

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 ІІI курсу

групи КВ-22

Гречишкіна К.Д.

Перевірив:

Павловський В.І.

Київ – 2024

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

*Завдання* роботи полягає в 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.

**Опис предметної галузі**

Обрана предметна галузь «Бібліотечна система управління книгами» охоплює автоматизацію рутинних процесів, та покращення ефективності обслуговування читачів і забезпечення зручного доступу до книг.

Основна мета – спростити управління ресурсами бібліотеки, забезпечити точний облік книг, покращити процеси пошуку, видачі та повернення книг. Це допоможе бібліотеці працювати більш ефективно, забезпечуючи якісне обслуговування клієнтів.

**Опис сутностей предметної галузі**

*При створенні концептуальної моделі обраної предметної галузі я виділила такі сутності:*

1. *Автор (Author)*

Атрибути: ID автора, прізвище, ім’я

Призначення: зберігає інформацію про авторів книг

1. *Книга(Book)*

Атрибути: ID книги, назва, рік видання

Призначення: зберігає інформацію про книги

1. *Читач (Reader)*

Атрибути: ID читача, прізвище, ім’я, адреса проживання

Призначення: зберігає інформацію про читачів

1. *Запозичення (Loan)*

Атрибути: ID запозичення, ID читача, ID книги, дата та час видачі, дата та час повернення, фактична дата та час повернення

Призначення: зберігає інформацію про процес запозичення книг читачам

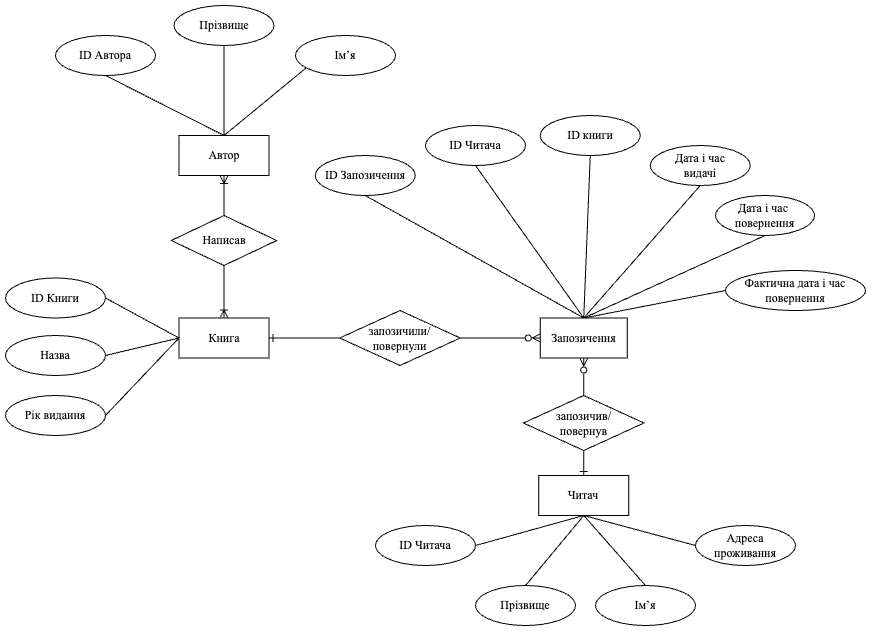
**Опис зв’язків між сутностями**

Між сутностями «Автор» та «Книга» - зв’язок M:N. Один автор може написати багато книг, а одна книга може мати багато авторів.

Між сутностями «Читач» та «Запозичення» - зв’язок 1:N. **Один читач може мати багато записів про запозичення книг**, а може й не мати**,** але кожен запис запозичення стосується лише одного читача.

«Книга» та «Запозичення» - зв’язок 1:N. **Одна книга може бути запозичена багато разів**, а може й не бути, але кожен запис про запозичення належить лише одній книзі.

Графічне подання концептуальної моделі «Сутність-зв’язок» зображено на рисунку 1.

Рисунок 1 – ER-діаграма, побудована за нотацією «Пташина лапка»

**Процес перетворення діаграми в таблицю**

Сутність «Автор» (Author) було перетворено в таблицю Author з первинним ключем *аuthor\_id ,* та атрибутами: first\_name, last\_name

Сутність «Книга» (Book) було перетворено в таблицю Book з первинним ключем*book\_id* , та атрибутами: title, publish\_year

Сутність «Читач» (Reader) було перетворено в таблицю Reader з первинним ключем *reader\_id ,* та атрибутами: first\_name, last\_name, address

Сутність «*З*апозичення» (Loan) було перетворено в таблицю Loan з первинним ключем *loan\_id* і зовнішніми ключами *reader\_id, book\_id* та атрибутами: **loan\_date\_and\_time,** return\_date\_and\_time**,** **actual\_return\_date\_and\_time**

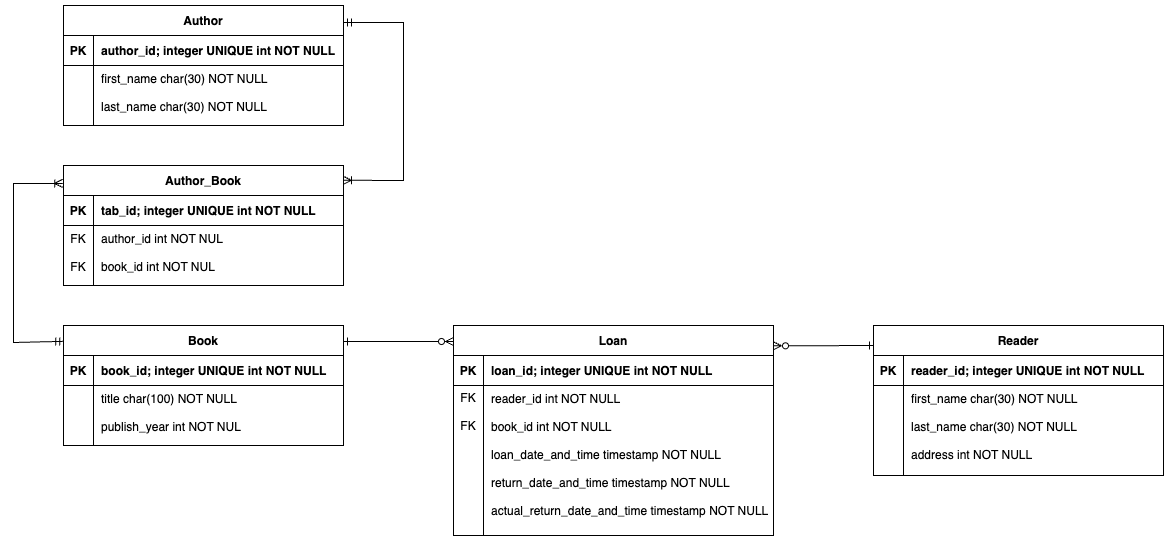
Оскільки в логічній моделі безпосередній зв’язок між сутностями M:N є неможливим, а в концептуальній моделі він існує між сутностями «Автор» (Author) i «Книга» (Book), то для його реалізації створено таблицю Author\_Book, первинним ключем *tab\_id,* та зовнішніми ключами *аuthor\_id* та *book\_id*

Опис сутностей та їх атрибутів наведено в таблиці 1

Таблиця 1 – Опис структури бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Author –** *містить у собі дані про авторів книг* | **аuthor\_id** –*унікальний ідентифікатор автора*  **first\_name** – *імʼя автора*  **last\_name** – *прізвище автора* | **integer** *(числовий)*  **character varying** (рядок)  **character varying** (рядок) |
| **Book –** *містить у собі дані про книгу* | **book\_id** – *унікальний ідентифікатор книги*  **title** – *назва книги*  **publish\_year** – *рік видання книги* | **integer** *(числовий)*  **character varying** (рядок)  **integer** *(числовий)* |
| **Reader****–** *містить дані про користувача бібліотеки* | **reader\_id** – *унікальний ідентифікатор читача*  **first\_name** – *імʼя читача*  **last\_name** – *прізвище читача*  **address** – *адреса проживання читача* | **integer** *(числовий)*  **character varying** (рядок)  **character varying** (рядок)  **character varying** (рядок) |
| **Loan** **–** *містить*  *інформацію про видачу книг користувачам бібліотеки* | **loan\_id** – *унікальний ідентифікатор видачі книги*  **reader\_id** – *унікальний ідентифікатор читача*  **book\_id** – *унікальний ідентифікатор книги*  **loan\_date\_and\_time** – *дата та час видачі*  **return\_date\_and\_time** – *датата час повернення*  **actual\_return\_date\_and\_time** – *фактична дата та час повернення* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)*  **timestamp** (*дата та час*)  **timestamp** (*дата та час*)  **timestamp** (*дата та час*) |
| **Author\_Book –** *містить інформацію відповідність автора і книги яку він написав* | **id\_tab** – *унікальний ідентифікатор*  *таблиці відповідності*  **аuthor\_id** – *унікальний ідентифікатор автора*  **book\_id** – *унікальний ідентифікатор книги* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |

Графічне подання логічної моделі «Сутність-зв’язок» зображено на рисунку 2.

Рисунок 2 – Схема бази даних

**Функціональні залежності**

**Author:**

author\_id first\_name, last\_name

**Book:**

book\_id title, publish\_year

**Reader:**

reader\_id first\_name, last\_name, address

**Loan:**

loan\_id reader\_id,book\_id, **loan\_date\_and\_time,** return\_date**\_and\_time, actual\_return\_date\_and\_time**

book\_id→ reader\_id, **loan\_date\_and\_time,** return\_date**\_and\_time, actual\_return\_date\_and\_time**

reader\_id→ book\_id, **loan\_date\_and\_time,** return\_date**\_and\_time, actual\_return\_date\_and\_time**

**Author\_Book:**

id\_tab author\_id, book\_id

author\_id→ book\_id

book\_id → author\_id

**Відповідність схеми бази даних нормальним формам:**

Схема відповідає 1НФ, тому що:

* Кожна таблиця має первинний ключ (Primary key, PK) та набір атрибутів, які ідентифікують запис
* Кожен атрибут має лише одне значення, а не множину значень.

*Приклад*: У таблиці "Читачі" атрибути **first\_name, last\_name, address** та reader\_id мають атомарні значення, оскільки кожна клітинка містить лише одне значення.

Схема відповідає 2НФ, тому що:

* Вона відповідає вимогам 1НФ;
* Кожен неключовий атрибут функціонально залежить від усього ключа, а не його частини

*Приклад*: У таблиці "Автор" атрибути author\_id, first\_name, last\_name повністю залежать від author\_id, який є первинним ключем цієї таблиці.

Схема відповідає 3НФ, тому що:

* Вона відповідає вимогам 2НФ;
* Дані в таблиці усі неключові атрибути залежать винятково від основного ключа і не залежать один від одного.

*Приклад*: У таблиці "Читачі" атрибути **first\_name, last\_name, address**

не залежать один від одного**.** Тобто адреса не залежить від імені.

Підсумовуючи, схема бази даних відповідає нормальним формам (1НФ, 2НФ та 3НФ), що забезпечує її ефективність, та мінімізує дублювання даних, запобігає аномаліям вставки, оновлення або видалення. Це дозволяє забезпечувати стабільність і логічну цілісність при взаємодії з даними.

Схема бази даних у pgAdmin 4 зображено на рисунку 3.

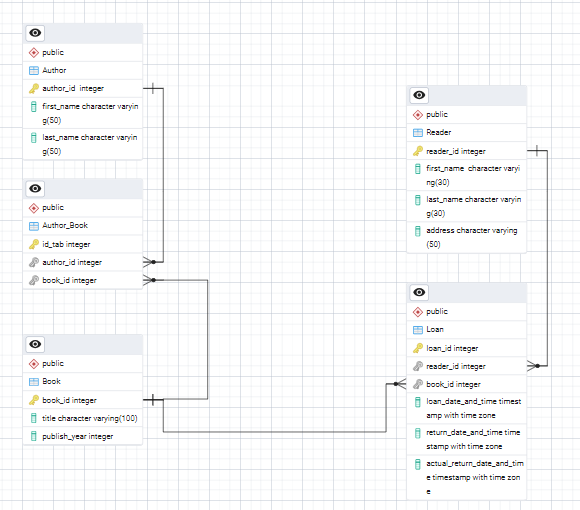
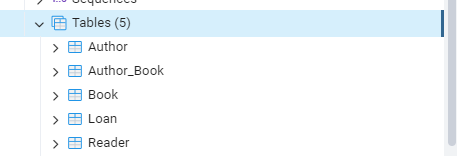
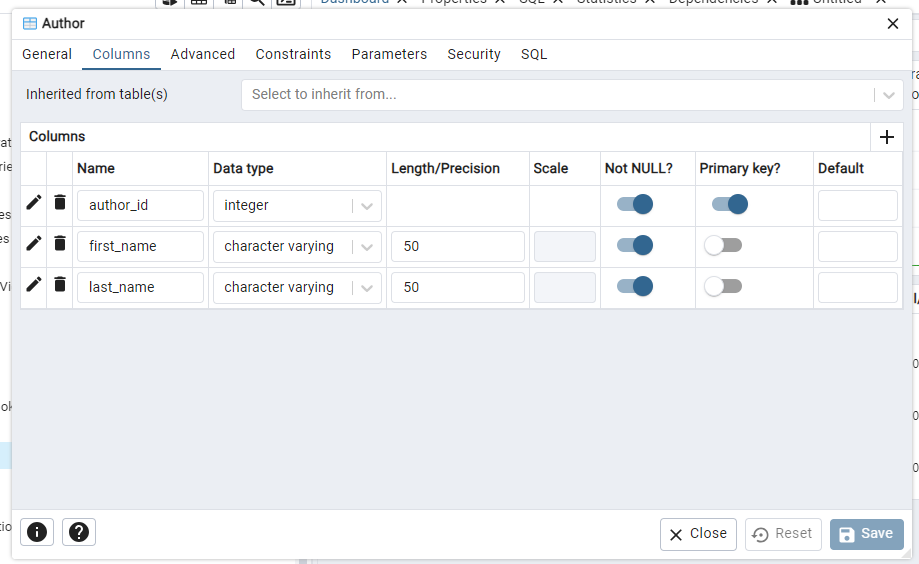
****

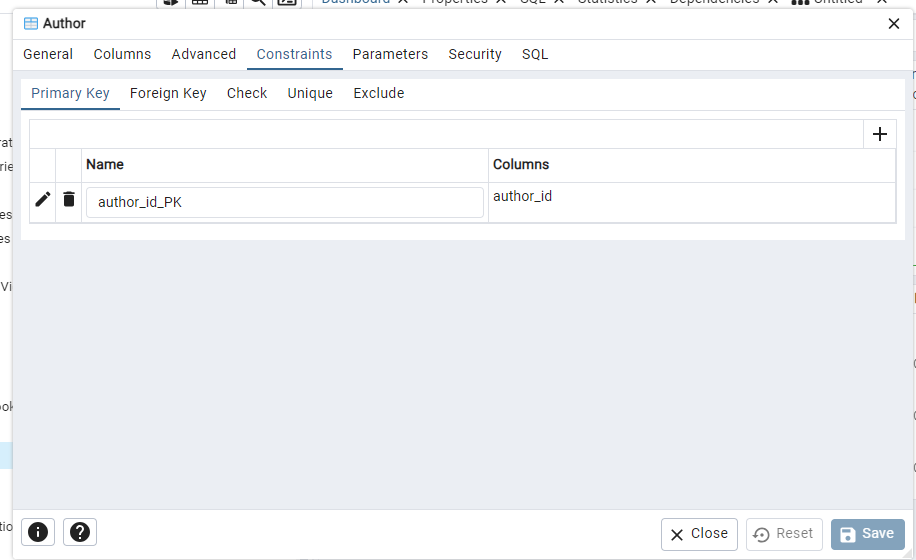
Рисунок 3 - Схема бази даних у pgAdmin 4

**Таблиці бази даних у pgAdmin4**

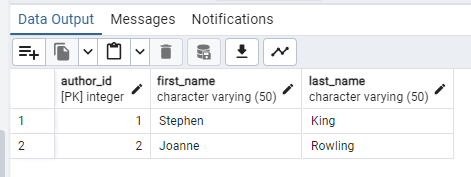
****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Author** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

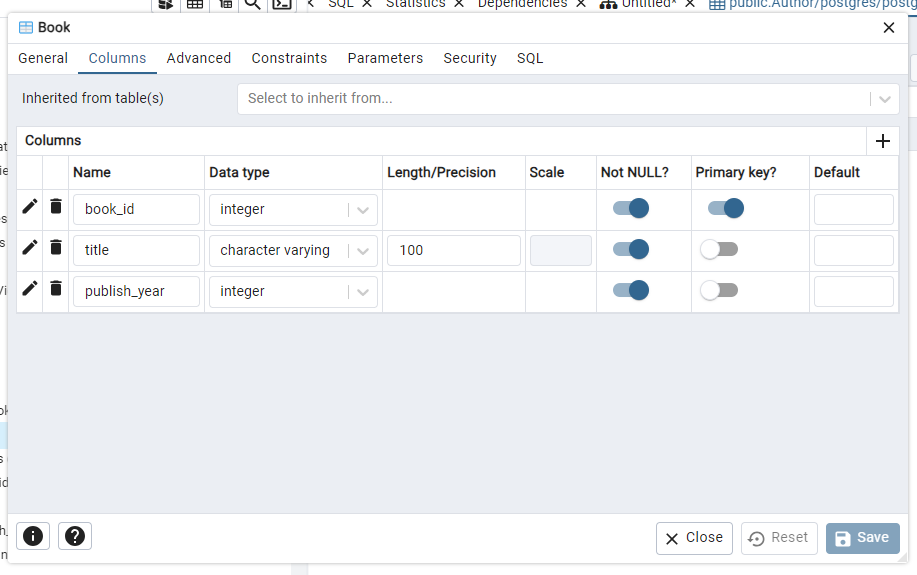


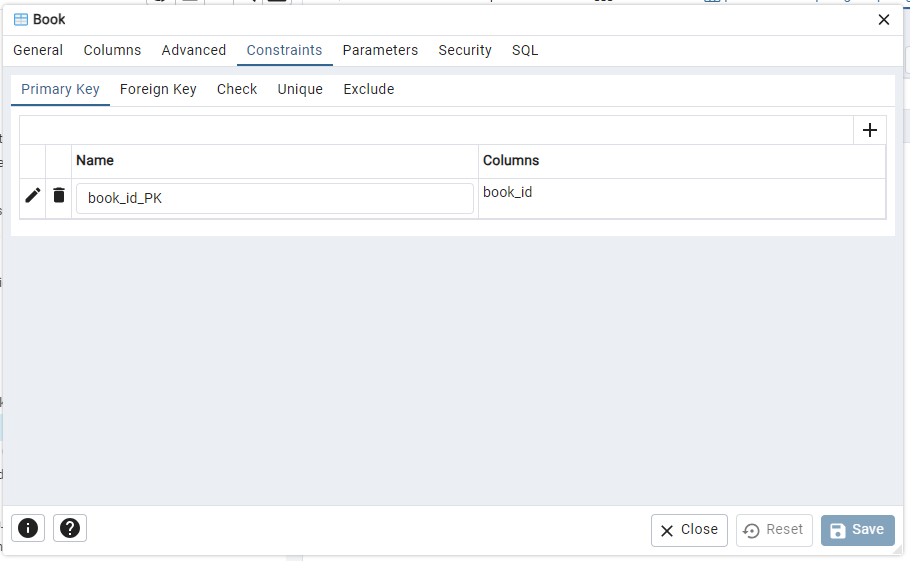


У цієї таблиці немає зовнішніх ключів (FK)

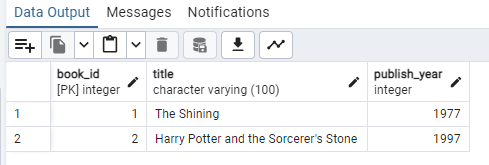


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Book** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

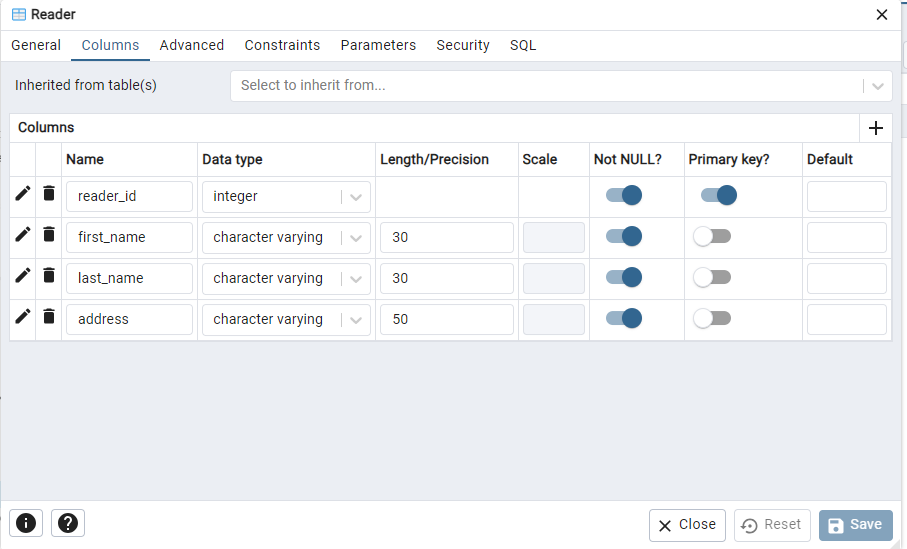


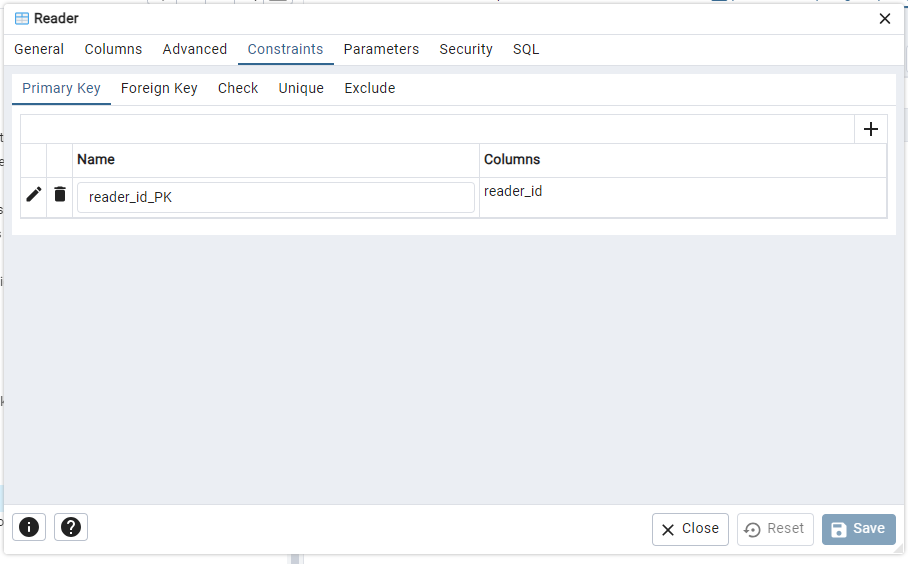


У цієї таблиці немає зовнішніх ключів (FK)

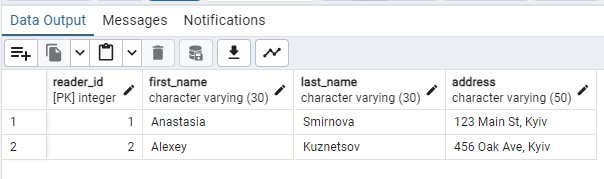


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Reader** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

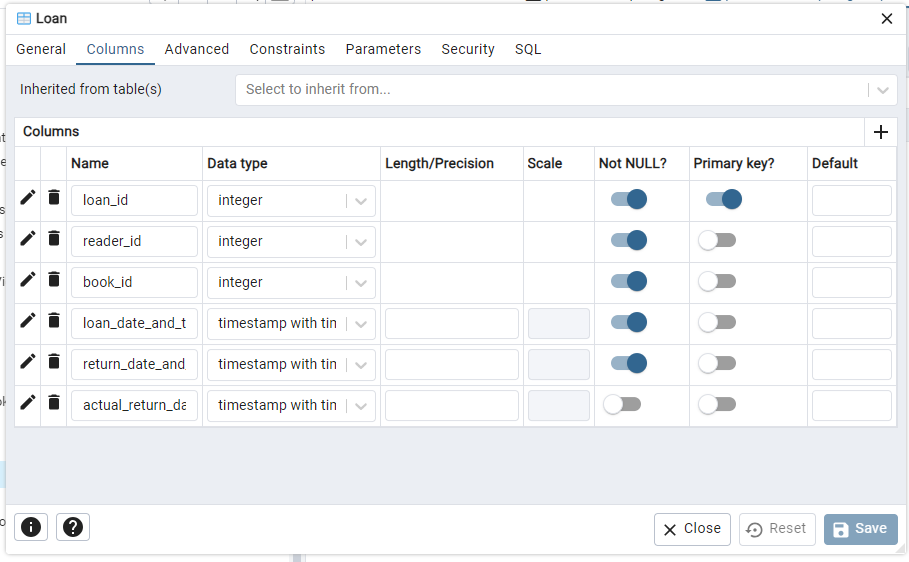


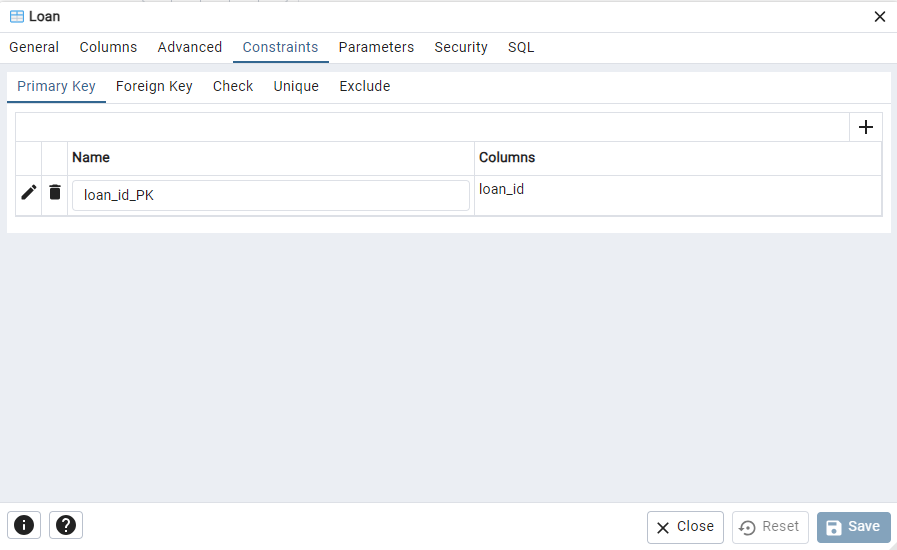


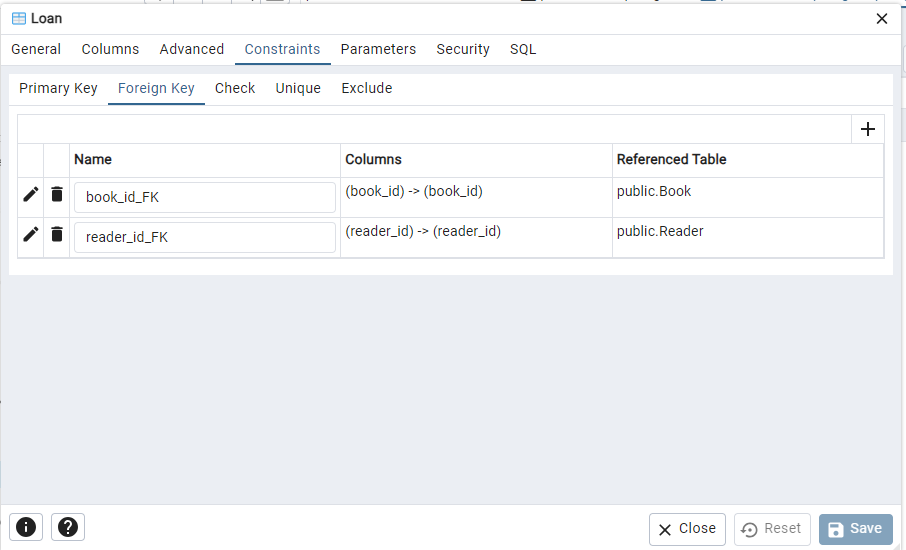
У цієї таблиці немає зовнішніх ключів (FK)

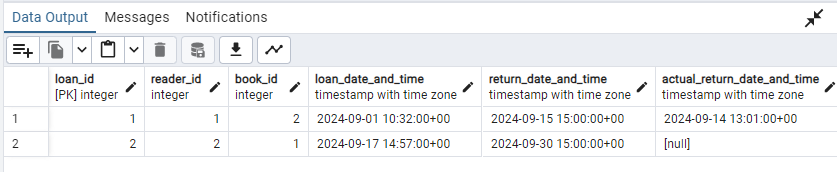


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Loan** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

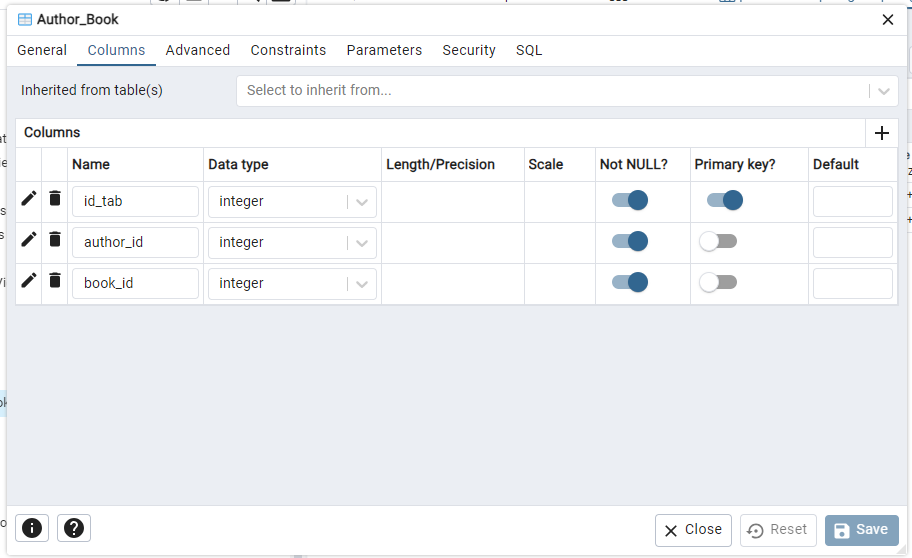


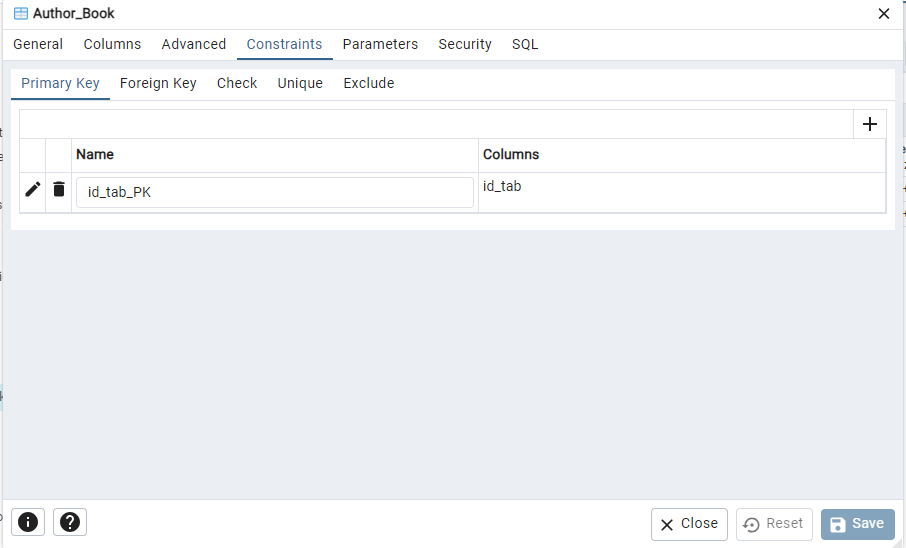


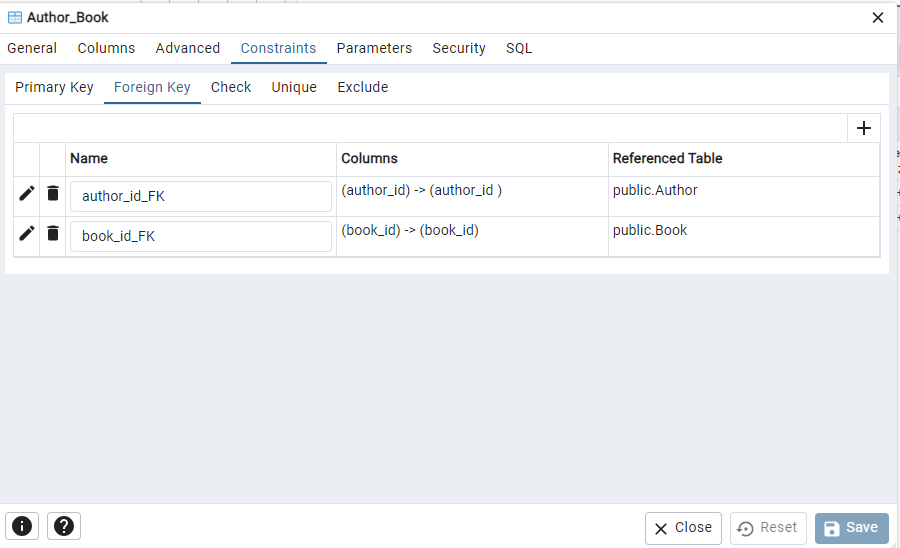


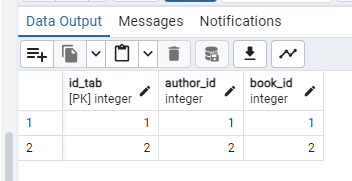


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Author\_Book** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_









**Таблиці в коді SQL**

-- Table: public.Author

-- DROP TABLE IF EXISTS public."Author";

CREATE TABLE IF NOT EXISTS public."Author"

(

"author\_id " integer NOT NULL,

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

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

CONSTRAINT "аuthor\_id\_PK" PRIMARY KEY ("author\_id ")

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Author"

OWNER to postgres;

-- Table: public.Book

-- DROP TABLE IF EXISTS public."Book ";

CREATE TABLE IF NOT EXISTS public."Book "

(

book\_id integer NOT NULL,

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

publish\_year integer NOT NULL,

CONSTRAINT "book\_id\_PK" PRIMARY KEY (book\_id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Book "

OWNER to postgres;

-- Table: public.Reader

-- DROP TABLE IF EXISTS public."Reader";

CREATE TABLE IF NOT EXISTS public."Reader"

(

reader\_id integer NOT NULL,

"first\_name " character varying(30) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(30) COLLATE pg\_catalog."default" NOT NULL,

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

CONSTRAINT "reader\_id\_PK" PRIMARY KEY (reader\_id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Reader"

OWNER to postgres;

-- Table: public.Loan

-- DROP TABLE IF EXISTS public."Loan";

CREATE TABLE IF NOT EXISTS public."Loan"

(

loan\_id integer NOT NULL,

reader\_id integer NOT NULL,

book\_id integer NOT NULL,

loan\_date\_and\_time timestamp with time zone NOT NULL,

return\_date\_and\_time timestamp with time zone NOT NULL,

actual\_return\_date\_and\_time timestamp with time zone,

CONSTRAINT "loan\_id\_PK" PRIMARY KEY (loan\_id),

CONSTRAINT "book\_id\_FK" FOREIGN KEY (book\_id)

REFERENCES public."Book " (book\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "reader\_id\_FK" FOREIGN KEY (reader\_id)

REFERENCES public."Reader" (reader\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Loan"

OWNER to postgres;

-- Table: public.Author\_Book

-- DROP TABLE IF EXISTS public."Author\_Book";

CREATE TABLE IF NOT EXISTS public."Author\_Book"

(

id\_tab integer NOT NULL,

author\_id integer NOT NULL,

book\_id integer NOT NULL,

CONSTRAINT "id\_tab\_PK" PRIMARY KEY (id\_tab),

CONSTRAINT "author\_id\_FK" FOREIGN KEY (author\_id)

REFERENCES public."Author" ("author\_id ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT "book\_id\_FK" FOREIGN KEY (book\_id)

REFERENCES public."Book " (book\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Author\_Book"

OWNER to postgres;

Посилання на GitHub*:* [*https://github.com/ccartix/LoanBook*](https://github.com/ccartix/LoanBook)