

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

ФПМ групи КВ-23

Мамчиць Микола Валентинович

Перевірив: Петрашенко А.В.

Київ – 2024

*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 кожну з таблиць засобами pgAdmin 4.

**Nickname Telegram: @b3lqa**

Посилання на репозиторій: <https://github.com/Mykola6117/lab1-DB>

**Опис предметної області «Платформа для зберігання історій хвороб пацієнтів»**

Перелік сутностей та їх призначення:

1.Лікар:

Призначення: представляє лікарів, що ведуть медичні історії та ставлять діагнози.

Основні атрибути:

Лікар\_id: унікальний ідентифікатор лікаря.

П.І.Б: повне ім'я лікаря.

Спеціалізація: спеціалізація лікаря.

2.Історія\_Хвороби:

Призначення: зберігає інформацію про медичні записи пацієнтів, їх діагнози, а також лікаря, що проводив обстеження.

Основні атрибути:

Історія\_хвороби\_id: унікальний ідентифікатор медичної історії.

Назва хвороби : назва діагнозу.

Опис: опис діагнозу або лікування.

3.Пацієнт:

Призначення: представляє пацієнтів, які проходять медичне обстеження та лікування.

Основні атрибути:

Пацієнт\_id: унікальний ідентифікатор пацієнта.

П.І.Б: повне ім'я пацієнта.

стать: стать пацієнта.

Дата народження: дата народження пацієнта.

4.Рецепт:

Призначення: представляє рецепти на ліки, виписані лікарями для пацієнтів у межах їх медичної історії.

Основні атрибути:

Рецепт\_id: унікальний ідентифікатор рецепту.

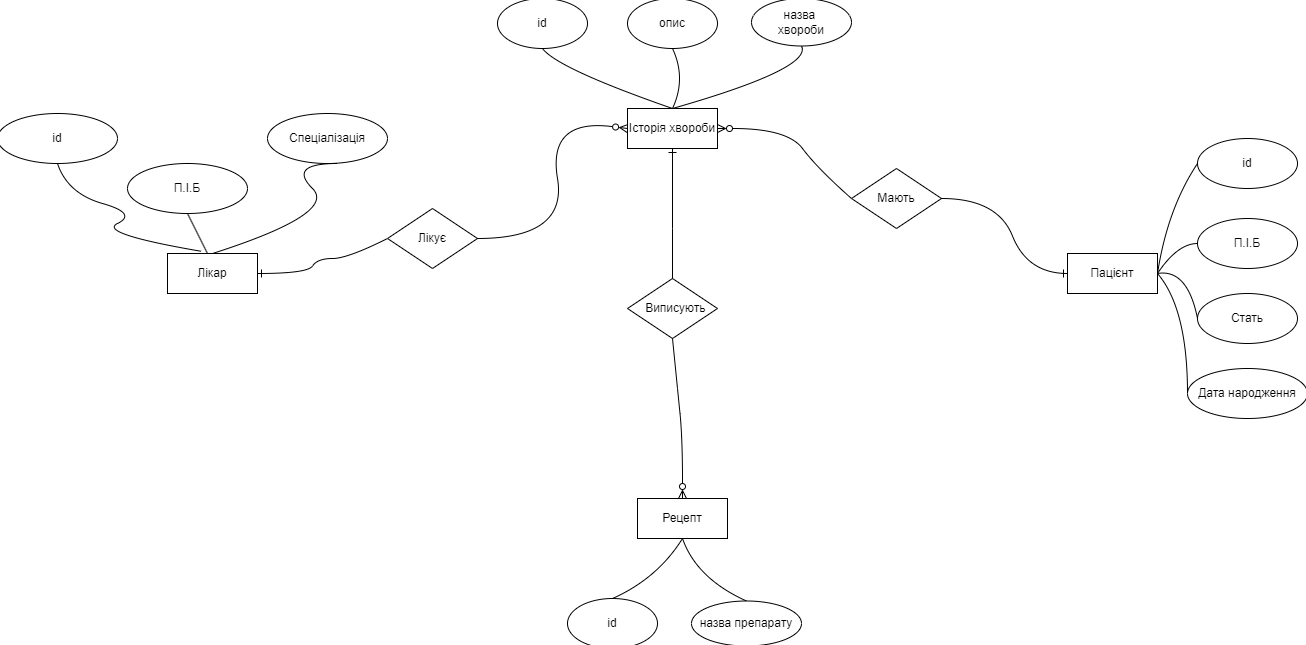
Назва препарату: назва виписаних ліків.

**Опис зв’язків між сутностями предметної області**

Сутність “Лікар” має зв’язок 1:N по відношенню до сутності “Історії хвороби” оскільки один лікар може написати багато історій хвороби.

Сутність “Пацієнт” має зв'язок 1:N по відношенню до “Історії хвороби”, оскільки пацієнт може мати безліч історій хвороби.

Сутність “Історія Хвороби” має зв'язок має зв’язок 1:N по відношенню до “Рецепт”, оскільки одна історія хвороби може містити багато рецептів.

ER-діаграма, побудована за нотацією Чена (draw.io)

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

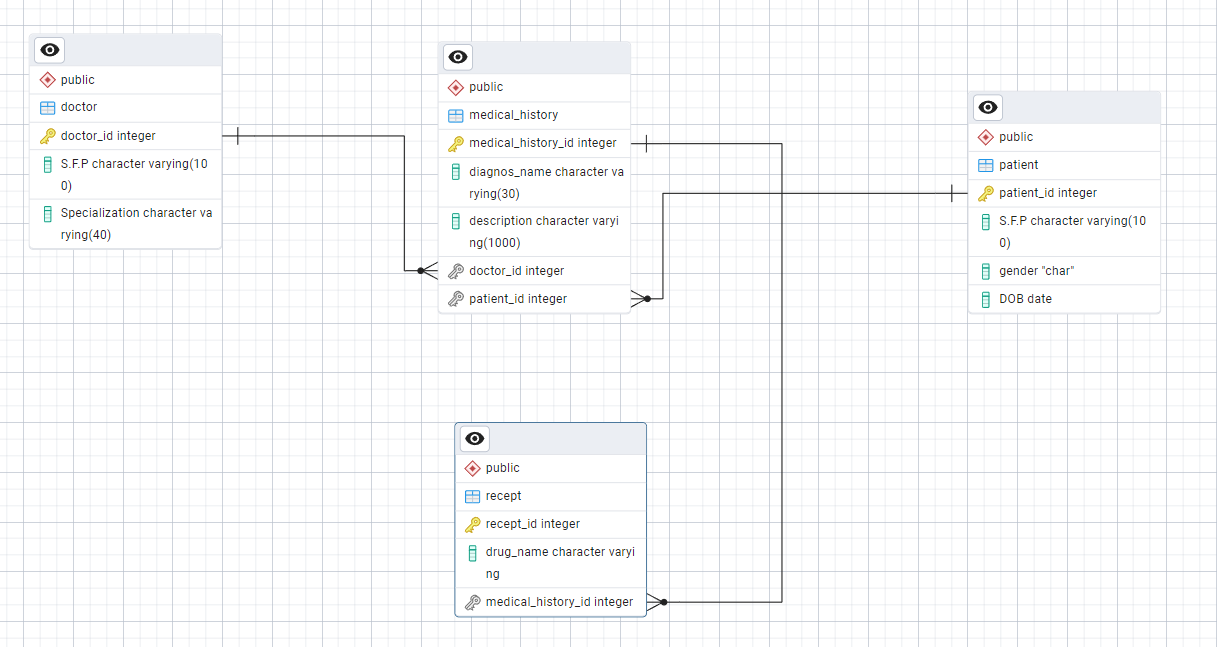
Сутність “Лікар” перетворено у таблицю “doctor”.

Сутність “Історія хвороби” перетворено у таблицю “medical\_history”.

Сутність “Пацієнт” перетворено у таблицю “patient”.

Сутність “Рецепт” перетворено у таблицю “recept”.

**Логічна модель (схема)**



**Опис об’єктів бази даних у вигляді таблиці**

|  |  |  |  |
| --- | --- | --- | --- |
| Сутність | Атрибут | Опис | Тип даних |
| doctor | doctor\_id | Унікальний ідентифікатор лікаря | integer (числовий) |
| doctor | S.F.P | Повне ім'я лікаря. Не допускає NULL | character varying (рядок) |
| doctor | Specialization | Спеціалізація лікаря. Не допускає NULL | character varying (рядок) |
| medical\_history | medical\_history\_id | Унікальний ідентифікатор медичної історії | integer (числовий) |
| medical\_history | diagnos\_name | Назва діагнозу. Не допускає NULL | character varying (рядок) |
| medical\_history | description | Опис діагнозу або лікування. Не допускає NULL | character varying (рядок) |
| medical\_history | doctor\_id | Ідентифікатор лікаря (зв'язок з таблицею doctor) | integer (числовий) |
| medical\_history | patient\_id | Ідентифікатор пацієнта (зв'язок з таблицею patient) | integer (числовий) |
| patient | patient\_id | Унікальний ідентифікатор пацієнта | integer (числовий) |
| patient | S.F.P | Повне ім'я пацієнта. Не допускає NULL | character varying (рядок) |
| patient | gender | Стать пацієнта | char (символ) |
| patient | DOB | Дата народження пацієнта | date (дата) |
| recept | recept\_id | Унікальний ідентифікатор рецепту | integer (числовий) |
| recept | drug\_name | Назва ліків. Не допускає NULL | character varying (рядок) |
| recept | medical\_history\_id | Ідентифікатор медичної історії (зв'язок з medical\_history) | integer (числовий) |

**Функціональні залежності для кожної таблиці**

 **Таблиця doctor**:

* doctor\_id→"S.F.P",Specialization  
  (Ідентифікатор лікаря визначає його повне ім'я та спеціалізацію).

 **Таблиця medical\_history**:

* medical\_history\_id→diagnos\_name, description, doctor\_id, patient\_id  
  (Ідентифікатор медичної історії визначає назву діагнозу, опис, а також лікаря та пацієнта).
* doctor\_id→(атрибути таблиці doctor: "S.F.P", Specialization) через зовнішній ключ dtomh.
* patient\_id→(атрибути таблиці patient: "S.F.P", gender, DOB) через зовнішній ключ ptomh.

 **Таблиця patient**:

* patient\_id→"S.F.P",gender,DOB  
  (Ідентифікатор пацієнта визначає його повне ім'я, стать та дату народження).

 **Таблиця recept**:

* recept\_id→drug\_name,medical\_history\_id  
  (Це означає, що кожен унікальний рецепт ідентифікується через recept\_id та має конкретне значення для назви препарату та пов'язаний із певною медичною історією.).

medical\_history\_id → medical\_history.medical\_history\_id: Кожен запис у recept пов'язаний з певною медичною історією через зовнішній ключ.

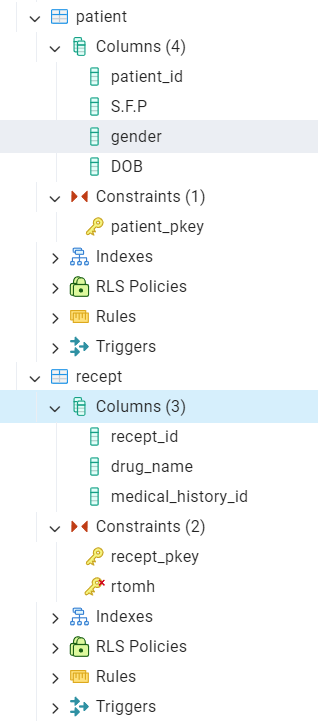
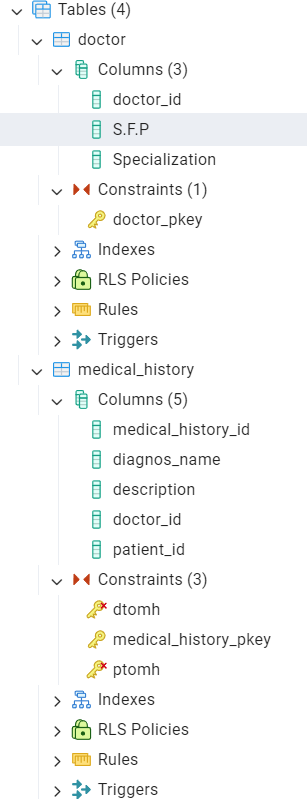
**Відповідність схеми бази даних нормальним формам:**

Схема відповідає 1НФ, тому що в таблиці немає дубльованих рядків, у кожній комірці зберігається атомарне значення, у кожному стовпчику зберігаються дані одного типу.

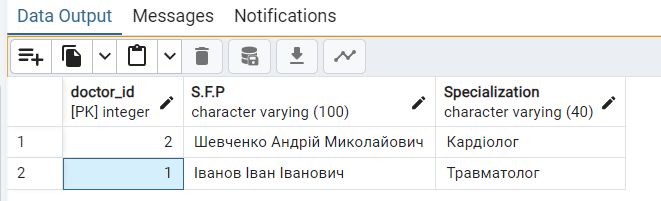
Схема відповідає 2НФ, тому що вона відповідає 1НФ, має первинний ключ від якого залежать всі не ключові стовпчики таблиці.

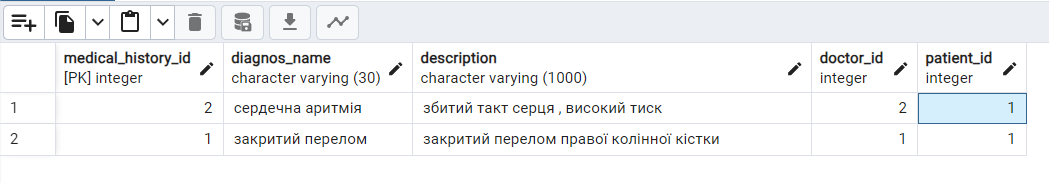
Схема відповідає 3НФ, тому що вона відповідає 2НФ і всі не ключові атрибути таблиці залежать від первинного ключа.

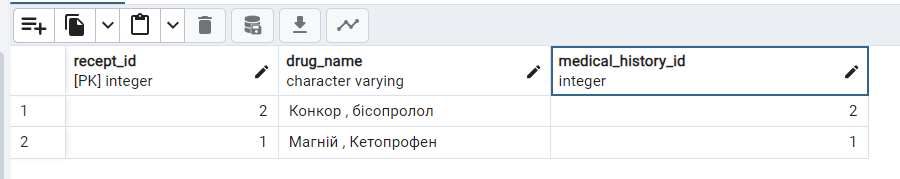
**Таблиці бази даних у pgAdmin4**

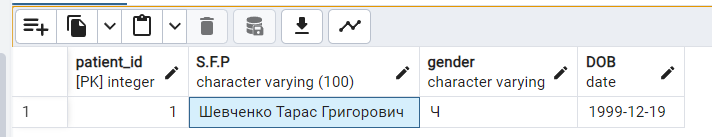


**Фотографії вмісту таблиць**







****

**SQL-текст опису БД**

**BEGIN;**

**CREATE TABLE IF NOT EXISTS public.doctor**

**(**

**doctor\_id integer NOT NULL,**

**"S.F.P" character varying(100) COLLATE pg\_catalog."default" NOT NULL,**

**"Specialization" character varying(40) COLLATE pg\_catalog."default" NOT NULL,**

**CONSTRAINT doctor\_pkey PRIMARY KEY (doctor\_id)**

**);**

**CREATE TABLE IF NOT EXISTS public.medical\_history**

**(**

**medical\_history\_id integer NOT NULL,**

**diagnos\_name character varying(30) COLLATE pg\_catalog."default" NOT NULL,**

**description character varying(1000) COLLATE pg\_catalog."default" NOT NULL,**

**doctor\_id integer NOT NULL,**

**patient\_id integer NOT NULL,**

**CONSTRAINT medical\_history\_pkey PRIMARY KEY (medical\_history\_id)**

**);**

**CREATE TABLE IF NOT EXISTS public.patient**

**(**

**patient\_id integer NOT NULL,**

**"S.F.P" character varying(100) COLLATE pg\_catalog."default" NOT NULL,**

**gender "char" NOT NULL,**

**"DOB" date NOT NULL,**

**CONSTRAINT patient\_pkey PRIMARY KEY (patient\_id)**

**);**

**CREATE TABLE IF NOT EXISTS public.recept**

**(**

**recept\_id integer NOT NULL,**

**drug\_name character varying COLLATE pg\_catalog."default" NOT NULL,**

**medical\_history\_id integer NOT NULL,**

**CONSTRAINT recept\_pkey PRIMARY KEY (recept\_id)**

**);**

**ALTER TABLE IF EXISTS public.medical\_history**

**ADD CONSTRAINT dtomh FOREIGN KEY (doctor\_id)**

**REFERENCES public.doctor (doctor\_id) MATCH SIMPLE**

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

**NOT VALID;**

**CREATE INDEX IF NOT EXISTS fki\_dtomh**

**ON public.medical\_history(doctor\_id);**

**ALTER TABLE IF EXISTS public.medical\_history**

**ADD CONSTRAINT ptomh FOREIGN KEY (patient\_id)**

**REFERENCES public.patient (patient\_id) MATCH SIMPLE**

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

**NOT VALID;**

**CREATE INDEX IF NOT EXISTS fki\_ptomh**

**ON public.medical\_history(patient\_id);**

**ALTER TABLE IF EXISTS public.recept**

**ADD CONSTRAINT rtomh FOREIGN KEY (medical\_history\_id)**

**REFERENCES public.medical\_history (medical\_history\_id) MATCH SIMPLE**

**ON UPDATE NO ACTION**

**ON DELETE NO ACTION**

**NOT VALID;**

**END;**