

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# **Кафедра системного програмування та спеціалізованих комп’ютерних систем**

**Лабораторна робота №1**

з дисципліни

**«Бази даних і засоби управління»**

## **Тема:** «Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL»

Виконав: студент ІІI курсу

ФПМ групи КВ-94

Зінчук М. С.

Перевірив: Петрашенко А.В.

Київ 2021

**Загальне завдання:**

*У звіті щодо пункту №1 завдання має бути:*

* перелік сутностей з описом їх призначення;
* графічний файл розробленої моделі «сутність-зв’язок»;
* назва нотації.

*У звіті щодо пункту №2 завдання має бути:*

* опис процесу перетворення (наприклад, “сутність А було перетворено у таблицю А, а зв’язок R (M:N) зумовив появу додаткової таблиці R1 тощо);
* схему бази даних у графічному вигляді **з назвами таблиць (!) та зв’язками між ними.**

*У звіті щодо пункту №3 завдання має бути:*

* пояснення (**обґрунтування!**) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. Пояснення ***полягає у наведенні функціональних залежностей***, що демонструють висновки. У випадку невідповідності надати опис необхідних змін у схемі;
* У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

*У звіті щодо пункту №4 завдання має бути:*

* навести копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці (доступне у закладці “Columns” та “Constraints” властивостей “Properties” таблиць дерева об’єктів у pgAdmin4);
* навести копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL. Таблиці на зображенні обов'язково **повинні мати назву**!

**Варіант завдання:**

Блог(користувачі, дописи, коментарі, рейтинги);

**Завдання №1:**

**Перелік сутностей з описом їх призначення:**

В даній роботі в нас є три 4 сутності: Users, Posts, Comments, Likes.

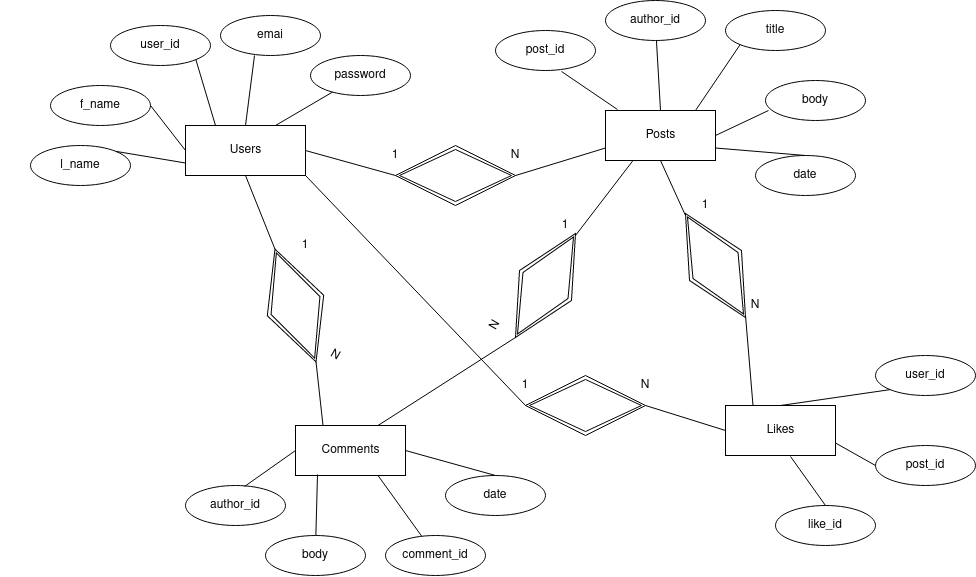
Перша сутність *Users* слугує для ведення обліку користувачів блогу, шляхом їх унікальної ідентифікації. Також для визначення даних користувача, таких як ім’я, прізвище, мейл.

Друга сутність *Posts* призначена для ведення обліку дописів, шляхом їх унікальної ідентифікації, та визначення дати допису, тіла допису, заголовку і автора допису.

Третя сутність *Comments* призначена для ведення обліку коментарів, шляхом їх унікальної ідентифікації, та визначення дати, автора, тексту коментаря і до якого посту він належить.

Четверта сутність *Likes* призначена для ведення обліку лайків, шляхом їх унікальної ідентифікації, та визначення поста до якого він відноситься, та юзера який поставив лайк.

**Графічний файл розробленої моделі «сутність-зв’язок»:**

****

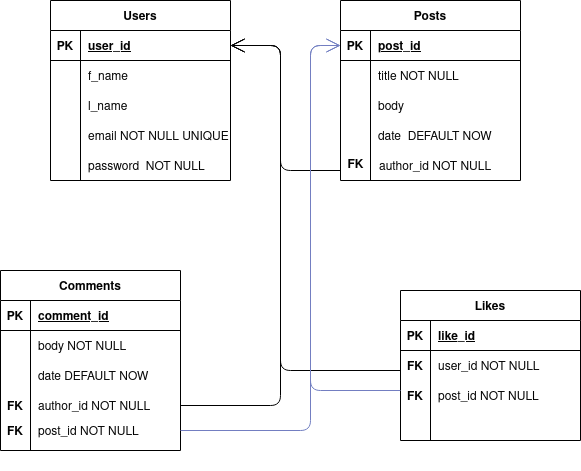
Використана нотація Чена

**Завдання №2:**

**Опис процесу перетворення:**

Сутність “Users” було перетворено у таблицю “Users”, сутність “Post” було перетворено у таблицю “Posts”, сутність “Comments” було перетворено у таблицю “Comments”, сутність “Likes” було перетворено у таблицю “Likes”. Переносимо всі атрибути, які належали сутностям у відповідні таблиці. Усі первинні ключі сутностей стають первинними ключами таблиць. Зв’язок 1:N є обов’язковим, тому первинний ключ сутності на стороні 1 додається атрибутом до сутності на стороні N та стає зовнішнім ключем.

**Схема бази даних у графічному вигляді:**



**Завдання №3:**

**Пояснення щодо відповідності схеми бази даних нормальним формам:**

Схема бази даних задовольняє умовам першої нормальної форми, оскільки всі атрибути в таблиці є атомарними, відсутні багатозначні та композитні атрибути, кожна таблиця має Primary key.

Схема бази даних задовольняє умовам другої нормальної форми, так як вона знаходиться в першій нормальній формі, та кожний не ключовий атрибут залежить від єдиного ключа.

Схема бази даних задовольняє умовам третьої нормальної форми, оскільки вона задовольняє умовам 2НФ та атрибути не залежать від інших атрибутів, які не є первинними ключами, тобто немає транзитивних функціональних залежностей не ключових атрибутів від ключових.

Users(user\_id, f\_name, l\_name, email, password)

Primary keys: user\_id.

Functional dependencies:

user\_id - > f\_name

user\_id - > l\_name

user\_id - > email

user\_id - > password

user\_id - > f\_name, l\_name, email, password

Posts(post\_id, title, body, date, author\_id)

post\_id - > title

post\_id - > body

post\_id - > date

post\_id - > author\_id

post\_id - > title, body, date, author\_id

Comments(comment\_id, body, date, author\_id, post\_id)

comment\_id - > body

comment\_id - > date

comment\_id - > author\_id

comment\_id - > post\_id

comment\_id - > body, date, author\_id, post\_id

Likes(like\_id, post\_id, user\_id)

like\_id - > post\_id

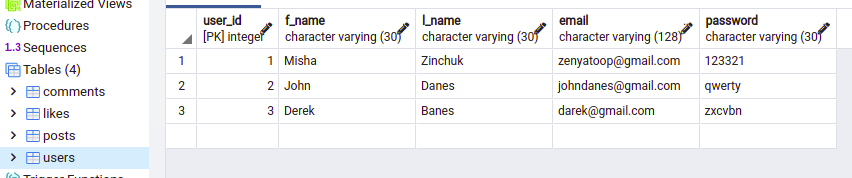
like\_id - > user\_id

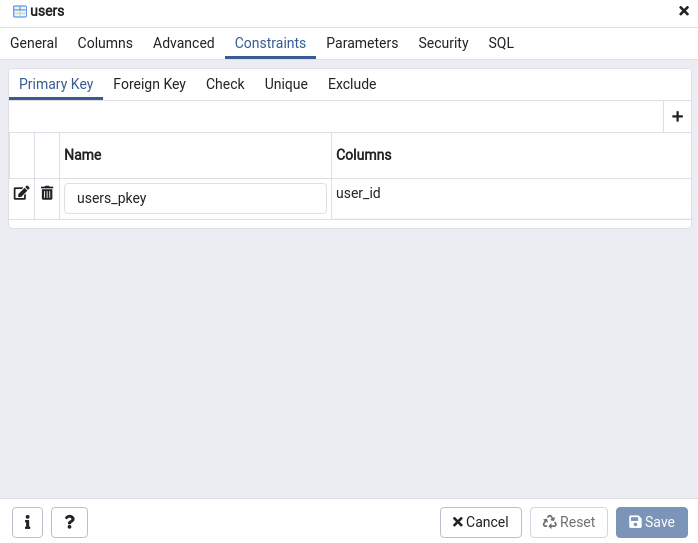
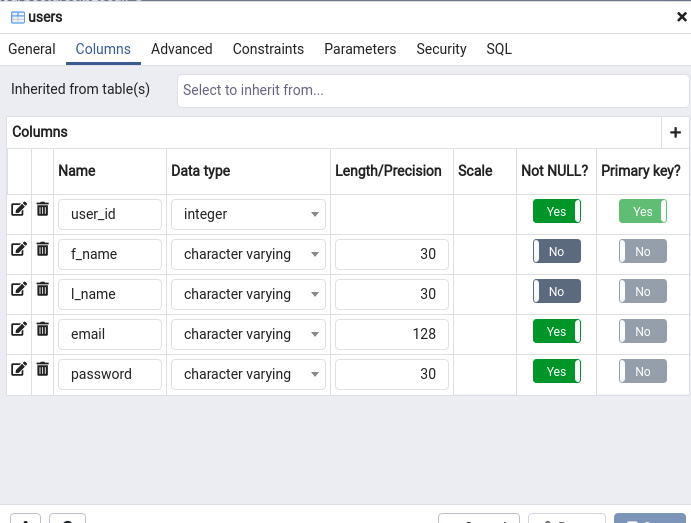
like\_id - > post\_id, user\_id

**Завдання №4:**

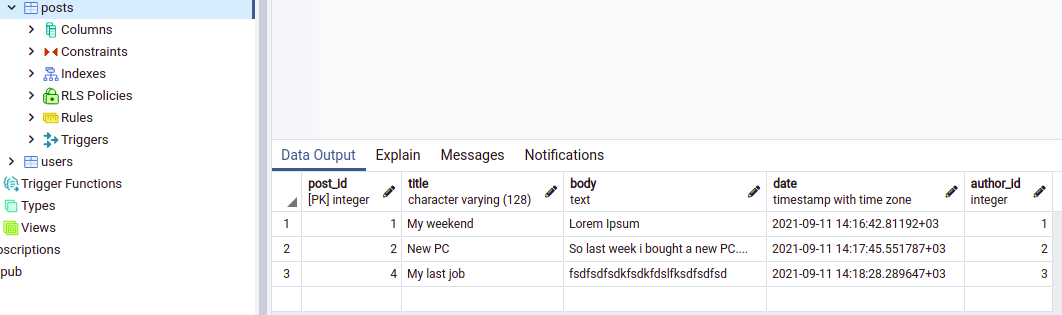
**Копії екрану з pgAdmin4, що відображають назви та типи стовпців та обмеження на стовпці:**

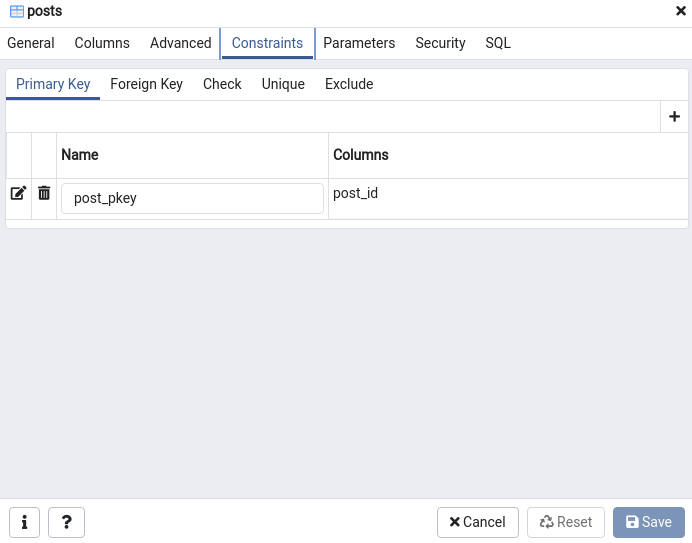
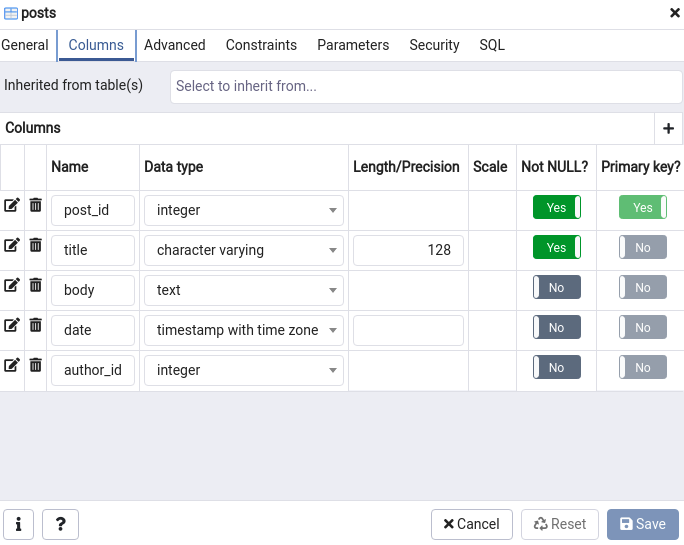
**Users:**

****

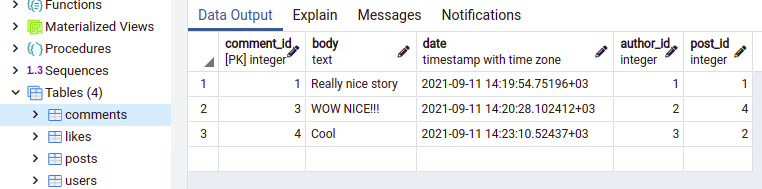
****

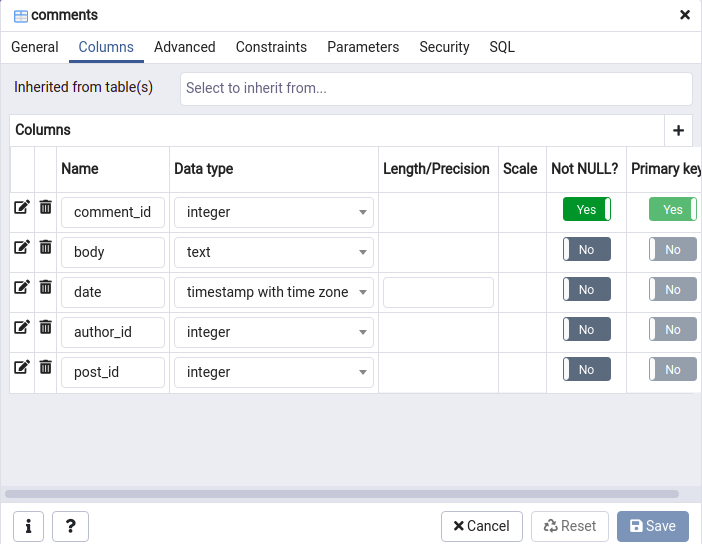
**Posts:**

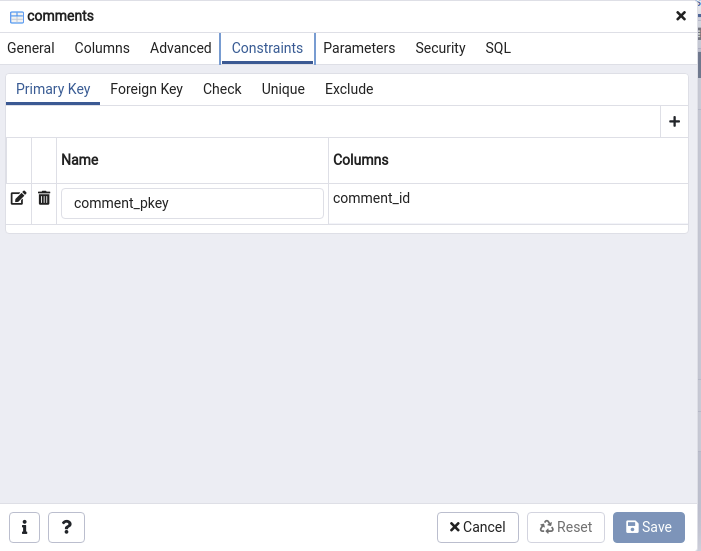




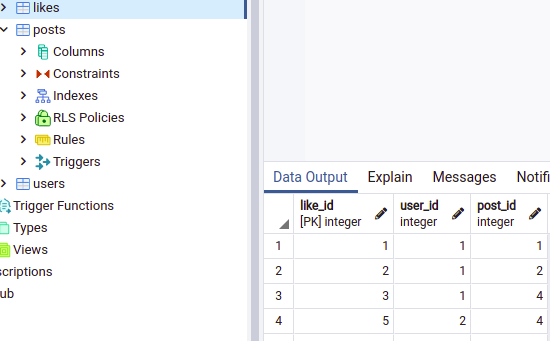
**Comments:**

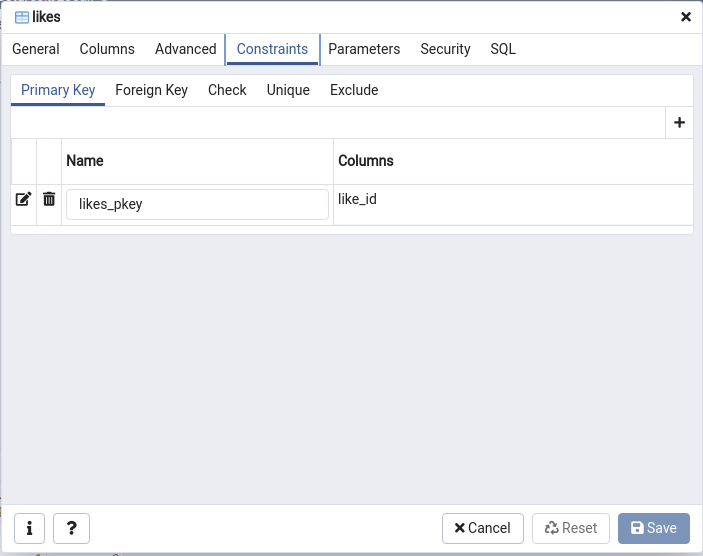
****

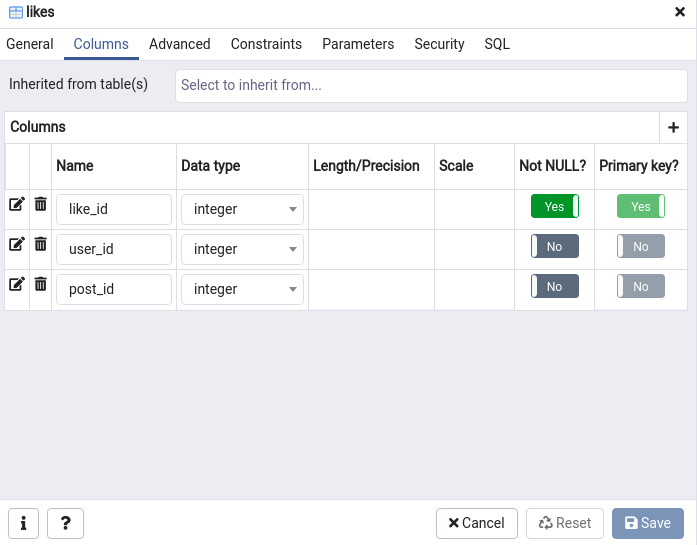
****

****

**Likes:**

****

****



**Контрольні запитання:**

Контрольні запитання

1. Сформулювати призначення діаграм типу «сутність-зв’язок».

Діаграми типу «сутність-зв’язок» слугують для створення моделі бази даних, перед її реалізацією. Вона дає змогу візуалізувати структуру бази даних, для подальшого її аналізу. Також це робиться для того, щоб мати змогу показати її клієнту, або людині яка не розбирається у програмуванні, щоб переконатися що всі вимоги були виконані.

1. Назвати основні об’єкти схеми PostgreSQL.

У PostgreSQL схема є простором імен, що містить іменовані об'єкти бази даних, такі як таблиці, подання, індекси, типи даних, функції, збережені процедури та оператори.

1. Навести приклади різних типів зв’язків у базах даних (1:1, 1:N, N:M).

- Один до одного: це тоді коли один екземпляр одного елемента може бути зв’язаний не більше ніж з одним екземпляром з другого елементу. Наприклад:

Одна людина може мати тільки один паспорт. Один автомобіль, може мати тільки один конкретний номерний знак.

- Один до багатьох: одна сутність асоціюється з багатьма іншими сутностями. Наприклад: користувач може мати багато постів, Сім’я може мати багато дітей.

- Багато до багатьох: Багато сутностей пов'язані з багатьма іншими сутностями. Наприклад: категорія товарів і товари(товар може належати до багатьох категорій, і в категорії може бути багато товарів), вибіркова дисципліна і студент(студент може мати багато вибіркових дисциплін, і вибіркову дисципліну можуть вибрати кілька студентів)