

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»**

Виконав: студент ІII курсу

ФПМ групи КВ-83

Ткачук Тарас

Київ – 2020

**Засоби оптимізації роботи СУБД PostgreSQL**

*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).

1. Створити та проаналізувати різні типи індексів у PostgreSQL.
2. Розробити тригер бази даних PostgreSQL.

**Варіант 23**

У другому завданні проаналізувати індекси GIN, Hash.

Умова для тригера – before update, delete.

Навігація по звіту:

1. [Завдання 1(перероблення програми в ORM модель)](#Завдання1);
2. [Завдання 2(створення індексів)](#Завдання2);
3. [Завдання 3(створення тригерів)](#Завдання3).

Завдання 1

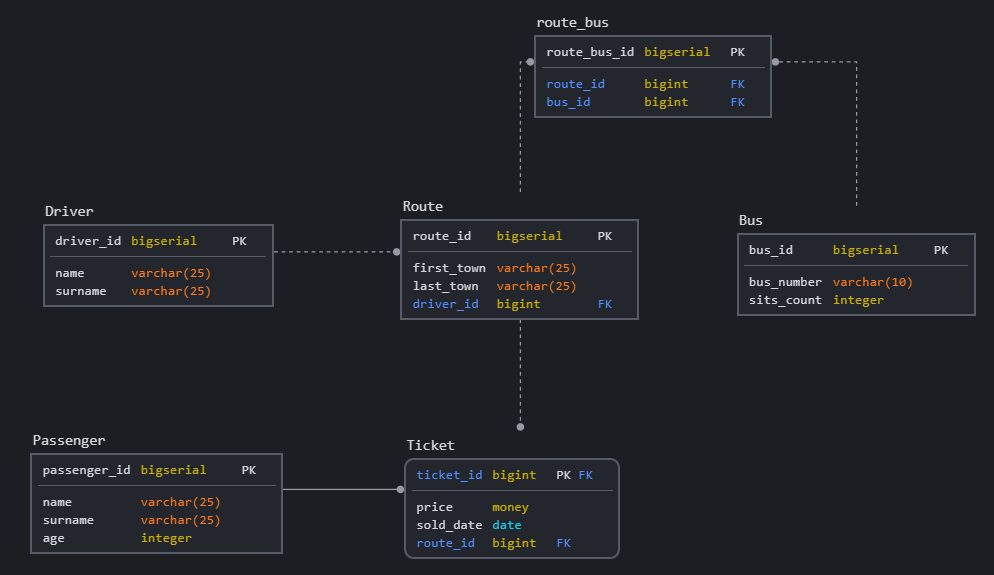


Рисунок 1 — Схема нормалізованої бази даних PostgreSQL на основі   
ER-моделі предметної області "Міжміське сполучення".

Тепер розглянемо реалізацію даної бази даних за допомогою мови програмування Python і бібліотеки sqlalchemy.

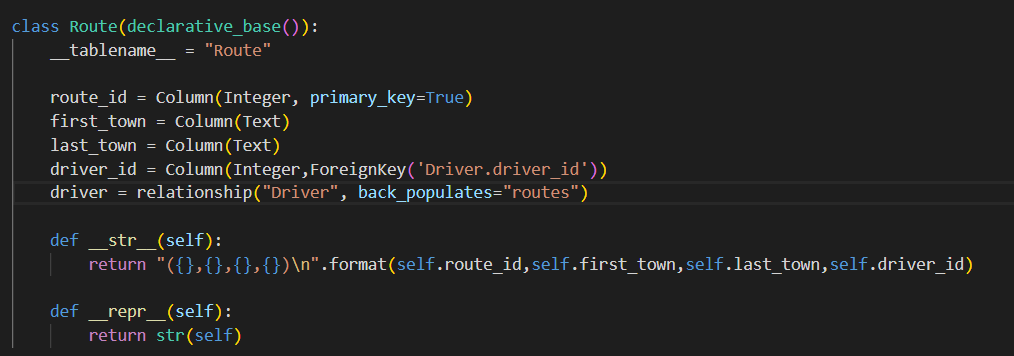


Рисунок 2 — Реалізація таблиці Route.

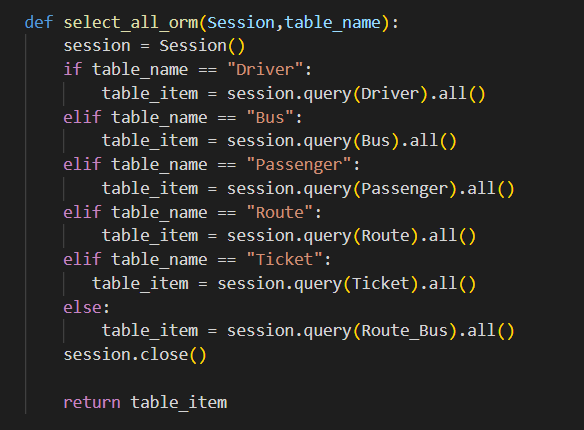


Рисунок 3 — Функція вибірки записів з таблиці

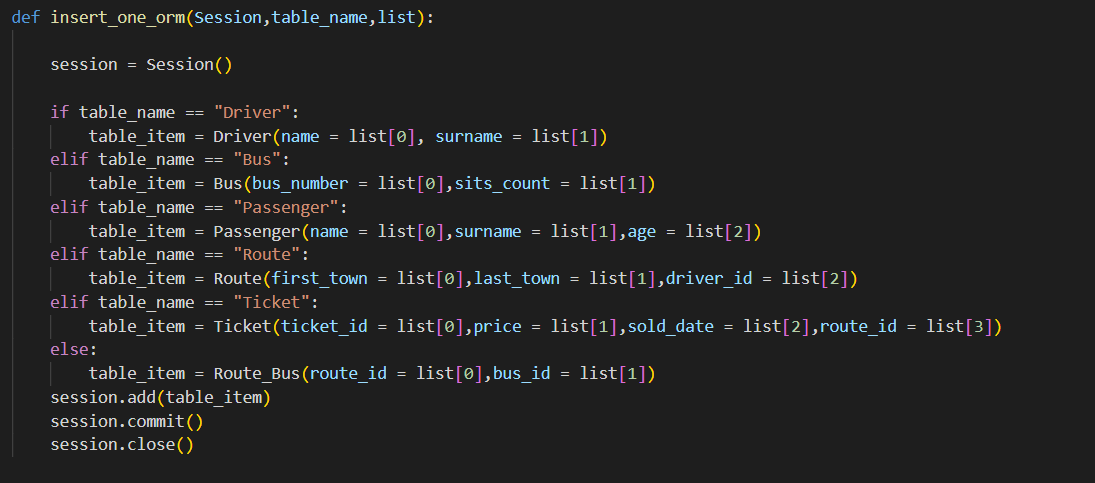


Рисунок 4 — Функція додавання запису у таблицю

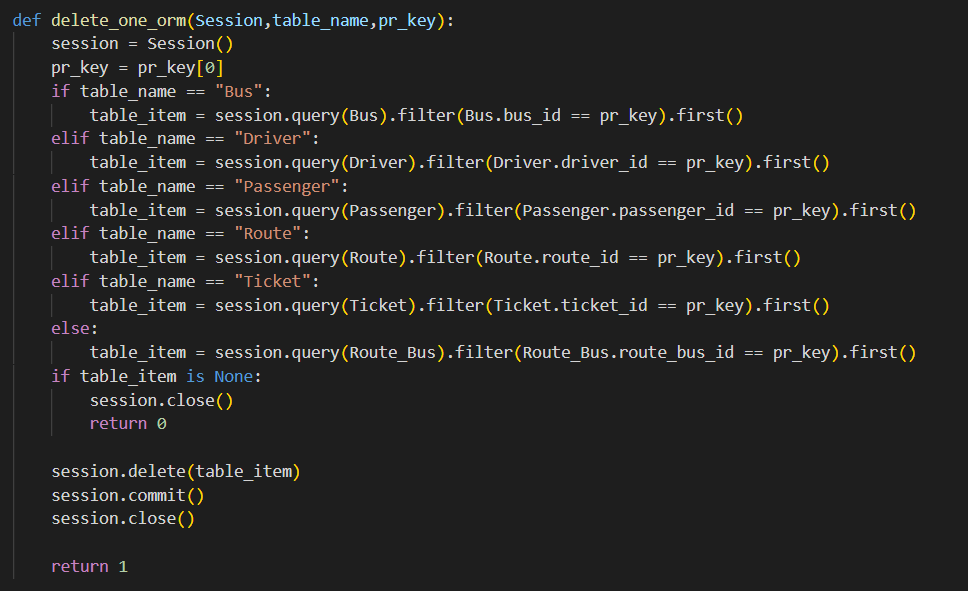


Рисунок 5 — Функція видалення запису з таблицю

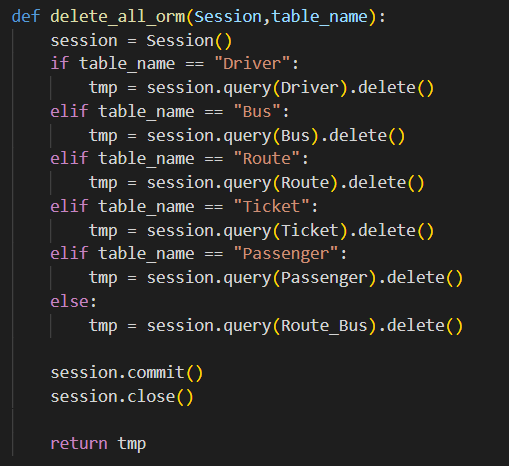
****

Рисунок 6 — Функція видалення всіх записів з таблиці

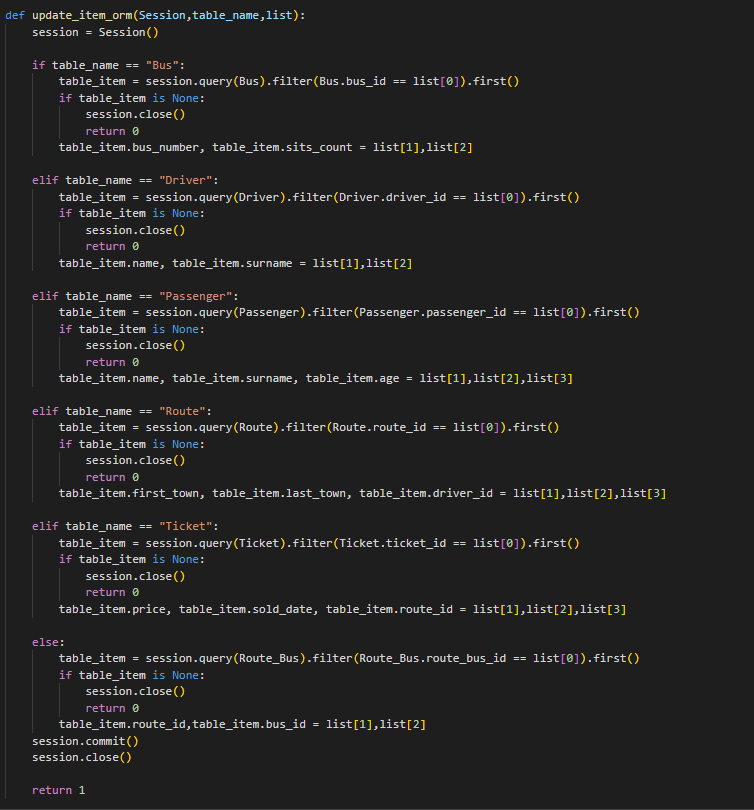


Рисунок 7 — Функція оновлення запису в таблиці

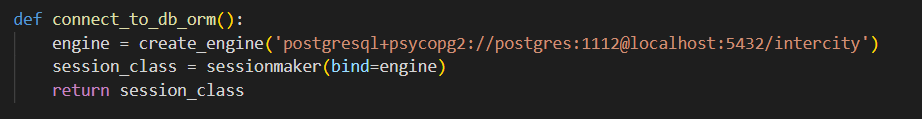


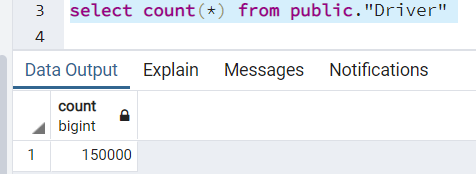
Рисунок 8 — Функція під’єднання до бази даних

Завдання 2

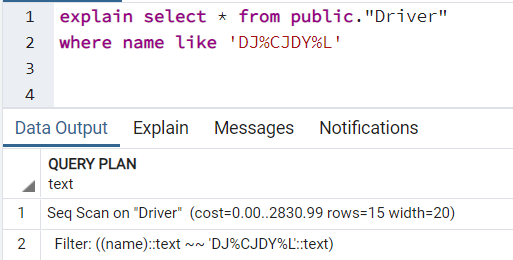
**GIN**

GIN означає узагальнений інверсований індекс. GIN призначений 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.

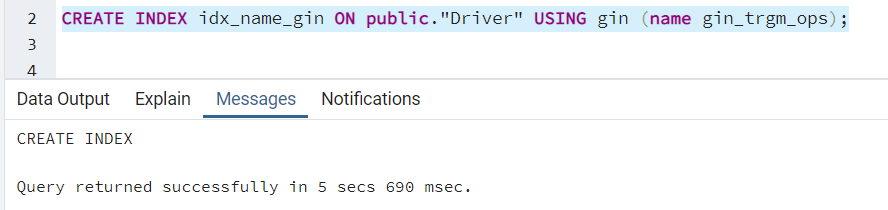
Для цього завдання створимо 150000 рядків в таблиці Driver.



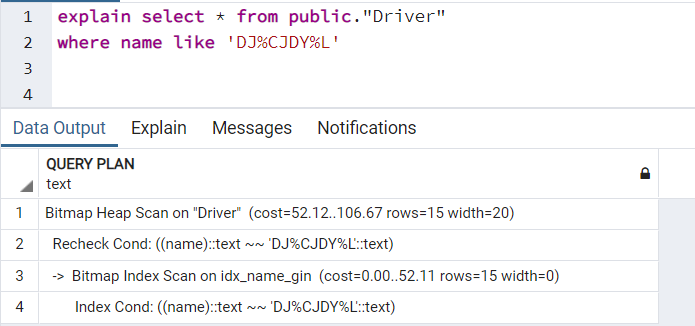
Для перевірки давайте виконаємо вибірку усіх записів, де   
name = 'DJSCJDYWL'. В результаті ми отримали cost = 2830 при Seq Scan.



Тепер створимо GIN індекс.



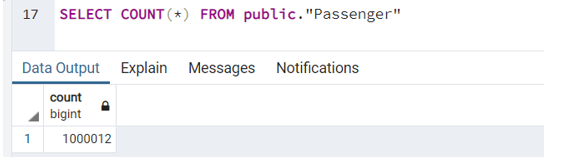
Тепер виконаємо той самий запит. І отримуємо cost = 52 при Bitmap Index Scan.



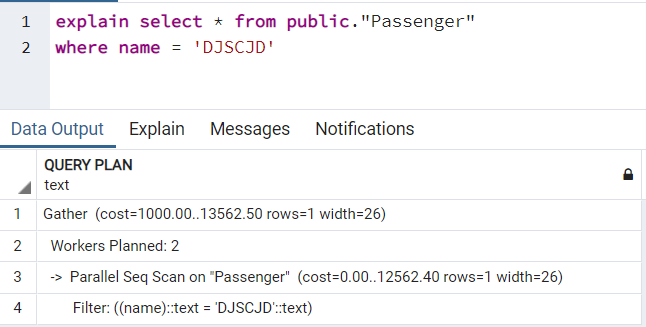
При використанні GIN індексу отримали пришвидшення в 54 раз.

**Hash**

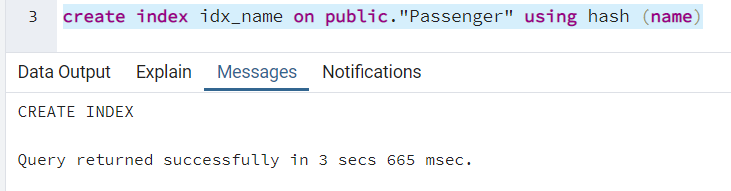
Для цього завдання створимо 1000000 випадкових рядків в таблиці Passenger



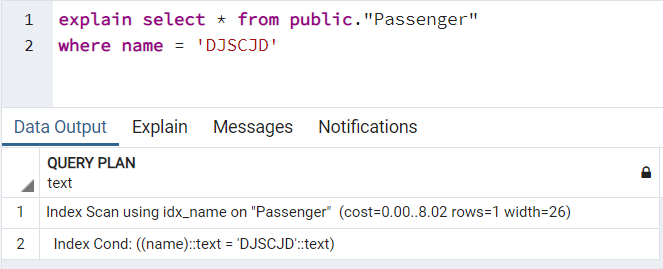
Для перевірки давайте виконаємо вибірку усіх записів, де   
name = 'DJSCJD'. В результаті ми отримали cost = 12562 при Parallel Seq Scan.



Тепер створимо індекс по колонці name.

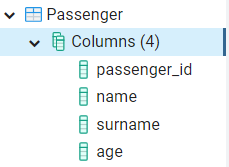
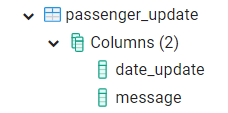


Далі виконаємо той же запит. І в результаті, ми отримуємо   
cost = 8 при Index Scan.

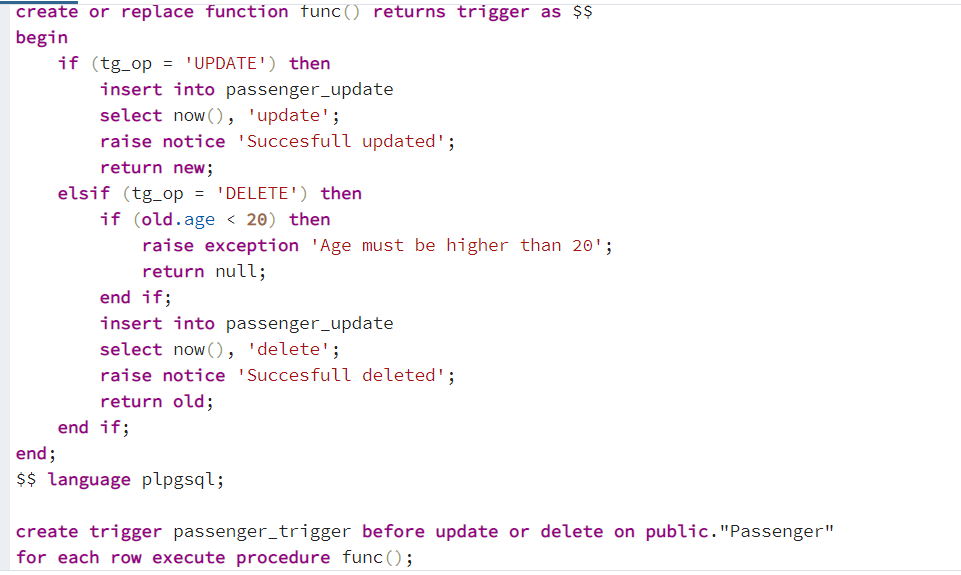


Отже, ми з’ясували, що після створення індексу, пошук та відображення результату покращився приблизно в 1570 разів. Тобто, при застосуванні індексу Hash, відбувається явне покращення роботи пошуку.

Завдання 3

В моєму завдання необхідно створити тригери before update та before delete. Працювати ми будемо з таблицею Passenger  
 

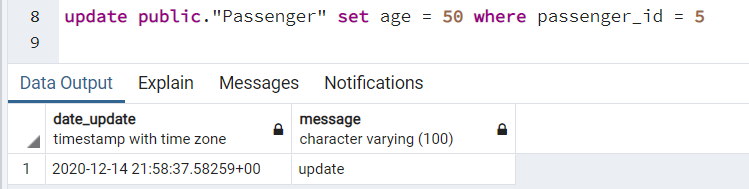
Створимо простий тригер, який при оновленні або видаленні рядка записує інформацію в таблицю passenger\_update.



Як видно з рисунка, ми запрограмували тригер, на те, щоб при оновленні запису у таблиці Passenger, у таблицю passenger\_update записувався рядок повідомленням про оновлення і датою оновлення. А при видаленні, відбувалася перевірка, якщо age < 20, то видалення не відбувається, в іншому випадку, запис видаляється, і відповідна відмітка з’являється у таблиці passenger

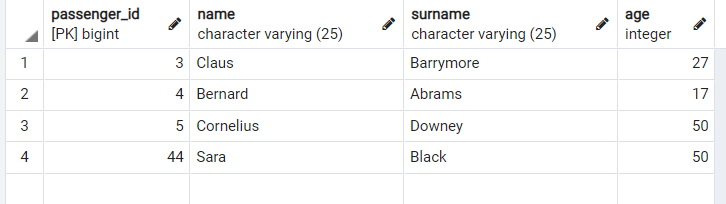
Розглянемо випадки роботи тригера.

1. Для випадку оновлення.

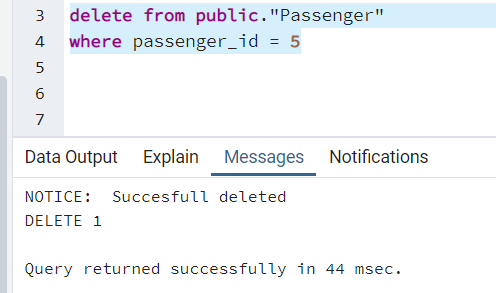


2. При випадку видалення рядку.

Початкова таблиця:



Коли дані задовольняють умову.



Та випадок коли age менше 20.

