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 III курсу

ФПМ групи КВ-83

Бусленко Віталій

Перевірив: Павловський В. І.

Київ – 2020

*Мета роботи:* здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

*Загальне завдання роботи полягає у наступному*:

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

Варіант 3

У другому завданні проаналізувати індекси GIN, Hash.

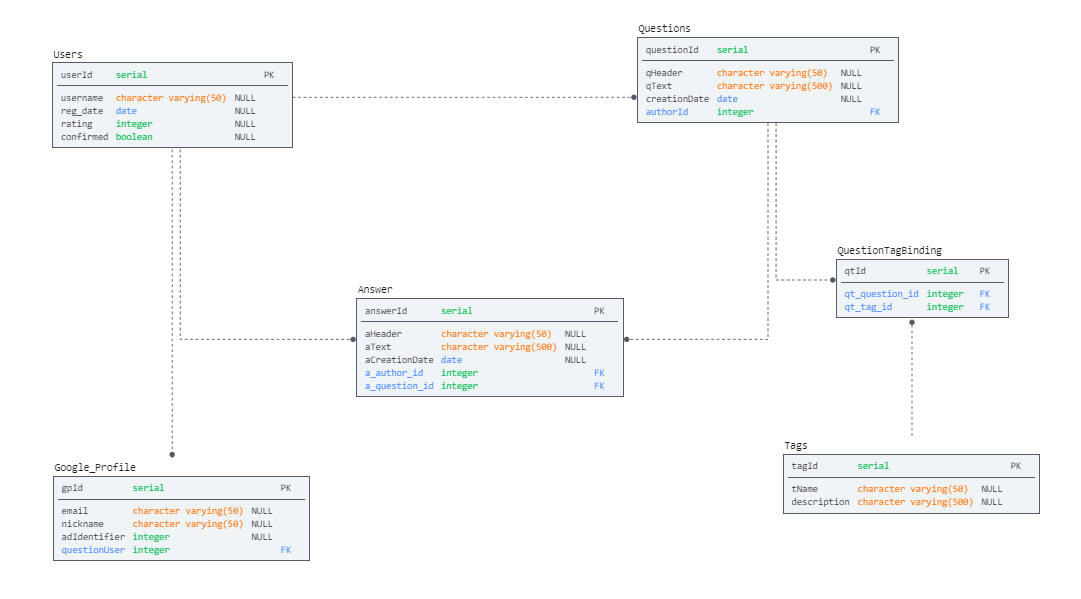
Умова для тригера – before delete, update.

Навігація по звіту

1. [Завдання 1.](#Завдання1)
2. [Завдання 2.](#завдання2)
3. [Завдання 3.](#завдання3)
4. [Програмний код.](#Ілюстрації)

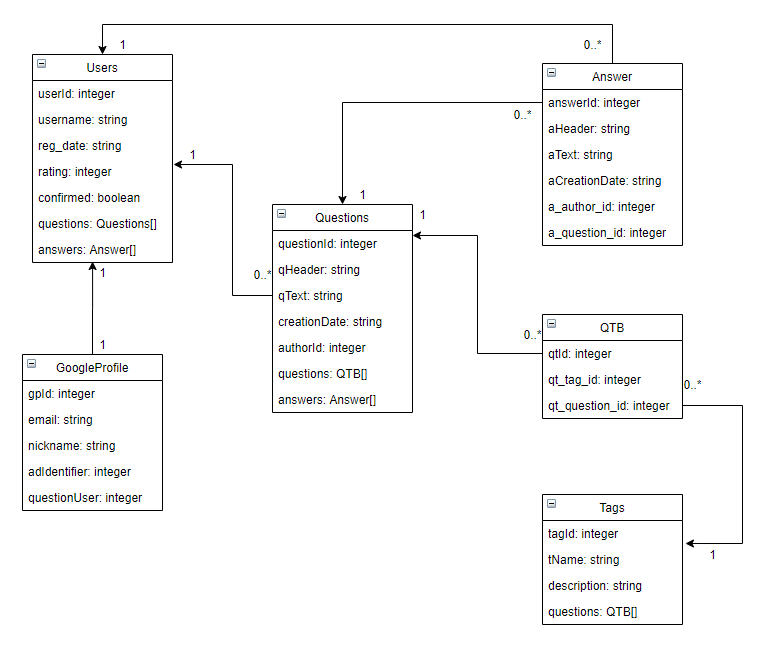
**Завдання 1**

Нижче (Рис. 1) наведено логічну модель бази даних:



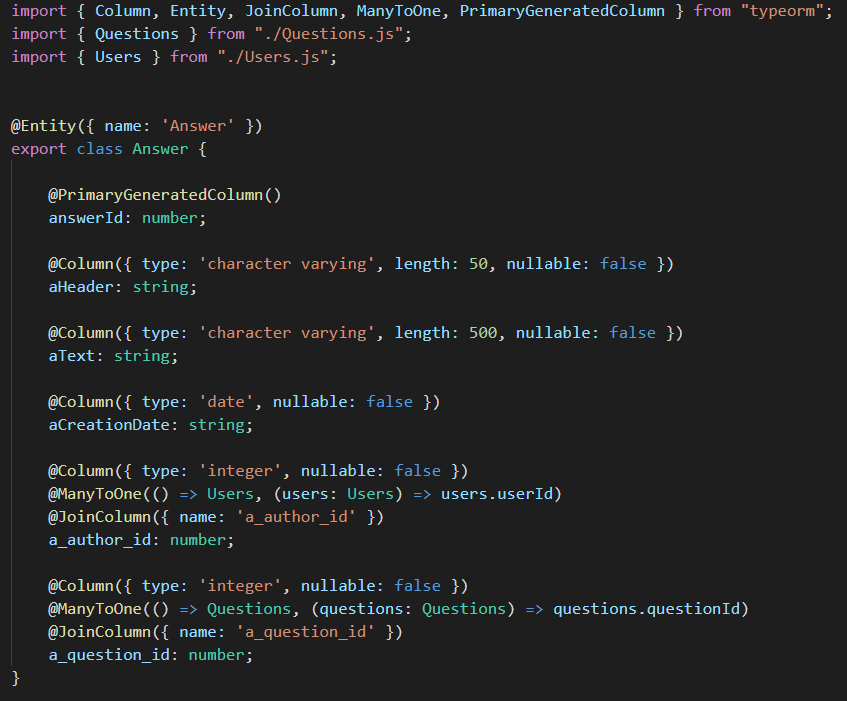
Для перетворення модулів моделей програми, створених в 2 лабораторній роботі, у вигляд об’єктно-реляційної моделі було використано бібліотеку TypeORM.

Зобразимо сутнісні класи програми:



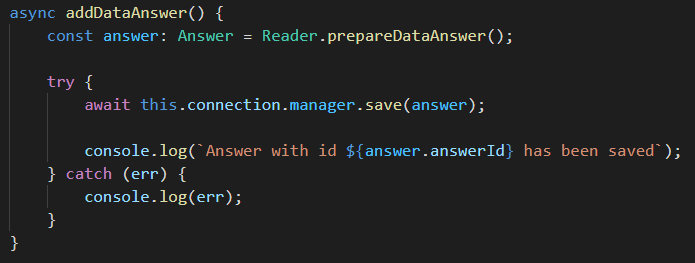
Продемонструємо код лише для одного класу Answer:

Класс Answer, де за допомогою декораторів бібліотеки TypeORM описується зв’язок класу із таблицею бази даних.

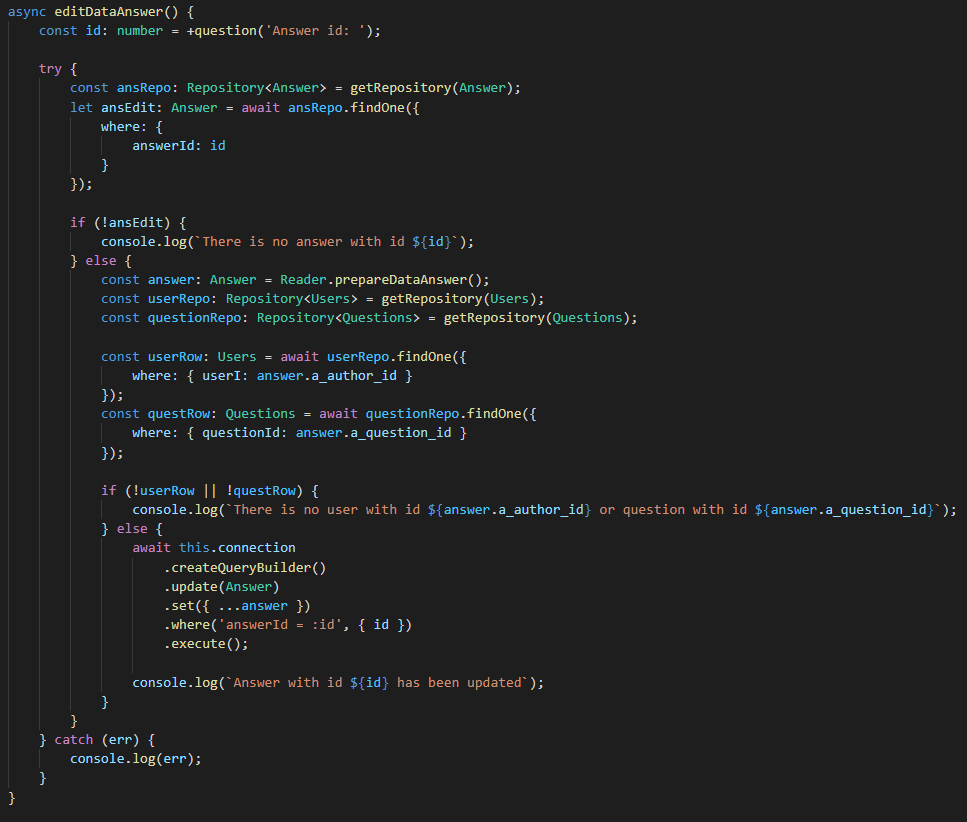


Кожний клас таблиці «обслуговує» клас «сервіс», що містить методи для роботи з відповідною таблицею.

Метод для вставки у таблицю:



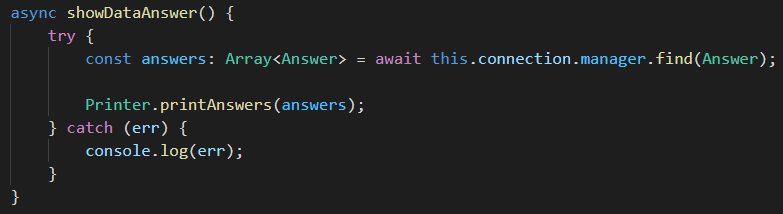
Метод для редагування запису:



Метод для видалення запису:

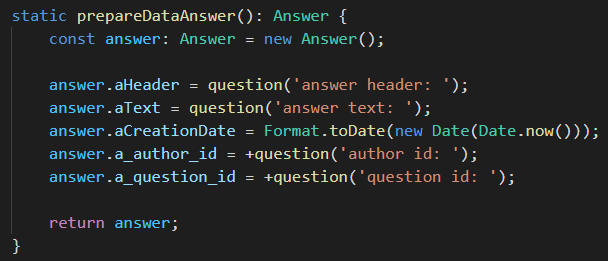


Метод для виведення вмісту таблиці:

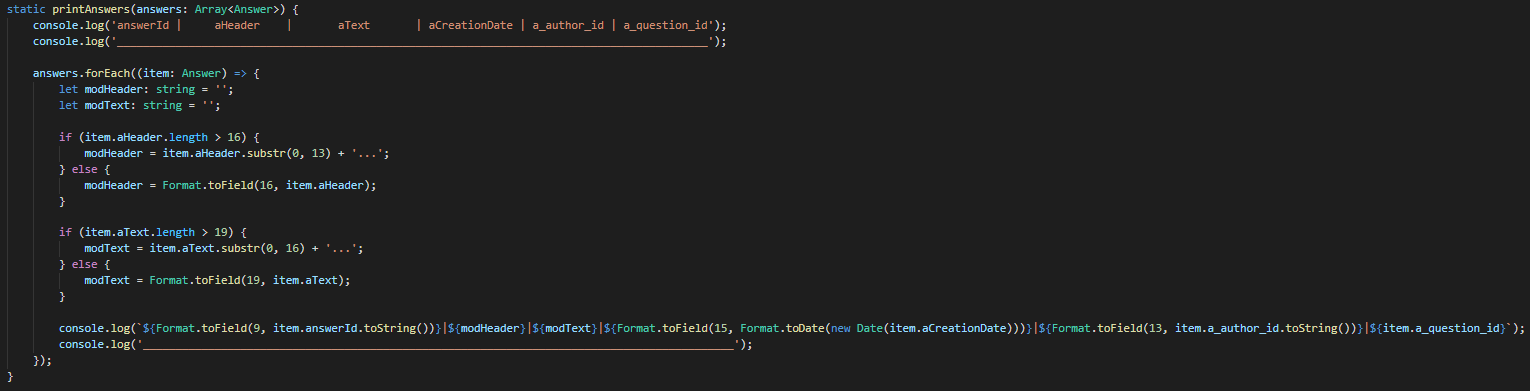


Окрім сервісів, також було створено допоміжні класи, використання методів частини з них(Printer, Reader) можна помітити у сервісах. Продемонструємо методи цих класів для роботи з поточним класом – Answer:

Метод класу Reader для зчитування інформації з консолі та формування об’єкту:



Метод класу Printer для форматування виводу даних із таблиці:



Інші класи та сервіси реалізовані по аналогії.

**Завдання 2**

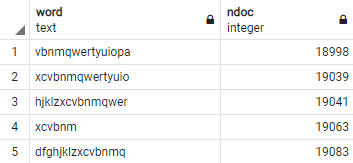
GIN

Для дослідження індексу була створена таблиця, яка має дві колонки типу text та одну колонку типу tsvector. У таблицю було занесено 1000000 записів.

Знайдемо слова, які зустрічаються найрідше:



Результат:



Здійснимо пошук по цих словах.



Результат:



У неіндексованій таблиці:



У індексованій таблиці:



Hash

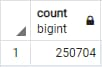
Багато сучасних мов програмування включають хеш-таблиці в якості базового типу даних. Зовні це виглядає, як звичайний масив, але в якості індексу використовується не ціле число, а будь-який тип даних (наприклад, рядок). Хеш-індекс в PostgreSQL влаштований схожим чином.

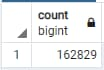
Для дослідження індексу була створена таблиця records, яка має дві колонки: record\_id типу integer та rec\_text текстового типу. Вони проіндексовані як Hash. У таблицю було занесено 1000000 записів.

Виконаємо запити для пошуку:



Результат:





У неіндексованій таблиці:





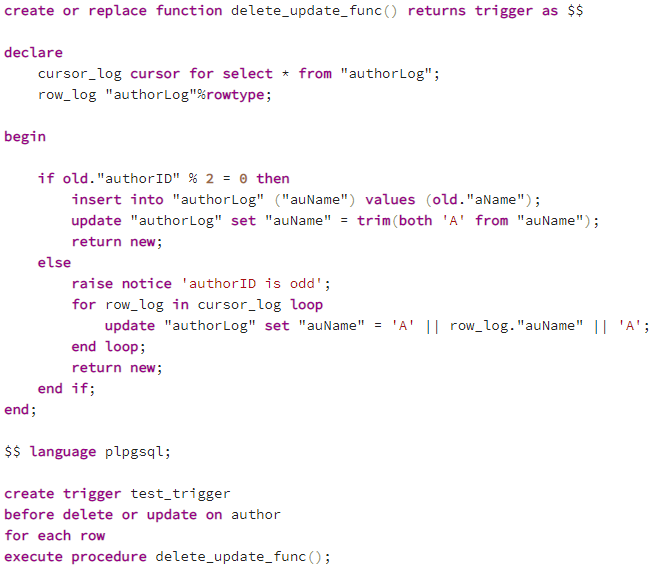
У індексованій таблиці:





**Завдання 3**

Тригер:



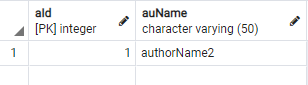
Принцип роботи

Тригер спрацьовує перед видаленням з таблиці чи при оновлені у таблиці author. Якщо значення ідентифікатора запису, який видаляється або оновлюється, парне, то цей запис заноситься у додаткову таблицю authorLog. Також, з кожного значення auName видаляються символи “A” на початку і кінці. Якщо значення ідентифікатора непарне, то до кожного значення auName у таблиці authorLog додається “A” на початку і кінці.

Видаляємо запис із author:



authorLog:



Оновлюємо запис:

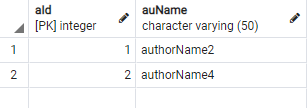




Повторно видаляємо запис:



authorLog:



**Ілюстрації програмного коду на Github**