МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

# Лабораторна робота № 3

з дисципліни “Бази даних”

тема “ Засоби оптимізації роботи СУБД PostgreSQL ”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) II курсу  групи КП-03  Винничук Богдан Андрійович (*прізвище, ім’я, по батькові*)  Варіант №14 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Радченко Костянтин Олександрович (*прізвище, ім’я, по батькові*) |

Київ 2021

**Мета роботи**

Метою роботи є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

**Постановка завдання**

Завдання роботи полягає у наступному:

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи №2

у вигляд об’єктно-реляційної проекції (ORM).

2. Створити та проаналізувати різні типи індексів у PostgreSQL.

3. Розробити тригер бази даних PostgreSQL.

Вимоги до пункту завдання №1

Для перетворення функцій, що реалізують запити до об’єктної бази даних,

необхідно встановити бібліотеку sqlAlchemy, налаштувати програму на роботу

з ORM, розробити класи-сутності для об’єктів-сутностей, представлених

відповідними таблицями БД та пов’язаних зв’язками 1:М, М:М та 1:1 виконати

опис схеми бази даних. Особливу увагу приділити контролю зовнішніх зв’язків

між таблицями засобами ORM.

Замінити виклики запитів мовою SQL на відповідні запити засобами

SQLAlchemy по роботі з об’єктами. Обов’язковим є реалізація вставки,

вилучення та редагування екземплярів класів-сутностей. Розробка запитів на

генерацію даних та пошук екземплярів класів-сутностей вітається, але не є

обов’язковою.

Інтерфейси функцій (вхідні та вихідні аргументи функцій модуля “Модель”)

мають залишитись без змін.

Вимоги до пункту завдання №2

Відповідно до варіанту індексування продемонструвати на прикладах запитів

SQL SELECT підвищення швидкодії їх виконання з використанням індексів, а

також пояснити чому для деяких випадків індексування використовувати

недоцільно. При цьому для наочного представлення слід використати функцію

генерування рандомізованих даних з лабораторної роботи No2, створивши

необхідну кількість тестових даних. Навести 4-5 прикладів запитів SELECT (із виведенням результуючих даних), що містять фільтрацію, агрегатні функції,

групування та сортування (у необхідних комбінаціях).

Корисні посилання: Hash, B-tree, GIN, BRIN.

Вимоги до пункту завдання №3

Створити тригер бази даних PostgreSQL відповідно до варіанта. Тригерна

функція має включати обробку запису, що модифікується (вставляється або

вилучається), умовні оператори, курсорні цикли та обробку виключних

ситуацій. Виконати відлагодження тригера при різних вхідних даних, навівши

2-3 приклади його використання.

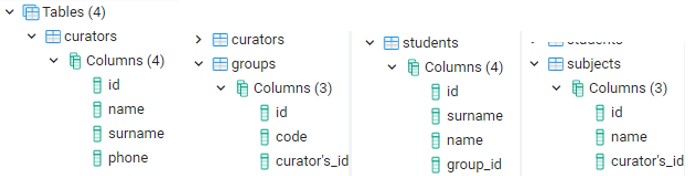
Варіант №14.

Види індексів: Btree, Hash

Умови для тригера: after insert, update

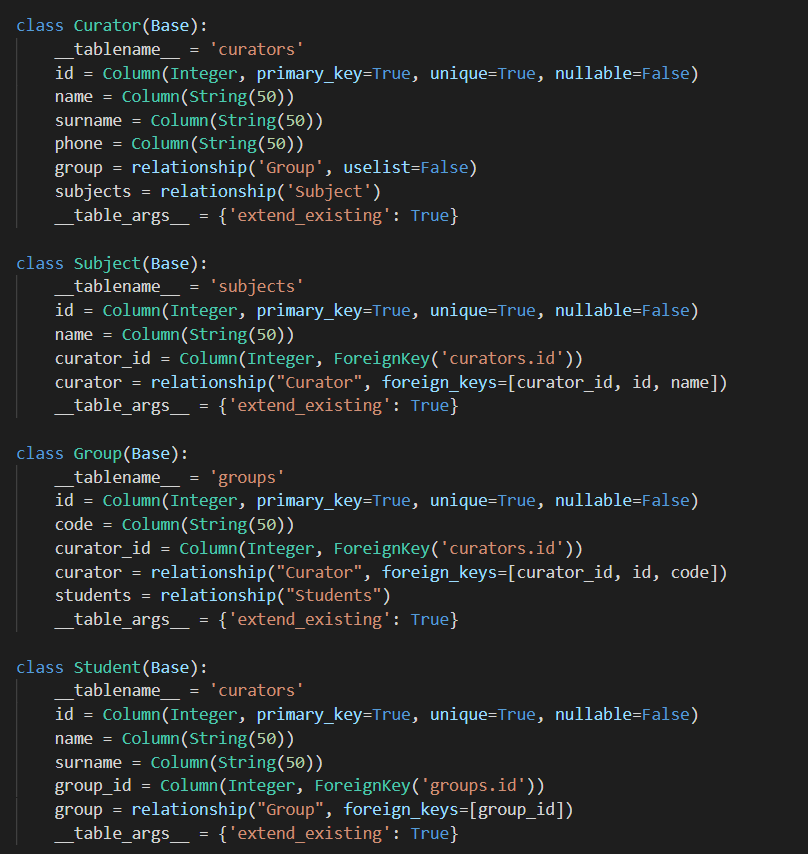
**Хід роботи:**

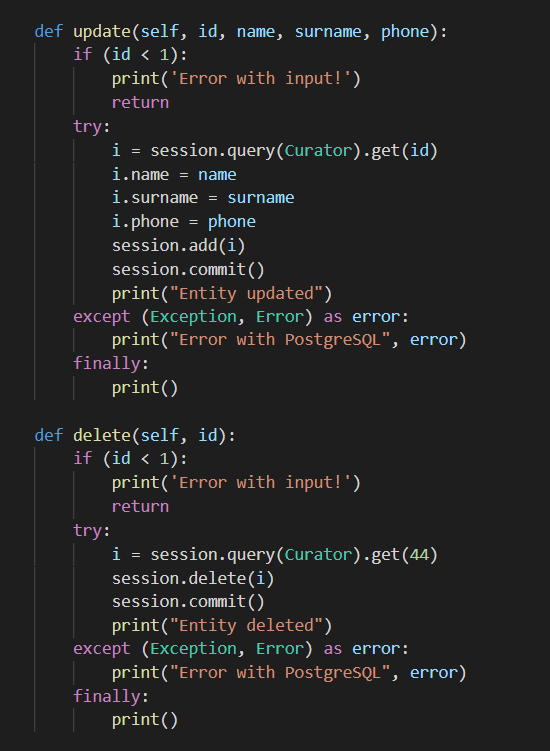
1. Таблиці бази даних





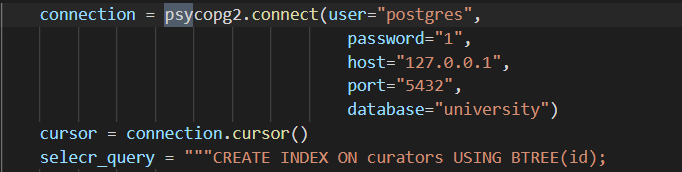
Класи ORM



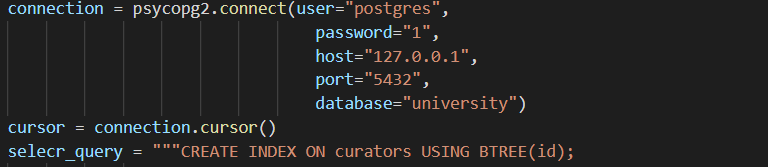
Приклади команд

1. Приклади індексування

Команди створення індексів:

BTree: 

B-tree – це збалансоване дерево пошуку з кількома гілками, тому воно оптимальне, якщо використовується для пошуку.

HASH: 

Хеш-функція використовується зокрема у структурах даних — хеш-таблицях, широко вживаних у програмному забезпеченні для швидкого пошуку даних. Хеш-функції використовуються для оптимізації таблиць та баз даних користуючись з того, що в однакових записів однакові значення хеш-функції. Такий підхід пошуку дублікатів ефективний у файлах великого розміру.

Приклади запитів і результатів:

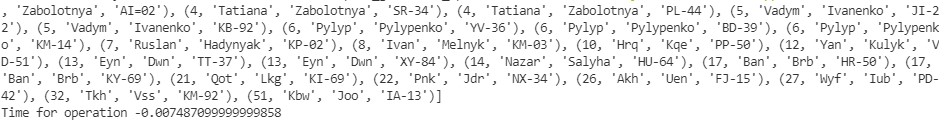
1.



2.

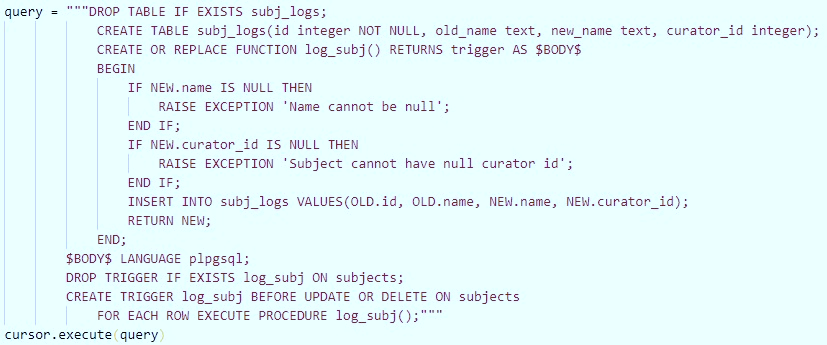


3.



1. Тригери:

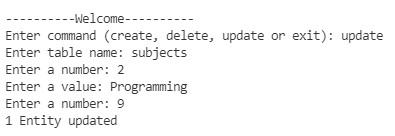
Команди створення тригера

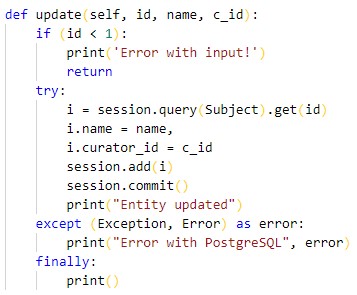




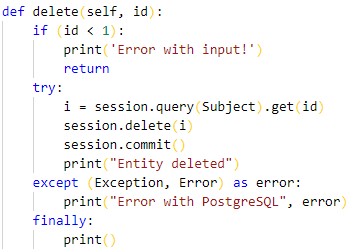
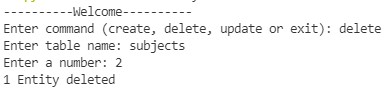
Приклади роботи:

1. Редагування:





1. Вилучення:



**Контрольні запитання:**

* 1. Сформулювати призначення та задачі об’єктно-реляційної проекції (ORM).

Призначенням є пов'язати бази даних з концепціями об'єктноорієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». Необхідно забезпечити роботу з даними в термінах класів, а не таблиць даних, і, навпаки, перетворити терміни та дані класів на дані, придатні для зберігання в СУБД. Необхідно також забезпечити інтерфейс для CRUD- операцій над даними. Загалом, необхідно позбавитися необхідності писати SQL-код для взаємодії в СУБД.

* 1. Проаналізувати основні види індексів у PostgreSQL (BTree, BRIN, GIN, Hash): призначення, сфера застосування, переваги та недоліки.

Незважаючи на всі відмінності між типами індексів (названими також методами доступу), зрештою будь-який з них встановлює відповідність між ключем (наприклад, значенням проіндексованого стовпця) та рядками таблиці, в яких цей ключ зустрічається. У PostgreSQL використовуються такі основні види індексів: BTree, BRIN, GIN, Hash. Рядки ідентифікуються за допомогою TID (tuple id), який складається з номера блоку файлу та позиції рядка всередині блоку. Тоді, знаючи ключ або деяку інформацію про нього, можна швидко прочитати ті рядки, в яких може знаходитися інформація, що цікавить нас, не переглядаючи всю таблицю повністю. Важливо розуміти, що індекс, прискорюючи доступ до даних, натомість потребує певних витрат на свою підтримку.

* 1. Пояснити призначення тригерів та функцій у базах даних.

Тригер запускається сервером автоматично при спробі зміни даних в таблиці, з якою він зв’язанний. Функції такого типу дозволяють реалізувати складнішу логіку.

**Висновки**

Під час виконання лабораторної роботи я здобула практичні навички використання засобів оптимізації СУБД PostgreSQL.