



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 III курсу  
групи KB-22  
Ковкін В. В.  
Перевірив:  
Павловский В. І.

Київ – 2024

*Метою роботи є здобуття вмінь проектування бази даних 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 (3 НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin4 та внести декілька рядків даних у кожен з таблиць засобами pgAdmin4.

### **Модель «сутність-зв'язок»**

Обрана предметна галузь передбачає систему управління процесом вакцинації населення, зберігання даних про пацієнтів, медичні установи, вакцини та лікарів.

#### **Сутності**

Згідно цієї області для побудови бази даних було виділено наступні сутності:

Громадянин з атрибутами: код громадянина, ім'я, адреса, номер телефону. Призначення: збереження інформації про громадян які мають отримати вакцинацію.

Вакцина з атрибутами: код вакцини, необхідна кількість доз. Призначення: збереження інформації про вакцини, що використовуються в процесі вакцинації.

Лікар з атрибутами: код лікаря, ім'я, номер телефону. Призначення: збереження інформації про медичних працівників, що здійснюють вакцинацію.

Медичний заклад з атрибутами: код закладу, адреса. Призначення: збереження інформації про медичні заклади, де проводиться вакцинація.

#### **Опис зв'язків**

Один медичний заклад може мати багато лікарів і один лікар може працювати в різних медичних закладах. Тому між сутностями Медичний заклад і Лікар існує зв'язок М:Н.

В одному медичному закладі може бути доступно багато вакцин і також вакцини можуть зберігатися у різних медичних закладах. Тому між сутностями Медичний заклад і Вакцина існує зв'язок М:Н.

Для вакцинації кожен раз використовується один громадянин, одна вакцина, один лікар і один медичний заклад. Для цього зв'язку додана сутність **Вакцинація**. **Вакцинація** це сутність-зв'язок 4-х сутностей **Громадянин – Вакцина – Лікар – Лікарня**.

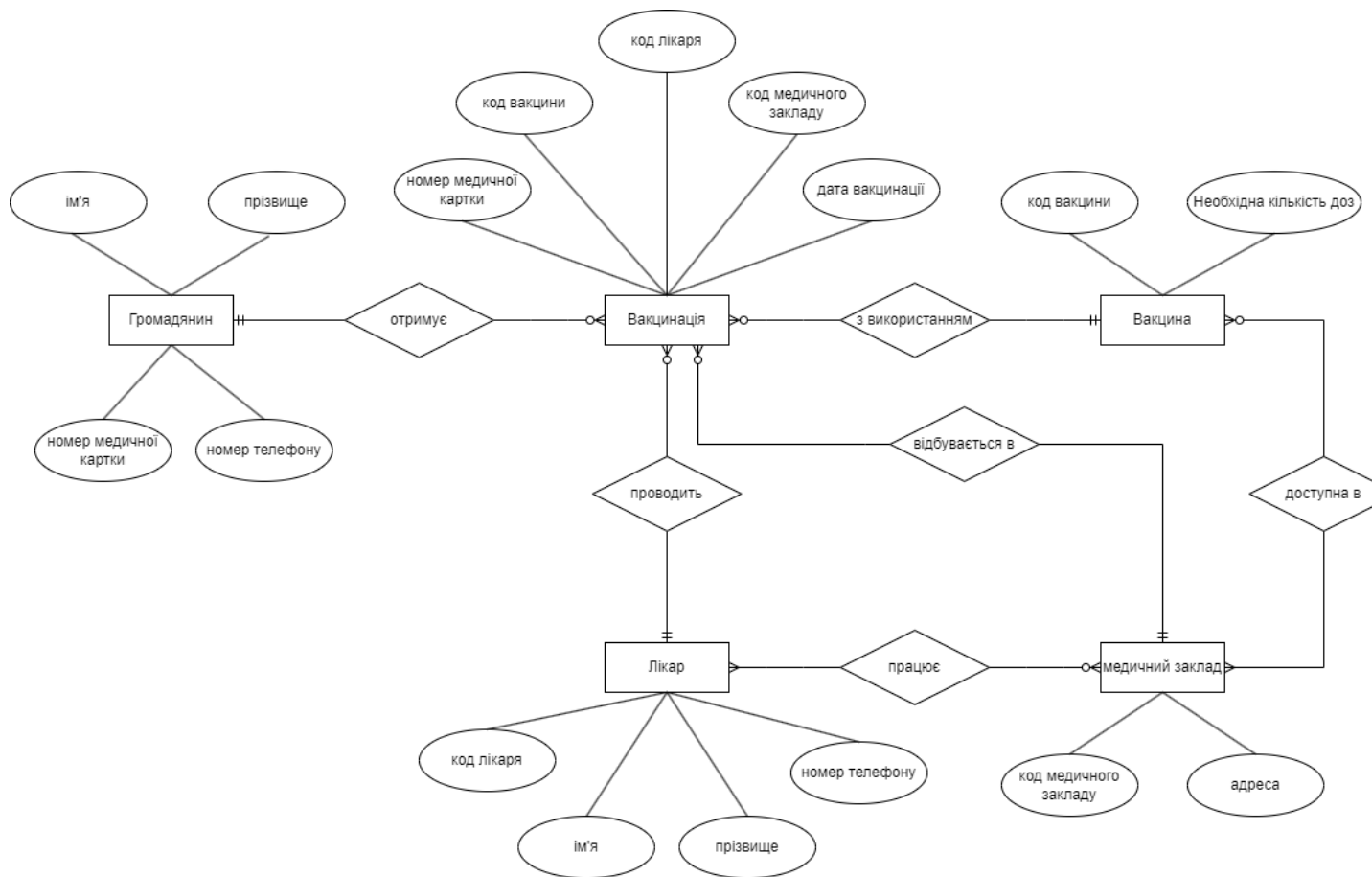


Рисунок 1 - ER-діаграма, побудована за нотацією Чена

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

В логічній моделі неможливий безпосередній зв'язок M:N, а в концептуальній моделі він існує між сутностями Лікар і Медичний заклад та Медичний заклад і Вакцина. Для їх представлення було введено допоміжні таблиці Лікар\_МедичнийЗаклад, Вакцина\_Медичний заклад.

Тернарний зв'язок буде перетворено на сутність Вакцинація, з якою кожна сутність у зв'язку, а саме Громадянин, Лікар і Вакцина будуть мати зв'язок 1:N.

Назви таблиць і полів було перекладено англійською, для зручності наведено таблицю нижче.

Таблиця 1 – Опис структури бази даних «Система управління вакцинацією населення»

Сутність	Атрибут	Тип атрибуту
<b>Citizen</b> – містить дані про громадян	<b>citizen_id</b> – унікальний ідентифікатор того, хто має отримати вакцинацію <b>name</b> – ім'я громадянина <b>phone</b> – номер телефону громадянина <b>address</b> – адреса громадянина	<b>integer</b> (числовий)  <b>character varying</b> (рядок)  <b>character varying</b> (рядок)  <b>character varying</b> (рядок)
<b>Vaccination</b> – зберігає дані про кожну здійснену вакцинацію.	<b>vaccination_id</b> – унікальний ідентифікатор вакцинації <b>citizen_id</b> – унікальний	<b>integer</b> (числовий)  <b>integer</b> (числовий)

	<p>ідентифікатор того, хто отримав вакцинацію  <b>doctor_id</b> – унікальний ідентифікатор лікаря, який проводив вакцинацію  <b>vaccine_id</b> – унікальний ідентифікатор вакцини, яка була використана  <b>clinic_id</b> – унікальний ідентифікатор медичного закладу  <b>date</b> – дата вакцинації</p>	<p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p> <p><b>date</b></p>
<b>Vaccine</b> – містить інформацію про вакцини	<p><b>vaccine_id</b> – унікальний ідентифікатор вакцини  <b>dosage</b> – необхідна кількість вакцинацій цією вакциною</p>	<p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p>
<b>Doctor</b> – містить інформацію про лікарів (тих, хто здійснює/надає вакцинацію)	<p><b>doctor_id</b> -- унікальний ідентифікатор лікаря  <b>name</b> – ім'я лікаря  <b>phone</b> – номер телефону лікаря</p>	<p><b>character varying</b> (рядок)</p> <p><b>character varying</b> (рядок)</p> <p><b>character varying</b> (рядок)</p> <p><b>integer</b> (числовий)</p>
<b>Clinic</b> – містить інформацію про медичні заклади в яких працюють лікарі	<p><b>clinic_id</b> – унікальний ідентифікатор медичного закладу  <b>address</b> – фізична адреса медичного закладу</p>	<p><b>integer</b> (числовий)</p> <p><b>character varying</b> (рядок)</p>
<b>Doctor_Clinic</b> – містить інформацію про відповідність лікаря і лікарні, де він працює	<p><b>id_tab</b> – унікальний ідентифікатор таблиці відповідності  <b>doctor_id</b> – унікальний ідентифікатор лікаря  <b>hospital_id</b> – унікальний ідентифікатор лікарні</p>	<p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p>
<b>Vaccine_Clinic</b> – містить інформацію про наявність вакцин у медичних закладах	<p><b>table_id</b> – унікальний ідентифікатор таблиці  <b>vaccine_id</b> – унікальний ідентифікатор вакцини  <b>clinic_id</b> – унікальний ідентифікатор медичного закладу</p>	<p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p> <p><b>integer</b> (числовий)</p>

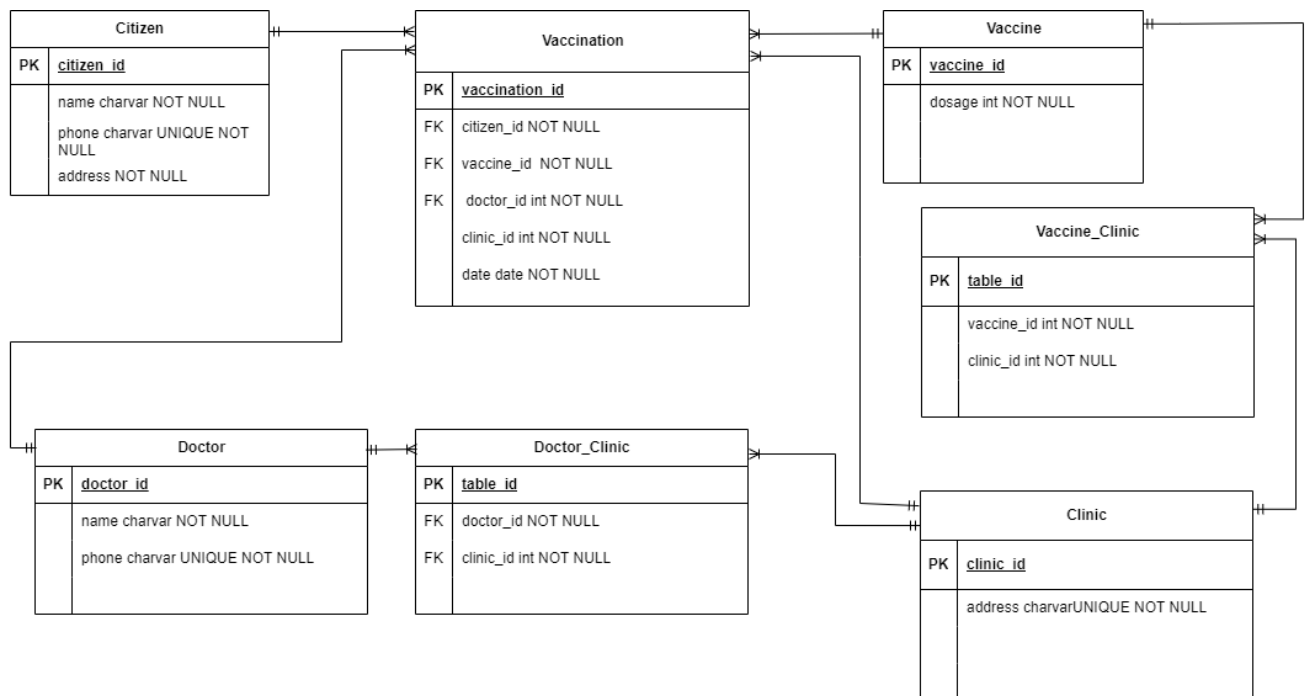


Рисунок 2 – Схема бази даних  
Функціональні залежності

### Citizen:

**citizen\_id** – унікальний ідентифікатор громадянина

**name** – ім'я громадянина

**phone** – номер телефону

**address** – адреса

**citizen\_id** → name, phone, address

citizen\_id → phone → name, address (транзитивне відношення, але включає в себе ключовий атрибут)

phone → citizen\_id, name, address

### Doctor:

**doctor\_id** – унікальний ідентифікатор лікаря

**name** – ім'я лікаря

**phone** – номер телефону лікаря

**doctor\_id** → name, phone

phone → doctor\_id, name

### Vaccine:

**vaccine\_id** – унікальний ідентифікатор вакцини

**dosage** – потрібна кількість доз

**vaccine\_id** → dosage

### Clinic:

**clinic\_id** – унікальний ідентифікатор лікарні

**address** – адреса медичного закладу

**clinic\_id** → address

### Vaccine\_Clinic:

**table\_id** – унікальний ідентифікатор таблиці

**vaccine\_id** – унікальний ідентифікатор вакцини FK

**clinic\_id** – унікальний ідентифікатор медичного закладу FK

**table\_id** → clinic\_id, vaccine\_id

clinic\_id, vaccine\_id → table\_id

### Doctor\_Clinic

**table\_id** – унікальний ідентифікатор таблиці

**doctor\_id** – унікальний ідентифікатор лікаря FK

clinic\_id – унікальний ідентифікатор медичного закладу FK

table\_id → doctor\_id, vaccine\_id

clinic\_id, vaccine\_id → table\_id

### **Vaccination:**

**vaccination\_id** → **citizen\_id, vaccine\_id, doctor\_id, clinic\_id, date**

citizen\_id, vaccine\_id, doctor\_id, clinic\_id, date → vaccination\_id

Схема відповідає 1НФ, тому що:

Кожен атрибут в кожній таблиці є атомарним. Тобто кожна клітинка містить єдине значення і кожен запис є унікальний.

Схема відповідає 2НФ, тому що:

Кожний неключовий атрибут функціонально залежить від цілого ключа. Кожна таблиця схеми має унікальний ідентифікатор від якого залежать всі її неключові атрибути.

Схема перебуває в 3НФ, тому що:

Схема перебуває в 2НФ

Жодна таблиця не містить неключових атрибутів, які б залежали від інших неключових атрибутів.

## **Таблиці бази даних у pgAdmin4**

### **Citizen**

The screenshot shows the pgAdmin4 interface for the 'citizen' table. The 'Columns' tab is selected, showing the following table structure:

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	citizen_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	address	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	phone	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

At the bottom of the interface, there are buttons for 'Close', 'Reset', and 'Save'.







doctor

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s) 

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	doctor_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	phone	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns
	doctor_pk	doctor_id

Close

Reset

Save

Clinic

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Inherited from table(s) 

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
<div></div>	clinic_id	integer			<div></div>	<div></div>	
<div></div>	address	character varying			<div></div>	<div></div>	

Close

Reset

Save

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Name	Columns
clinic_pk	clinic_id

Close

Reset

Save

Doctor\_Clinic

doctor\_clinic

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		table_id	integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		doctor_id	integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
		clinic_id	integer		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

doctor\_clinic

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Foreign Key

	Name	Columns	Referenced Table	
		clinic_id	(clinic_id) -> (clinic_id)	public.clinic
		doctor_id	(doctor_id) -> (doctor_id)	public.doctor

Close

Reset

Save

Vaccine\_Clinic



vaccine\_clinic

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

+

	Name	Columns	Referenced Table
	clinic_id	(clinic_id) -> (clinic_id)	public.clinic
	vaccine_id	(vaccine_id) -> (vaccine_id)	public.vaccine

Close

Reset

Save

## Vaccination

vaccination

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Inherited from table(s) 

Select to inherit from...

Columns

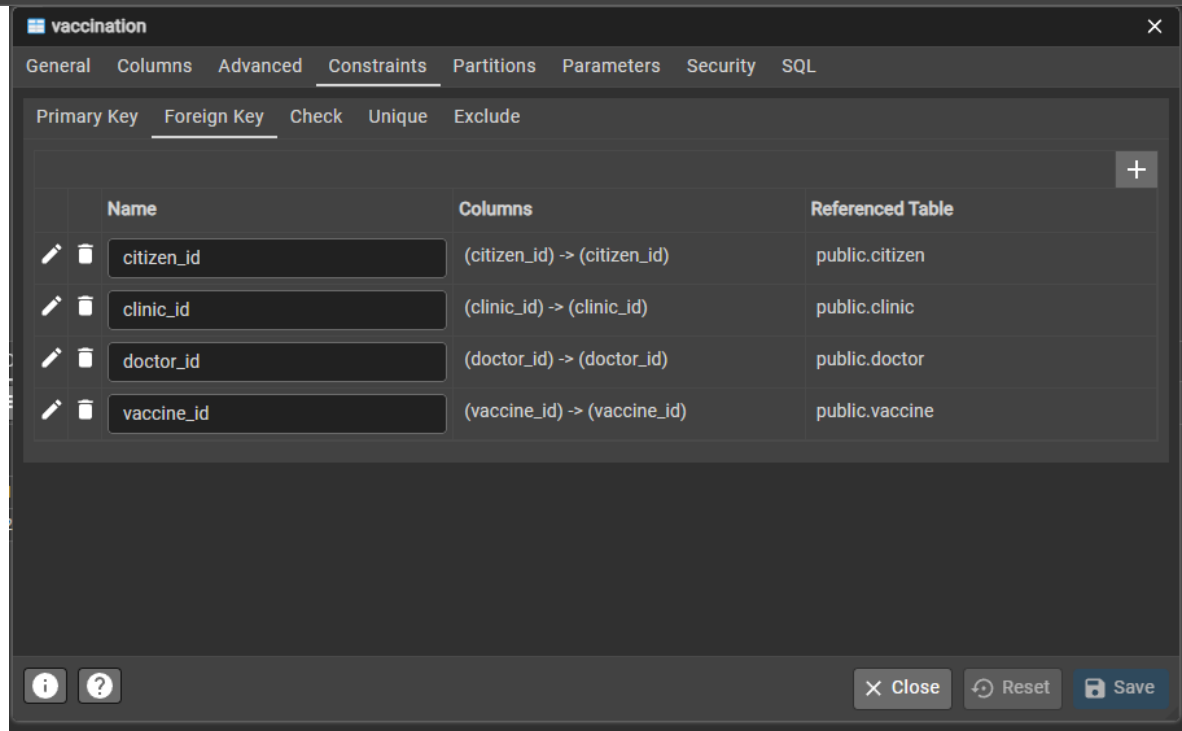
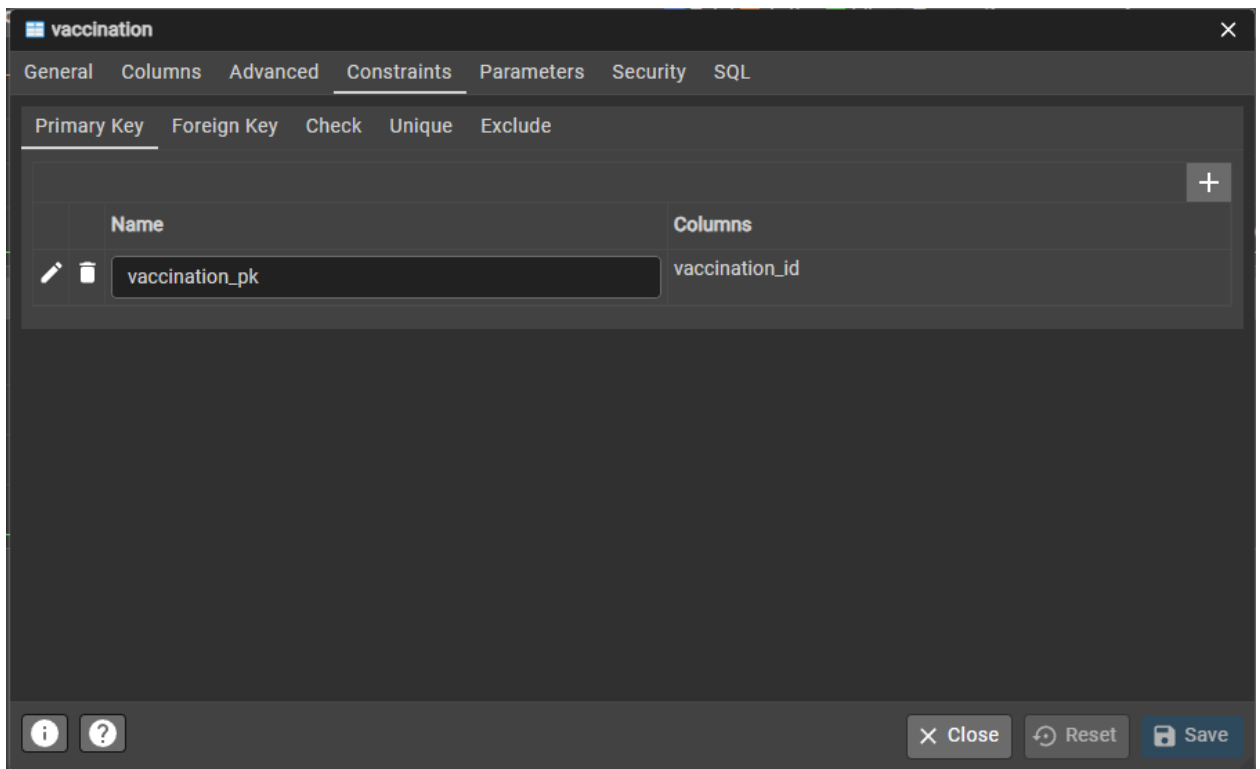
+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	vaccination_id	integer   v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	citizen_id	integer   v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	doctor_id	integer   v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	vaccine_id	integer   v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	clinic_id	integer   v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	date	date   v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close

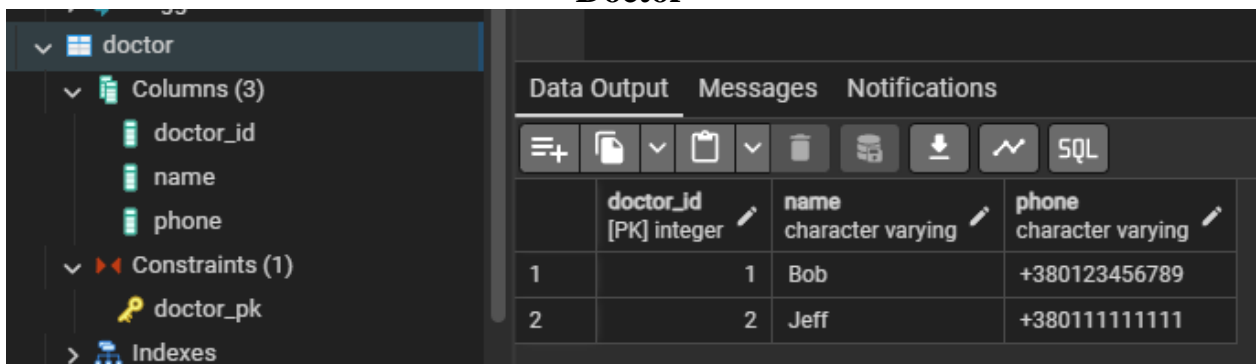
Reset

Save



## Таблиці заповнені даними

### Doctor



### Citizen

citizen

Columns (4)

citizen\_id

name

address

phone

Constraints (2)

citizen\_pk

unique\_phone

Data Output

Messages

Notifications

citizen\_id

[PK] integer

name

character varying

address

character varying

phone

character varying

1

1

Jeff

Kyiv, Random Street, 1

+380333333333

Clinic

clinic

Columns (2)

clinic\_id

address

Constraints (1)

clinic\_pk

Indexes

RLS Policies

Rules

Data Output

Messages

Notifications

clinic\_id

[PK] integer

address

character varying

1

1

Kyiv, Random Clinic str. 1

2

2

Kyiv, Random Clinic str. 2

Doctor\_Clinic

doctor\_clinic

Columns (3)

table\_id

doctor\_id

clinic\_id

Constraints (3)

clinic\_id

doctor\_clinic\_pk

doctor\_id

Indexes

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

📦

⬇️

📈

SQL

	table_id [PK] integer	doctor_id integer	clinic_id integer
1	1	1	1
2	2	1	2
3	3	2	1

Vaccination

vaccination

Columns (4)

vaccination\_id

citizen\_id

doctor\_id

vaccine\_id

Constraints (4)

citizen\_id

doctor\_id

vaccination\_pk

vaccine\_id

Data Output

Messages

Notifications

	vaccination_id [PK] integer	citizen_id integer	doctor_id integer	vaccine_id integer	clinic_id integer	date timestamp without time zone
1	1	1	1	1	1	2024-10-16
2	2	1	2	2	2	2024-10-16

Vaccine

	vaccine_id [PK] integer	dosage integer
1	1	5
2	2	3

### Vaccine\_clinic

	table_id [PK] integer	vaccine_id integer	clinic_id integer
1	1	1	1
2	2	1	2
3	3	2	2

### Висновки

У ході виконання лабораторної роботи було розроблено ER-модель предметної області, яка відповідає вимогам. Модель включає сутності з атрибутами і зв'язками між ними. Для побудови моделі використовувалась нотація Чена. Це дозволило правильно відобразити всі зв'язки між сутностями, зокрема зв'язки типу 1:N та M:N.

Після перетворення ER-моделі на реляційну схему бази даних, було проведено нормалізацію до третьої нормальної форми (3НФ). Схема відповідає 1НФ, оскільки всі атрибути атомарні. Вона також відповідає 2НФ, оскільки кожен неключовий атрибут залежить від повного первинного ключа. У 3НФ виключено транзитивні залежності, забезпечено правильну функціональну залежність атрибутів.

За допомогою PostgreSQL та інструмента pgAdmin4 були створені таблиці бази даних, визначені типи даних атрибутів, встановлені первинні та зовнішні ключі, а також обмеження на стовпці (NOT NULL, UNIQUE). Це гарантувало цілісність даних та відповідність заданим обмеженням.

У результаті роботи було заповнено таблиці бази даних реальними даними, що дозволило перевірити коректність створеної структури та виконання базових операцій додавання і збереження даних у системі.

**Посилання на репозиторій Github:** [https://github.com/kovkinvladyslav/bd\\_labs.git](https://github.com/kovkinvladyslav/bd_labs.git)