A black and white drawing of a building

Description automatically generated

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 ІІI курсу

групи КВ-22

Вакульчук Ярослав

Перевірив:

Павловский В. І.

Київ – 2024

**Мета:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання:**

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 області**

Дана предметна область – онлайн-платформа для обміну книгами між користувачами. Вона охоплює інструменти для моніторингу, зберігання та управління інформацією про книги, користувачів, їхні оголошення та заявки на обмін. Система дозволяє користувачам створювати оголошення про обмін книгами, переглядати доступні книги інших користувачів та подавати заявки на обмін.

**Опис сутностей**

Для побудови концептуальної моделі обраної предметної області, були виділені такі сутності:

1. Користувач (User)

Атрибути ідентифікатор користувача, ім’я, прізвище, електронна пошта.

Призначення: збереження інформації про зареєстрованих користувачів платформи.

1. Книга (Book)

Атрибути: ідентифікатор книги, назва, автор, жанр, стан.

Призначення: збереження інформації про книги, доступні для обміну.

1. Оголошення (Listing)

Атрибути: ідентифікатор оголошення, статус оголошення, дата створення.

Призначення: збереження інформації про оголошення, створені користувачами, статус оголошення.

**Опис зв’язків між сутностями**

Зв’язок-сутність «Користувач» - «Книга» є зв’язком 1:N. Один користувач може додати кілька книг для обміну, але кожна книга належить тільки одному користувачу.

Зв’язок-сутність «Книга» - «Оголошеня» є зв’язком 1:1. Одна книга може бути пов’язана лише з одним оголошенням.

Зв’язок-сутність «Користувач» - «Оголошення» є зв’язком M:N. Один користувач може створити кілька оголошень на обмін та кожне оголошення може прийняти декілька користувачів.

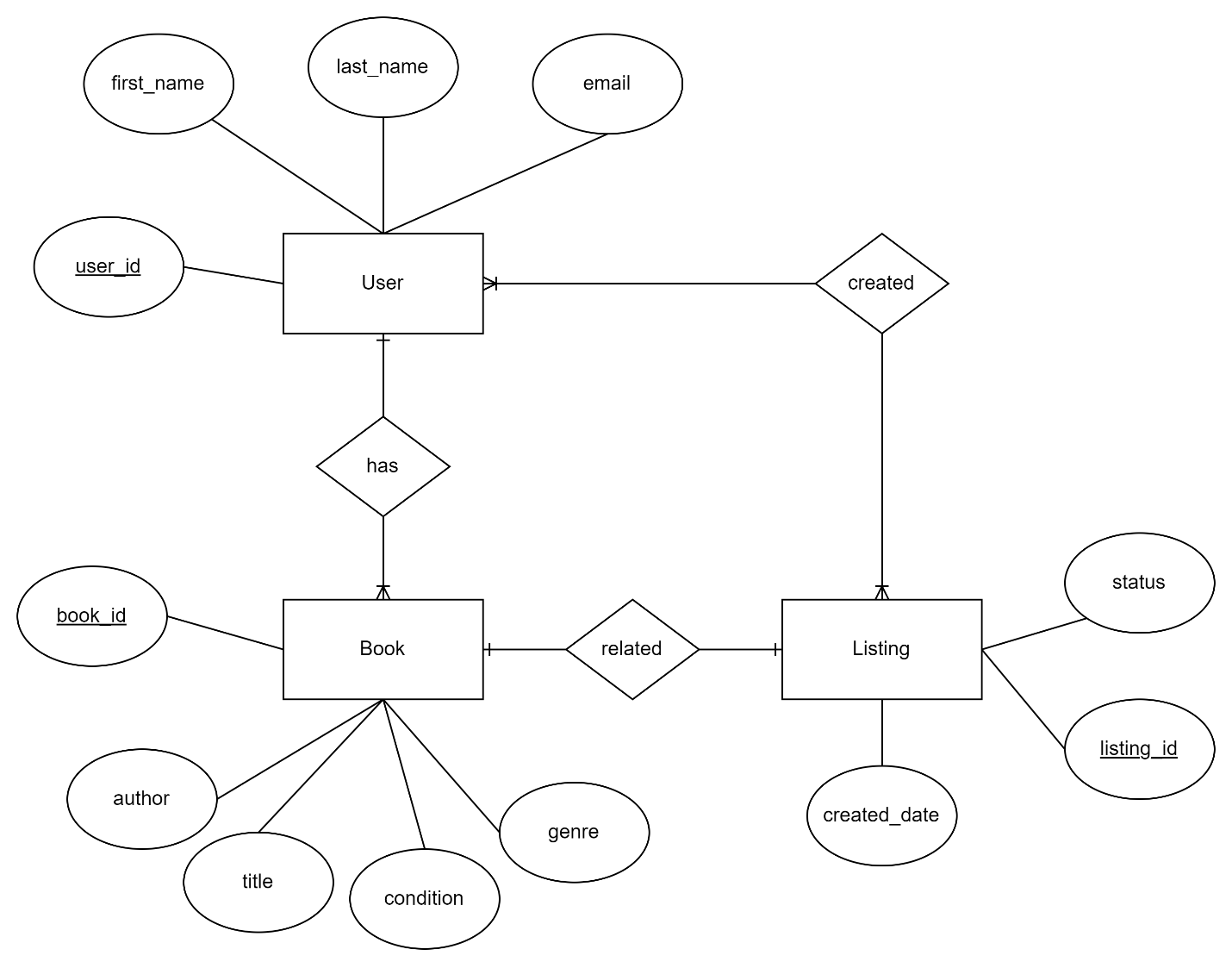


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Сутність User перетворено в таблицю User з первинним ключем user\_id, атрибутами first\_name, last\_name та email.

Сутність Book перетворено в таблицю Book з первинним ключем book\_id, атрибутами title, author, genre, condition, зовнішнім ключем user\_id.

Сутність Listing перетворено в таблицю Listing з первинним ключем listing\_id, атрибутом created\_date та зовнішнім унікальнім ключем book\_id.

Зв’язок-сутність «Користувач» - «Оголошення», яке є зв’язком M:N, перетворено в таблицю Request, для перетворення в зв’язки 1:N, з первинним ключем request\_id та зовнішніми ключами user\_id та listing\_id.

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

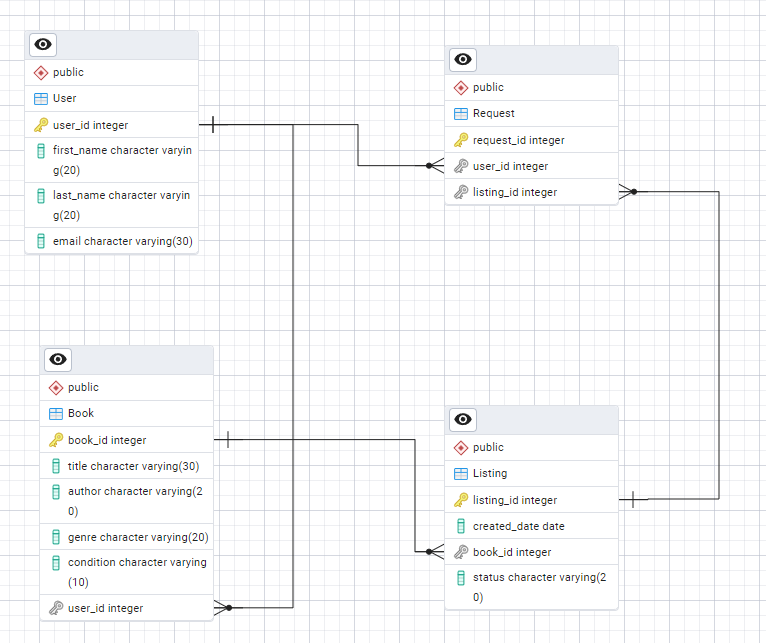


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| *User* – містить дані про користувача. | *user\_id* – унікальний ідентифікатор користувача. | *integer* (число) |
| *first\_name* – ім’я. | *character varying* (рядок) |
| *last\_name* – прізвище. | *character varying* (рядок) |
| *email* – електронна пошта. | *character varying* (рядок) |
| *Book* – містить дані про книжку. | *book\_id* – унікальний ідентифікатор книжки. | *integer* (число) |
| *title* – назва книжки. | *character varying* (рядок) |
| *author* – автор книжки. | *character varying* (рядок) |
| *genre* – жанр книжки. | *character varying* (рядок) |
| *condition* – стан книжки. | *character varying* (рядок) |
| *user\_id* – ідентифікатор користувача. | *integer* (число) |
| *Listing* – містить дані про оголошення. | *listing\_id* – унікальний ідентифікатор оголошення. | *integer* (число) |
| *created\_date* – дата створення заявки. | *date* (дата) |
| *status –* статус оголошення. | *character varying* (рядок) |
| *book\_id* – унікальний ідентифікатор книжки. | *integer* (число) |
| *Request* – містить інформацію про заявку. | *request\_id* – унікальний ідентифікатор заявки. | *integer* (число) |
| *user\_id* – ідентифікатор користувача. | *integer* (число) |
| *listing\_id* – ідентифікатор оголошення. | *integer* (число) |

**Функціональні залежності для кожної таблиці**

1. User:
   1. user\_id → {first\_name, last\_name, email}
2. Book:
   1. book\_id → {title, author, genre, condition, user\_id}
3. Listing:
   1. listing\_id → {created\_date, book\_id}
4. Request:
   1. request\_id → {user\_id, listing\_id}

Ці функціональні залежності вказують на те, які атрибути в кожній таблиці визначаються від інших атрибутів. Це важливо для нормалізації та управління базою даних.

Транзитивні функціональні залежності виникають, коли один атрибут функціонально визначає інший через інший атрибут. Іншими словами, якщо A визначає B, а B визначає C, то ми можемо сказати, що A транзитивно визначає C. Дані таблиці не мають транзитивних функціональних залежностей.

**Відповідність схеми нормальним формам**

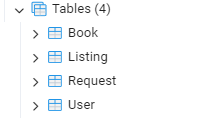
Вона відповідає 1НФ, оскільки всі атрибути є атомарними (недільними) та мають просту структуру, не містять складних об'єктів або масивів.

Вона відповідає 2НФ, оскільки всі неключові атрибути залежать від цілісних ключів цих таблиць. Наприклад, атрибут "title" в таблиці "Book" залежить від ідентифікатора "book\_id".

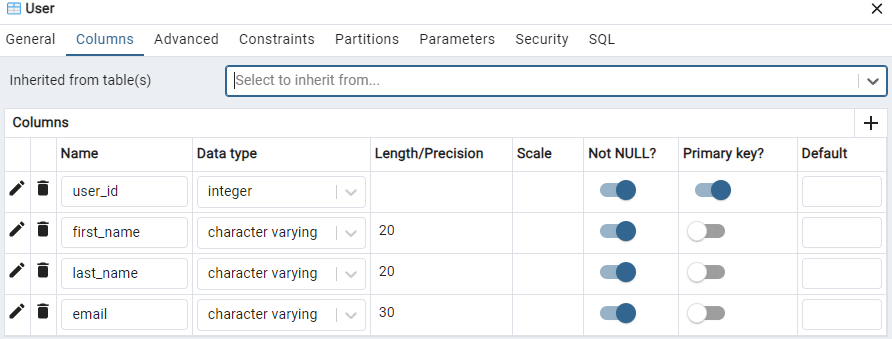
Вона також відповідає 3НФ, оскільки усі неключові атрибути мають бути функціонально залежними від первинного ключа, тобто не може бути взаємозалежностей між неключовими атрибутами. Кожен атрибут безпосередньо пов'язаний з ключами, від яких він залежить, що і виконується у нас.

Загальний висновок полягає в тому, що ця база даних добре структурована і нормалізована, що робить її ефективною та надійною для зберігання і обробки даних.

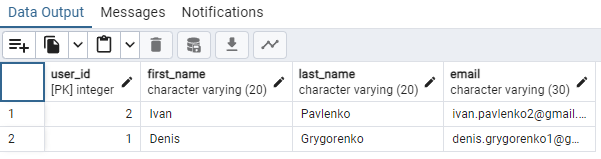
**Таблиці бази даних у pgAdmin4**



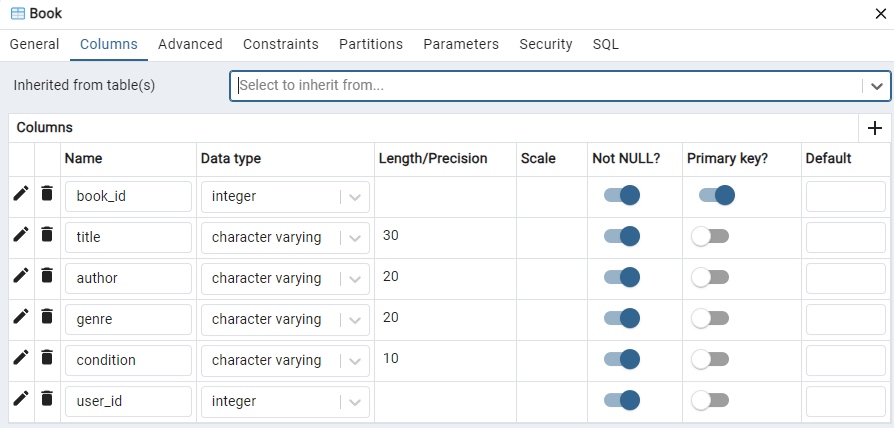
User

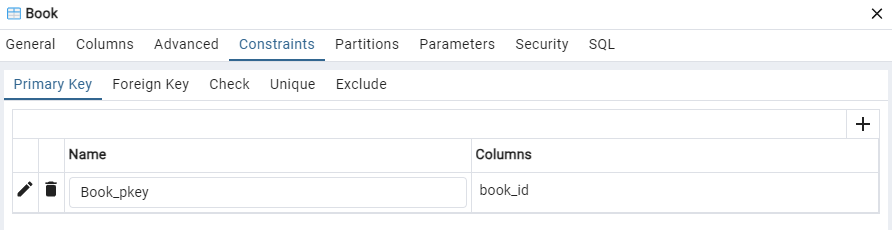




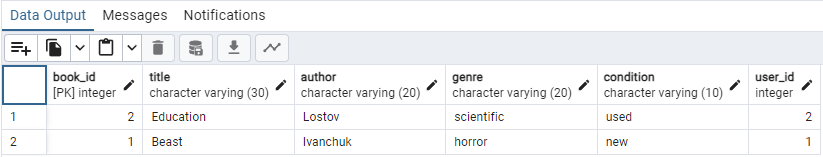


Book

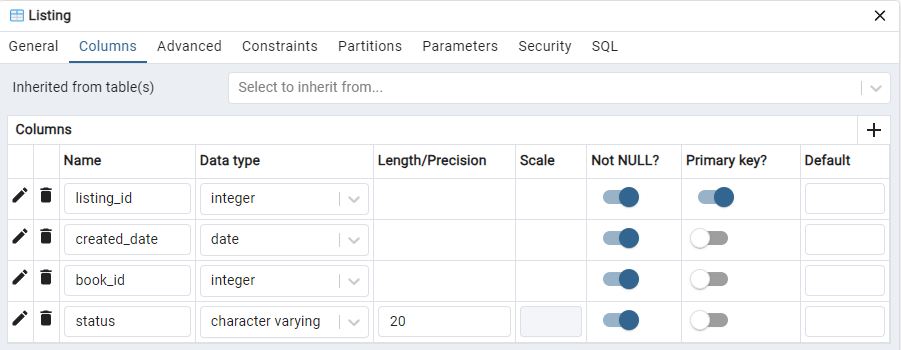


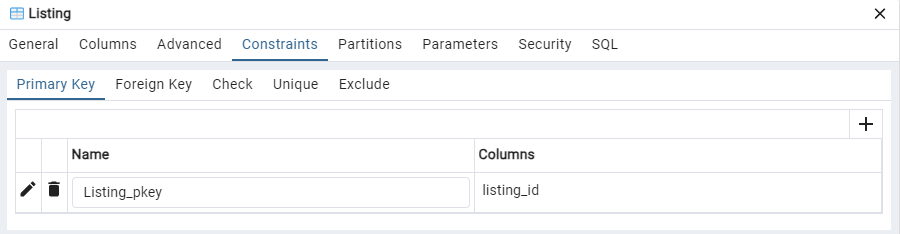


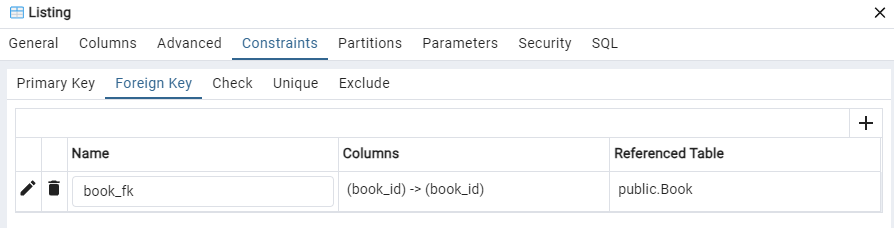


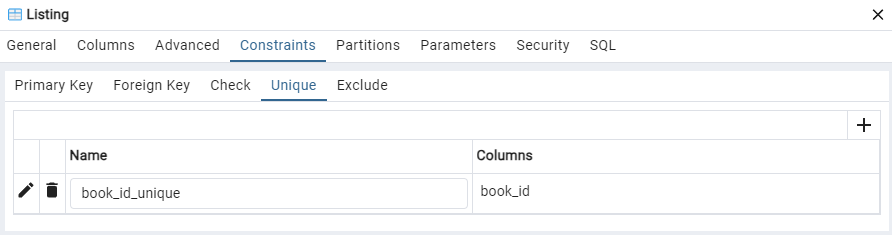


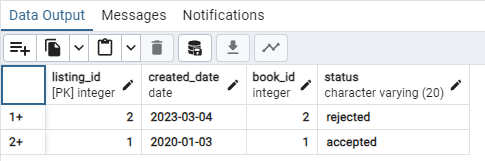
Listing



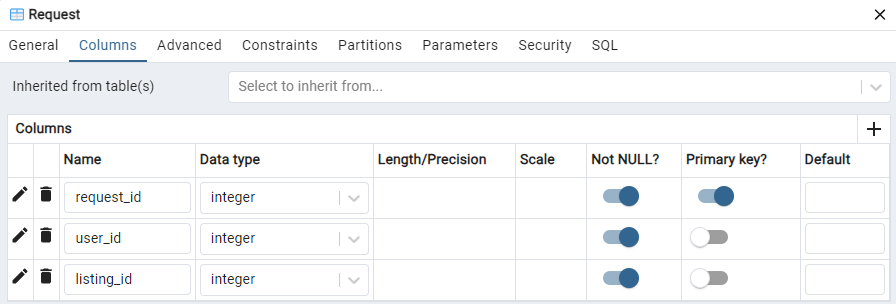


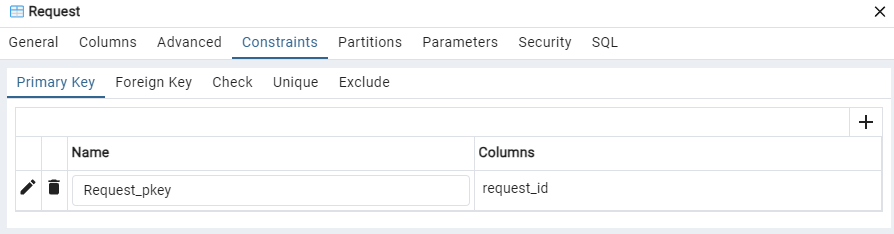




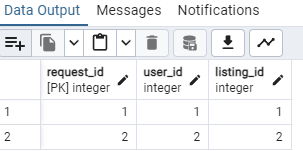


Request









**Висновок**

У ході лабораторної роботи було створено базу даних для онлайн-платформи обміну книгами між користувачами, яка містить інформацію про користувачів, книги, оголошення та запити на обмін, а також їх взаємозв’язки. Для графічного відображення структури бази даних використовувалася ER-діаграма у нотації Чена, що дозволило чітко позначити сутності, зв’язки та атрибути. Схема бази даних була перевірена на відповідність першій, другій та третій нормальним формам, що забезпечує надійне, структуроване та ефективне зберігання інформації. Додатково було створено таблицю "Request", яка реалізує зв’язок між оголошеннями та користувачами, дозволяючи кожному оголошенню включати кілька користувачів. Завершуючи, були додані скріншоти з pgAdmin4, що демонструють властивості стовпців, обмеження, зовнішні ключі та вміст таблиць бази даних у PostgreSQL. У підсумку, розроблена база даних повністю відповідає поставленим вимогам платформи для обміну книгами та готова до використання.

**Репозиторій Git:** https://github.com/Yaroslavvak/lab