

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”

Виконав:

студент ІII курсу

групи КВ-12

Пеленов І. О.

Перевірив: Павловский В. І.

2023

*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 кожному з таблиць засобами pgAdmin 4.

**Модель «сутність-зв’язок» галузі електронний каталог для зберігання та публікації наукових досліджень.**

**Сутності**

Для побудови бази даних було виділено наступні сутності:

**Автор (Author)** з атрибутами: ім’я автора, організація в якій перебуває автор, спеціалізація автору, код автора для зручності. Призначення: зберігати інформацію про автора наукової роботи

**Спеціалізація (specialization)** з атрибутами: галузь науки, освіта відображення рівня освіти, код спеціальності. Призначення: зберігати інформацію про спеціалізацію.

**Наукова\_робота (research\_work)** з атрибутами: код роботи, код теми, назва наукової роботи, дата публікації роботи. Призначення: зберігати інформації про саму наукову роботу.

**Тема (theme)** з атрибутами: код теми, терміни використані в роботі, опис наукової роботі, галузь в якій наукова робота. Призначення: зберігати інформацію про тему наукової роботи.

**Опис зв’язків**

Автор може мати декілька спеціальності, також на одну спеціалізацію може припадати одночасно декількох людей, Тому Автор та Спеціалізація мають відношення N:M.

Автор може написати декілька наукових робіт. А наукова робота може бути написати в співавторстві з іншими, але при цьому повинний бути, як мінімум, один автор. Тому Автор та Наукова робота має відношення N:M.

Наукова робота може мати лише одну тему. При цьому на одну тему можуть написати декілька наукових робіт. Тому Тема та Наукова робота має відношення 1:N.

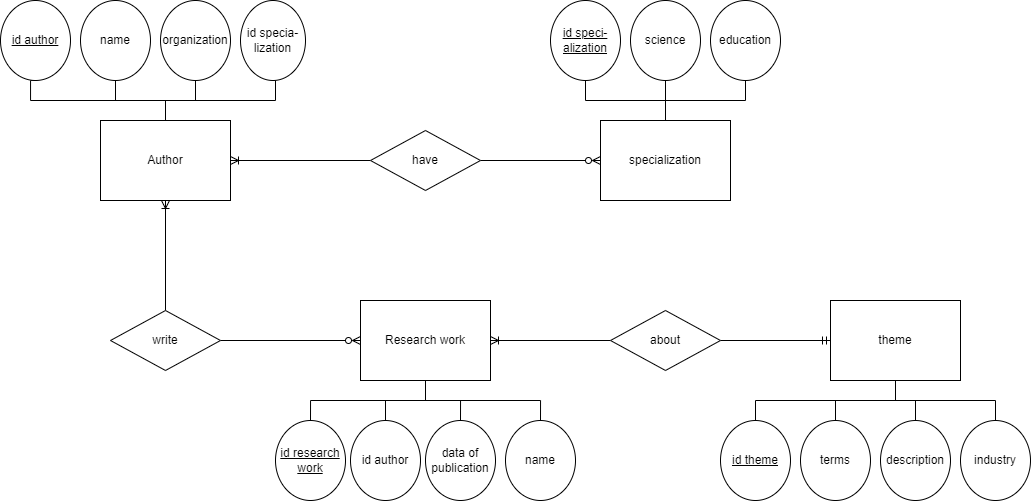


Рисунок 1 – ER-діаграма

**Перетворення концептуальної моделі у логічному модель та схему бази даних**

В логічній моделі неможливо представити зв’язок M:N, а в концептуальній моделі є відразу дві таких зв’язки: Автор-Спеціаліалізація та Автор-Наукова\_робота. Тому для вирішення цієї проблеми ведено дві додаткові сутності: Автор\_Спеціаліалізація та Автор\_Наукова\_робота.

Таблиця 1 – опис структури даних.

|  |  |  |
| --- | --- | --- |
| Відношення | Атрибут | Тип атрибута |
| author – *містить дані про автора* | id\_author – унікальний код автора  name – ім’я автора  organization – назва організація в якій перебуває автор  id\_specialization – ідентифікатор спеціальності | integer (числовий)  character varying (рядок)  character varying (рядок)  integer (числовий) |
| specialization – *містить дані про спеціалізацію* | id\_specialization – унікальний код спеціалізації  science – галузь науки  education – освіта | integer (числовий)  character varying (рядок)  character varying (рядок) |
| research\_work – *містить дані про наукову роботу* | id\_research – унікальний код  id\_author – ідентифікатор автора  id\_theme – тема роботи  name – назва роботи  data\_publication – дата публікації | integer (числовий)  integer (числовий)  integer (числовий)  character varying (рядок)  integer (числовий) |
| theme – *містить дані про тему роботи* | id\_theme – унікальний код теми  terms – терміни в роботі  description – короткий опис  industry – галузь роботи | integer (числовий)  character varying (рядок)  character varying (рядок)  character varying (рядок) |
| research\_work\_author – *містить відповідність автора та наукової роботи, яку він написав* | id\_tab\_rwa – унікальний код таблиці відповідності  research\_work\_id\_author – унікальний ідентифікатор наукової роботи  author\_id\_author – унікальний ідентифікатор автора | integer (числовий)  integer (числовий)  integer (числовий) |
| author\_specialization – *містить відповідність автора та спеціалізацію, яку він здобув* | id\_tab\_as – унікальний ідентифікатор таблиці відповідності  author\_id\_specialization – унікальний ідентифікатор автора  specialization\_id\_specialization – унікальний ідентифікатор спеціалізації | integer (числовий)  integer (числовий)  integer (числовий) |

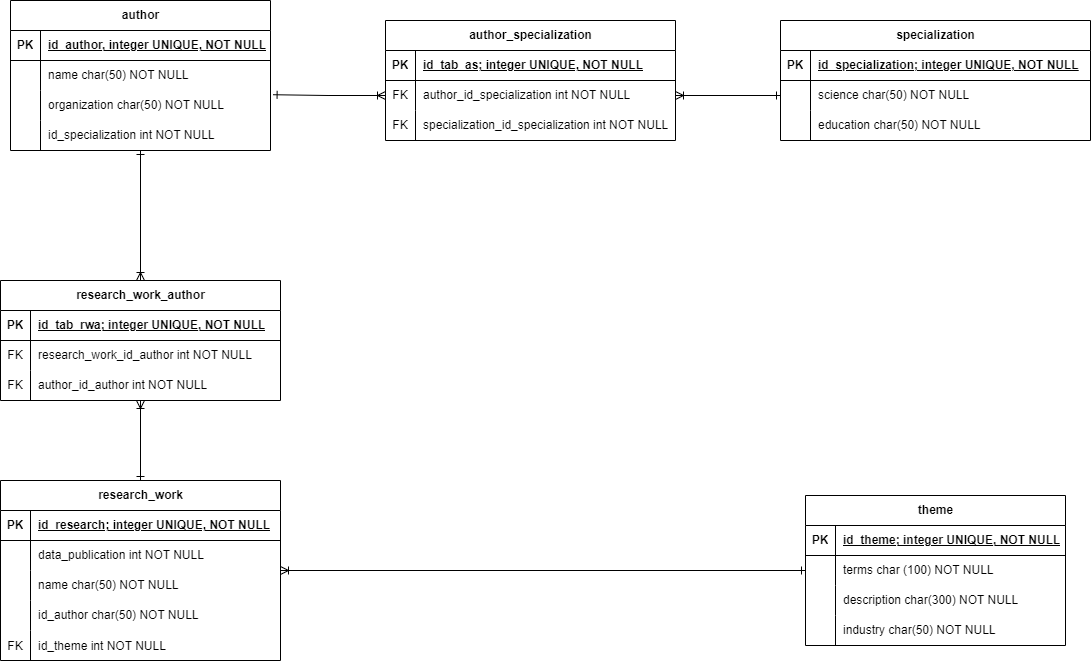


Рисунок 2 – Схеми бази даних

**Функціональні залежності**

author:

id\_author – *унікальний код автора*

name – *ім’я автора*

organization – *назва організація в якій перебуває автор*

id\_specialization – *ідентифікатор спеціальності*

id\_author → name, organization, id\_specialization

id\_author → name (ім’я автора залежить від коду автора)

id\_author → organization (місце роботи залежить від коду автора)

name → organization

id → name →organization (транзитивне відношення, але воно включає в себе ключовий атрибут)

specialization:

id\_specialization – *унікальний код спеціалізації*

science – *галузь науки*

education – *освіта*

id\_specialization → science, education

id\_specialization → science (галузь науки залежить від коду)

id\_specialization → education (освіта залужить від коду)

research\_work:

id\_research – *унікальний код*

id\_author – *ідентифікатор автора*

id\_theme – *тема роботи*

name – *назва роботи*

data\_publication – дата публікації

id\_research → id\_author, id\_theme, name, data\_publication

id\_research → name (назва роботи залежить від коду наукової роботи)

id\_research → data\_publication (дата публікації залежить від коду наукової роботи)

Є декілька транзитивних відношень, але вони включають в себе ключовий атрибут

theme:

id\_theme – *унікальний код теми*

terms – *терміни в роботі*

description – *короткий опис*

industry – *галузь роботи*

id\_theme → terms, description, industry

id\_theme → terms (терміни залежні від коду теми)

id\_theme → description (опис залежний від коду теми)

id\_theme → industry (галузь залежний від коду теми)

Схема відповідає 1НФ, тому що всі комірки сутностей мають лише одне значення, також кожен запис є унікальним.

Схема відповідає 2НФ, тому що відповідає 1НФ, а також у всіх сутностях неключові атрибути функціонально залежить від усього ключа.

Схема відповідає 3НФ, тому що відповідає 2НФ, а також кожен не простий атрибут не є транзитивно залежним від кожного ключа-кандидата.

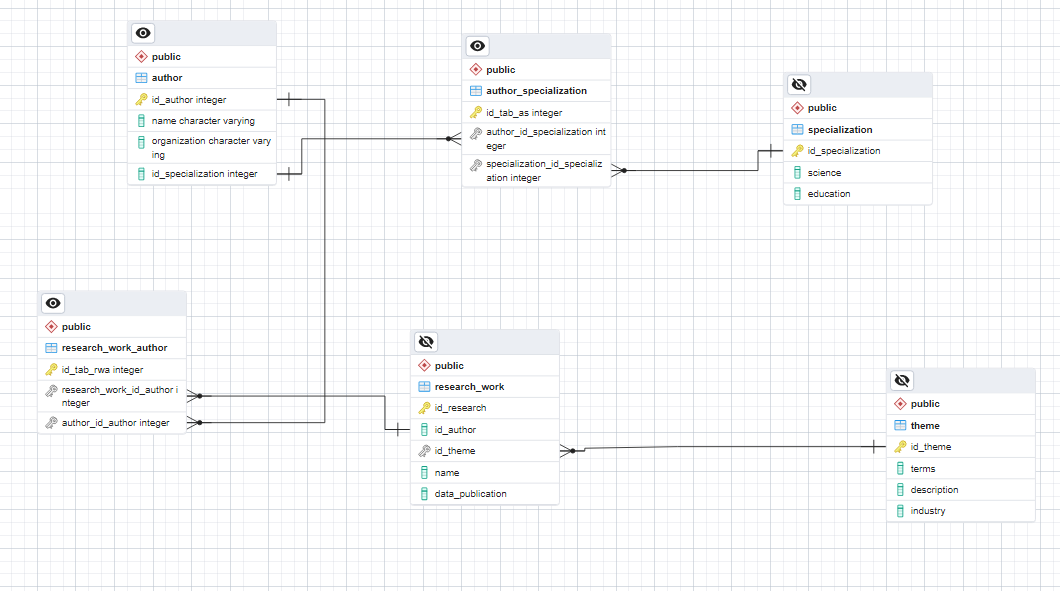
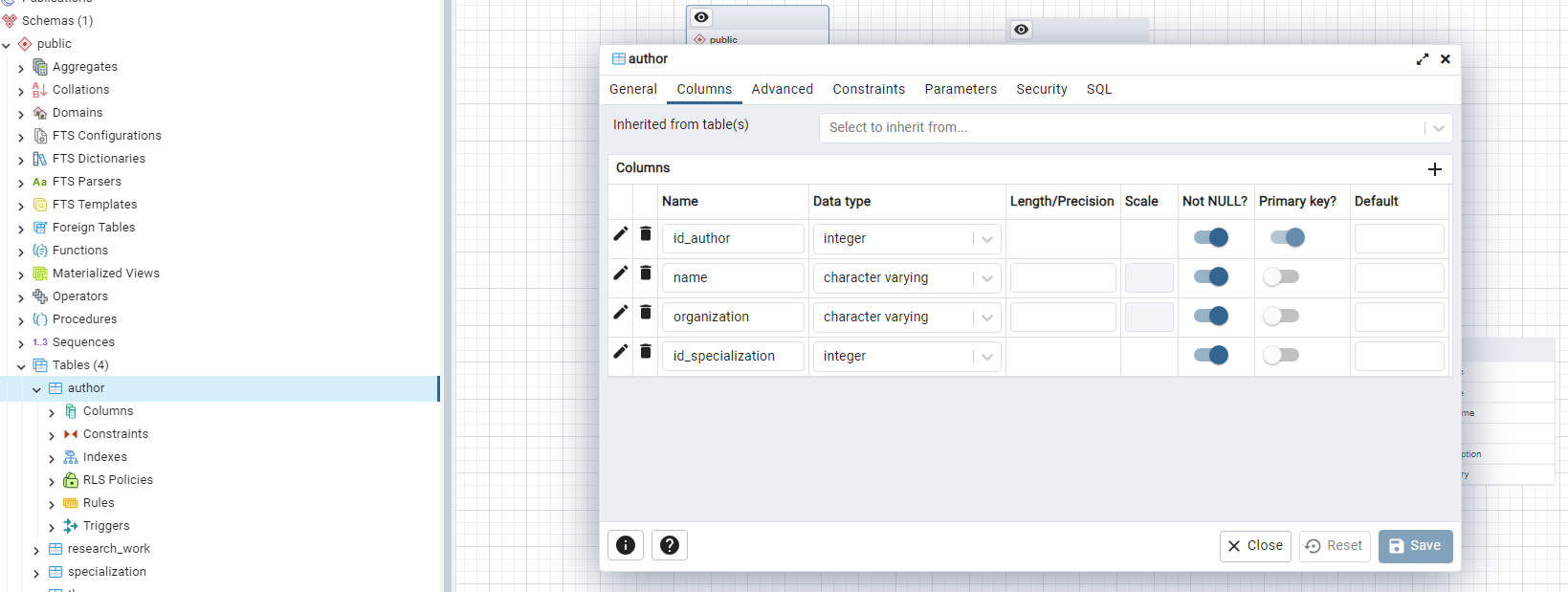
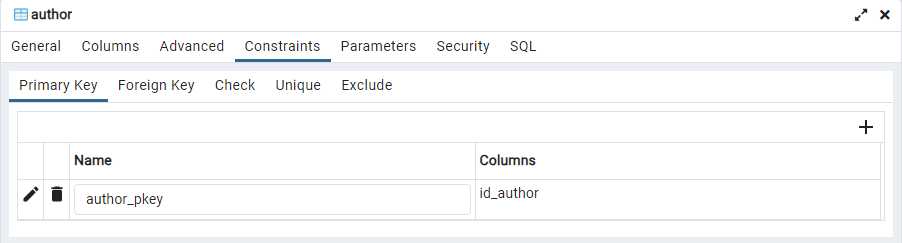


Рисунок 3 – Схема бази даних у pgAdmin 4

**Таблиці бази даних у pgAdmin 4**

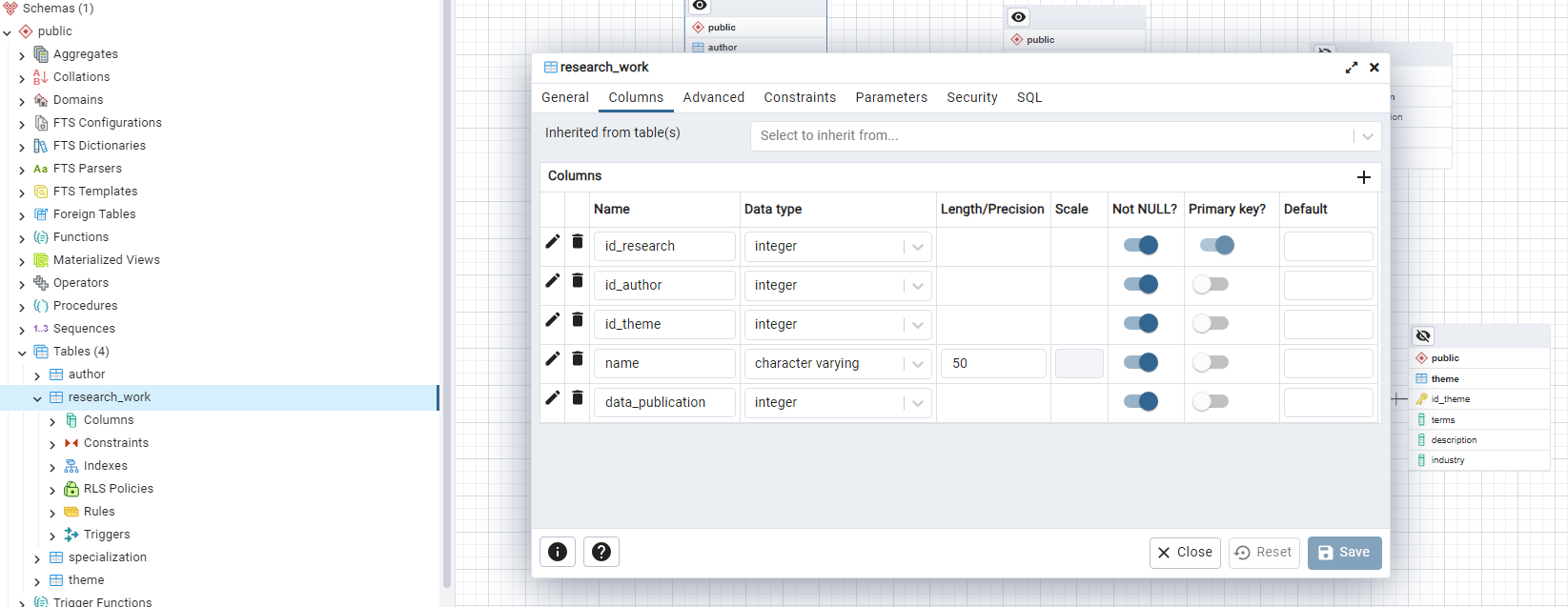
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Author\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

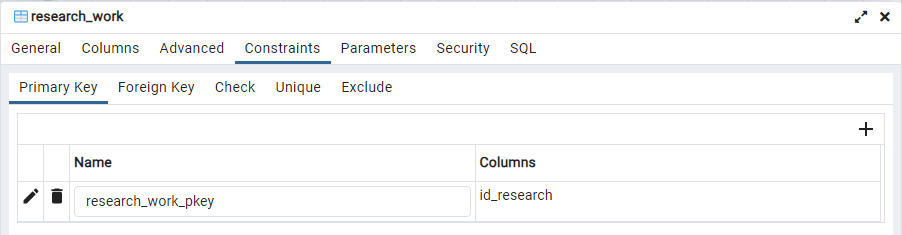


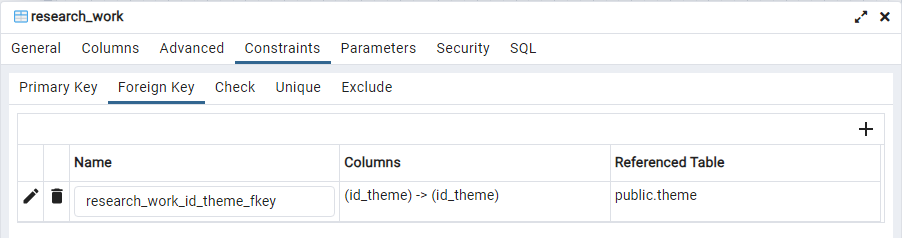


У цієї таблиці немає зовнішніх ключів (FK)

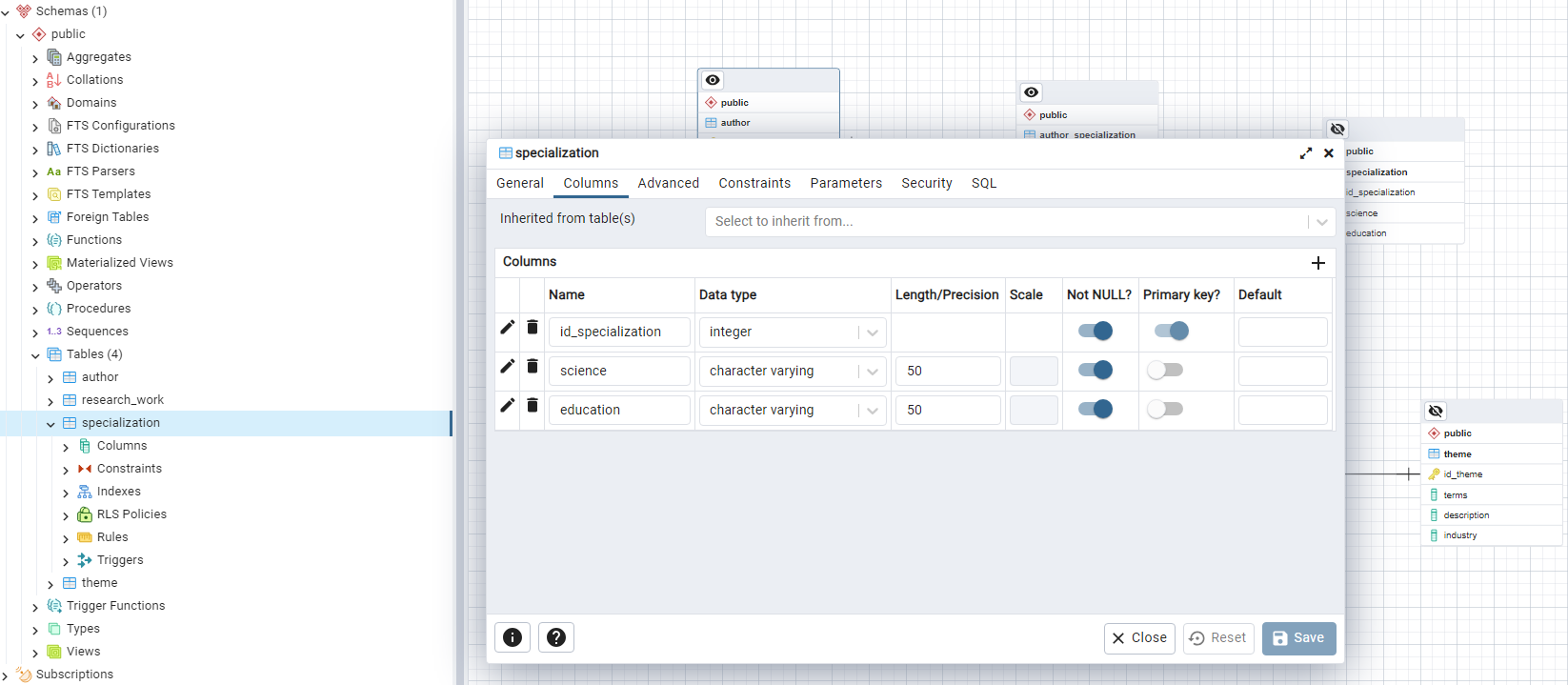
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_research\_work\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

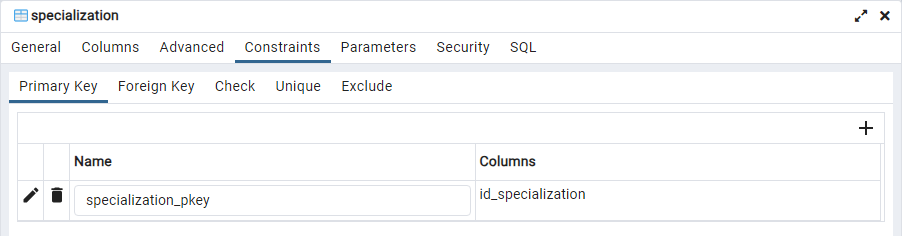






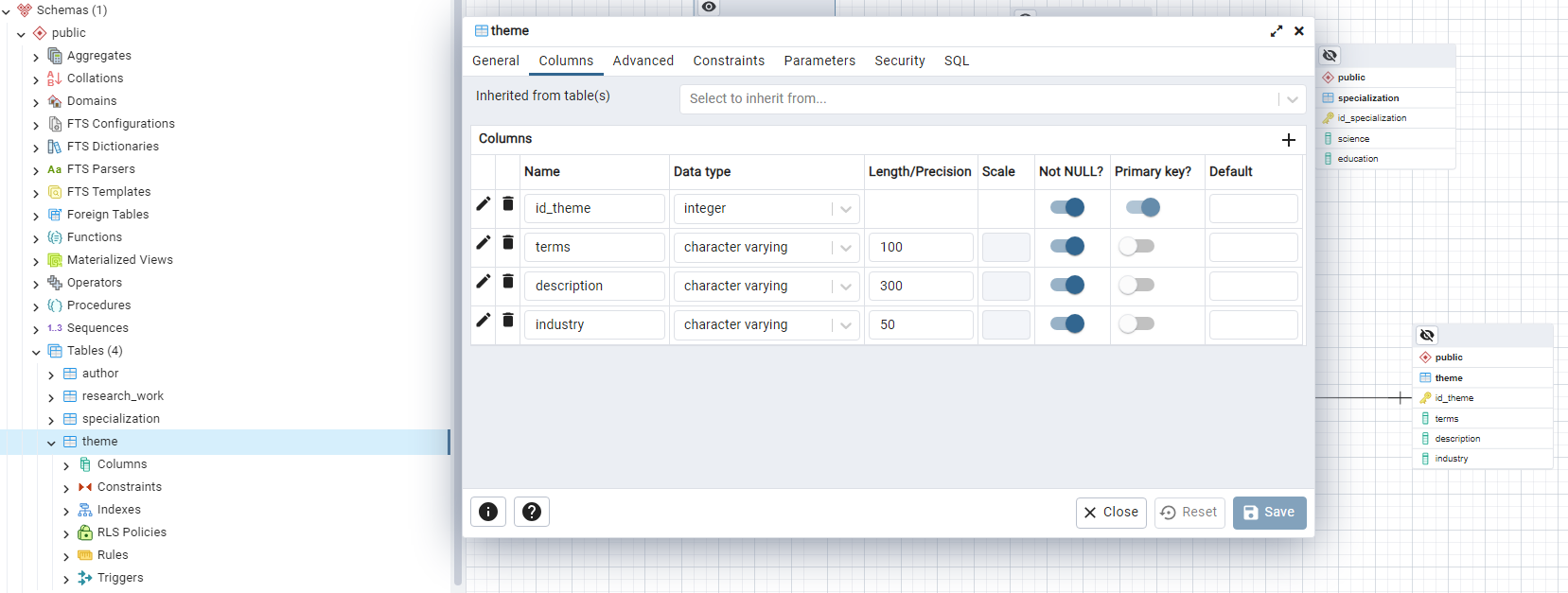
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_specialization\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

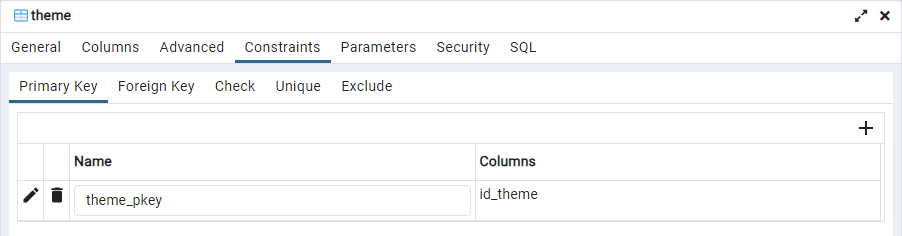




У цієї таблиці немає зовнішніх ключів (FK)

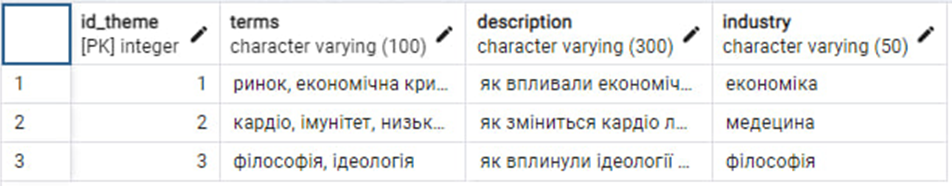
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_theme \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

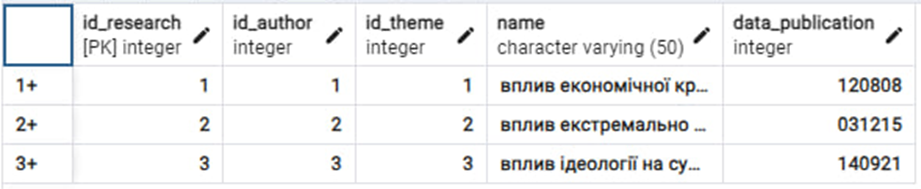


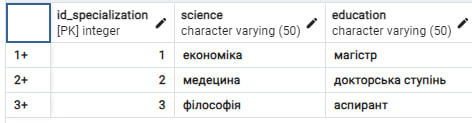


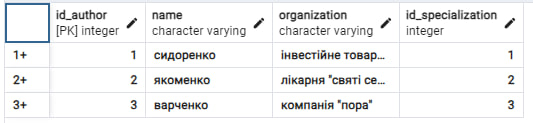
У цієї таблиці немає зовнішніх ключів (FK)

**Таблиці заповнені даними (уривки знімків екрану pgAdmin 4)**









Таблиці в коді SQL

-- This script was generated by the ERD tool in pgAdmin 4.

-- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public.research\_work

(

id\_research integer NOT NULL,

id\_author integer NOT NULL,

id\_theme integer NOT NULL,

name character varying(50) COLLATE pg\_catalog."default" NOT NULL,

data\_publication integer NOT NULL,

CONSTRAINT research\_work\_pkey PRIMARY KEY (id\_research)

);

CREATE TABLE IF NOT EXISTS public.specialization

(

id\_specialization integer NOT NULL,

science character varying(50) COLLATE pg\_catalog."default" NOT NULL,

education character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT specialization\_pkey PRIMARY KEY (id\_specialization)

);

CREATE TABLE IF NOT EXISTS public.theme

(

id\_theme integer NOT NULL,

terms character varying(100) COLLATE pg\_catalog."default" NOT NULL,

description character varying(300) COLLATE pg\_catalog."default" NOT NULL,

industry character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT theme\_pkey PRIMARY KEY (id\_theme)

);

CREATE TABLE IF NOT EXISTS public.author

(

id\_author integer NOT NULL,

name character varying COLLATE pg\_catalog."default" NOT NULL,

organization character varying COLLATE pg\_catalog."default" NOT NULL,

id\_specialization integer NOT NULL,

CONSTRAINT author\_pkey PRIMARY KEY (id\_author)

);

CREATE TABLE IF NOT EXISTS public.author\_specialization

(

id\_tab\_as integer NOT NULL,

author\_id\_specialization integer NOT NULL,

specialization\_id\_specialization integer NOT NULL,

PRIMARY KEY (id\_tab\_as)

);

CREATE TABLE IF NOT EXISTS public.research\_work\_author

(

id\_tab\_rwa integer NOT NULL,

research\_work\_id\_author integer NOT NULL,

author\_id\_author integer NOT NULL,

PRIMARY KEY (id\_tab\_rwa)

);

ALTER TABLE IF EXISTS public.research\_work

ADD CONSTRAINT research\_work\_id\_theme\_fkey FOREIGN KEY (id\_theme)

REFERENCES public.theme (id\_theme) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

ALTER TABLE IF EXISTS public.author\_specialization

ADD FOREIGN KEY (author\_id\_specialization)

REFERENCES public.author (id\_specialization) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.author\_specialization

ADD FOREIGN KEY (specialization\_id\_specialization)

REFERENCES public.specialization (id\_specialization) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.research\_work\_author

ADD FOREIGN KEY (research\_work\_id\_author)

REFERENCES public.research\_work (id\_author) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.research\_work\_author

ADD FOREIGN KEY (author\_id\_author)

REFERENCES public.author (id\_author) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

END;