НАЦIОНАЛЬНИЙ ТЕХНIЧНИЙ УНIВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛIТЕХНIЧНИЙ IНСТИТУТ IМЕНI IГОРЯ СIКОРСЬКОГО» ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування і спеціалізованих комп'ютерних систем**

**Лабораторна робота №3**

з дисципліни  «Бази даних і засоби управління»

Тема: «**здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL**»

Виконали: студенти ІІІ курсу

ФПМ групи КВ-81

 Ядуха Б.В.

Викладач: ПетрашенкоА.В.

*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.

*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.

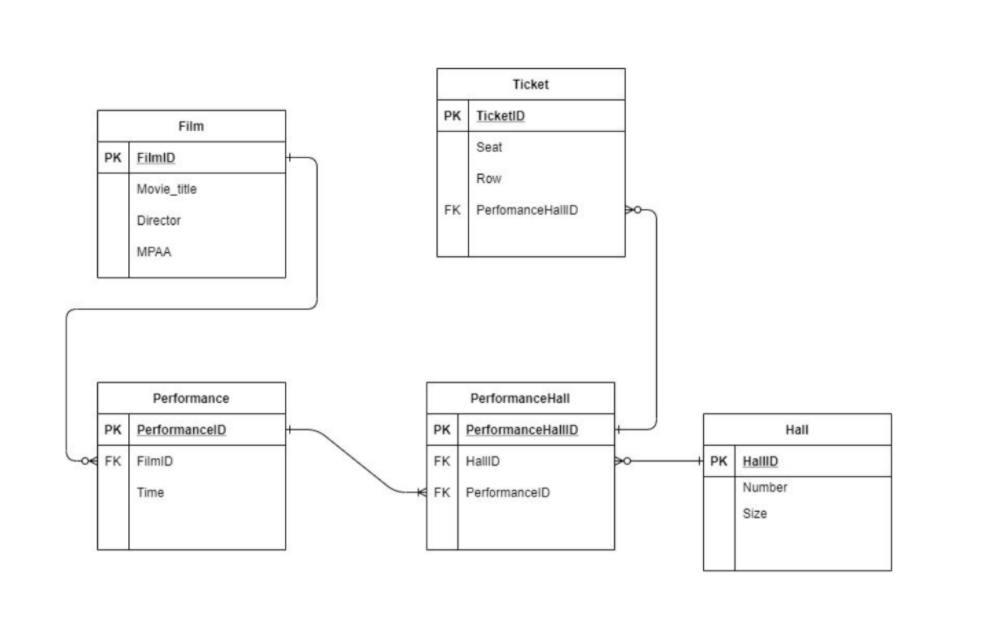
Посилання на github: https://github.com/Ent71/database-labs

**Варіант 26:**

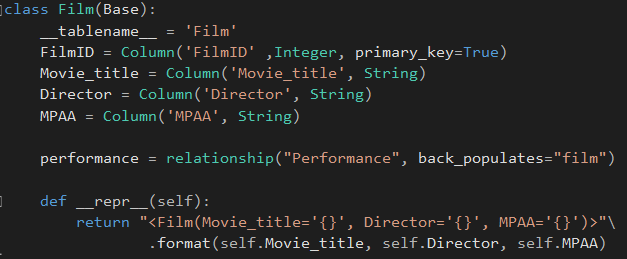
*Види індексів - Btree, BRIN*

*Умови для тригера - before update, delete*

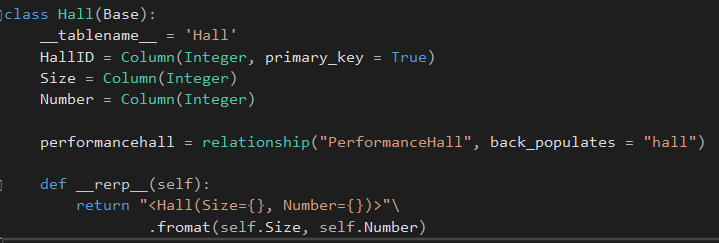
***Зв’язки між таблицями***



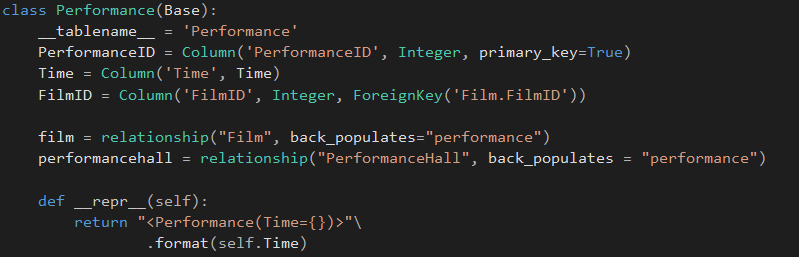
Таблиця **Film**:



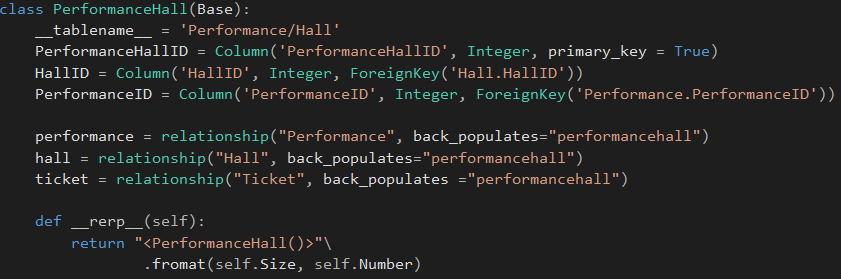
Таблиця **Hall**:



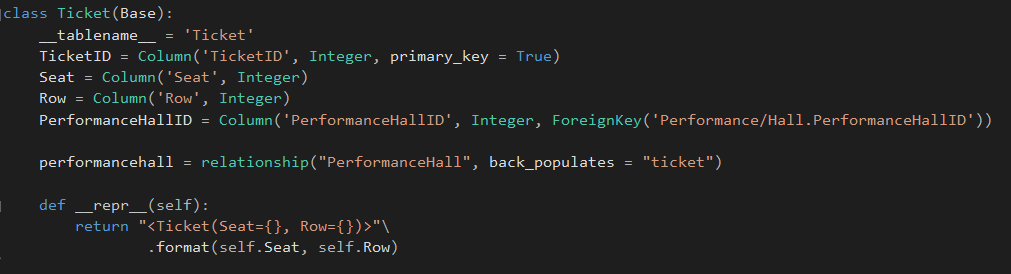
Таблиця **Performance**:



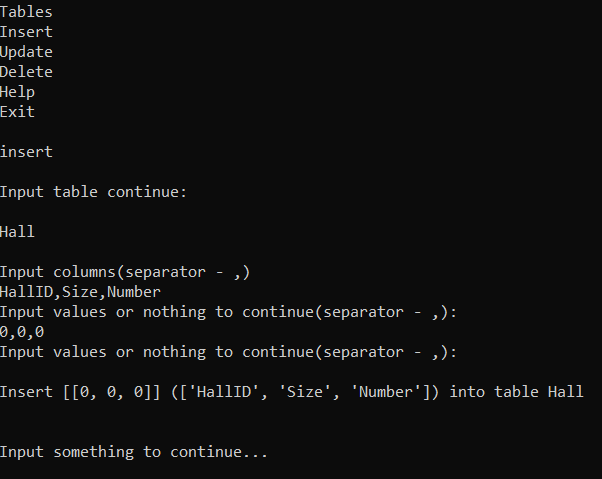
Таблиця **Performance**/**Hall**:

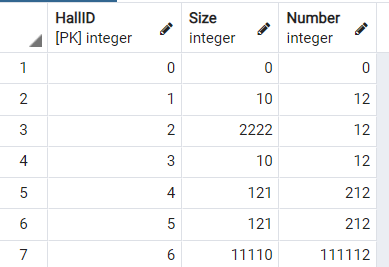


Таблиця **Ticket**:

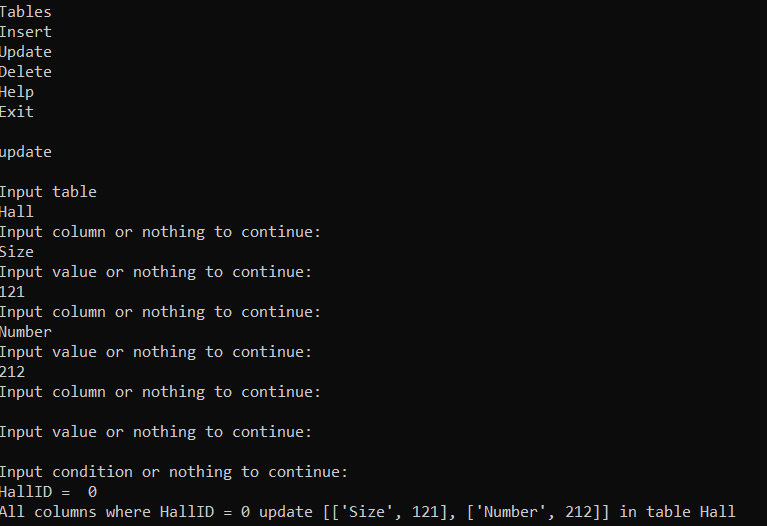


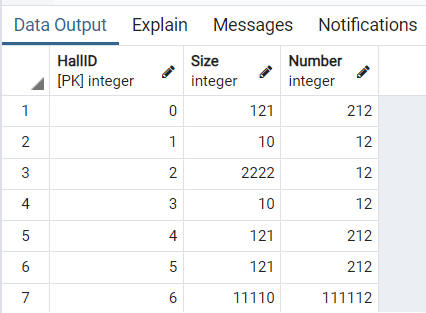
Приклад роботи **Insert**:



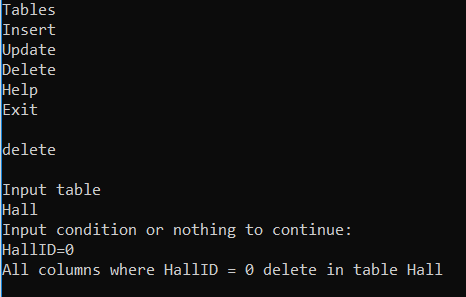


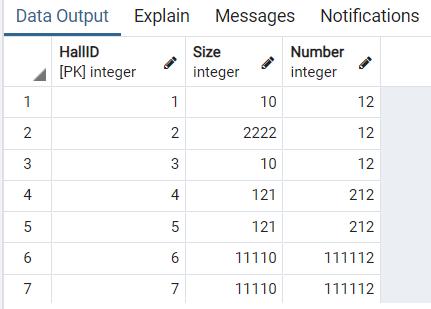
Приклад роботи **Update**:





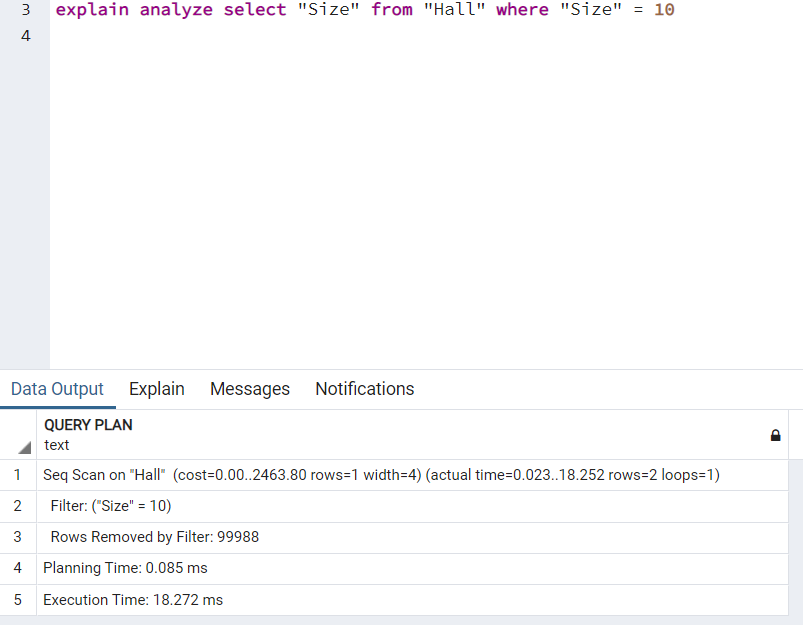
Приклад роботи **Delete**:





Індекси

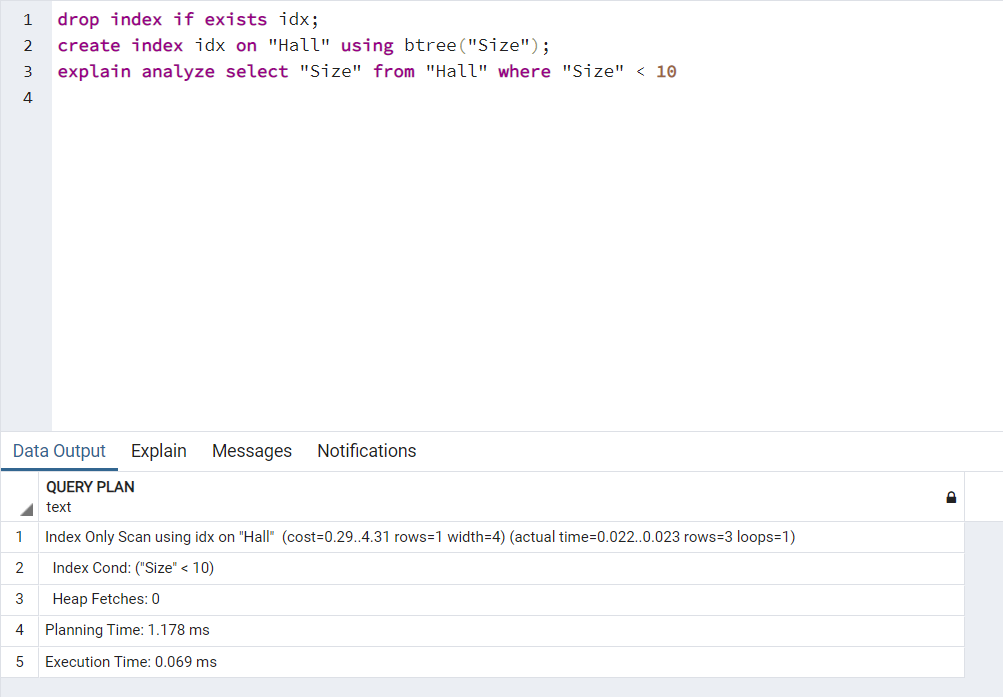
Без індексів:



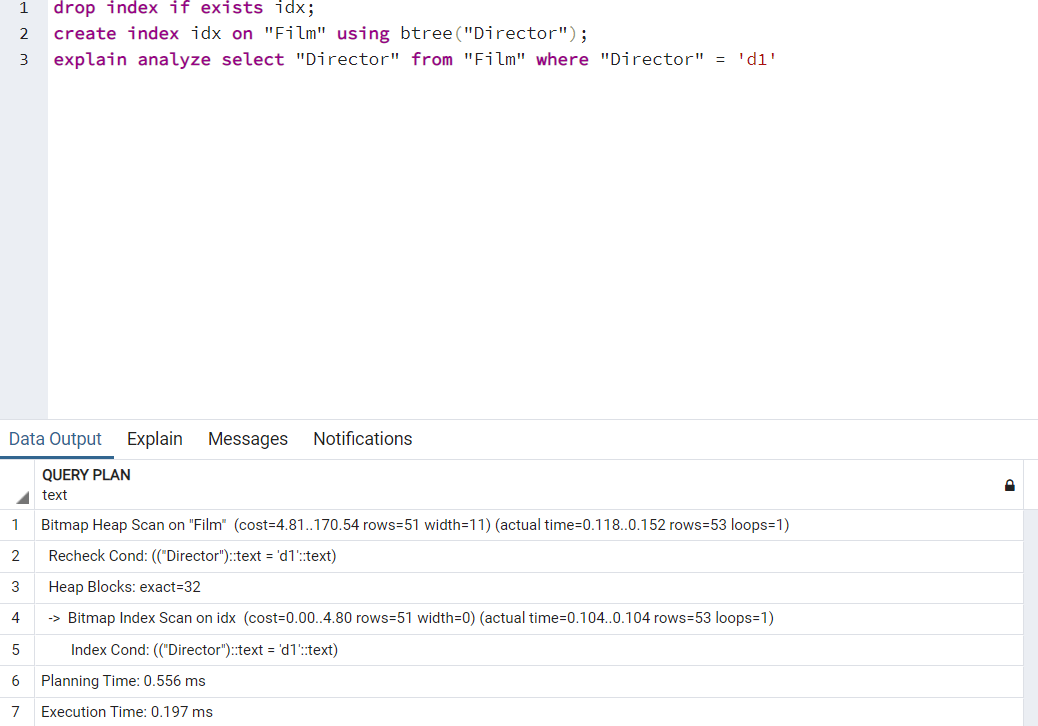
Індекс не використовується при вибірці усієї інформації:



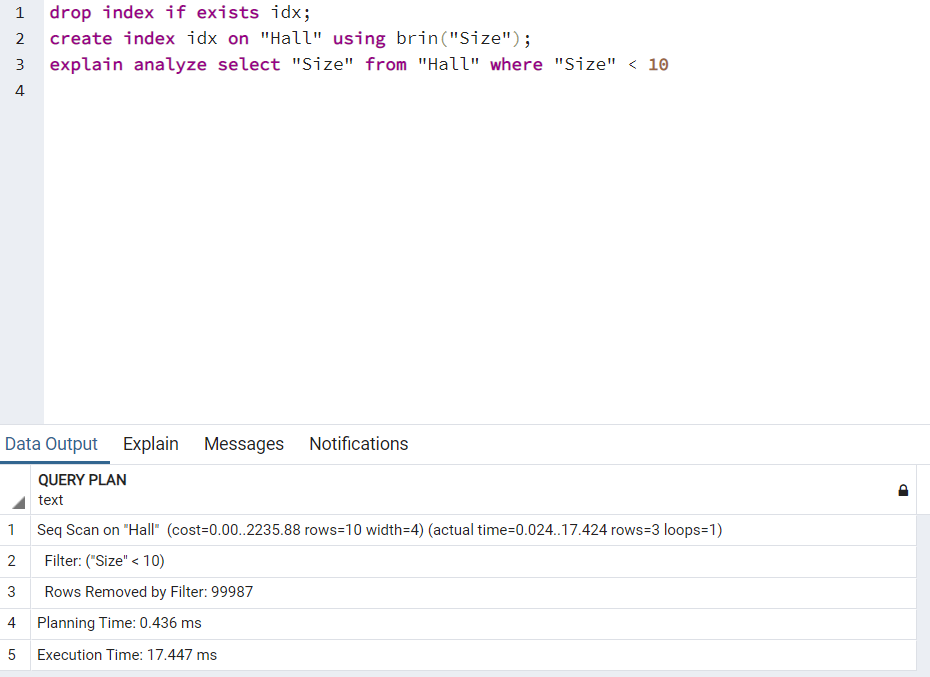
Індекс використовується при пошуку числа(btree ефективний при порівнянні, рівності, пошуку в діапазоні):



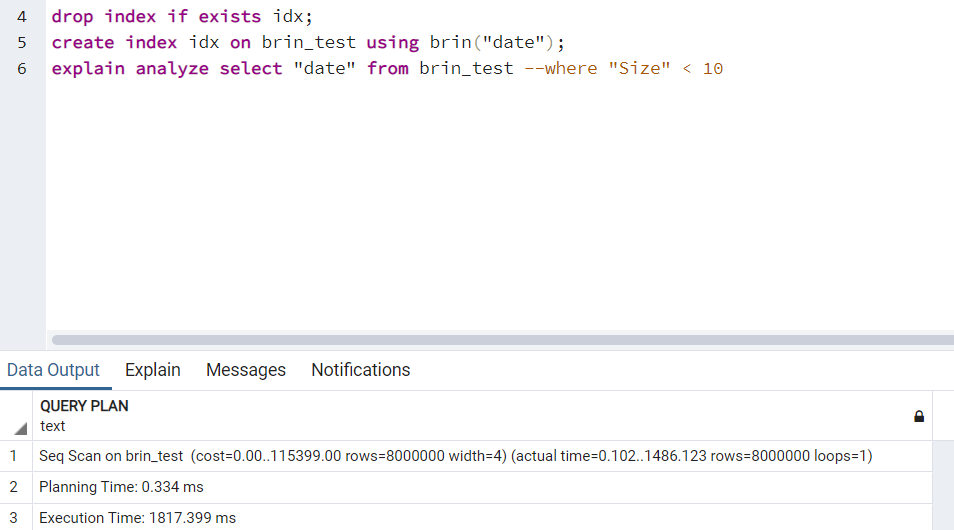
Індекс не використовується при пошуку рядка(Btree неефективний при пошуку рядка)



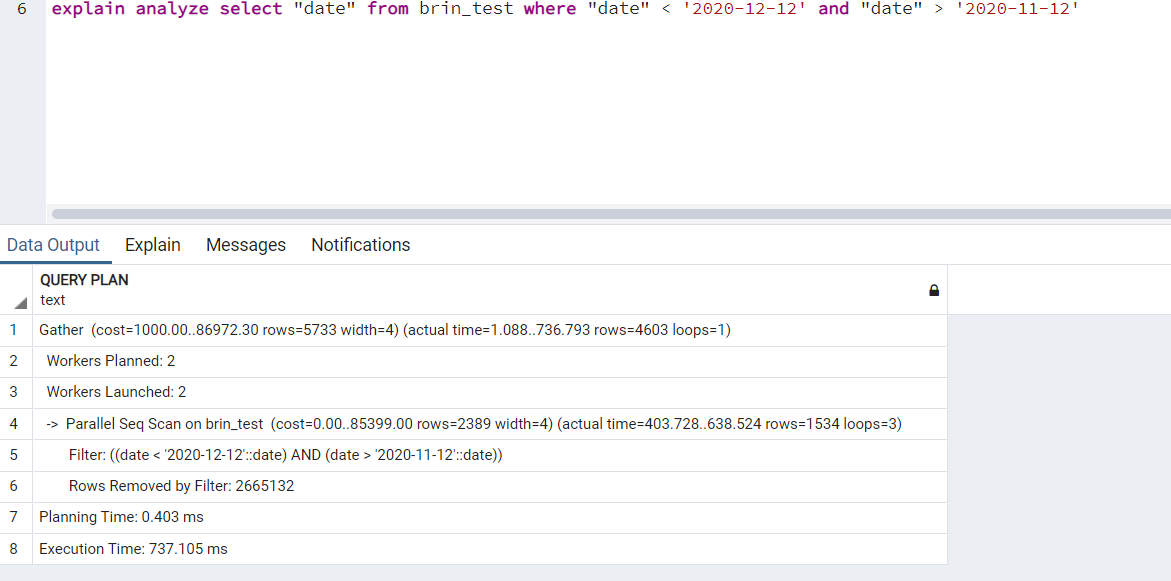
Brin використовується для дуже великих таблиць(не використовується при 100000 тому було створено нову таблицю для швидшого її заповнення(колонка якої була типу date для більшої імовірності використання Brin))



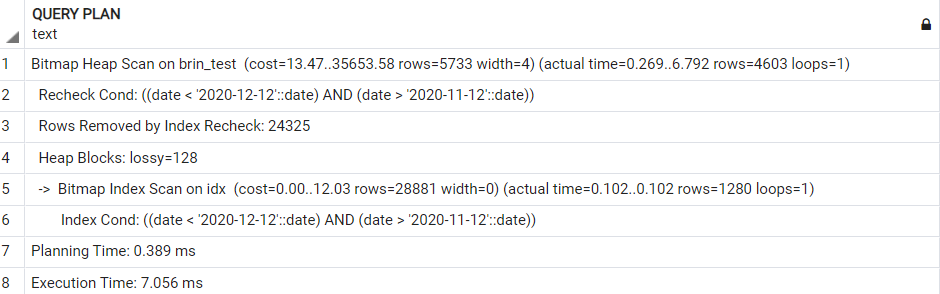
Не використовується при виборі усьої або майже усьої таблиці:



Без індексу:



З індексом Brin:

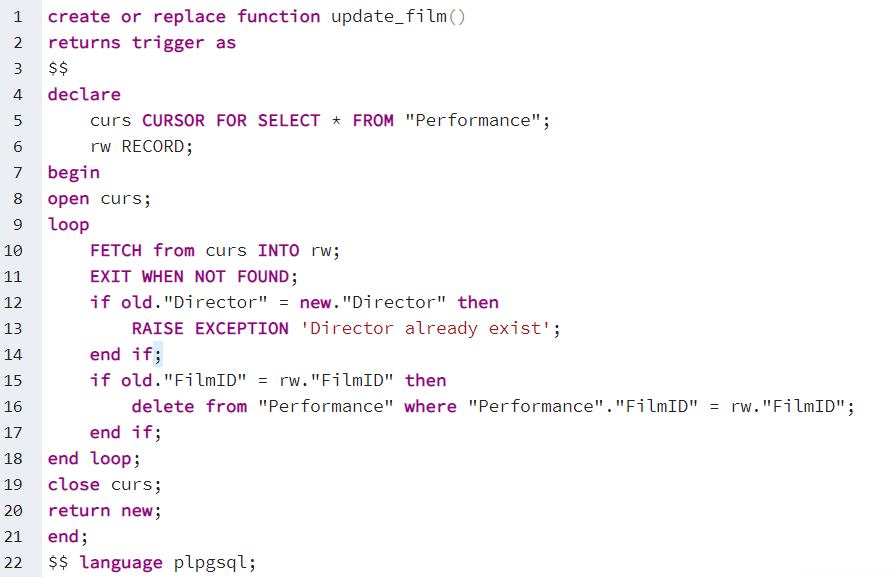


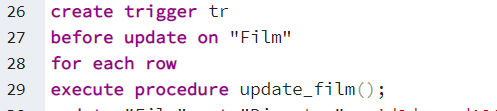
Brin працює лише на дуже великих таблицях, але набагато менший за Btree

**Тригер**

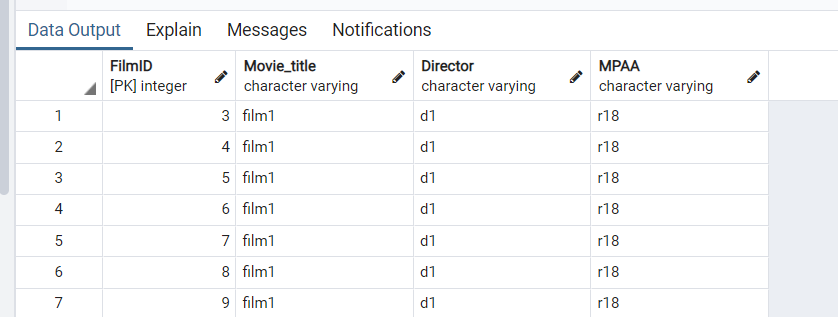
*Умови для тригера - before update, delete*

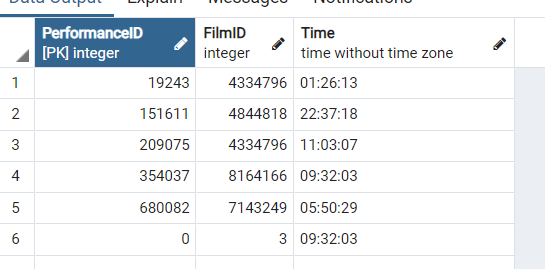
*Тригер:*

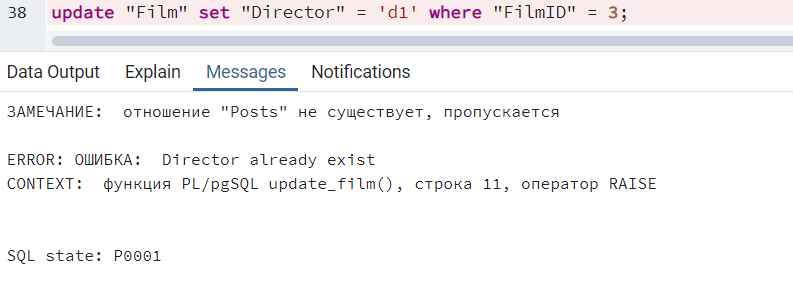




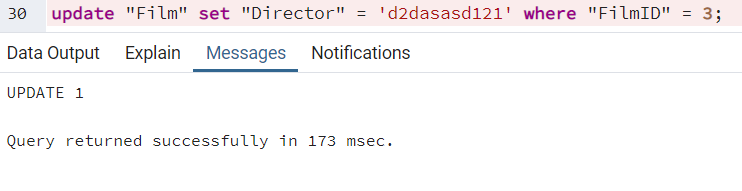
Таблиці перед використанням тригера:

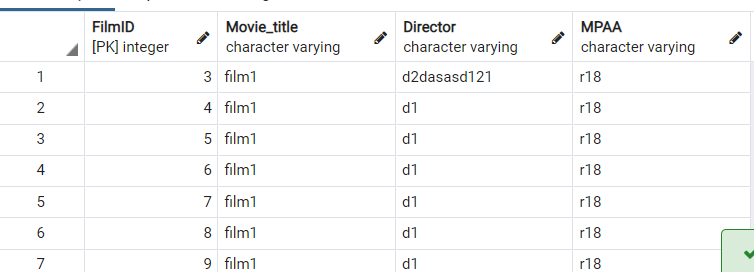


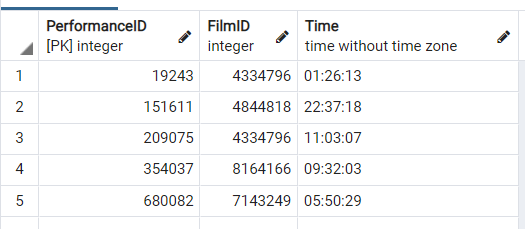




Викликано виключення таблиці не змінилися







При зміні рядка таблиці Film видаляються усі рядки в яких згадується цей рядок в таблиці Performance.