Изображение выглядит как текст

Автоматически созданное описание

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**

на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”

Виконала:

студентка ІII курсу

групи КВ-94

Романенко М. В.

Перевірив:

Петрашенко А. В.

Київ – 2021

*Метою роботи* є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

*Завдання* роботи полягає у наступному:

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Завдання 1** – модель «сутність-зв’язок» галузі постів якоїсь соціальної мережи

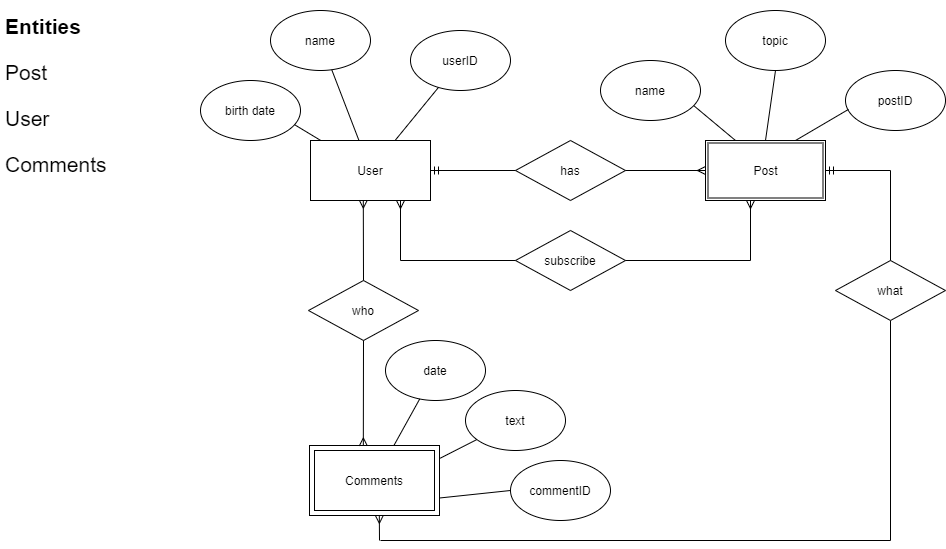


Рисунок 1. ER-діаграма, побудована за нотацією "Пташиної лапки"

**Сутності, їх призначення та опис зв'язків:**

Маємо три сутності: User, Post, Comments. Сутність User описує користувачів соціальної мережи. Кожен користувач має ім’я, дату народження та свій ID. Сутність Post – це пости цих самих користувачів. Тобто кожен окремий користувач може мати скільки завгодно постів в своєму блозі, але скільки завгодно користувачів можуть бути підписаними на ці пости. Кожний пост також має свій ID, назву поста та тему, на яку написано пост. Третя сутність – це Comments. Це коментарі під постами. Коментар має два зв’язки: чий коментар (користувача) та що це за коментар (тобто під яким саме постом). Багато користувачів можуть написати скільки завгодно коментарів, а також під одним постом може бути багато коментарів. Коментар має свій ID, свою дату та сам текст.

**Завдання 2** – перетворення графічного вигляду бази даних у схему бази даних PostgreSQL

Изображение выглядит как текст, снимок экрана, квитанция

Автоматически созданное описание

Рисунок 2. Схема бази даних у графічному вигляді

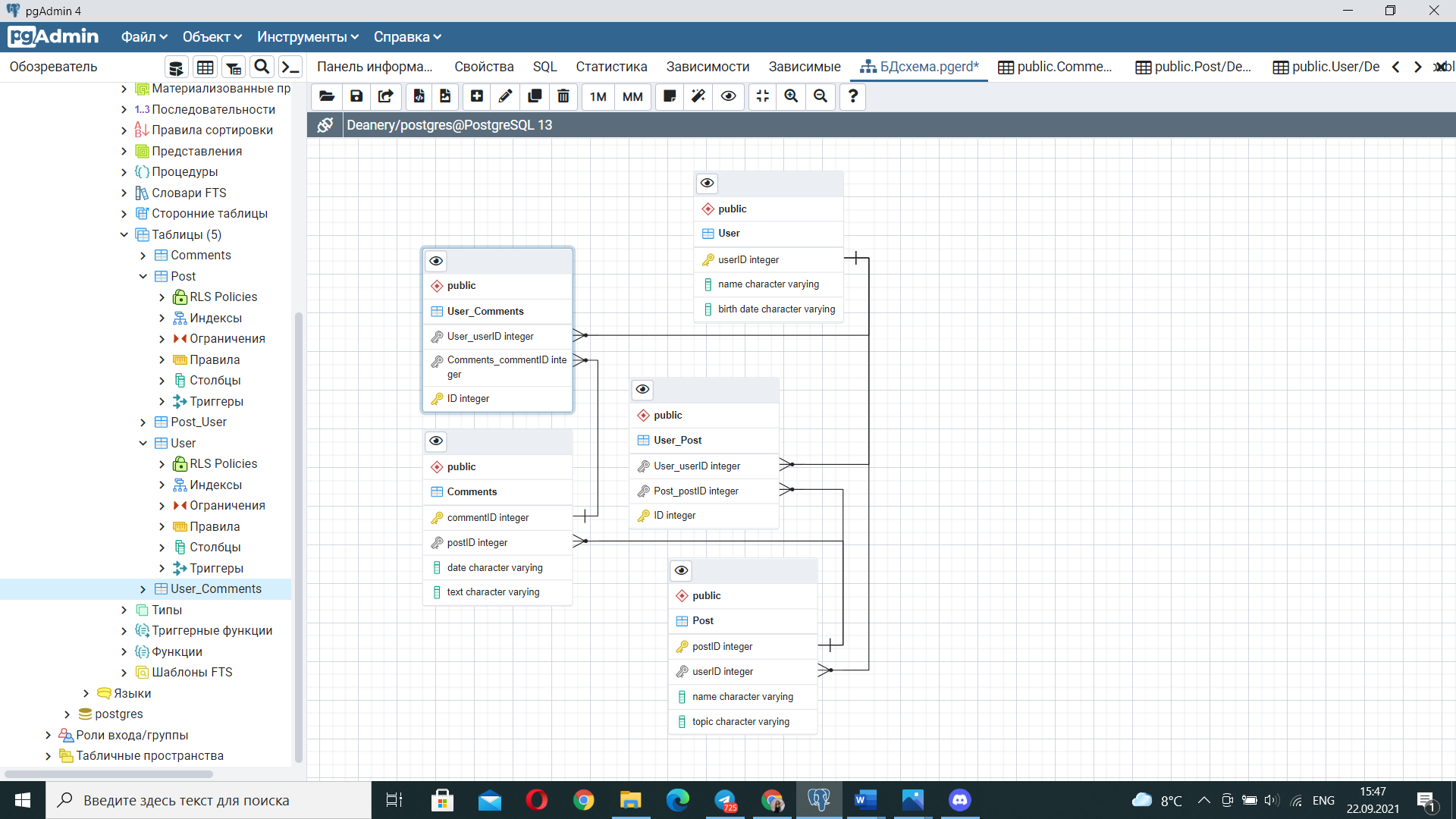


Рисунок 3. Схема бази даних у pgAdmin 4

**Опис процесу перетворення**

Сутність User було перетворено в таблицю User. Сутність Post було перетворено в таблицю Post, зв’язок has (1:N) із сутністю User зумовив появу у ній зовнішнього ключа userID. Сутність Comments було перетворено в таблицю Comments, зв’язок what (1:N) із сутністю Post зумовив появу у ній зовнішнього ключа postID. Зв’язок subscribe (N:M) зумовив появу додаткової таблиці User\_Post. Зв’язок who (N:M) зумовив появу додаткової таблиці User\_Comments.

**Завдання 3** - нормалізацію схеми бази даних до третьої нормальної форми

**Функціональні залежності:**

User (*userID*, name, birth date):

userID → name

userID → birth date

userID → name, birth date

Post (*postID*, userID, name, topic):

postID → userID

postID → name

postID → topic

postID → userID, name, topic

Comments (*commentID*, postID, date, text):

commentID → postID

commentID → date

commentID → text

commentID → postID, date, text

User\_Post (User\_userID, Post\_postID, *ID*):

ID → Post\_postID

ID → User\_userID

ID → Post\_postID, User\_userID

User\_Comments (User\_userID, Comment\_commentID, *ID*):

ID → Comment\_commentID

ID → User\_userID

ID → Comment\_commentID, User\_userID

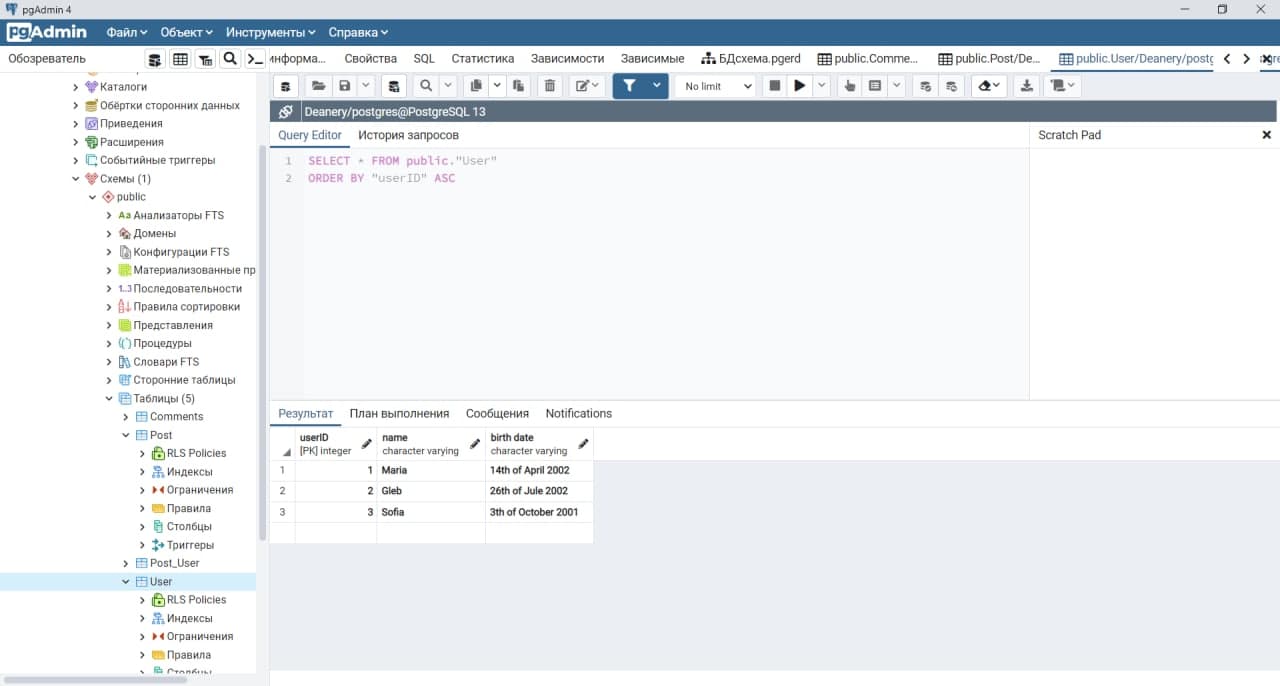
Схема бази даних відповідає нормальній формі НФ1, тому що всі атрибути таблиці є атомарними, кожна таблиця має primary key та кожна таблиця має мінімальний набір атрибутів.

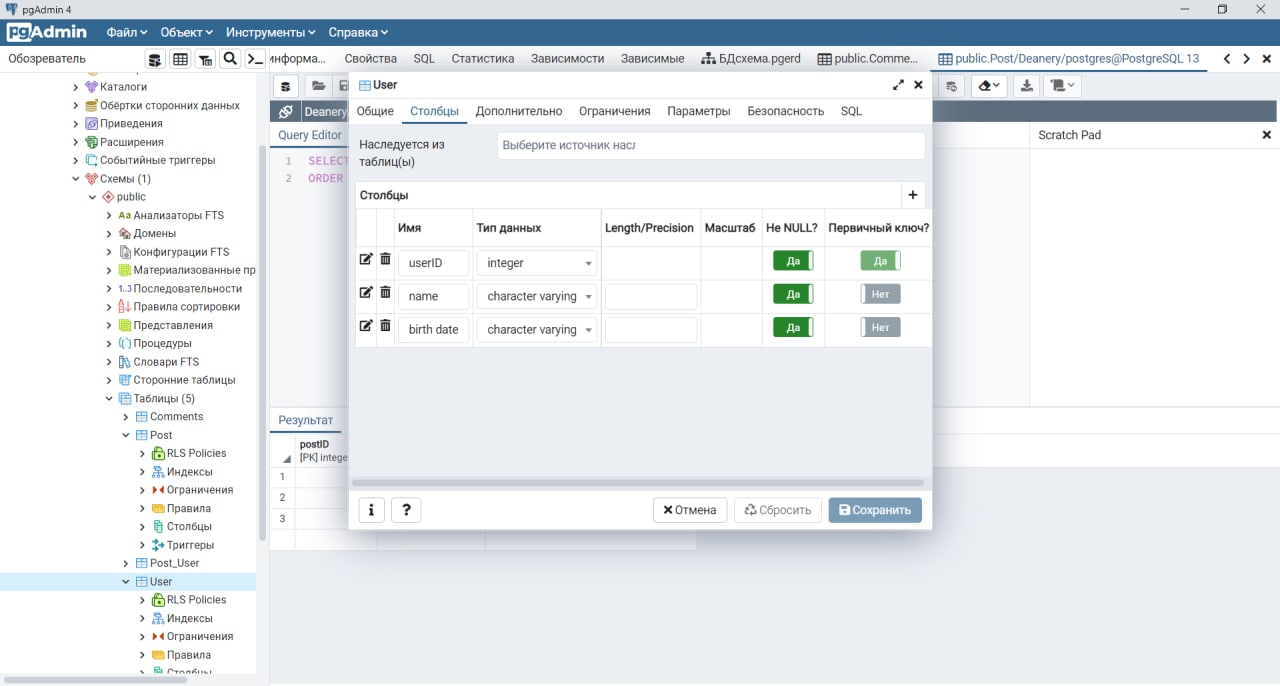
Схема бази даних відповідає нормальній формі НФ2, тому що вона відповідає нормальній формі НФ1 і кожен неключовий атрибут функціонально залежить від цілого ключа, а не від його частини.

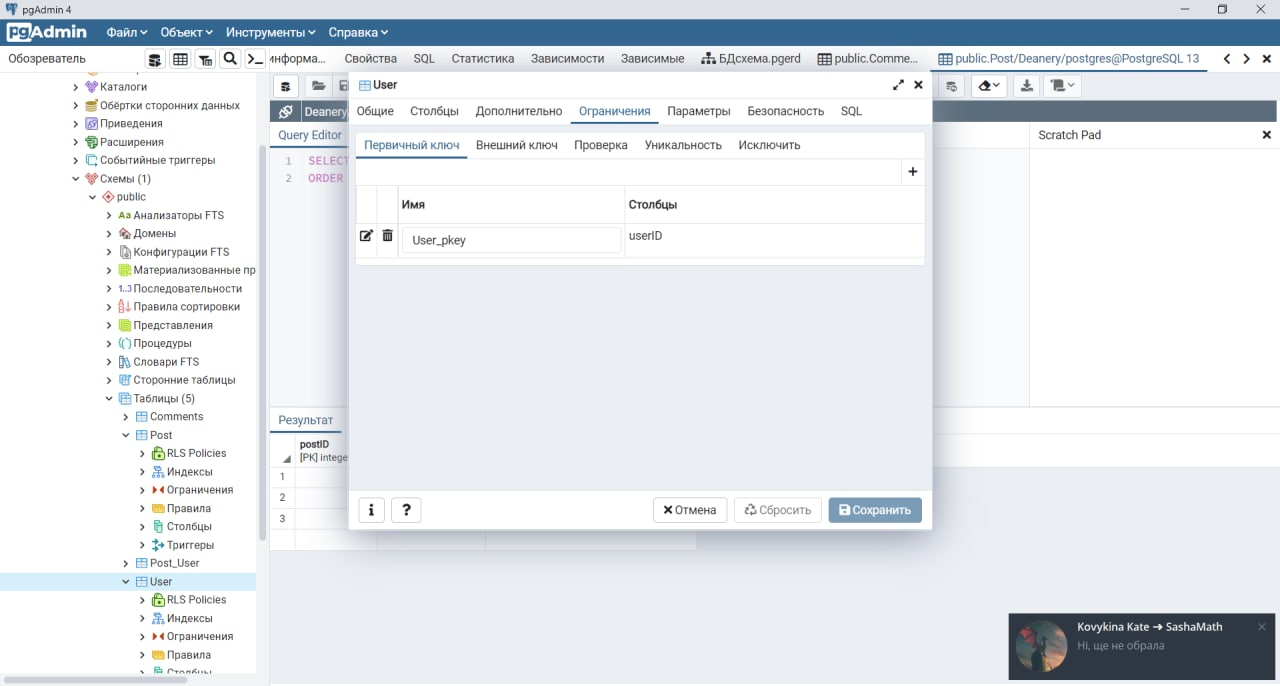
Схема бази даних відповідає нормальній формі НФ3, тому що вона відповідає нормальній формі НФ2 і дані в таблиці залежать від primary key.

**Завдання 4** – таблиці бази даних у pgAdmin 4

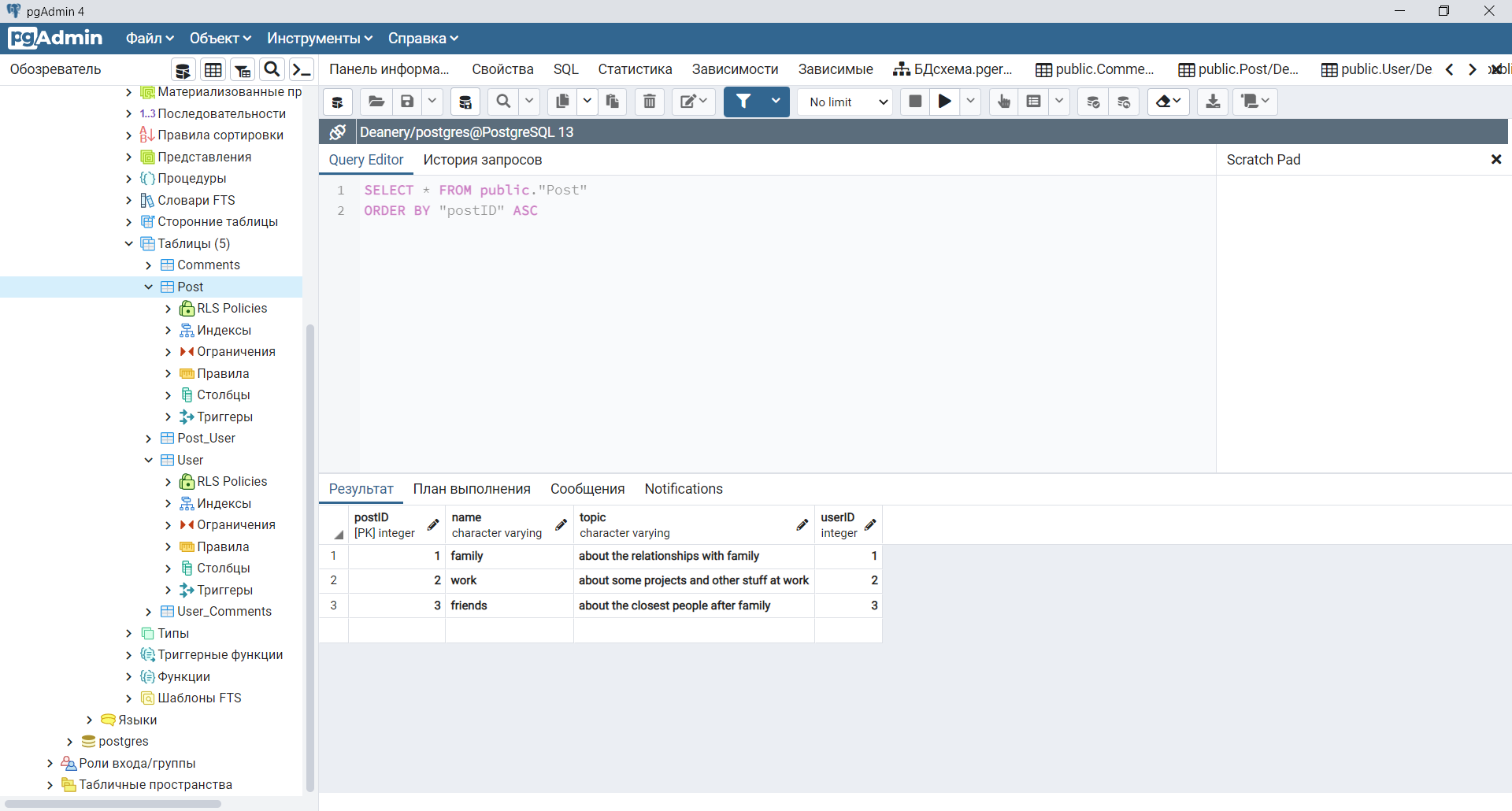
User:

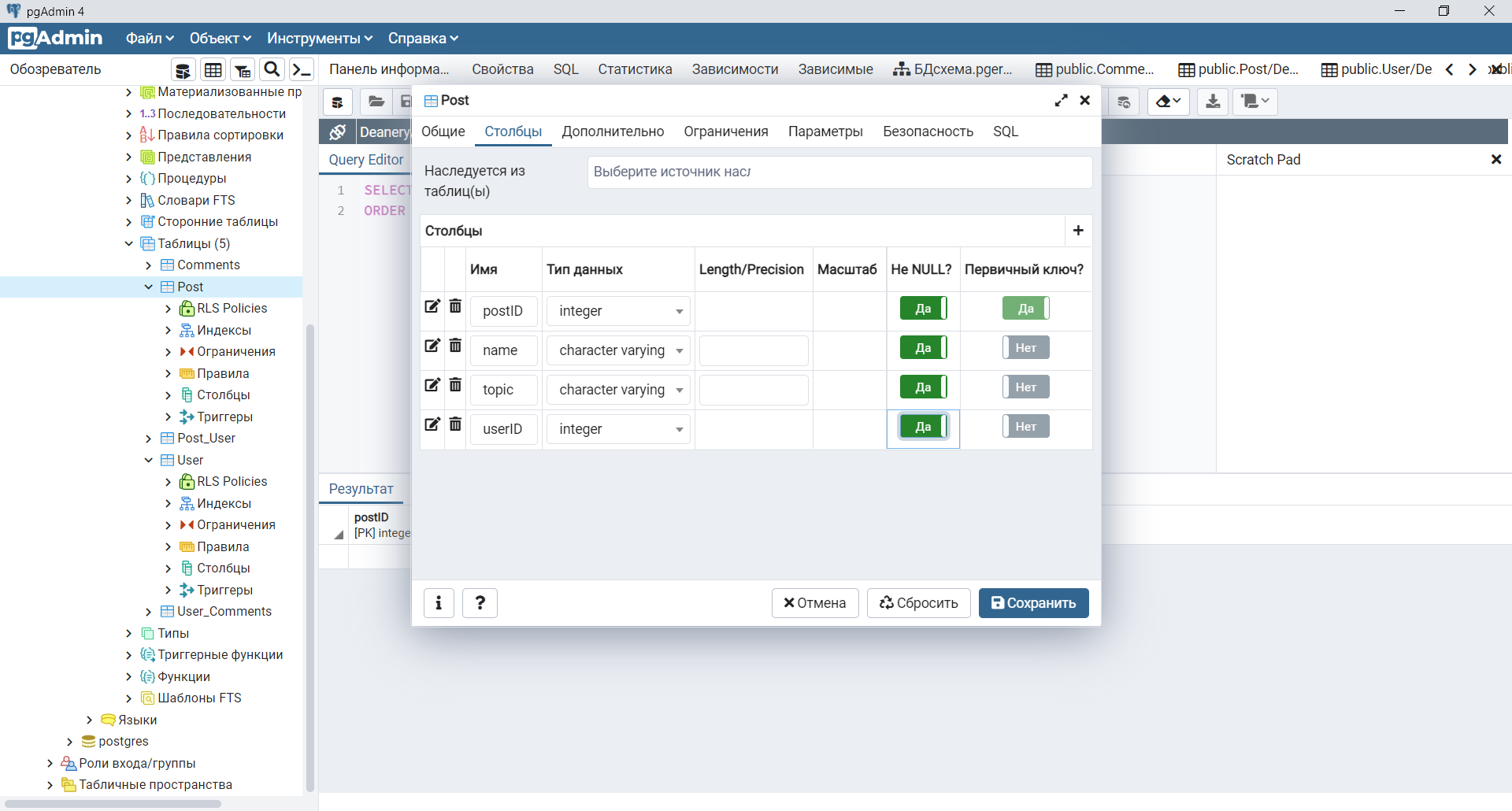


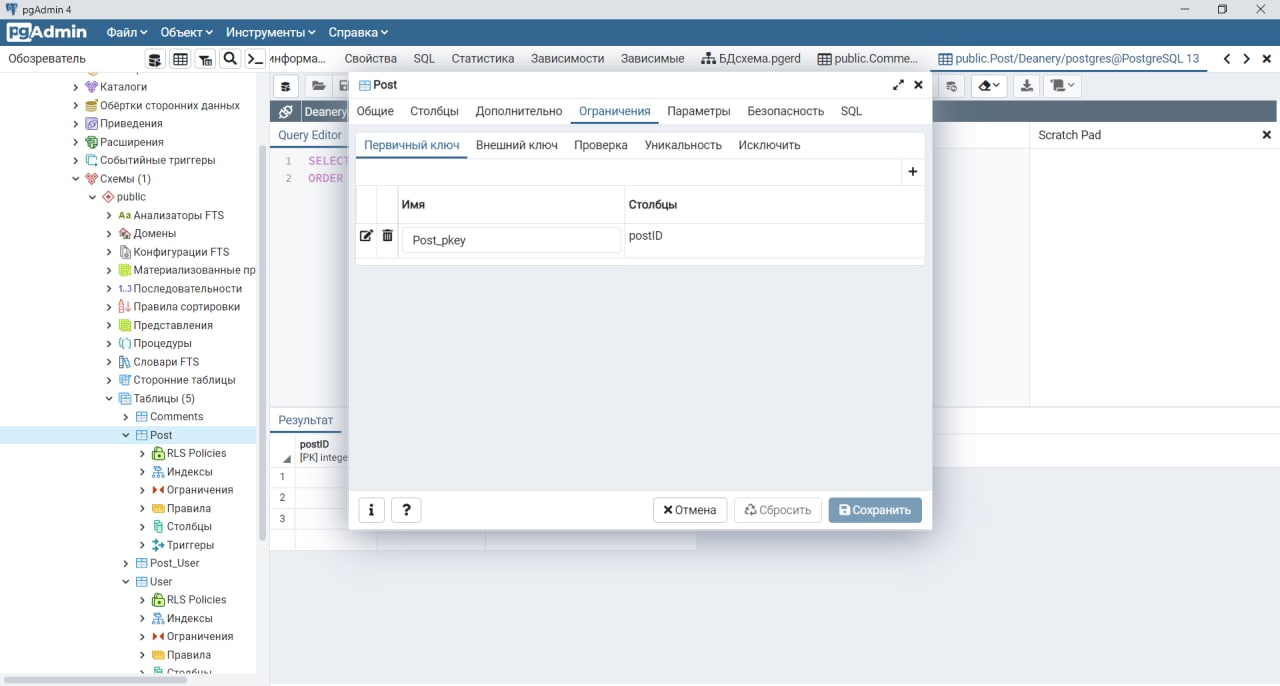


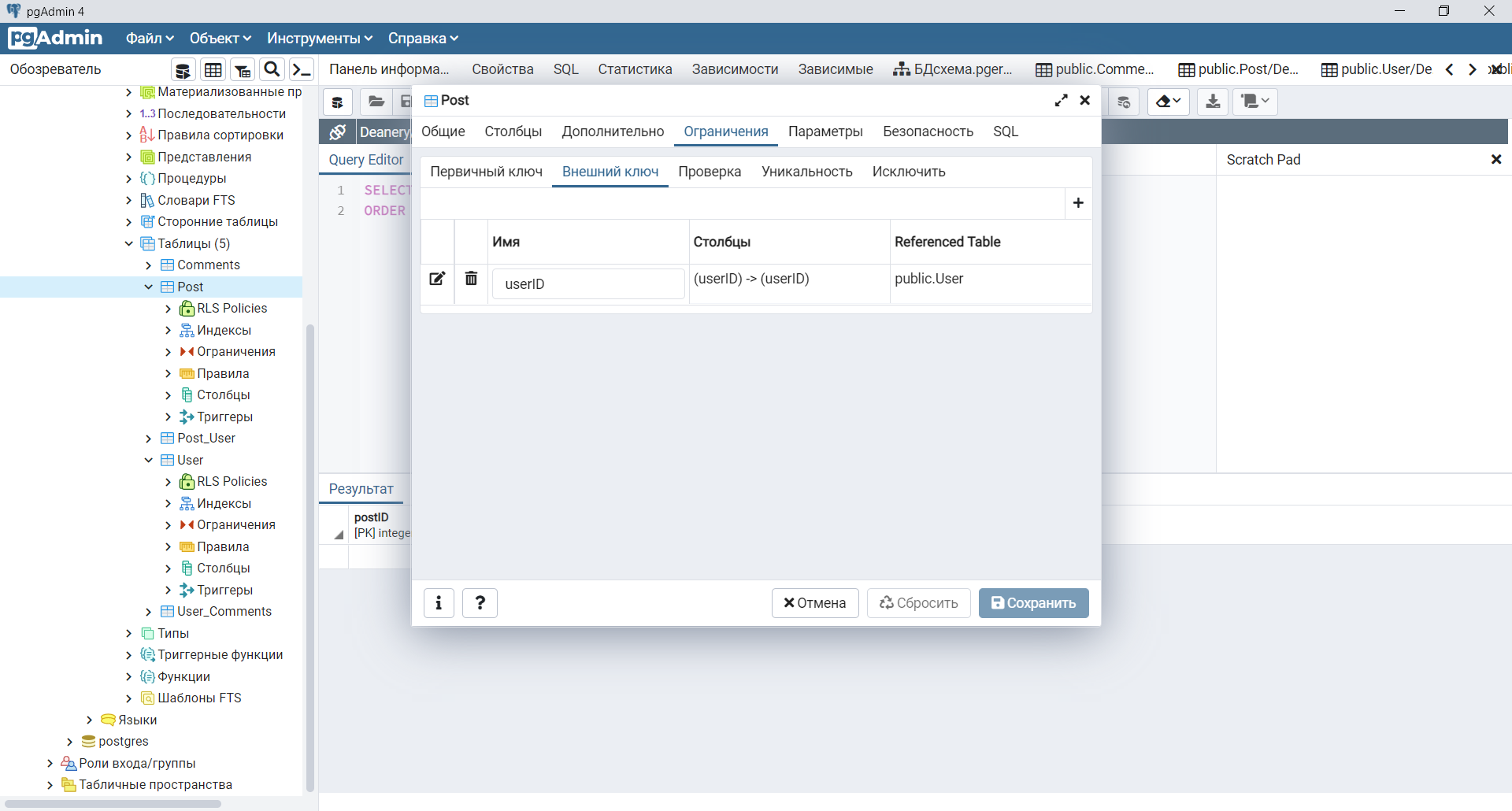


Post:

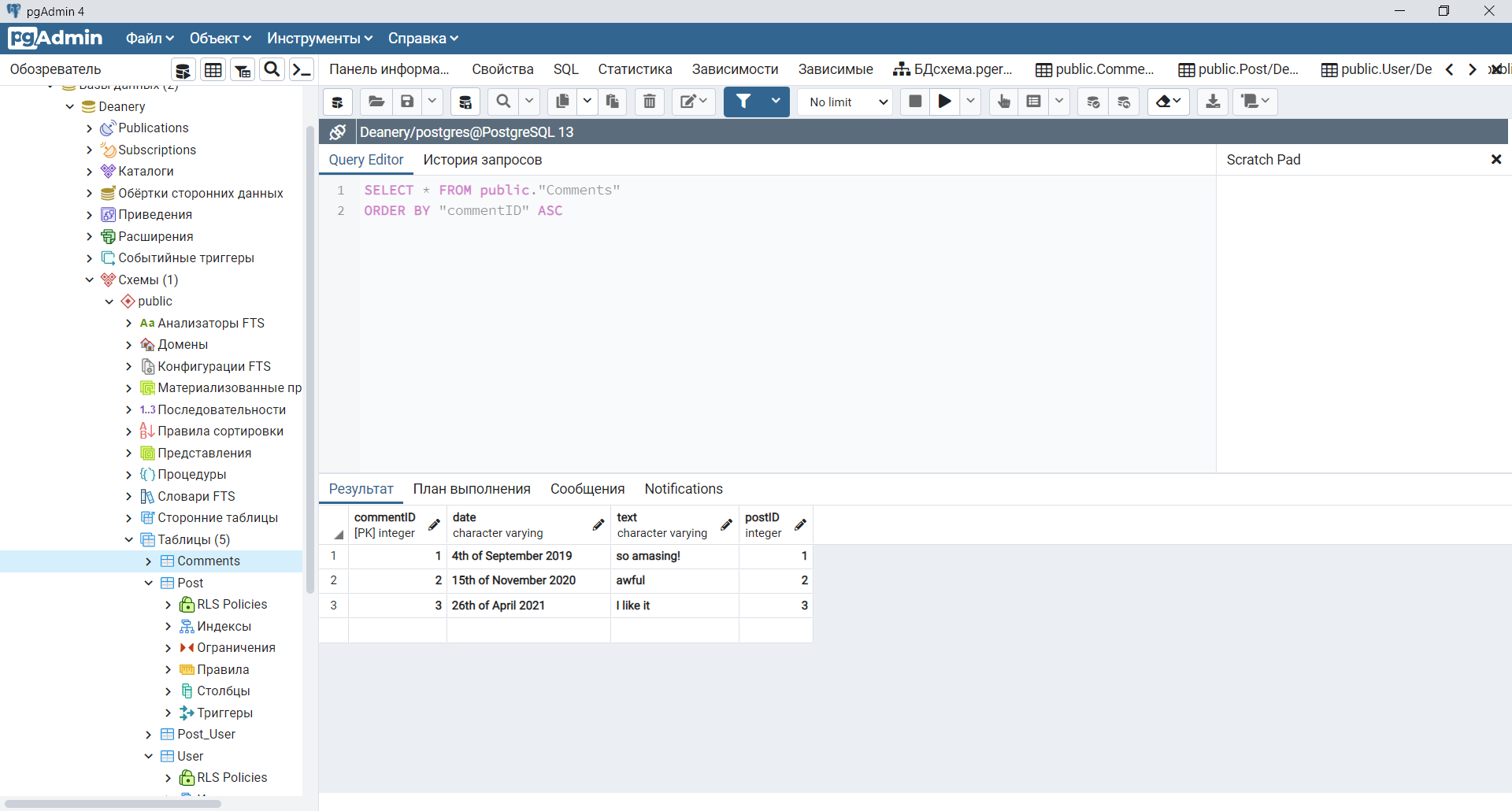


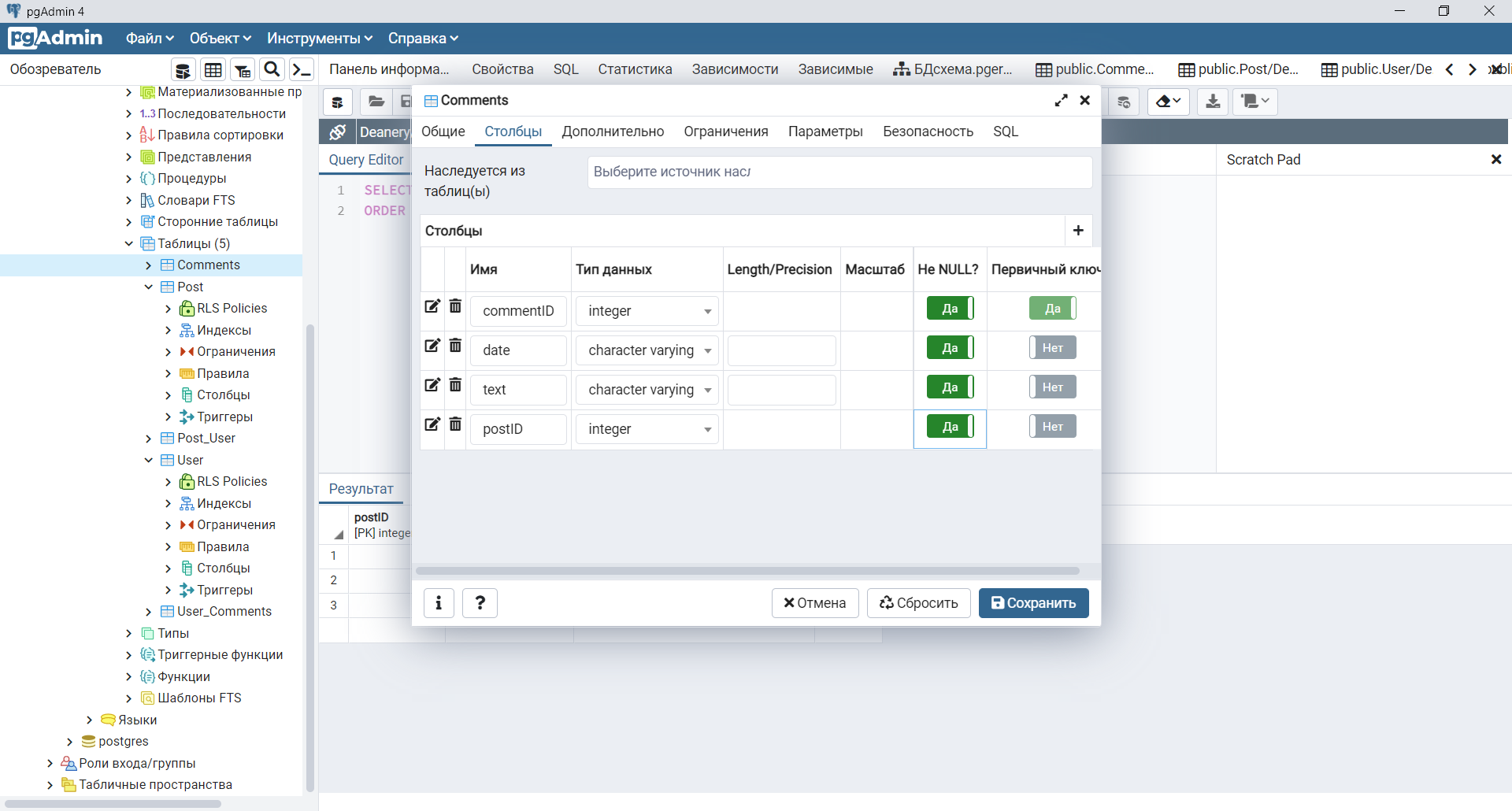


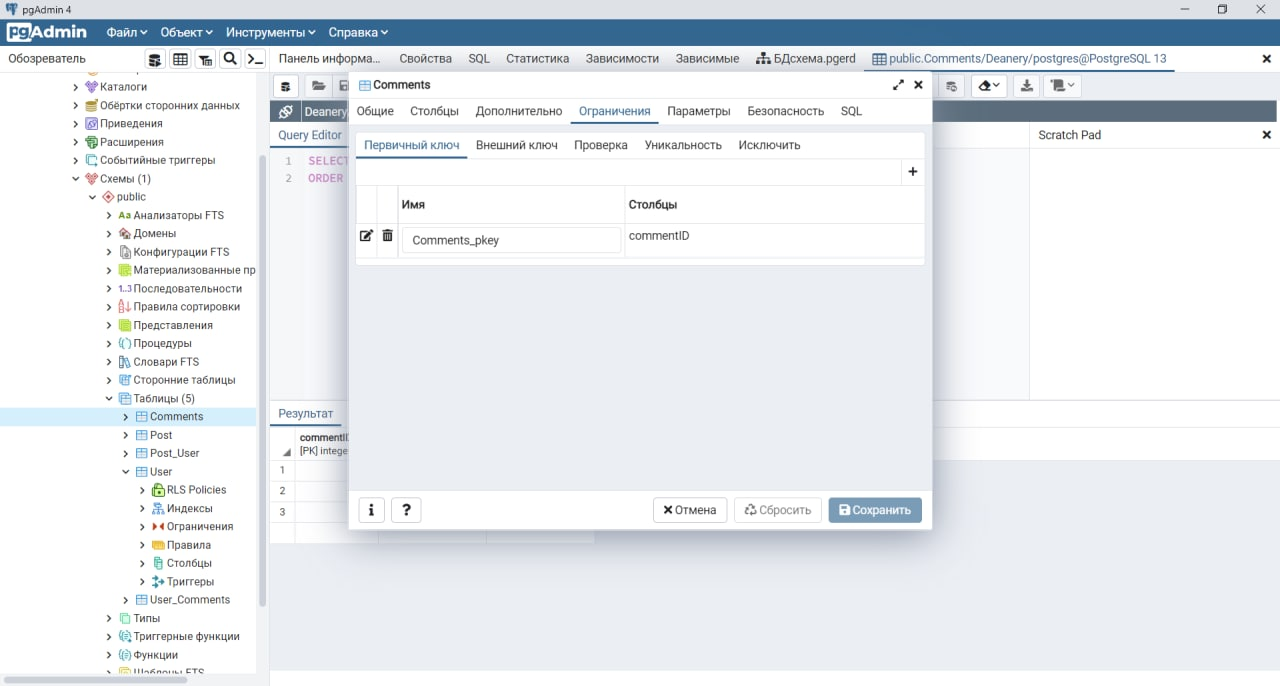


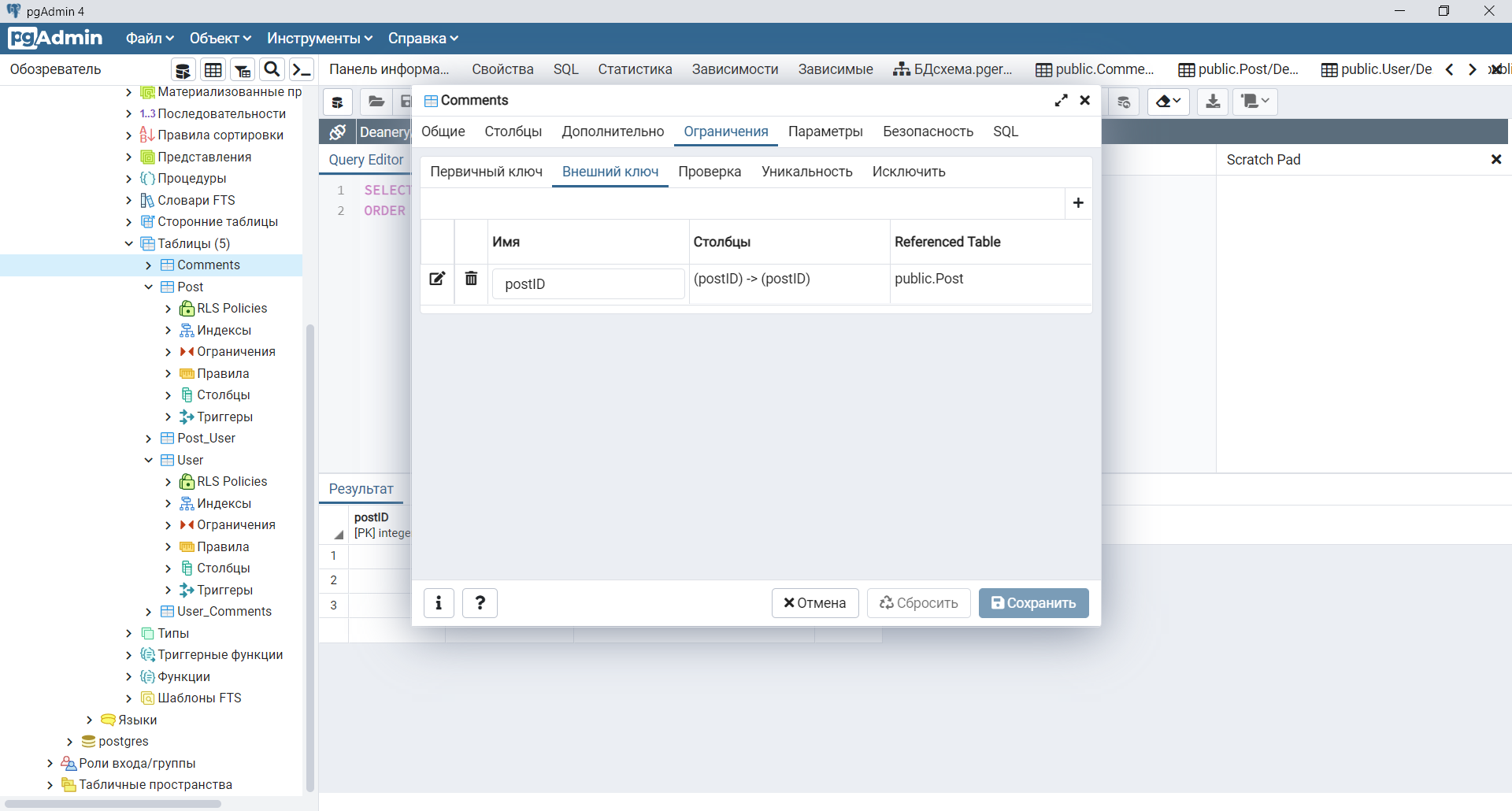


Comments:

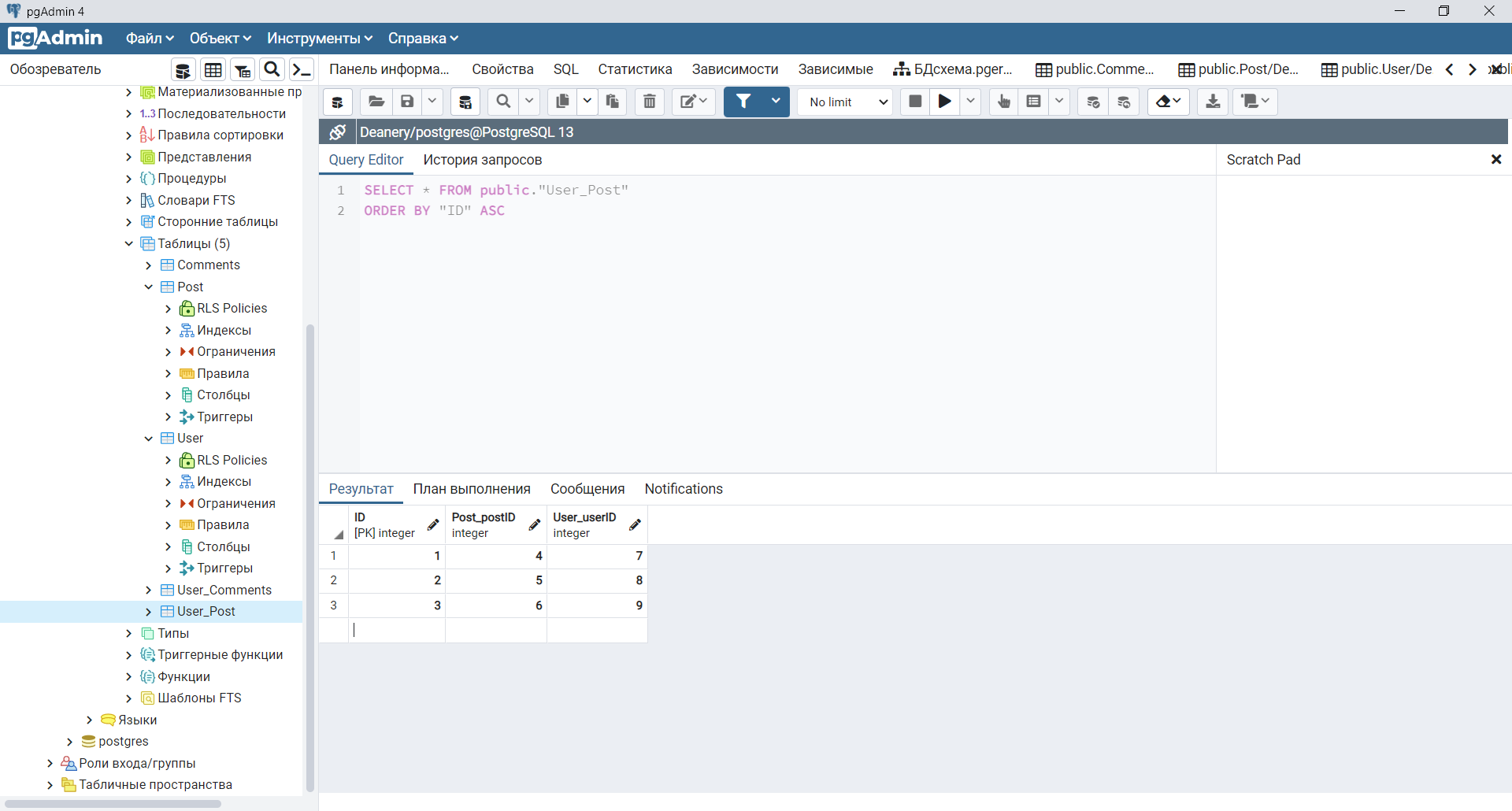


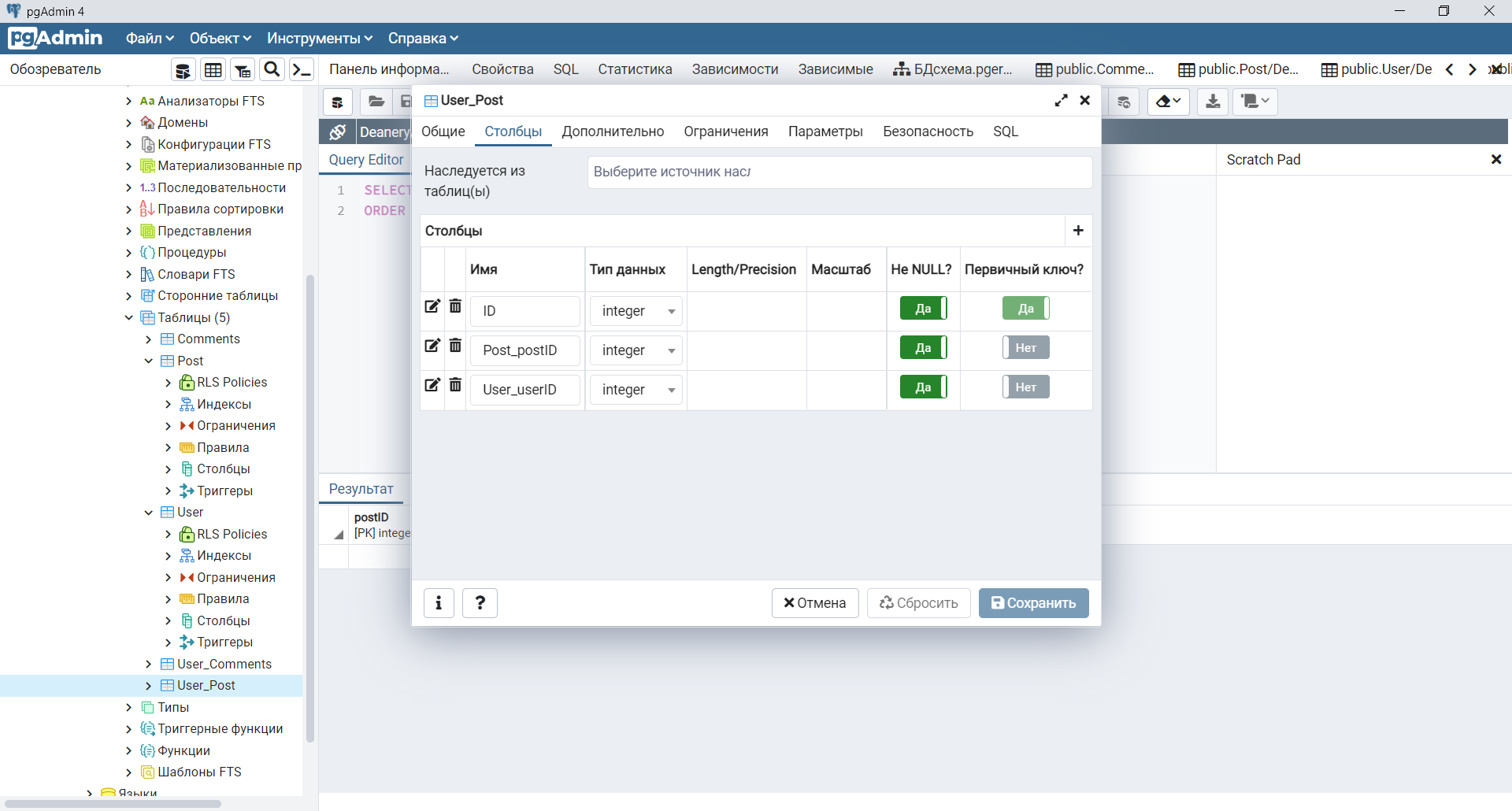


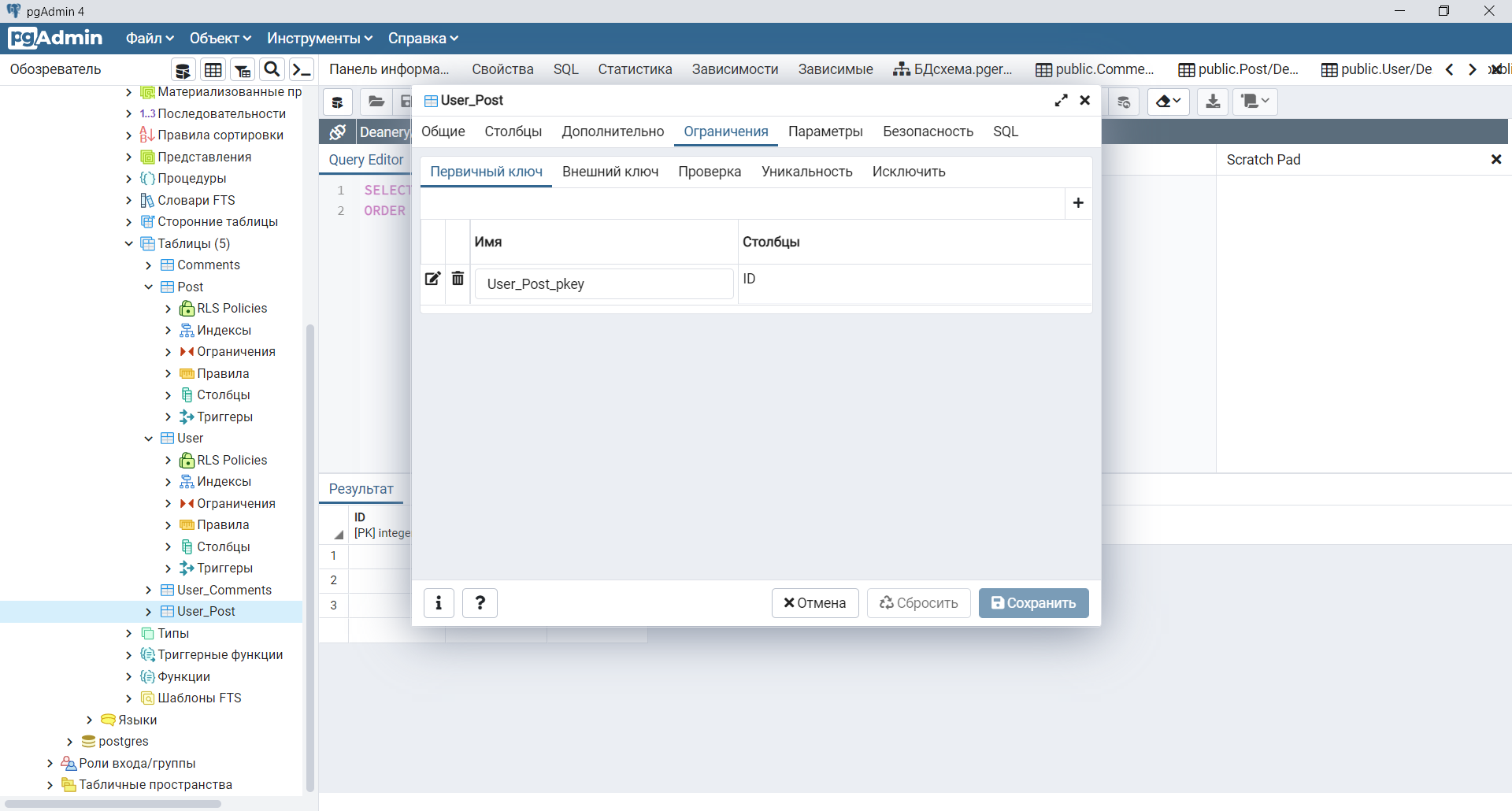


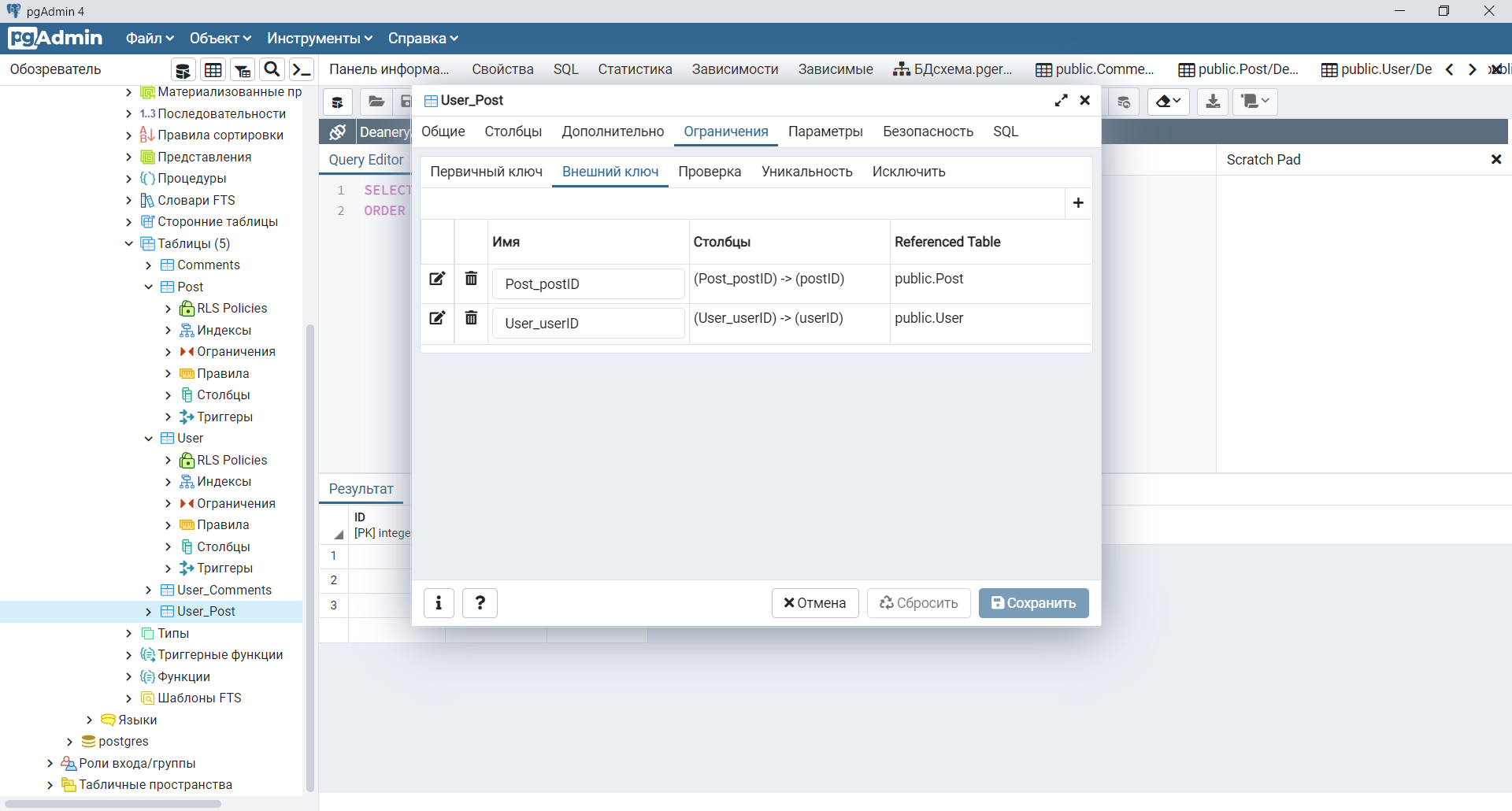


User\_Post:









User\_Comments:

