

Завдання 3. Розробка архітектури проекту та DB

1. Визначити які дані будуть зберігатися в базі даних. Розробити схему бази даних з таблицями у вигляді відповідної UML діаграми. Для кожної таблиці вказати дані, які там будуть зберігатися, типи, а також залежності між таблицями.

1.Users (Користувачі)

- id (PK, INT, AUTO_INCREMENT)
- name (VARCHAR) - ім'я
- email (VARCHAR, UNIQUE)
- password (VARCHAR, hashed) - пароль
- about (TEXT, optional) - про_себе
- region (VARCHAR) - область
- district (VARCHAR) - район
- city (VARCHAR) - місто
- role (ENUM: user / admin) - користувач / адміністратор

Логіка: тут зберігаються акаунти всіх людей, які працюють із системою.

2.Genres (Жанри)

- id (PK, INT)
- name (VARCHAR) - назва

Наприклад: фантастика, детектив, роман тощо.

3.BooksGenres (КнигиЖанри — зв'язок N:M)

- book_id (FK → Books.id)
- genre_id (FK → Genres.id)

Так можна призначити кілька жанрів одній книзі.

4. Posts (Оголошення)

- id (PK, INT)
- user_id (FK → Users.id)
- title (VARCHAR)
- author (VARCHAR)
- deal_type (ENUM: exchange / donation) - обмін / дарування
- description (TEXT) - опис
- photo_url (VARCHAR)

Це ключова таблиця — тут публікуються оголошення.

5. Complaints (Скарги)

- id (PK, INT)
- text (TEXT) - текст
- date (DATETIME) - дата
- post_id (FK → Posts.id)
- complainant_id (FK → Users.id) - id людини, яка подає скаргу

Адміністратор може їх розглядати.

Зв'язки між таблицями:

1. Users ↔ Posts (Один-до-багатьох)

- **Один** користувач може створити **багато** оголошень
- **Одне** оголошення належить **одному** користувачу

2. Posts ↔ Complaints (Один-до-багатьох)

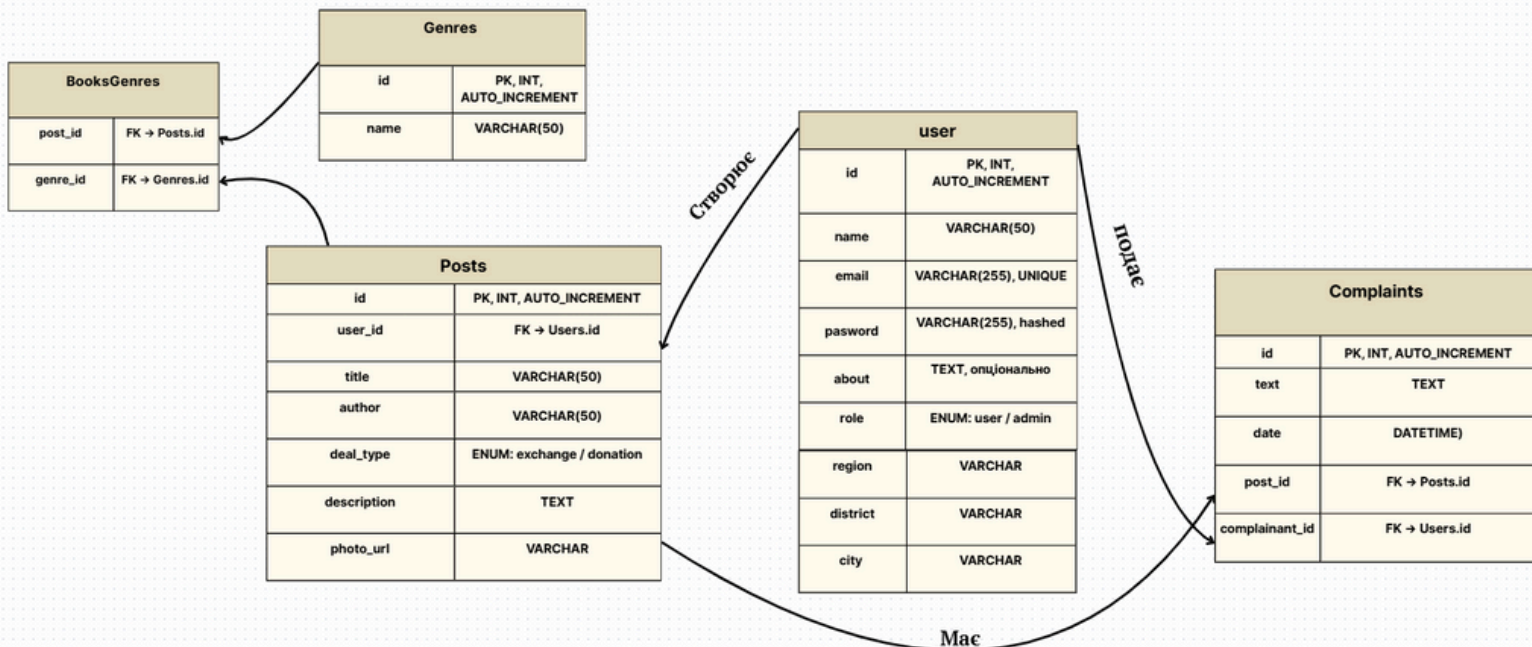
- **Одне** оголошення може мати **багато** скарг
- **Одна** скарга стосується **одного** оголошення

3. Users ↔ Complaints (Один-до-багатьох)

- **Один** користувач може подати **багато** скарг (як скаржник)
- **Одна** скарга подається **одним** користувачем

4. Posts ↔ Genres (Багато-до-багатьох)

- **Одне** оголошення книги може мати **кілька** жанрів
- **Один** жанр може належати **багатьом** оголошенням.
- Зв'язок реалізується через **проміжну таблицю BooksGenres**



Провайдер баз даних: PostgreSQL, Хостинг: Supabase

1. Створити базу даних, використовуючи написаний скрипт
2. Створити Console App (.NET Core) проект, Використовуючи ADO.Net підключити до проекту вашу базу даних.
3. На консоль вивести дані з таблицок. Створити метод в цій консольній апп, для заповнення бази даних тестовими рандомними даними через ADO.Net по 30-50 записів в кожній таблиці
4. Створити скрипт для генерації бази даних. Створити ER діаграму