

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»

Виконав:

студент ІII курсу ФПМ

групи КВ-04

Ільїнчик А.О.

Перевірив: доц. Петрашенко А. В.

Київ – 2023

# Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL

*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 концептуальну модель - 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;

1. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ);
2. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

# Концептуальна модель предметної області “Облік бібліотеки”(*пункт звіту №1*)

В концептуальній моделі предметної області “Облік книгозбірні” (Рисунок 1) виділено наступні сутності та зв'язки між ними:

1. Сутність “Читачі” з атрибутами: id, ім’я, прізвище, по батькові, адреса;
2. Сутність “Автори” з атрибутами: id, ім’я;
3. Сутність “Книги” з атрибутами: id, id на запис з властивостями книги;
4. Сутність “Властивості книг” з атрибутами: id, id автора, назва книги, дата публікації, ціна під заставу;
5. Сутність “Записи в абонемент” з атрибутами: id, id кристувача який запозичив книгу, id запозиченної книги, час видання книги, час до якого користувач повинен повернути книгу, час коли корустувач повернув книгу.

У бібліотеці може зберігатися багато екземплярів однієї книги, тому між сутностями “Властивості книг” та “Книги” зв’язок 1:N.

Один автор може написати декілька книг, але книга може бути написана одним автором, тому між сутностями “Автори” та “Властивості книг” зв’язок 1:N.

Читач може запозичити одночасно декілька книг, але одна книга в різних екземплярах або в різні періоди часу може бути взята багатьма читачами. Тому між сутностями “Книги” та “Користувачі” зв’язок R(N:M). Для релізації зв’язку N:M між сутностями “Книги” та “Користувачі” використовується сутність “Записи”. Сутність “Користувачі” та “Записи” мають зв’язок 1:N. Сутність “Книги” та “Записи” мають зв’язок 1:N.

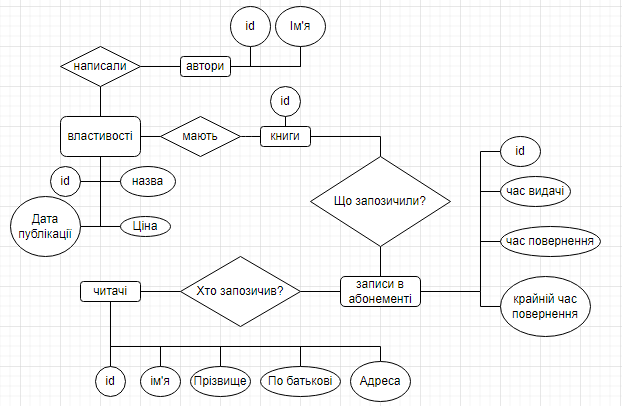


Рисунок 1 - Концептуальна модель предметної області “Облік книгозбірні”

**Логічна модель (схема) БД “Облік бібліотеки”(*пункт звіту №2*)**

У логічній моделі (Рисунок 2):

1.Сутність “Користувачі” перетворена в таблицю “Users”; 2.Сутність “Автори” перетворена в таблицю “Authors”; 3.Сутність “Властивості книг” перетворена в таблицю

“BooksData”;

4.Сутність “Книги” перетворена в таблицю “Books”; 5.Сутність “Записи в абонементі” перетворена в таблицю “Loan”.

Оскільки відношення між сутностями “Книги” та “Користувачі” R(N:M), то було створено додаткову таблицю під назвою “Loan”.

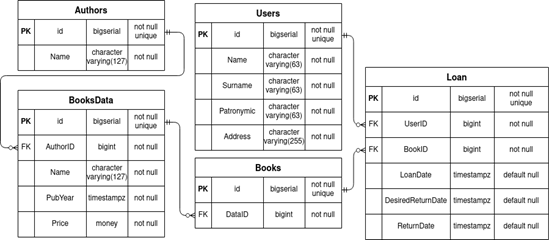
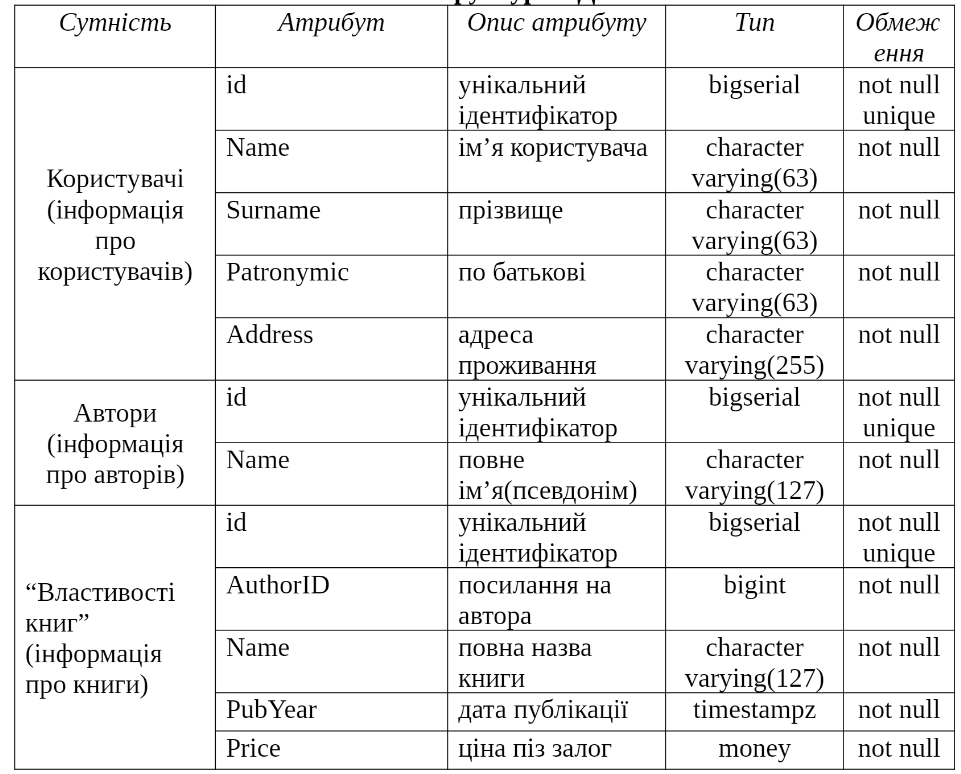
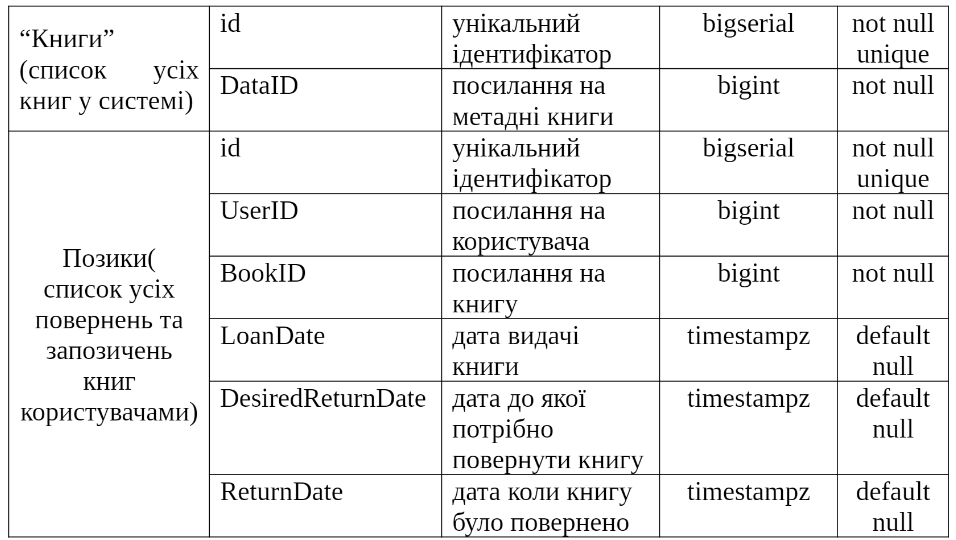


Рисунок 2 - Логічна модель предметної області “Облік бібліотеки”.

# Опис структури БД



****

**Відповідність нормальним формам(*пункт звіту №3*)**

Схема відповідає 1НФ, а саме:

1. Кожна таблиця має основний ключ (простий атрибут, що однозначно ідентифікує запис);
2. Кожний атрибут має лише одне значення, а не множину значень(масиви і списки в одному полі відсутні).
3. У таблиці відсутні ідентичні записи.
4. Кожна клітинка таблиці зберігає простий тип даних, тобто просте(не складене) значення.
5. Стовпці зберігають дані одного типу

Схема відповідає 2НФ, а саме: 1.Схема відповідає вимогам 1НФ;

2.Відсутність залежності не ключових полів від частини складеного ключа(не ключові атрибути таблиці “Loans” залежать лише від комбінації ідентифікатора книги та ідентифікатора користувача).

Схема відповідає 3НФ, а саме: 1.Схема відповідає вимогам 2НФ;

1. Дані в таблицях виняткого залежать від повного ключа(атрибут id у кожній таблиці), тобто відсутня залежність не ключових полів від інших не ключових полів.
2. Функціональні залежності:
   * Authors

1. id → Name (псевдонім конкретного автора залежить від його унікального ідентифікатора)

* + Books

1. id → DataID(FK) → AuthorID(FK) → Name (псевдонім автора книги; відношення транзитивне та послідовно включає в себе ключові атрибути)
2. id → DataID(FK) → Name (назва книги залежить від її унікального ідентифікатора; відношення транзитивне та послідовно включає в себе ключові атрибути)
3. id → DataID(FK) → PubYear (дата видавництва книги залежить від її унікального ідентифікатора; відношення транзитивне та послідовно включає в себе ключові атрибути)
4. id → DataID(FK) → Price (ціна книги залежить від її унікального ідентифікатора; відношення транзитивне та послідовно включає в себе ключові атрибути)
   * BooksData
5. id → AuthorID(FK) → Name (псевдонім автора книги; відношення транзитивне, але воно включає в себе ключовий атрибут)
6. id → Name (назва книги залежить від її унікального ідентифікатора)
7. id → PubYear (дата видавництва книги залежить від її унікального ідентифікатора)
8. id → Price (ціна книги залежить від її унікального ідентифікатора)
   * Users
9. id → Name (ім’я користувача залежить від його унікального ідентифікатора)
10. id → Surname (прізвище користувача залежить від його унікального ідентифікатора)
11. id → Patronymic (по батькові користувача залежить від його унікального ідентифікатора)
12. id → Address (адреса проживання користувача залежить від його унікального ідентифікатора)
    * Loan
13. id → UserID(FK) → Name (ім’я користувача, який запозичив книгу; відношення транзитивне та послідовно включає в себе ключові атрибути)
14. id → UserID(FK) → Surname (прізвище користувача , який запозичив книгу; відношення транзитивне та послідовно включає в себе ключові атрибути)
15. id → UserID(FK) → Patronymic (по батькові користувача , який запозичив книгу; відношення транзитивне та послідовно включає в себе ключові атрибути)
16. id → UserID(FK) → Address (адреса проживання користувача, який запозичив книгу; відношення транзитивне та послідовно включає в себе ключові атрибути)
17. id → BookID(FK) → DataID(FK) → AuthorID(FK) → Name (псевдонім автора, який написал запозичену книгу; відношення транзитивне та послідовно включає в себе ключові атрибути)
18. id → BookID(FK) → DataID(FK) → Name (назва книги, яку було запозичено; відношення транзитивне та послідовно включає в себе ключові атрибути)
19. id → BookID(FK) → DataID(FK) → PubYear (рік видавництва книги, яку було запозичено; відношення транзитивне та послідовно включає в себе ключові атрибути)
20. id → BookID(FK) → DataID(FK) → Price (ціна книги, яку було запозичено; відношення транзитивне та послідовно включає в себе ключові атрибути)
21. id → (UserID, BookID) → LoanDate (дата позики книги залежить від комбінації ідентифікаторів користувача, який запозичив книгу, та книги, яку запозичили; відношення транзитивне, залежить від пари ключових атрибутів)
22. id → (UserID, BookID) → DesiredReturnDate (бажана дата повернення книги залежить від комбінації ідентифікаторів користувача, який запозичив книгу, та книги, яку запозичили; відношення транзитивне, залежить від пари ключових атрибутів)
23. id→(UserID, BookID)→ReturnDate (дата повернення книги залежить від комбінації ідентифікаторів користувача, який повернув книгу, та книги, яку повернули; відношення транзитивне, залежить від пари ключових атрибутів)

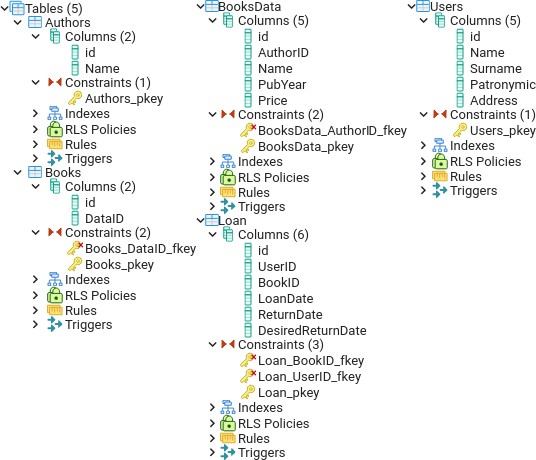
**Структура БД “Облік книгозбірні” в pgAdmin 4(*пункт звіту №4*)**

Рисунок 3 - Структура таблиць предметної області “Облік книгозбірні” в pgAdmin 4

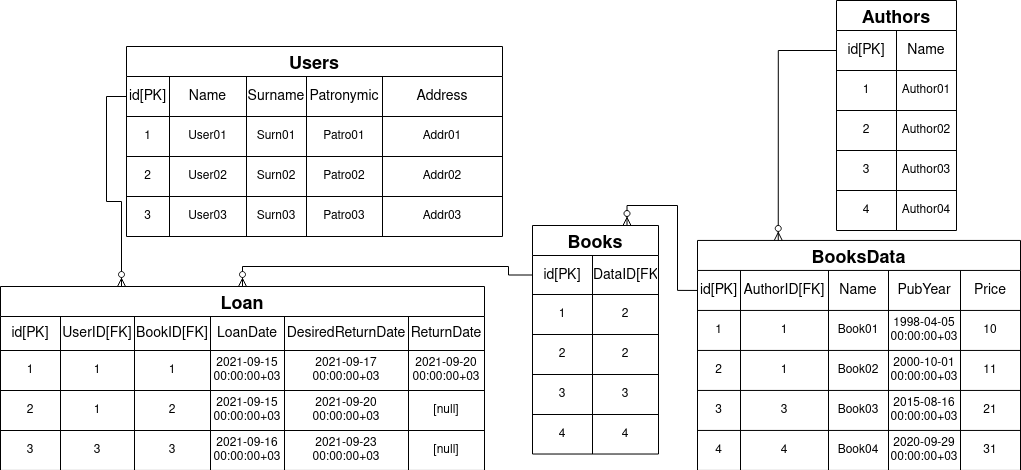


Рисунок 4 - Вміст таблиць бази даних.