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”*

Виконала:

студент ІІI курсу

групи 34

Гудима Кирило

Перевірив:

Павловский В. І.

Київ – 2025

**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**

Предметна область — система управління донорами крові.

Система призначена для автоматизації обліку донорів, проведених донацій, запасів крові, а також взаємодії між донорами, медичними працівниками та центрами переливання.

Основними функціями системи є:

* збереження інформації про донорів (їх персональні дані, групу крові, стан здоров’я, історію донацій);
* облік проведених донацій (дата, обсяг, тип крові, відповідальний лікар);
* управління медичними працівниками (лікарі, які проводять процедури забору крові);
* аналіз крові (група, резус-фактор, кількість, термін придатності).

Система забезпечує прозоре відстеження історії донорства, підвищує безпеку та ефективність управління кров’ю.

**Опис сутностей**

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

1. Донор (Donor)

Атрибути: ідентифікатор донора; ПІБ донора; картка донора.

Призначення: збереження основної інформації про донорів крові.

1. Лікар (Doctor)

Атрибути**:** ідентифікатор лікаря; ПІБ лікаря, досвід.  
Призначення**:** збереження інформації про медичних працівників, які проводять донації.

1. Донація(Donation)

**Атрибути:** ідентифікатор донації; час донації; обсяг крові; **Призначення:** збереження інформації про факти здачі крові донорами під контролем лікарів.

1. Скринінг крові(Blood\_Screen)

**Атрибути:** ідентифікатор скринінгу; результат; група крові; **Призначення:** збереження інформації про стан крові.

**Опис зв’язків між сутностями**

Зв’язок-сутність “Донор” – “Донація” є зв'язок N:M. Один донор може бути зареєстрований на кілька донацій, і одна донація має багато донорів.

Зв’язок-сутність “Донація” – “Доктор” зв'язок N:M. Один доктор може проводити багато донацій, і донація може мати багато різних лікарів.

Зв’язок-сутність “Скрин крові” – “Донація” зв'язок 1:N. Одна донація може мати багато скринів, але скрин може бути лише для певної донації.

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

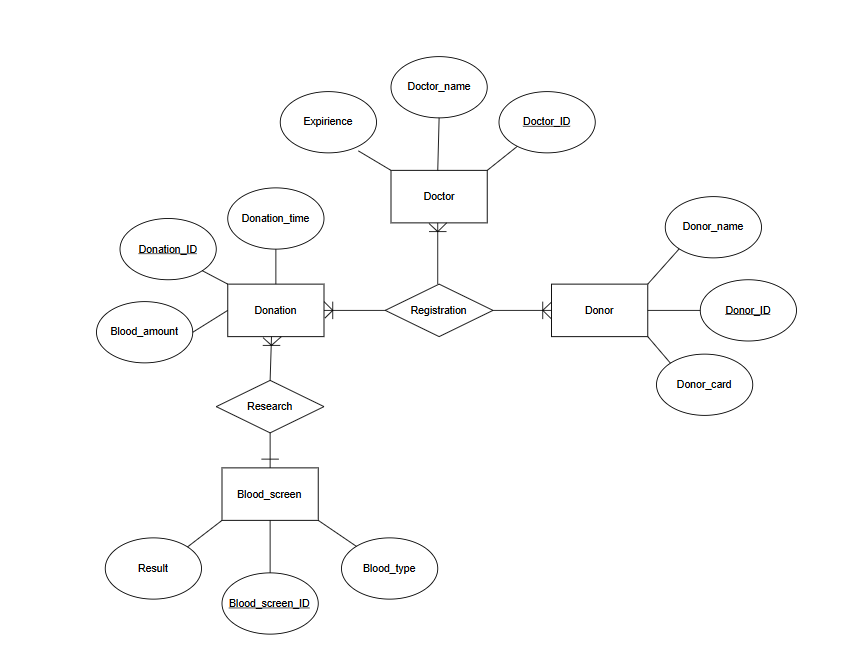


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Сутність **“Donor”** перетворено в таблицю **“Donor”**, що має первинний ключ **Donor\_ID** та атрибути **Donor\_Name**, **Donor\_Card**.

Сутність **“Doctor”** перетворено в таблицю **“Doctor”**, що має первинний ключ **Doctor\_ID** та атрибути **Doctor\_Name, Expirience**.

Сутність **“Donation”** перетворено в таблицю **“Donation”**, що має первинний ключ **Donation\_ID**, атрибути **Donation\_time**, **Blood\_amount**.

Оскільки в логічній моделі безпосередній зв’язок **N:M** є неможливим, а в концептуальній моделі він може існувати між сутностями **Donor,** **Donation** і **Doctor**, для його реалізації було створено проміжну таблицю **Registration** (реєстрація) з первинним ключем Registration\_ID, з атрибутами date, та зовнішніми ключами Doctor\_ID, Donor\_ID та Donation\_ID.

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

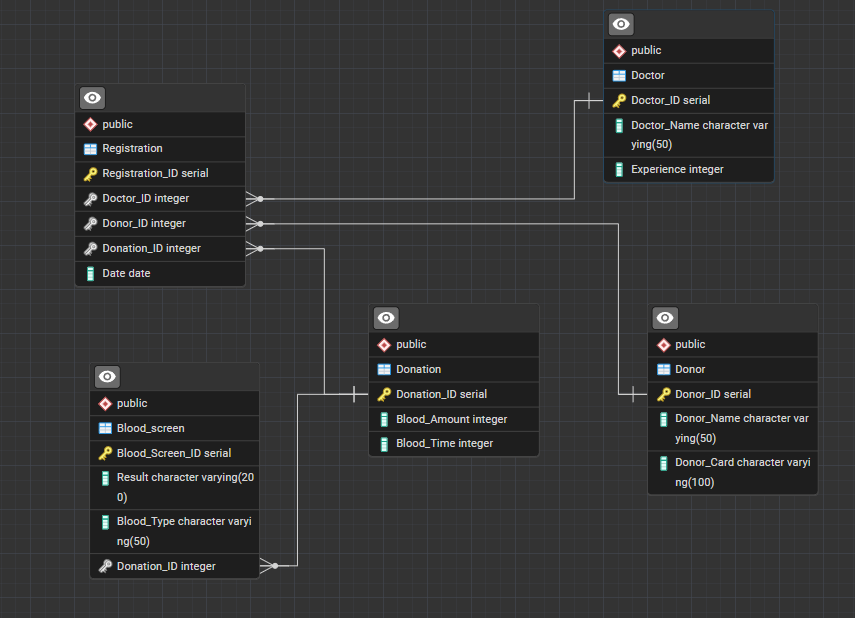


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| Docror – містить дані про доктора | Doctor\_ID - унікальний індефікатор доктора | Integer(число) |
| Doctor\_Name – ПІБ професора | Character varying(рядок) |
| Experience – досвід роботи в роках | Integer(число) |
| Donation – містить дані про донацію | Donation\_ID - унікальний індефікатор донації | Integer(число) |
| Donation\_time – час донації | Integer(число) |
| Blood\_amount – об’єм крові | Integer(число) |
| Donor\_Id - унікальний індефікатор донора | Integer(число) |
| Donor – містить дані про студента | Donor\_ID - унікальний індефікатор донора | Integer(число) |
| Donor\_Name – ПІБ донора | Character varying(рядок) |
| Donor\_Card – картка донор | Character varying(рядок) |
| Registration – містить дані про регістрацію | Registration\_ID - унікальний індефікатор регістрації | Integer(число) |
| Donor\_ID - унікальний індефікатор донора | Character varying(рядок) |
| Date – факт(дата) реєстрації | Integer(число) |
| Doctor\_ID - унікальний індефікатор доктора | Integer(число) |
| Donation\_ID - унікальний індефікатор донації | Integer(число) |
|  |  |  |
| Blood\_screen – скринінг крові(аналіз) | Blood\_screen \_ID - унікальний індефікатор скринінгу | Integer(число) |
| Blood\_type – група крові | Character varying(рядок) |
| Result – результат | Character varying(рядок) |

**Функціональні залежності для кожної таблиці**

1. **Donation:**
2. **Donation\_ID – {Donation\_time, blood\_amount}**
3. **Donation\_time – {Blood\_amount}**
4. Donor:
5. Donor\_ID – {Donor\_Name, Donor\_card}
6. Donor\_Name – {Donor\_card}
7. Doctor:
8. Doctor\_ID – {Doctor\_Name, Experience}
9. Experience – {Doctor\_Name}
10. Donation:
11. Donation\_ID – {Donation\_time, Blood\_amount }
12. Donation\_time – {Blood\_amount}
13. Registration:
14. Registration\_ID – { Doctor\_ID, Donation\_ID, data, Donor\_ID}
15. Blood\_screen:
16. Blood\_screen\_ID – {Result, Blood\_type}
17. Result – {Blood\_type}

**Відповідність схеми нормальним формам**

База даних відповідає **першій нормальній формі (1НФ)**, оскільки всі її атрибути є атомарними, тобто неподільними. Вони мають просту структуру та не містять складних об’єктів чи масивів.

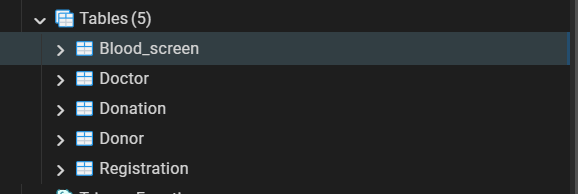
Вона також задовольняє вимоги **другої нормальної форми (2НФ)**, адже всі неключові атрибути повністю залежать від усього первинного ключа. Наприклад, атрибут "Name" у таблиці "Course" залежить від ідентифікатора "Course\_ID".

Крім того, база відповідає **третій нормальній формі (3НФ)**, оскільки між неключовими атрибутами відсутні транзитивні залежності — кожен із них безпосередньо залежить лише від первинного ключа.

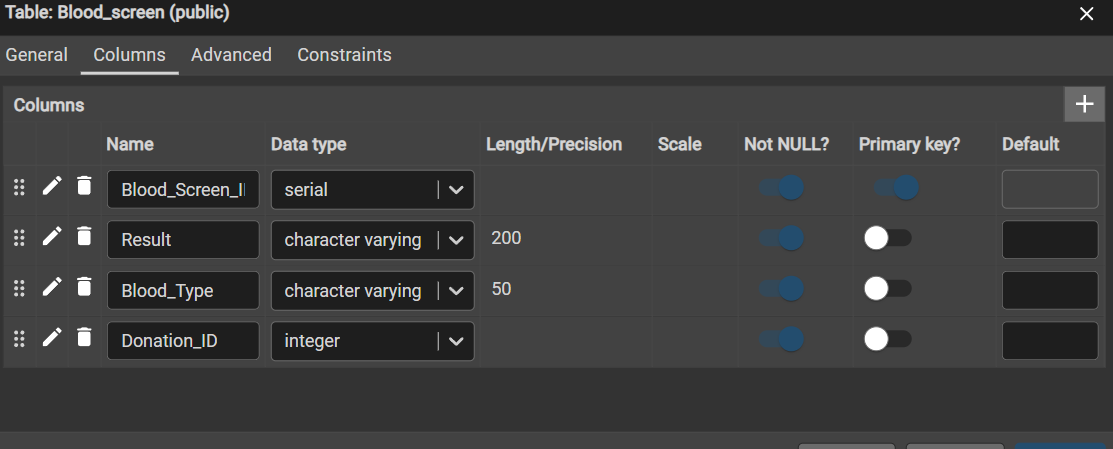
**Отже**, база даних є якісно структурованою та повністю нормалізованою, що забезпечує її ефективність, надійність і зручність у зберіганні та обробці інформації.

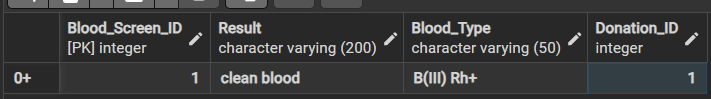
**Таблиці бази даних у pgAdmin4**

Таблиці у pgAdmin4

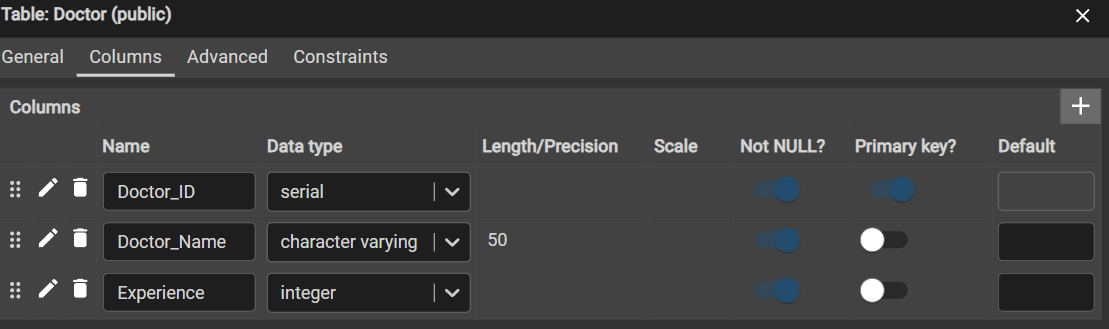


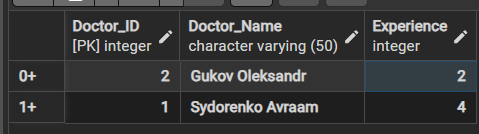
Таблиця Blood\_Screen



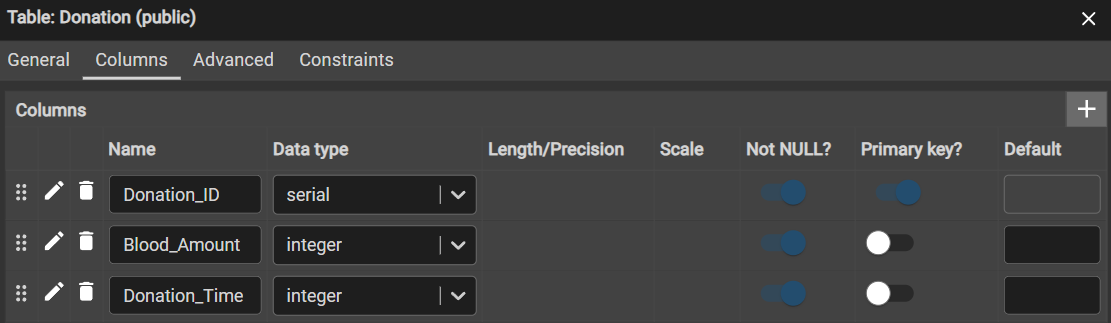


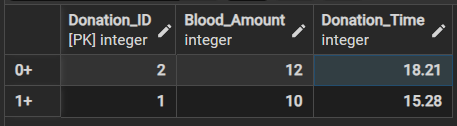
Таблиця Doctor



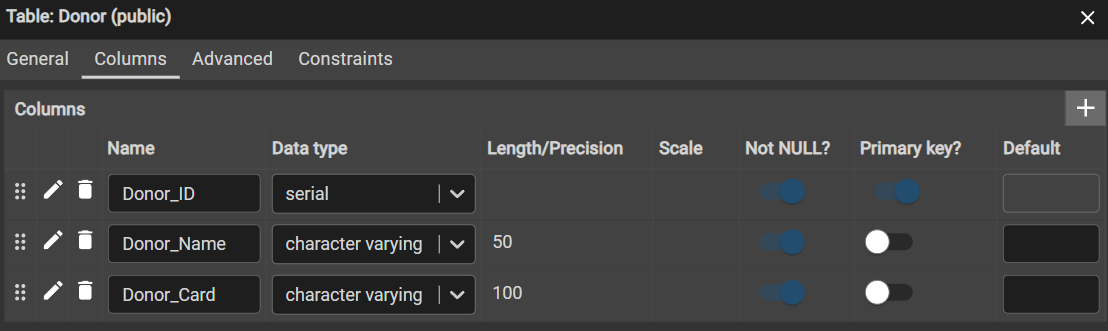


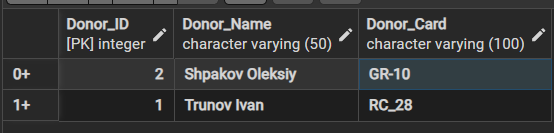
Таблиця Donation



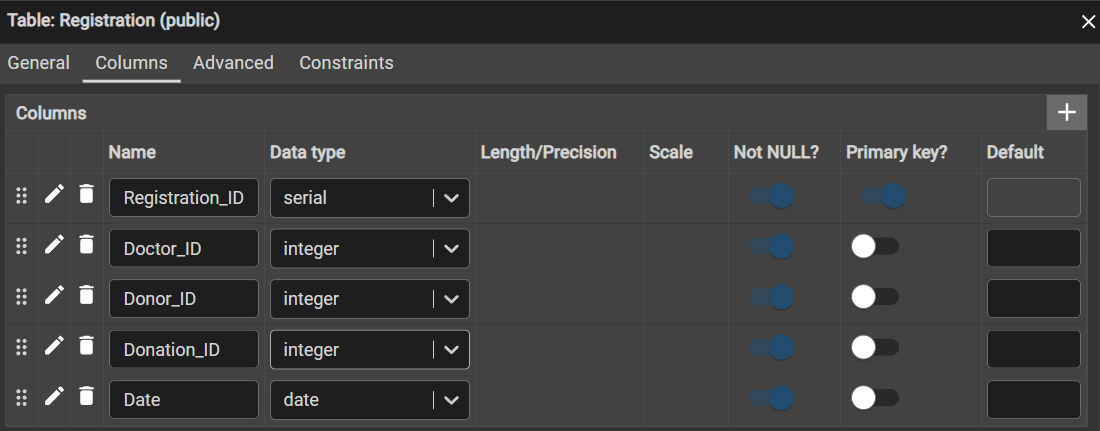


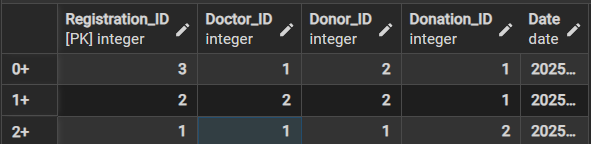
Таблиця Donor





Таблиця Registration





**Таблиці в коді SQL**

-- This script was generated by the ERD tool in pgAdmin 4.

-- Please log an issue at https://github.com/pgadmin-org/pgadmin4/issues/new/choose if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public."Blood\_screen"

(

"Blood\_Screen\_ID" serial NOT NULL,

"Result" character varying(200) COLLATE pg\_catalog."default" NOT NULL,

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

"Donation\_ID" integer NOT NULL,

CONSTRAINT "Blood\_screen\_pkey" PRIMARY KEY ("Blood\_Screen\_ID")

);

CREATE TABLE IF NOT EXISTS public."Donation"

(

"Donation\_ID" serial NOT NULL,

"Blood\_Amount" integer NOT NULL,

"Donation\_Time" integer NOT NULL,

CONSTRAINT "Donation\_pkey" PRIMARY KEY ("Donation\_ID")

);

CREATE TABLE IF NOT EXISTS public."Registration"

(

"Registration\_ID" serial NOT NULL,

"Doctor\_ID" integer NOT NULL,

"Donor\_ID" integer NOT NULL,

"Donation\_ID" integer NOT NULL,

"Date" date NOT NULL,

CONSTRAINT "Registration\_pkey" PRIMARY KEY ("Registration\_ID")

);

CREATE TABLE IF NOT EXISTS public."Doctor"

(

"Doctor\_ID" serial NOT NULL,

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

"Experience" integer NOT NULL,

CONSTRAINT "Doctor\_pkey" PRIMARY KEY ("Doctor\_ID")

);

CREATE TABLE IF NOT EXISTS public."Donor"

(

"Donor\_ID" serial NOT NULL,

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

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

CONSTRAINT "Donor\_pkey" PRIMARY KEY ("Donor\_ID")

);

ALTER TABLE IF EXISTS public."Blood\_screen"

ADD CONSTRAINT fk\_donation\_screen FOREIGN KEY ("Donation\_ID")

REFERENCES public."Donation" ("Donation\_ID") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE;

ALTER TABLE IF EXISTS public."Registration"

ADD CONSTRAINT fk\_doctor FOREIGN KEY ("Doctor\_ID")

REFERENCES public."Doctor" ("Doctor\_ID") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE;

ALTER TABLE IF EXISTS public."Registration"

ADD CONSTRAINT fk\_donation FOREIGN KEY ("Donation\_ID")

REFERENCES public."Donation" ("Donation\_ID") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE;

ALTER TABLE IF EXISTS public."Registration"

ADD CONSTRAINT fk\_donor FOREIGN KEY ("Donor\_ID")

REFERENCES public."Donor" ("Donor\_ID") MATCH SIMPLE

ON UPDATE CASCADE

ON DELETE CASCADE;

END;

**Висновок**

Під час виконання лабораторної роботи була створена база даних для системи управління процесом донорства крові. Розроблена база даних містить інформацію про лікарів, донорів, донації крові, результати аналізів та їх взаємозв’язки. Для візуального представлення структури бази даних була побудована ER-діаграма, яка відображає зв’язки між сутностями за допомогою нотації Чена. Діаграма дозволяє наочно побачити логічну структуру даних та взаємодію між таблицями.

Схема бази даних була перевірена на відповідність першій, другій та третій нормальним формам (1НФ, 2НФ, 3НФ). Усі таблиці задовольняють вимоги нормалізації, що забезпечує відсутність надлишкових даних і підвищує ефективність зберігання інформації.

У кінці роботи були представлені знімки екрана з pgAdmin4, які демонструють структуру таблиць, зовнішні ключі, обмеження цілісності та приклади заповнених записів.

Загалом, розроблена база даних повністю відповідає поставленим завданням, є структурованою, логічно побудованою та готовою до практичного використання в системі управління процесом донорства крові.

**Контакти:**

Посилання на репозиторій GitHub - https://github.com/kiril8245/Database\_Hudyma\_Kyrylo\_KV-34