МIНIСТЕРСТВО ОСВIТИ 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 ”

|  |  |
| --- | --- |
| Виконав | Перевірив |
| студент II курсу | “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р. |
| групи КП-93 | викладач |
| Катрич Владислав Сергійович | Петрашенко Андрій Васильович |
| (*прізвище,* *ім’я,* *по батькові*) | (*прізвище,* *ім’я,* *по батькові*) |

**Варіант № 1**

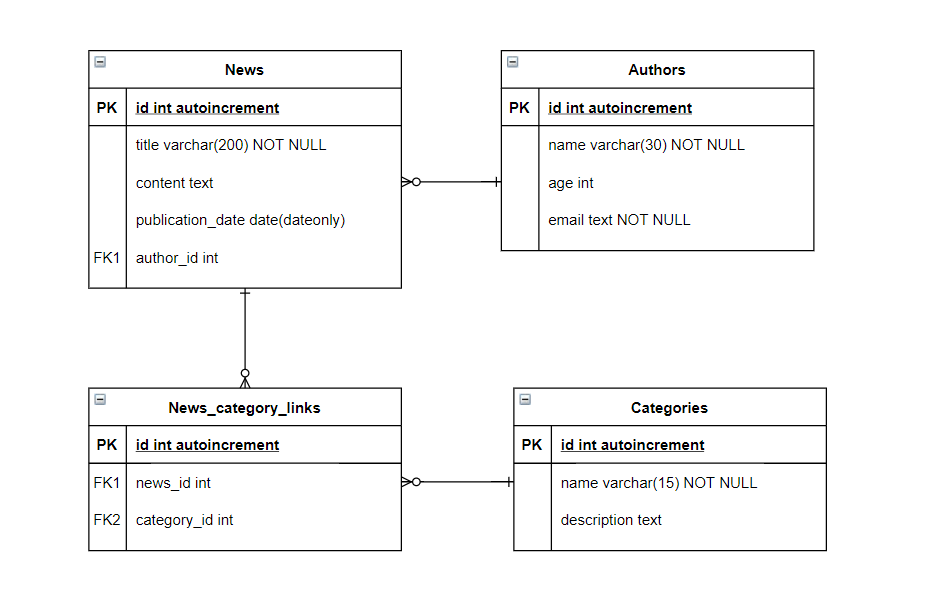
Київ 2020

***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.

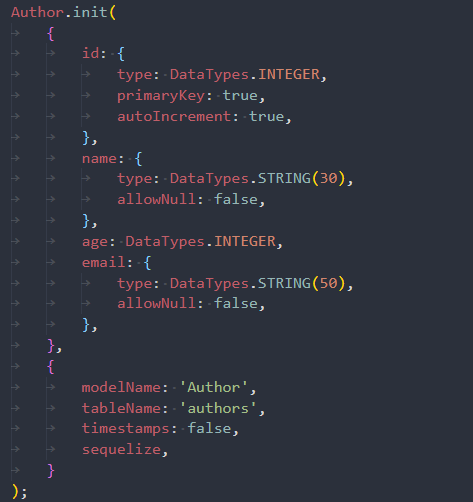
**№1)**

**Схема бази даних у вигляді таблиць:**



**Відповідні класи ORM:**

*Author:*



*News:*



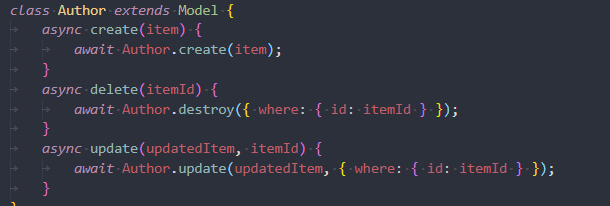
*Category:*

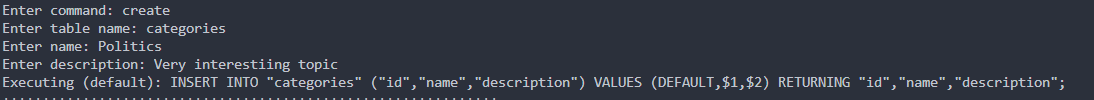


*CategoryNewsLinks:*



**Приклади запитів у вигляді ORM:**







**№2)** *(Btree, Hash)*

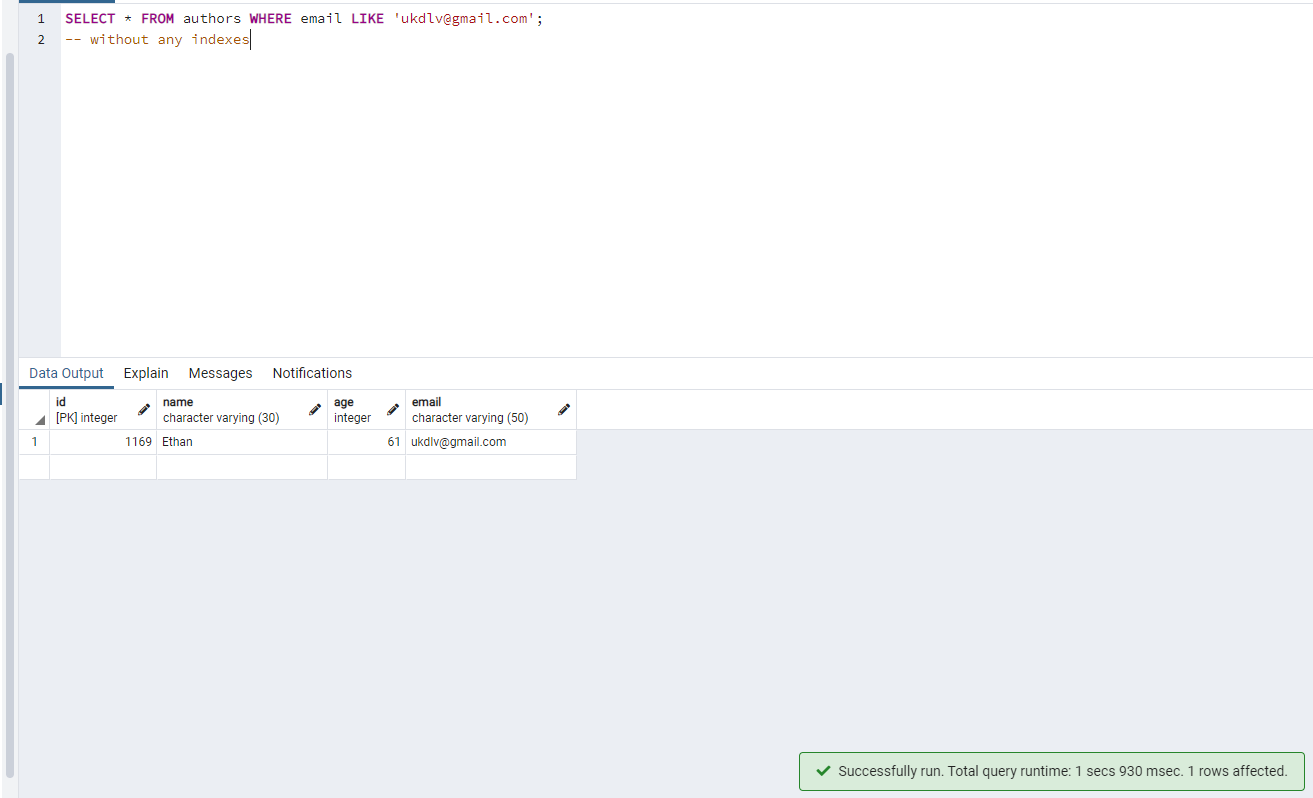
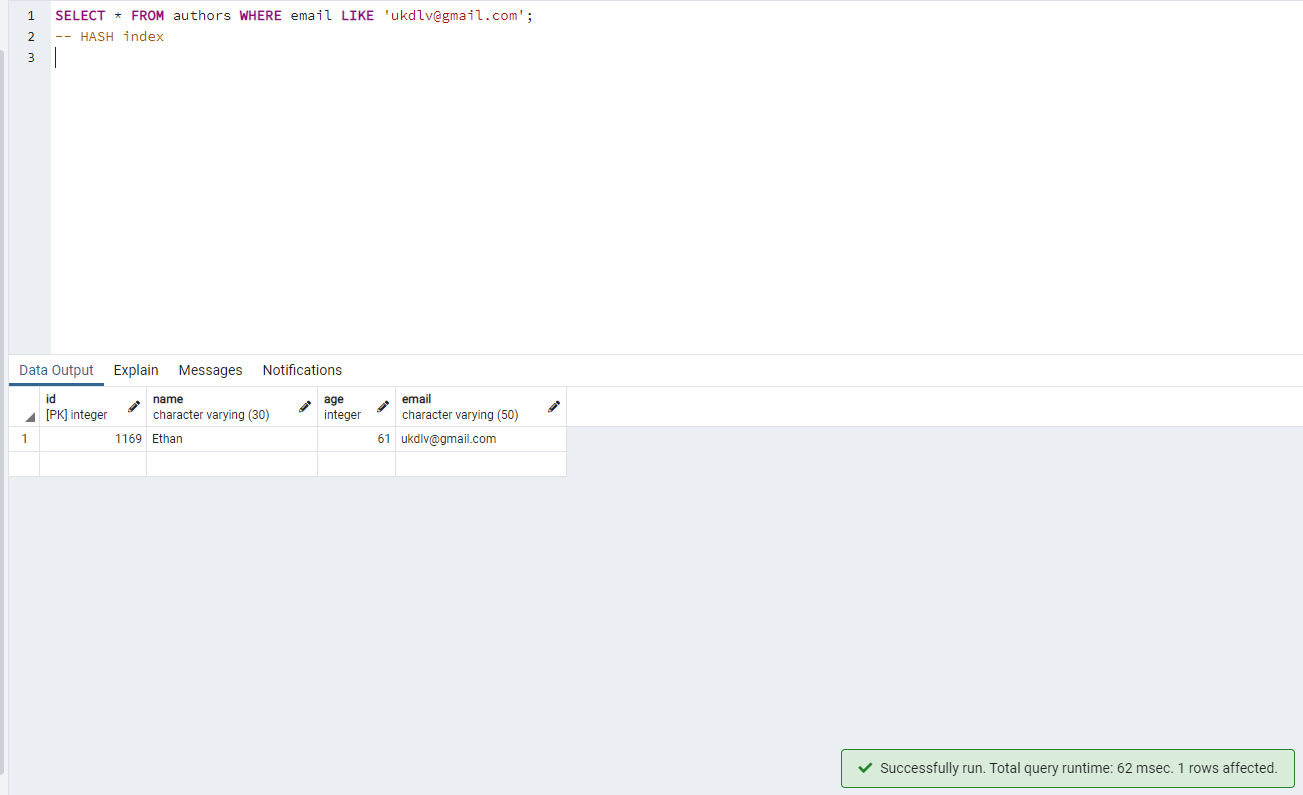
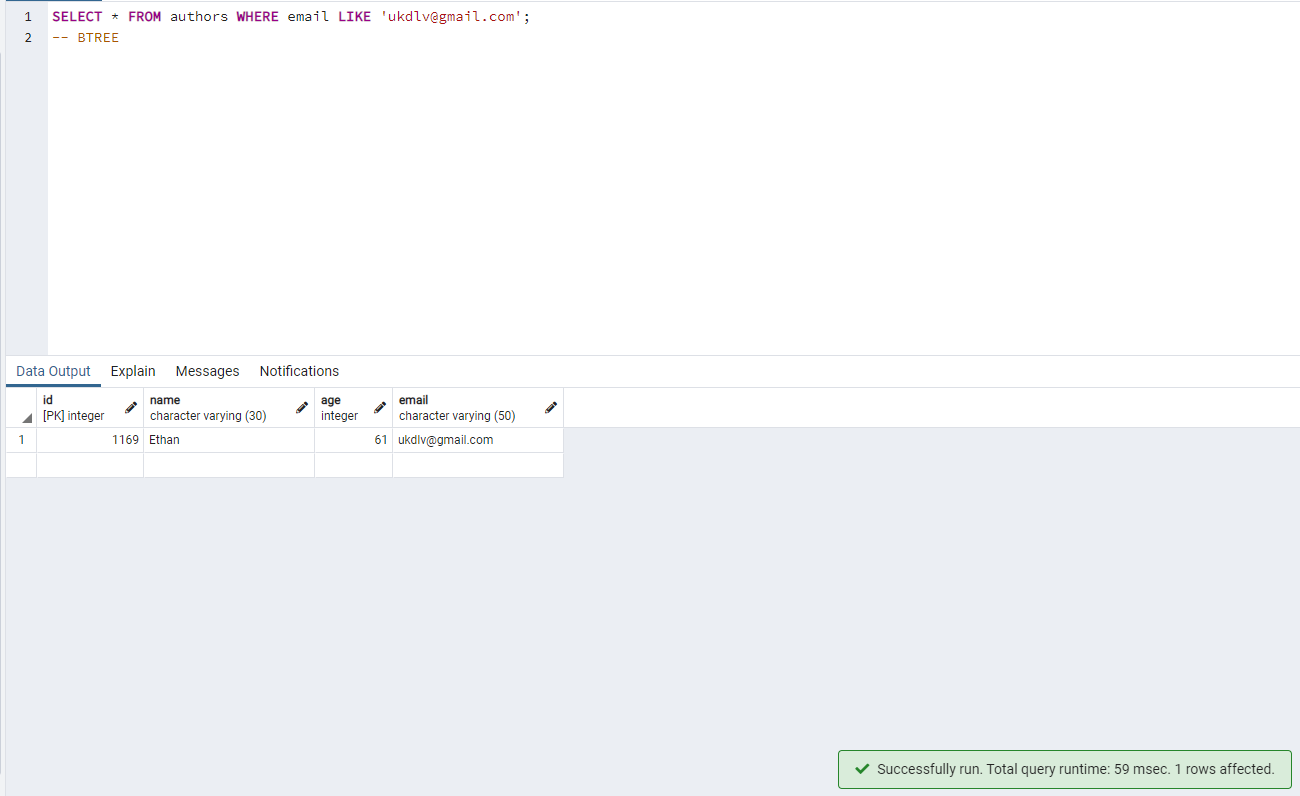
**Команди створення індексів:**

|  |
| --- |
|  |
|  |

**SQL тексти:**

|  |
| --- |
|  |
|  |

**Результати, час, аргументування:**



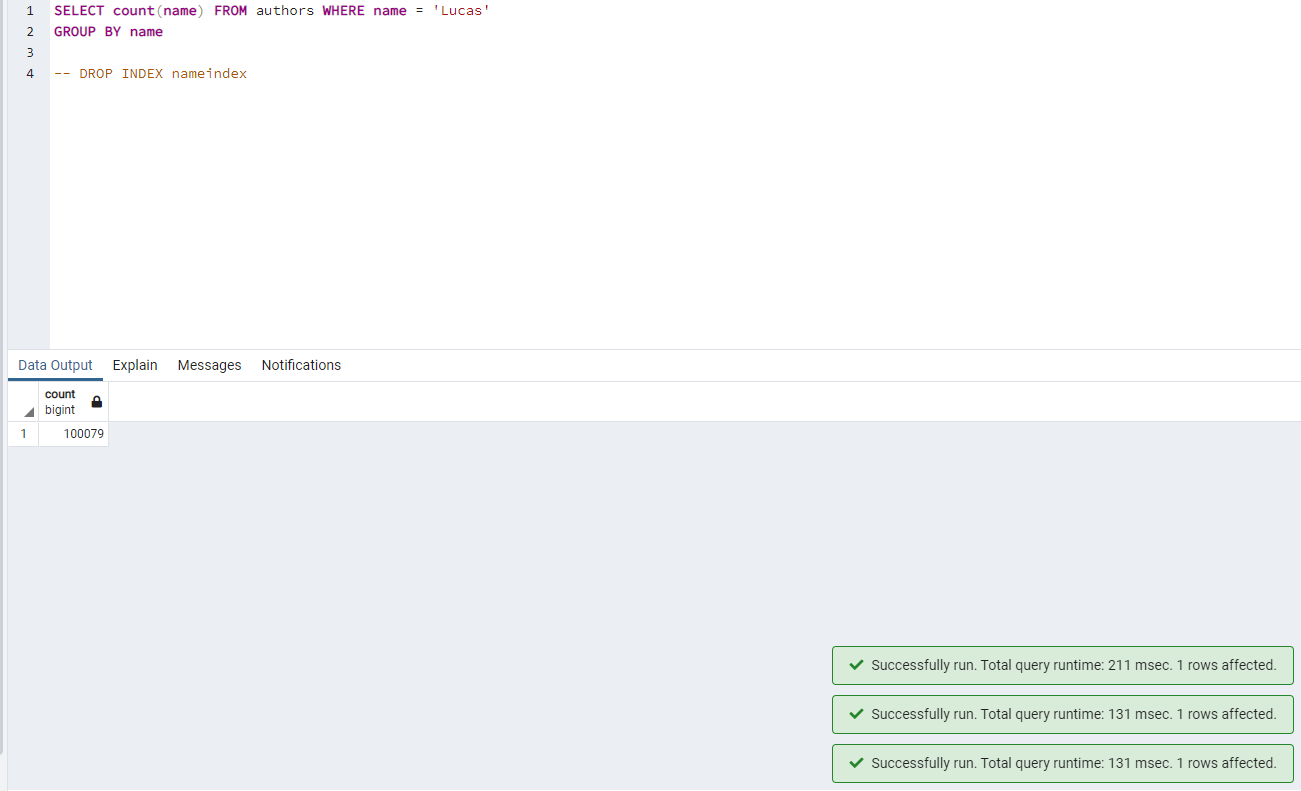
Даний запит відбувався при об’ємі записів розміром в 11 мільйонів. Добре видно, який результат ми отримуємо від пошуку використовуючи індекси.

Це пришвидшення відбувається за рахунок того, що ми не проходимося по всім записам, а використовуємо певні алгоритми, які залежать від того, який саме вид індексів ми обрали.

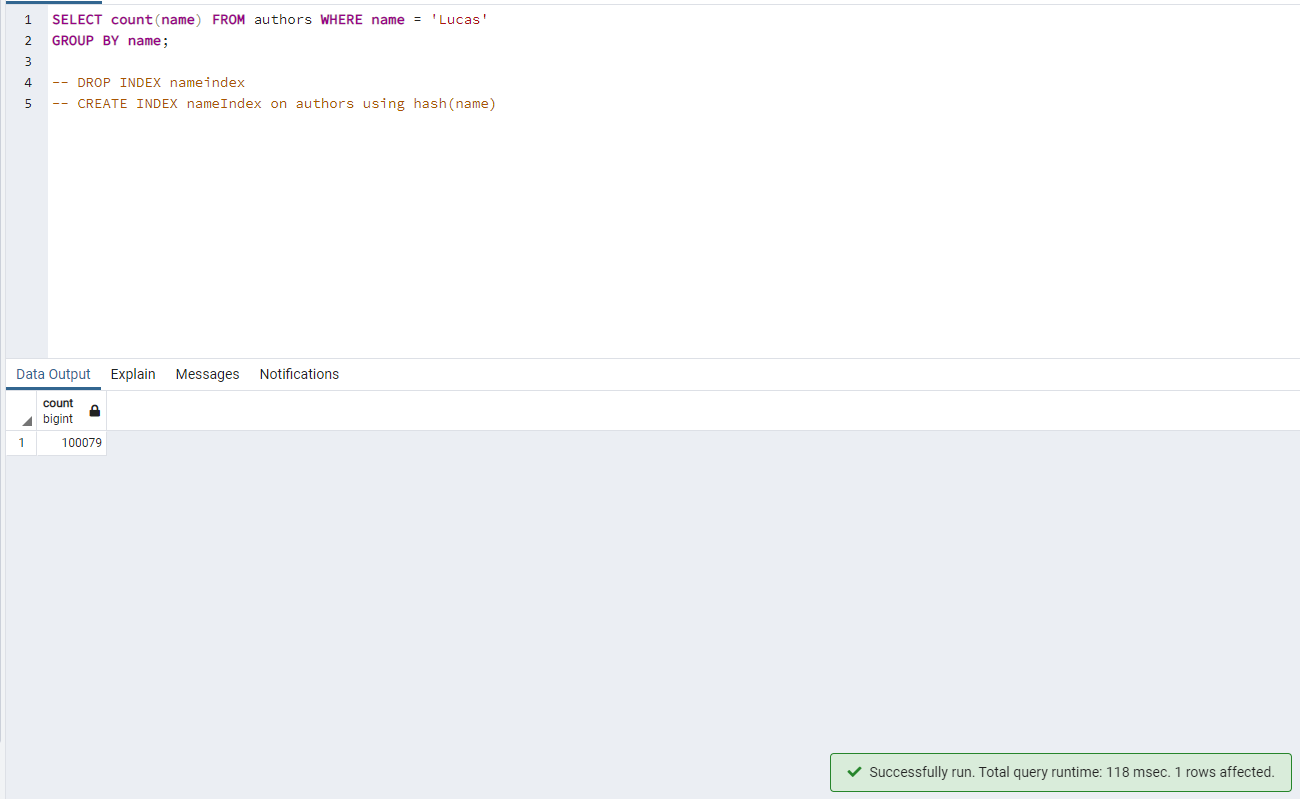
Але також бувають ситуації, коли індекси не прискорюють пошук. Зазвичай таке трапляється, коли дані або однакові, або їх занадто мало, щоб розкрити потенціал роботи з індексами.

Далі для тестування буде взято лише 300.000 записів

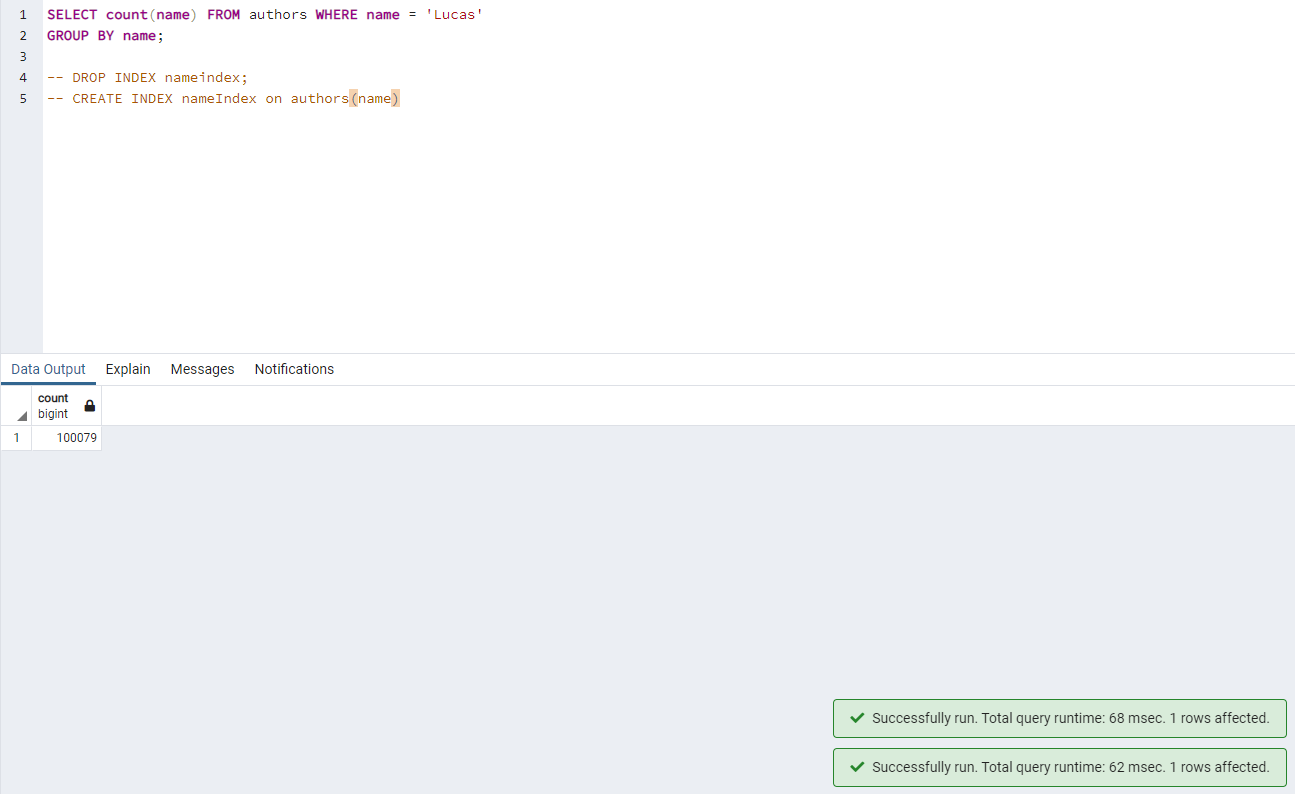
*Без використання індексів*



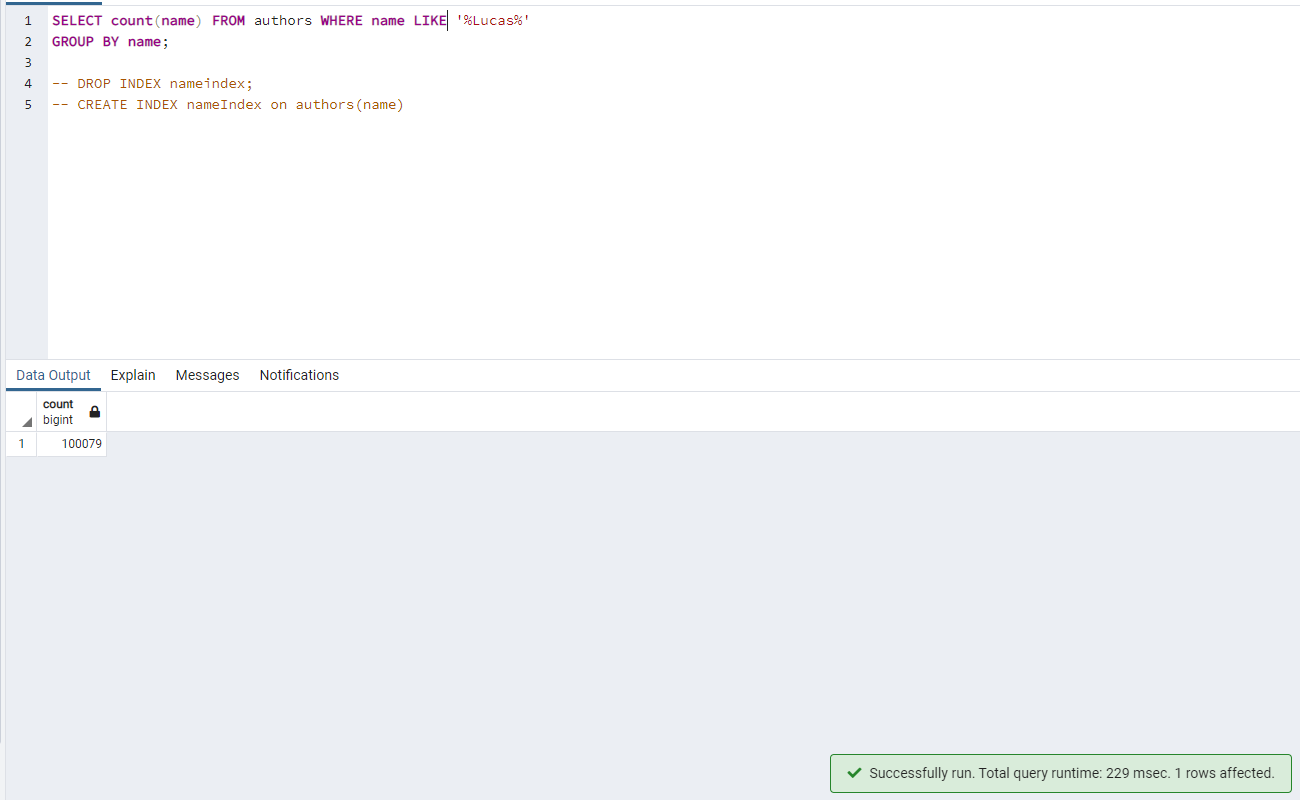
*З використанням індексу HASH по name*



*З використанням індексу BTREE по name*



Також можна помітити, шо при використанні LIKE ‘%VARIABLE%’ наші індекси не працюють. Наприклад:



**№3)** *(before insert, delete)*

**Команди, що ініціюють виконання тригера:**

|  |
| --- |
| before insert |
|  |

|  |
| --- |
| delete |
|  |

**Текст тригера:**

|  |
| --- |
| before insert & delete |
|  |

**Скріншоти зі змінами у таблицях бази даних:**

|  |
| --- |
| before insert |
|  |

|  |
| --- |
| delete |
|  |

**Контрольні запитання:**

1. Сформулювати призначення та задачі об’єктно-реляційної проекції (ORM).
2. Проаналізувати основні види індексів у PostgreSQL (*BTree, BRIN, GIN, Hash*): призначення, сфера застосування, переваги та недоліки.
3. Пояснити призначення тригерів та функцій у базах даних.

*Відповіді:*

1. Призначення ORM полягає в тому, що завдяки йому ми можемо спростити процес збереження об’єктів в реляційну базу даних і їх вилучення. Більшість ORM об’єктів покладаються саме на метадані бази даних і об’єктів, тобто об’єктам нічого не потрібно знати про структуру бази даних, а базі даних нічого про те, як дані організовані в додатку. ORM забезпечує повне розділення завдань в добре спроетованих додатках, при якому і база даних, і додаток можуть працювати з даними кожен у своїй вихідній формі.

2.

* 1. GIN – застосовується для обробки випадків, коли елементи, що підлягають індексації, є складеними значеннями, а запити, що обробляються індексом, повинні шукати значення елементів, які відображаються в складених елементах. Однією з переваг GIN є те, що він дозволяє розробляти власні типи даних із відповідними методами доступу. А також кожне значення ключа зберігається лише один раз, тому індекс GIN є дуже компактним для випадків, коли той самий ключ відображається багато разів.
  2. BTree – Btree забезпечує ефективний спосіб вставки та читання даних. Дерево надає можливості послідовного пошуку в додаток до двійкового пошуку, що надає базі даних більший контроль для пошуку значень без індексу в базі даних.
  3. Hash – перевага полягає в тому, що розмір індексів досить малий. А також в деяких випадках можуть бути навіть швидшими за BTree. А з недостатків є те, що хеш-функція не зберігається відношення порядку: з того, що значення хеш-функції одного ключа менше значення функції іншого ключа, можна зробити ніяких висновків про те, як впорядковані самі ключі. Тому хеш-індекс в принципі може підтримувати єдину операцію «дорівнює»:
  4. BRIN - Індекси BRIN ефективні, якщо впорядкування значень ключів слідує за організацією блоків на рівні зберігання. У найпростішому випадку це може вимагати фізичного впорядкування таблиці, яка часто є порядком створення рядків у ній, щоб відповідати порядку ключа. Ключі від сформованих порядкових номерів або створених даних є найкращими кандидатами для індексу BRIN. Оскільки індекс BRIN дуже малий, сканування індексу додає мало накладних витрат порівняно з послідовним скануванням, але може уникнути сканування великих частин таблиці, які, як відомо, не містять відповідних кортежів.

3. Взагалі функції як і в будь-якій мові програмування слугують для того, щоб можна було зробити так звану обгортку для якогось блоку коду і визивати його, коли завгодно. В цей час тригери це просто доповнення до функцій завдяки яким нам не потрібно їх визивати власноруч, напроти, ми можемо прив’язати певну функцію до певної події і цим самим наша функція буде визиватись(виконуватись) автоматично при певних умовах.