НАЦІОНАЛЬНИЙ  ТЕХНІЧНИЙ   УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №3

із дисципліни «Бази даних»

на тему **«Засоби оптимізації роботи СУБД PostgreSQL****»**

Виконав:

студент 2 курсу ФПМ групи КП-92

Остапенко Іван Петрович

Прийняв:

|  |  |
| --- | --- |
|  | Бали |
| Якість виконання |  |
| Термін здачі |  |
| Сумарний бал |  |

КИЇВ  —    2020

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

*Завдання* роботи полягає у наступному:

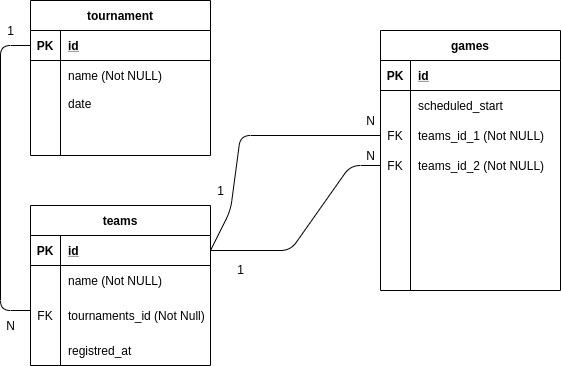
1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи №2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.

Посилання на репозиторій:

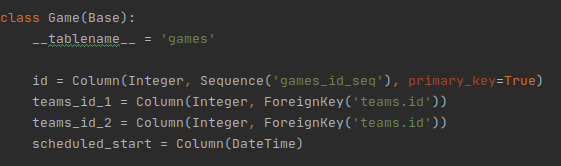
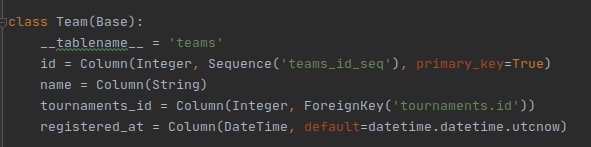
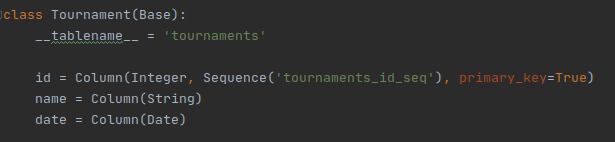
https://github.com/ostapenko-study/database-lab/tree/master/lab3

Завдання 1

Схема даних:



Класи ORM:

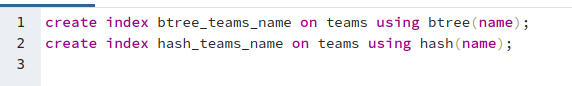


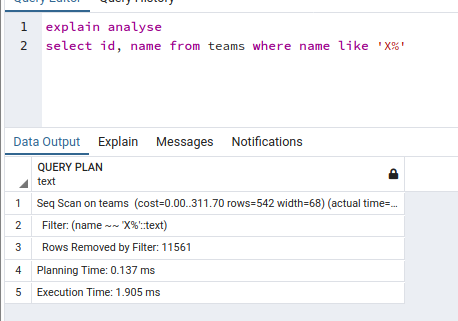
Інтерфейс модуля “Модель” не змінився, бо реалізований клас Storage не змінив свій інтерфейс. Замінені виклики SQL-запитів на відповідні запити засобами SQLAlchemy по роботі з об’єктами:

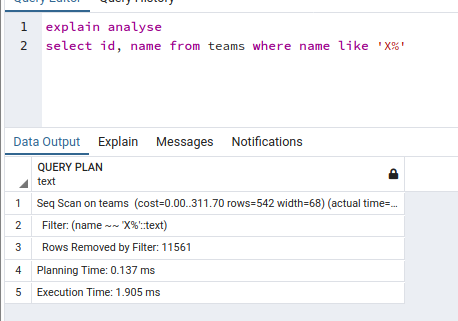
Рис. 1 - Частина файлу models.storage

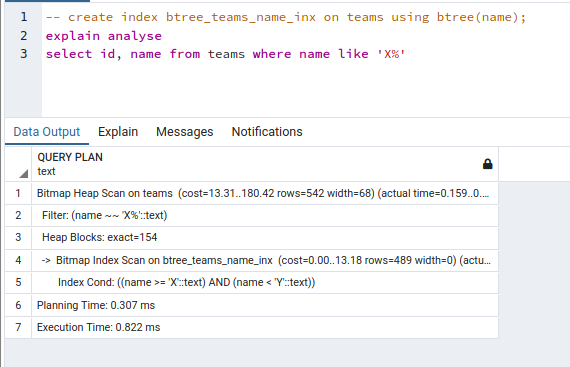
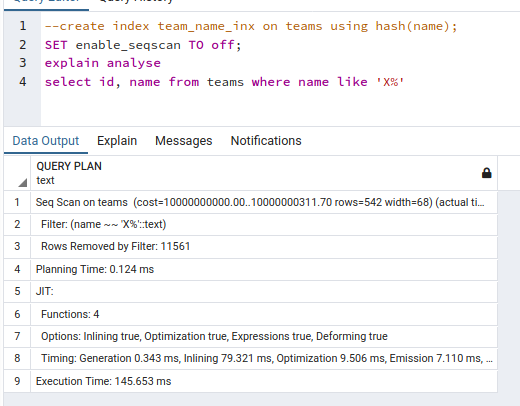
Завдання 2

Індекси:

Запит 1:

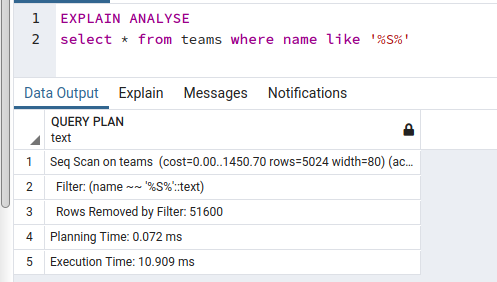
без використання hash та btree індексів:

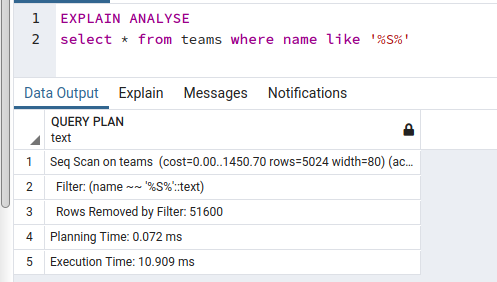
btree:

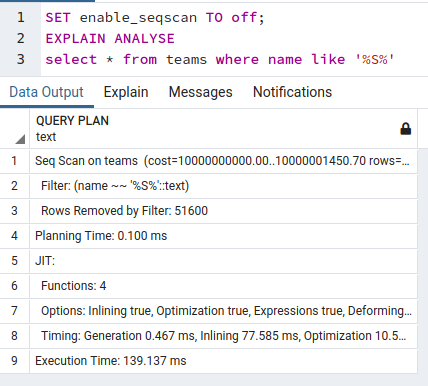
hash (опція SET enable\_seqscan TO off):

Висновок: при фільтруванні рядків за іменем, що починається на певну послідовність символів, ефективніше використовувати btree, а hash – не використовувати (даний hash-індекс не використовується для виконання даного запиту).

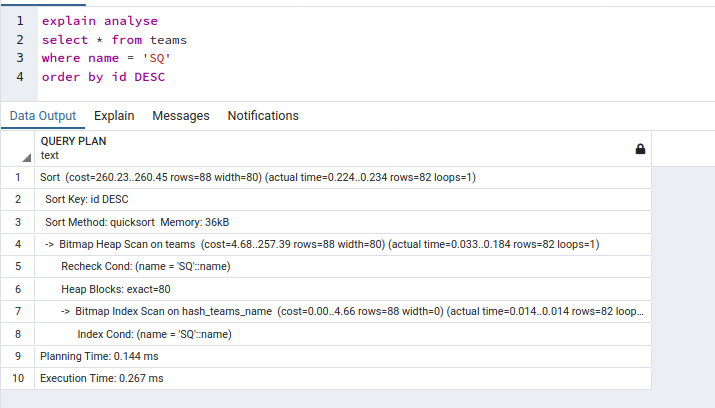
Запит 2:

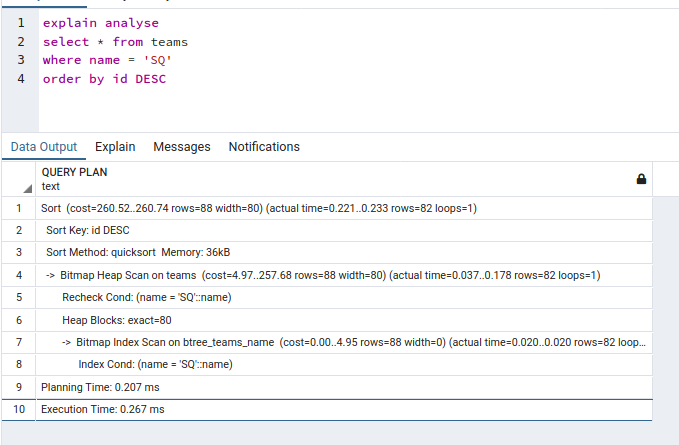
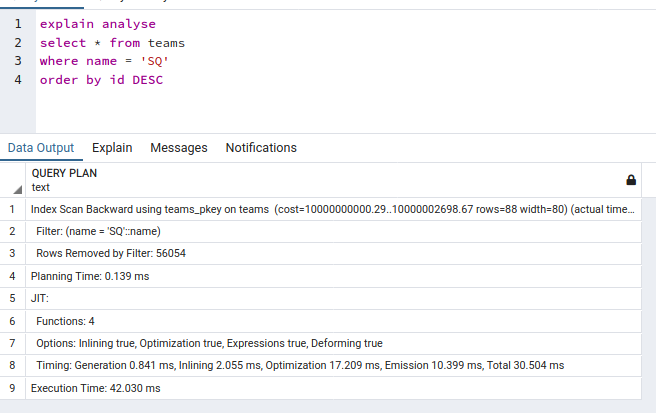
без використання hash та btree індексів:

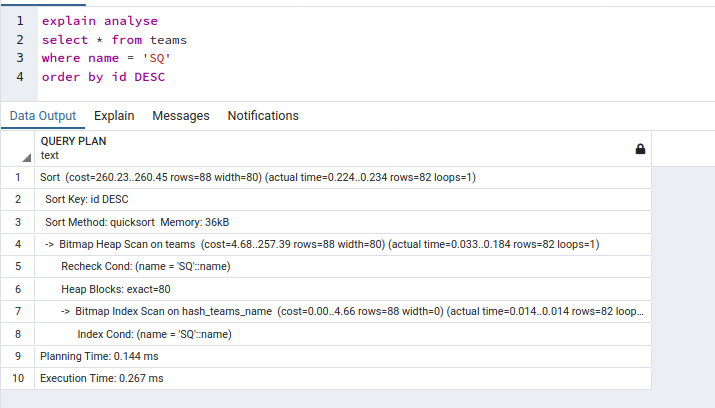
btree/hash:

Висновок: при фільтрувані запропоновані індекси btree та hash – не використовуються.

Запит 3:

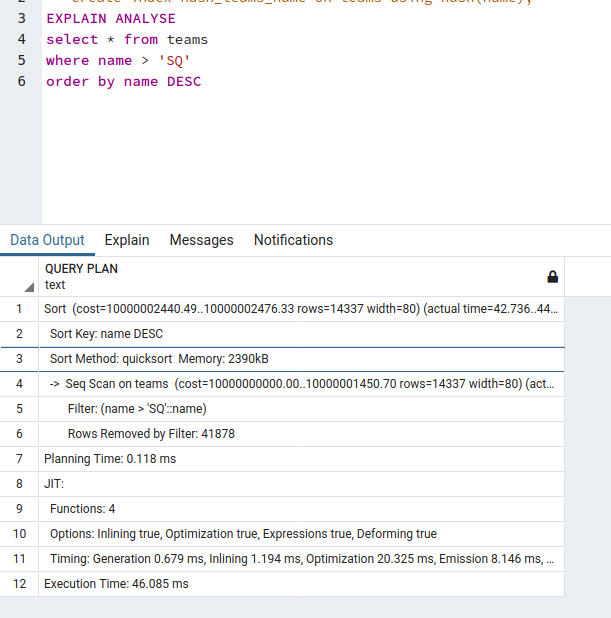
без використання hash та btree індексів:

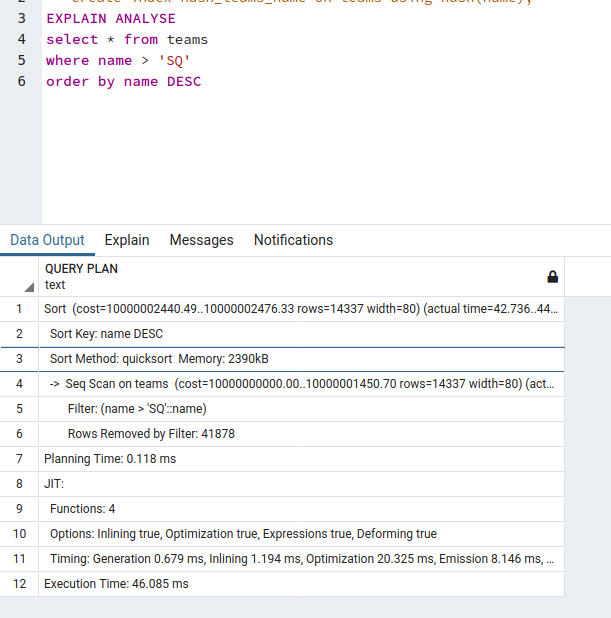
btree:

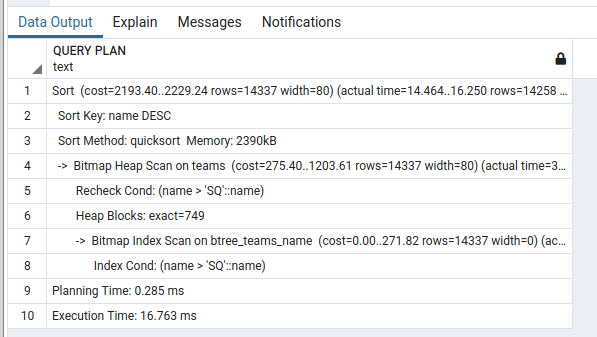
hash:

Висновки: індекси значно прискорюються виконання запиту з 42 ms до 0.267 ms

Запит 4:

без використання hash та btree індексів:

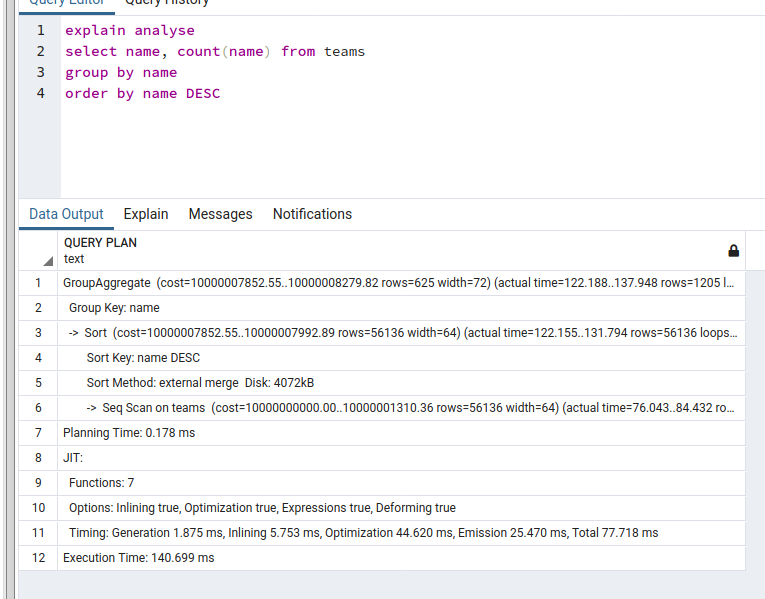
btree:

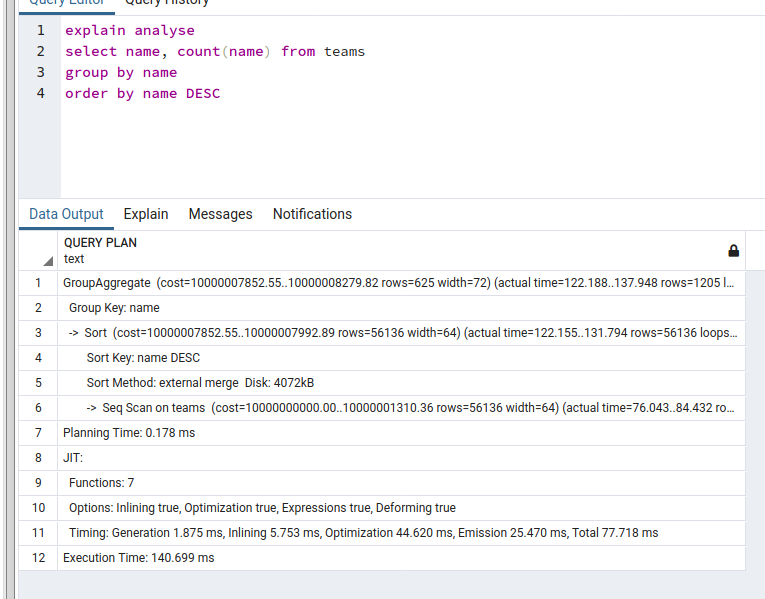


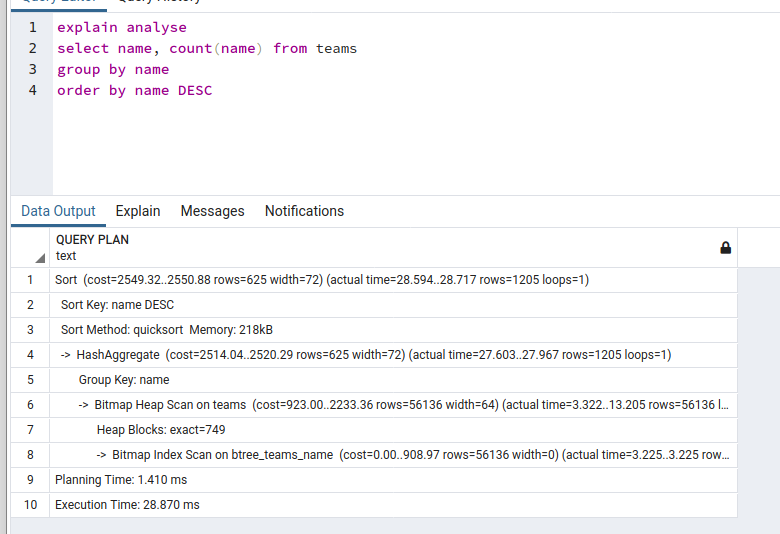
hash: не використовується

Висновок: btree прискорює виконання вибірки в заданому діапазоні (в даному запиті від ‘SQ’ до найбільшого). Послідовний пошук (без використання індексу) має приблизно однаковий час виконання для запиту 3 та 4.

Запит 5

без використання hash та btree індексів:

btree:

hash: не використовується

Висновок: btree прискорює виконання агрегатних функцій (з 140.7 до 28.9 ms), якщо групувати за полем, що індексується.

Висновок після аналізу результатів запитів:

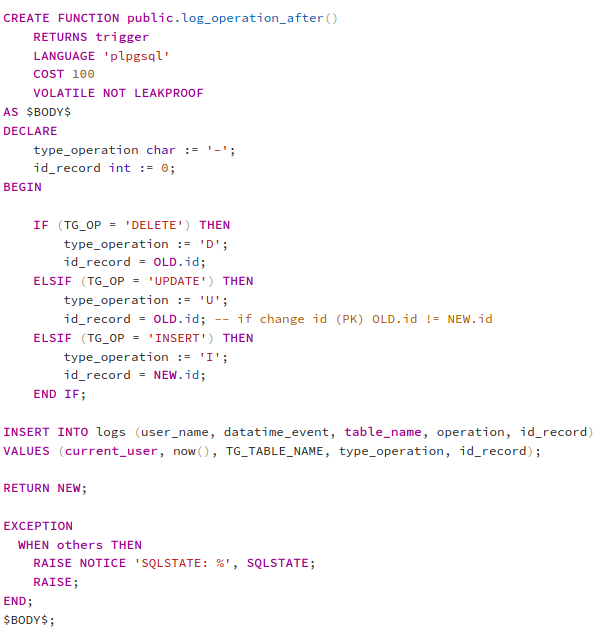
Hash-індекс ефективний, коли фільтруються дані за допомогою операції рівності (запит 3). Це пов’язано з фізичною реалізацією індекса на основі хеш-таблиці.

Btree-індекс ефективний у більшості випадків. Це пов’язано з фізичною реалізацією індекса на основі збалансованих дерев, що мають багато гілок.

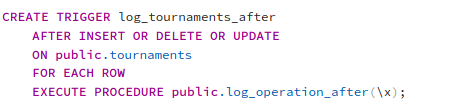
SeqScan ефективніший за запропоновані індекси у випадках специфічних шаблонів.

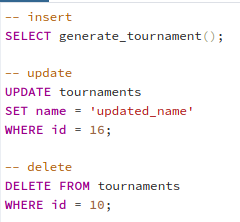
Завдання 3

Опис роботи тригерної функції: після операцій (INSERT, UPDATE, DELETE) над рядками таблиці, формується рядок у таблиці logs, в якому описано:

* користувача, що зробив операцію
* час виконання операції
* таблиця над якою виконувалась операція
* тип операції
* id рядка над яким виконувалася операція

Тригер реалізовано для таблиці tournamnets після INSERT, UPDATE, DELETE команд:

Команди, що ініціюють виконання тригера:

Останні рядки в таблиці logs:

