



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 III курсу

групи КВ-01

Рябенко Б. Ю.

Перевірив:

Київ – 2022

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 кожен з таблиць засобами pgAdmin 4.

Вимоги до ER-моделі

1. Сутності моделі предметної галузі мають містити зв'язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п'яти.
3. Передбачити наявність зв'язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow’s foot)”, UML.

Опис предметної області «Книжковий магазин»

Обрана предметна область передбачає опис параметрів книжок. Книжка має декілька жанрів, а також знаходиться у книжковій серії. Книжкові серії, у свою чергу, написані автором.

Опис сутностей предметної області

Для побудови бази даних для обраної області було виділено сутності, зображені на рисунку 1:

- 1) Книга (book), з атрибутами id книги, назва, ціна, кількість, дата публікації, id книжкової серії. Зберігає інформацію про книгу, серію та автора.
- 2) Серія (series), з атрибутами id серії, назва, id автора. Зберігає інформацію про Серію та автора.
- 3) Автор (author), з атрибутами id автора, ім'я. Зберігає інформацію про автора.
- 4) Жанр (genre), з атрибутами id жанру, назва. Зберігає інформацію про жанр.

Опис зв'язків між сутностями предметної області

Сутність «Книга» має зв'язок 1:N по відношенню до серії, тому що в одній серії можуть бути декілька книжок, але одна книжка не може відноситися до декількох серій.

Сутність «Серія» має зв'язок 1:N по відношенню до автора, тому що автор може мати декілька серій, але серія не може мати декілька авторів.

Сутність «Жанр» має зв'язок M:N по відношенню до книги, тому що одна книжка може мати декілька жанрів, та один і той самий жанр може бути у декількох книжках.

Концептуальна модель предметної області «Книжковий магазин»

Концептуальна модель наведена на рисунку 1.

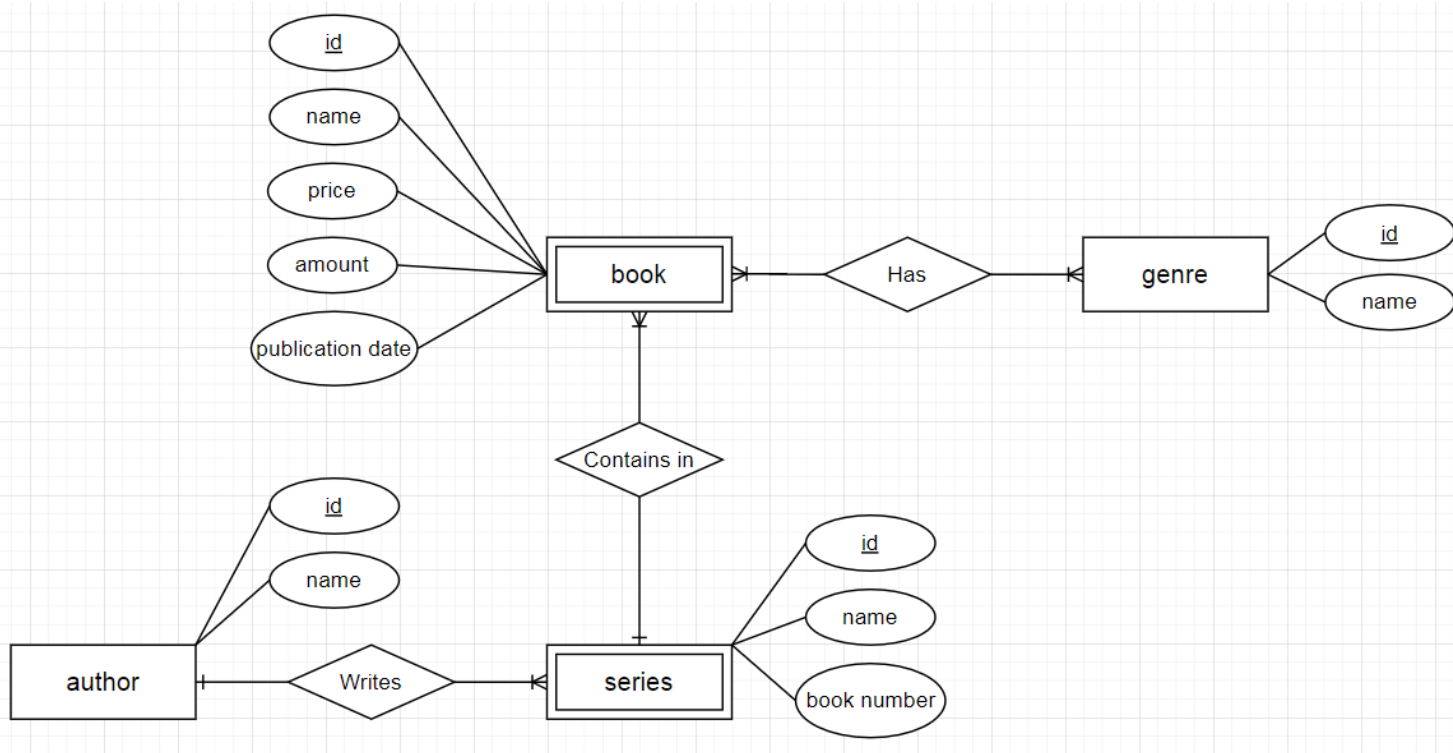


Рисунок 1 - ER-діаграма, побудована за нотацією Чена (інструмент: draw.io)

Перетворення концептуальної моделі у схему баз даних

1. Якщо зв'язок типу 1:N і клас приналежності сутності на стороні N є обов'язковим, то необхідно побудувати таблицю для кожної сутності. Первинний ключ сутності повинен бути первинним ключем відповідної таблиці. Первинний ключ сутності на стороні 1 додається як атрибут в таблицю для сутності на стороні N. Ми можемо спостерігати даний зв'язок між сутностями author і series, book і series.

2. Якщо зв'язок типу N:M, то необхідно побудувати три таблиці - по одній для кожної сутності і одну для зв'язку. Первинний ключ сутності

повинен бути первинним ключем відповідної таблиці. Таблиця для зв'язку серед своїх атрибутів повинна мати ключі обох сутностей. Даний зв'язок можна спостерігати між сутностями book і genre.

Логічну схему бази даних наведено на рисунку 2.

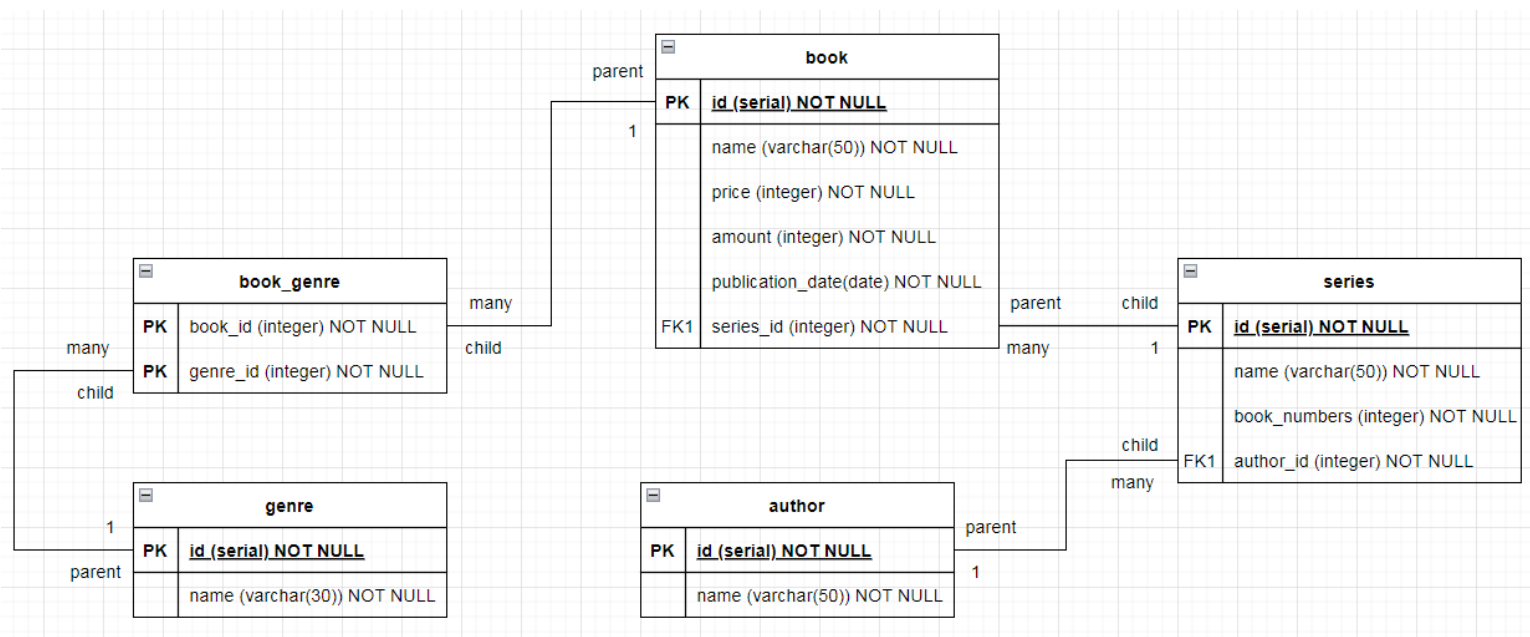


Рисунок 2 - Схема бази даних (інструмент: draw.io)

Опис об'єктів бази даних у вигляді таблиці

Сутність	Атрибут	Тип атрибуту
book – містить інформацію про всі книги	id – унікальний ідентифікатор книги name – назва книги price – ціна amount – кількість книжок publication_date – дата публікації series_id – зовнішній ключ серії	serial character varying integer integer date integer
series – містить інформацію про серію книг	id – унікальний ідентифікатор серії name – назва серії book_numbers – кількість книжок у серії author_id - зовнішній ключ автора	serial character varying integer integer
author – містить інформацію про автора	id – унікальний ідентифікатор автора name – ім'я автора	serial character varying
genre – містить інформацію про жанр	id – унікальний ідентифікатор жанру name – назва жанру	serial character varying
book_genre – містить інформацію про жанр книг	book_id – зовнішній ключ книги genre_id – зовнішній ключ жанру	integer integer

Функціональні залежності для кожної таблиці

BOOK:

id → name, price, amount, publication_date, series_id

id → name (назва книги залежить від коду книги)

id → price (ціна залежить від коду книги)

id → amount (кількість залежить від коду книги)

id → publication_date (дата залежить від коду книги)

id → series_id (код серії залежить від коду книжки)

SERIES:

id → name, book_numbers, author_id

id → name (назва серії залежить від коду серії)

id → book_numbers (кількість книг залежить від коду серії)

id → author_id (код автора залежить від коду серії)

AUTHOR:

id → name (ім'я автора залежить від коду автора)

GENRE:

id → name (назва жанру залежить від коду жанру)

Схема відповідає 1НФ, тому що:

1. В таблиці немає дубльованих рядків.
2. В кожній комірці зберігається атомарне (скалярне) значення.
3. В кожному стовпці зберігаються дані одного типу.

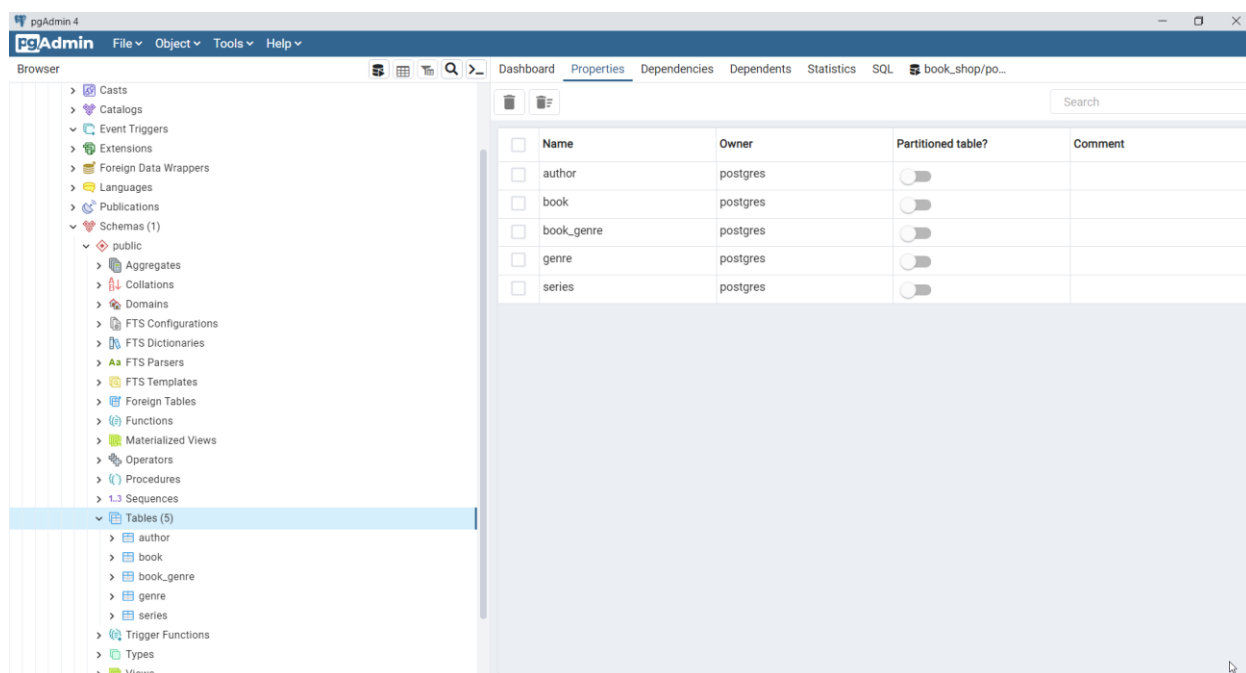
Схема відповідає 2НФ, тому що:

1. Вона відповідає 1НФ.
2. Має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що:

1. Вона відповідає 2НФ.
2. Всі не ключові атрибути таблиці залежать винятково від усього первинного ключа, а не його частини. Тобто кожен неключовий атрибут нетранзитивно (без посередника) залежить від первинного ключа

Фізична модель БД «Книжковий магазин» у pgAdmin3



Фотографії вмісту таблиць

Таблиця “author”

	id [PK] integer	name character varying (50)
1	1	Rick Riordan
2	2	Stephen King
3	3	Conan Doyle
4	4	Joanne Rowling

Таблиця “book”

	id [PK] integer	name character varying (50)	price integer	amount integer	publication_date date	series_id integer
1	9	The Two Dead Girls	512	60	1996-03-01	1
2	10	The Lightning Thief	513	35	2005-07-01	2
3	11	Harry Potter and the ...	456	184	2005-07-16	4
4	12	The Shining	634	89	1977-01-28	6
5	14	Night Journey	570	72	1996-07-25	1
6	15	The Adventure of the ...	99	12	2010-10-29	3
7	16	The Mark of Athena	312	56	2012-10-02	5

Таблиця “genre”

	id [PK] integer	name character varying (30)
1	1	Detective fiction
2	2	Short stories
3	3	Fantasy
4	4	Horror
5	5	Gothic
6	6	Dark Fantasy
7	7	Magic Realism
8	8	Greek mythology

Таблиця “series”

	id [PK] integer	name character varying (50)	book_numbers integer	author_id integer
1	1	The Green Mile	6	2
2	2	Percy Jackson and Th...	5	1
3	3	The Adventures of Sh...	12	3
4	4	Harry Potter	8	4
5	5	The Heroes of Olympus	5	1
6	6	The Shining	2	2

Таблиця “book_genre”

	book_id [PK] integer	genre_id [PK] integer
1	9	5
2	9	6
3	9	7
4	10	3
5	10	8
6	11	3
7	12	4
8	12	5
9	14	5
10	14	6
11	14	7
12	15	1
13	15	2
14	16	3
15	16	8

Sql-текст опису БД «Книжковый магазин»

```
-- Database: book_shop

-- DROP DATABASE IF EXISTS book_shop;

CREATE DATABASE book_shop
    WITH
        OWNER = postgres
        ENCODING = 'UTF8'
        LC_COLLATE = 'Russian_Ukraine.1251'
        LC_CTYPE = 'Russian_Ukraine.1251'
        TABLESPACE = pg_default
        CONNECTION LIMIT = -1
        IS_TEMPLATE = False;

-- Table: public.author

-- DROP TABLE IF EXISTS public.author;

CREATE TABLE IF NOT EXISTS public.author
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
        INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647
        CACHE 1 ),
    name character varying(50) COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT author_pkey PRIMARY KEY (id),
    CONSTRAINT author_name_key UNIQUE (name)
)

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
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
        INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647
        CACHE 1 ),
```



```

        name character varying(50) COLLATE
pg_catalog."default" NOT NULL,
        price integer NOT NULL,
        amount integer NOT NULL,
        publication_date date NOT NULL,
        series_id integer NOT NULL,
        CONSTRAINT book_pkey PRIMARY KEY (id),
        CONSTRAINT book_seriesid_fkey FOREIGN KEY
(series_id)
        REFERENCES public.series (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
    )

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.book
    OWNER to postgres;

-- Table: public.book_genre

-- DROP TABLE IF EXISTS public.book_genre;

CREATE TABLE IF NOT EXISTS public.book_genre
(
    book_id integer NOT NULL,
    genre_id integer NOT NULL,
    CONSTRAINT book_genre_pkey PRIMARY KEY (book_id,
genre_id),
    CONSTRAINT bookgenre_bookid_fkey FOREIGN KEY
(book_id)
        REFERENCES public.book (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE,
    CONSTRAINT bookgenre_genreid_fkey FOREIGN KEY
(genre_id)
        REFERENCES public.genre (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.book_genre
    OWNER to postgres;

```

```

-- Table: public.genre

-- DROP TABLE IF EXISTS public.genre;

CREATE TABLE IF NOT EXISTS public.genre
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
        INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647
        CACHE 1 ),
    name character varying(30) COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT genre_pkey PRIMARY KEY (id),
    CONSTRAINT genre_name_key UNIQUE (name)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.genre
    OWNER to postgres;

-- Table: public.series

-- DROP TABLE IF EXISTS public.series;

CREATE TABLE IF NOT EXISTS public.series
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY (
        INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647
        CACHE 1 ),
    name character varying(50) COLLATE
pg_catalog."default" NOT NULL,
    book_numbers integer NOT NULL,
    author_id integer NOT NULL,
    CONSTRAINT series_pkey PRIMARY KEY (id),
    CONSTRAINT series_name_key UNIQUE (name),
    CONSTRAINT series_authorid_fkey FOREIGN KEY
(author_id)
    REFERENCES public.author (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE CASCADE
)

TABLESPACE pg_default;

```

```
ALTER TABLE IF EXISTS public.series  
OWNER to postgres;
```