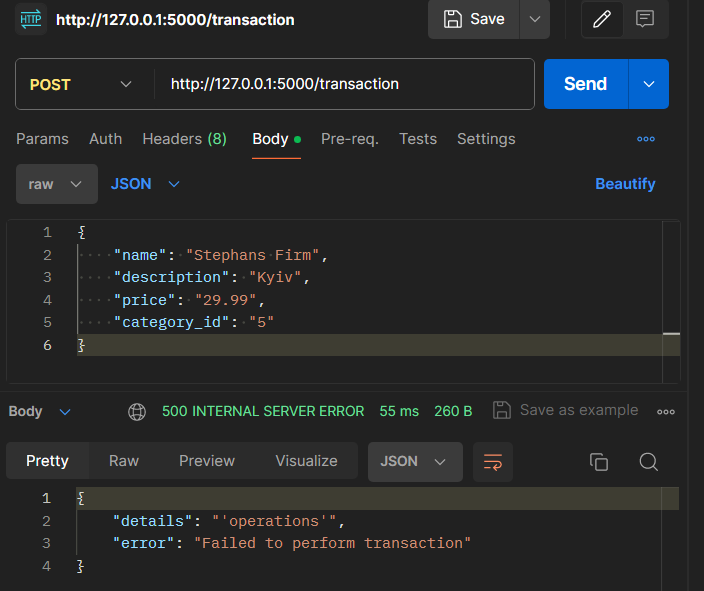
Запит виконався. Перевіряємо виконання тригеру:



Так, згідно тригера все виконалося, правда довелося внести певні корективи – не міг тригер видалити значення, що використовуються в інших таблицях:



Але загалом все правильно. Транзакція:



На жаль, виникли проблеми з виконанням транзакції.