НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих комп’ютерних систем

**Лабораторна робота №3**

з дисципліни

**«Бази даних і засоби управління»**

**Тема**: «Засоби оптимізації роботи СУБД PostgreSQL»

Виконав:   
студент групи КВ-84

Сітар І  
Перевірив:  
Петрашенко А.В.

Київ 2020

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

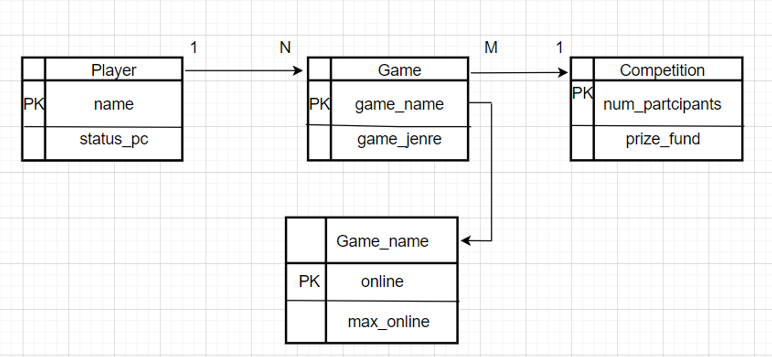
*Вимоги до пункту завдання №2*

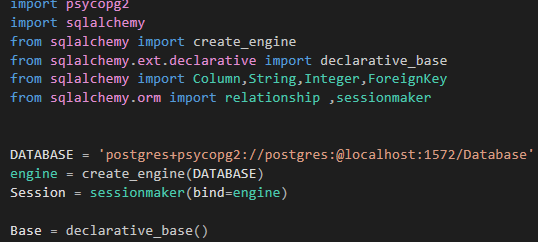
Відповідно до варіанту індексування продемонструвати на прикладах запитів SQL SELECT підвищення швидкодії їх виконання з використанням індексів, а також пояснити чому для деяких випадків індексування використовувати недоцільно. При цьому для наочного представлення слід використати функцію генерування рандомізованих даних з лабораторної роботи №2, 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).

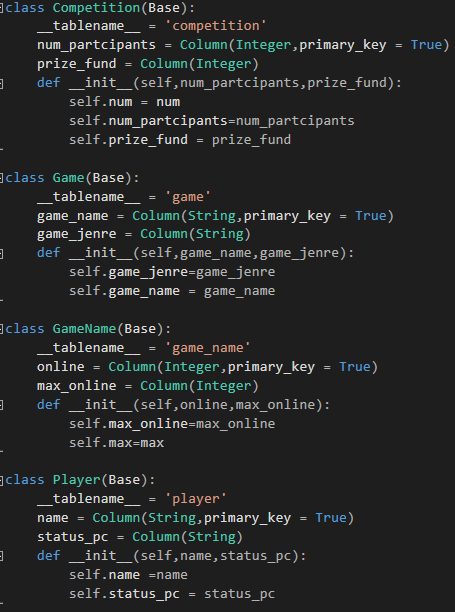
*Вимоги до пункту завдання №3*

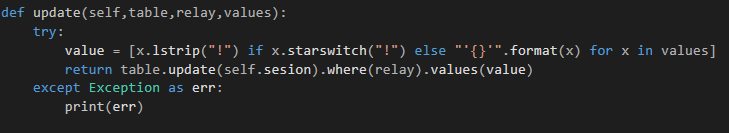
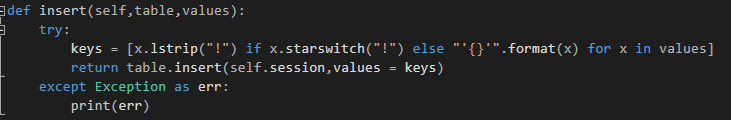
Створити тригер бази даних PostgreSQL відповідно до варіанта. Тригерна функція має включати обробку запису, що модифікується (вставляється або вилучається), умовні оператори, курсорні цикли та обробку виключних ситуацій. Виконати відлагодження тригера при різних вхідних даних, навівши 2-3 приклади його використання.

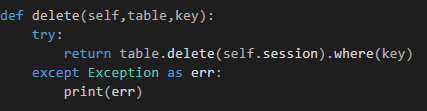




**Завдання 1** **Класи сутності**



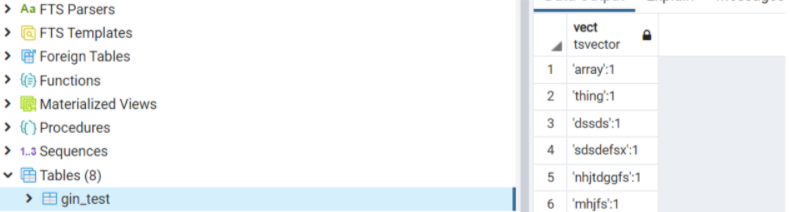
**Функції Insert,Delete,Update**



**Завдання 2**

GIN

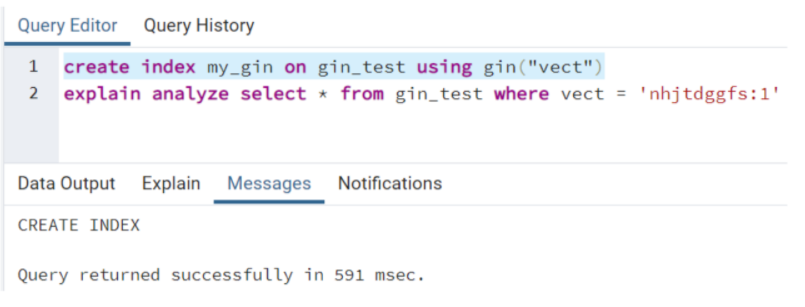
**Тестова таблиця**



Без індексу



З індексом

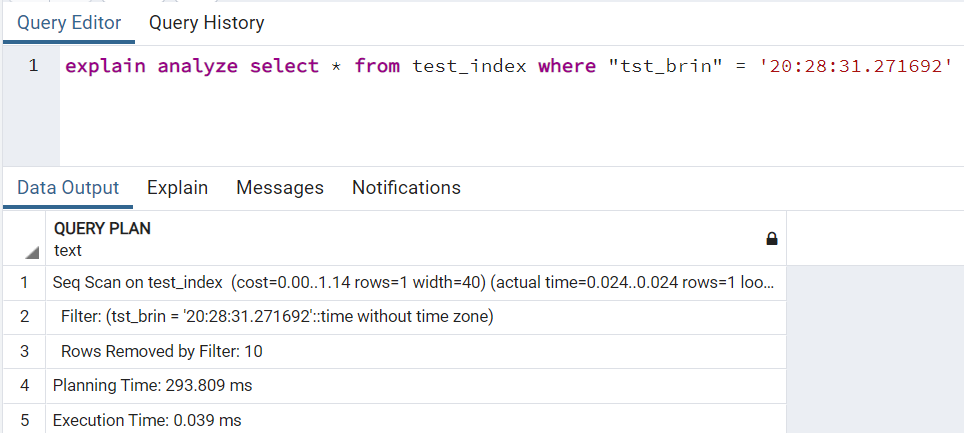




Індекс GIN покращив пошук,його основною задачею є прискорення повнотекстового пошуку.GIN добре підходить для даних,які не часто оновлюються.Для таблиць де зберігаються часто змінні дані не рекомендовано використовувати цей індекс,бо переіндексація може займати багато часу.

BRIN

Без індексу

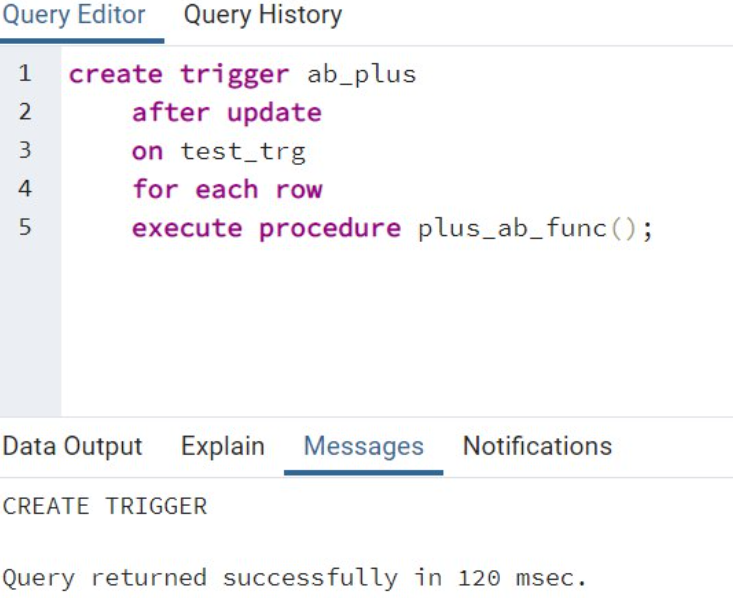


З індексом

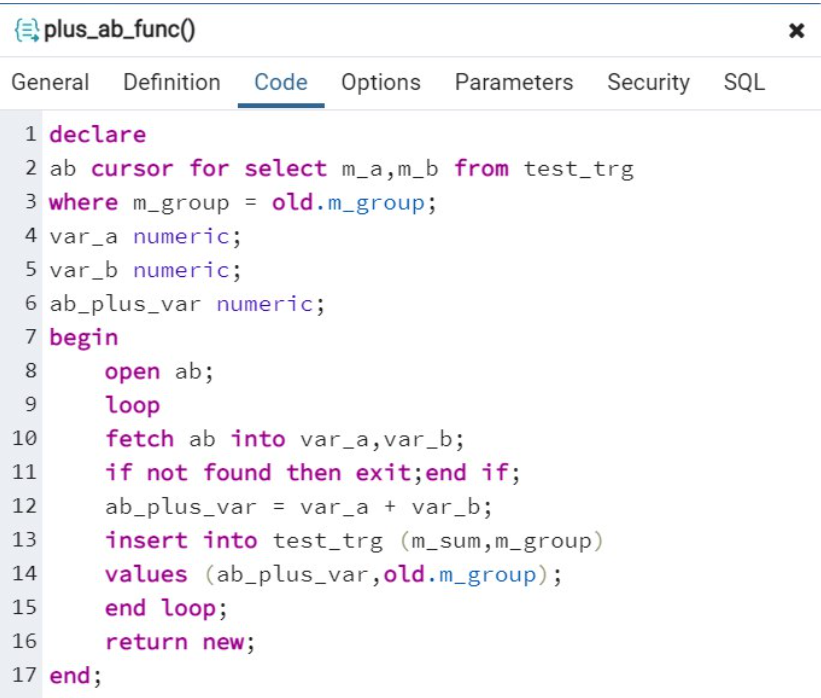


Спостерігаємо те , що застосування індексу BRIN пришвидшило пошук заданого значення.   
BRIN відмінно працює для стовпців, значення яких корелюють з їх фізичним розташуванням в таблиці.  
BRIN в основному призначений для таблиць великих і навіть величезних розмірів, які або взагалі не оновлюються, або оновлюються дуже незначно.

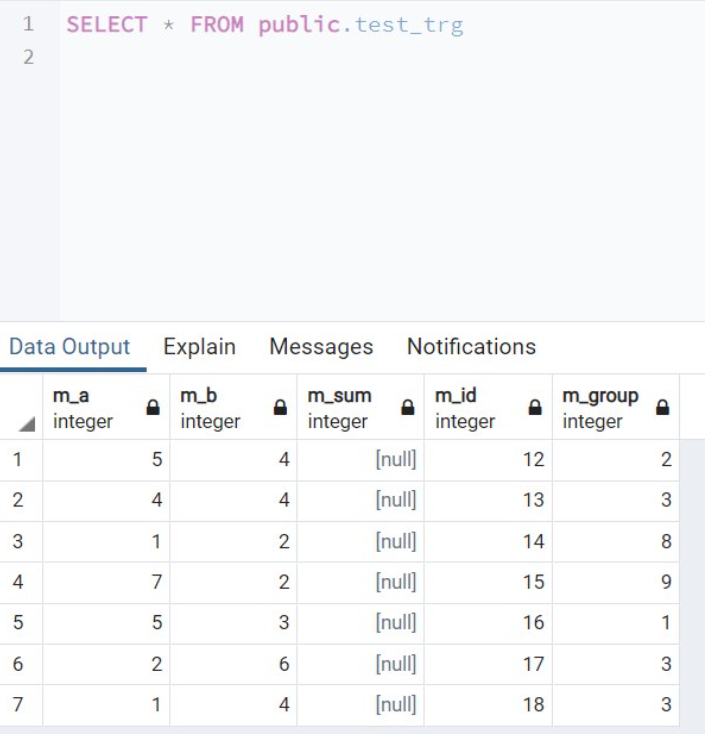
**Завдання 3**

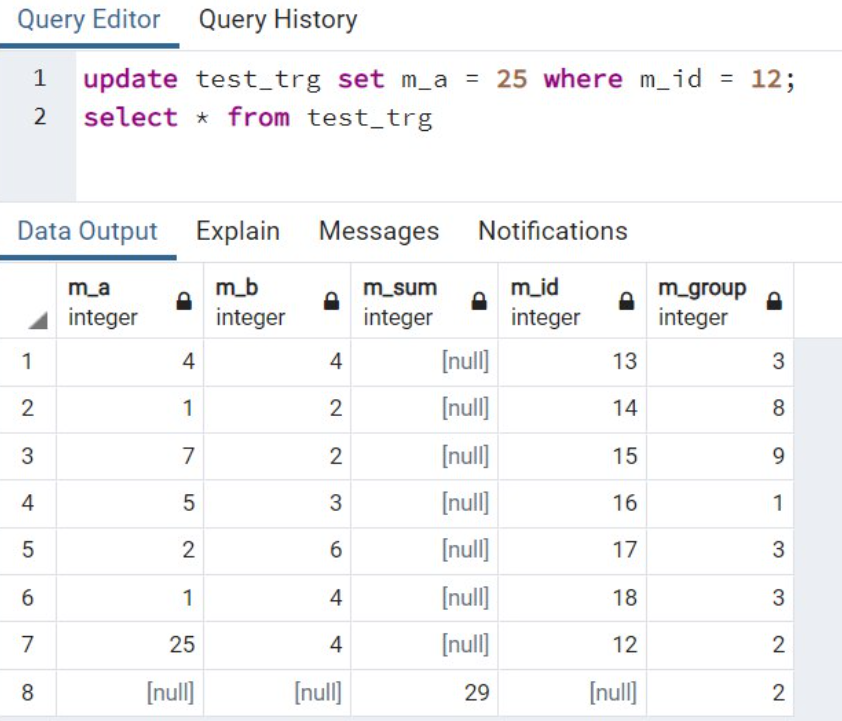


***Створення тригерної функції***



Перевірка роботи





У рядку 8 в стовбці m\_sum записана сума m\_a та m\_b.