



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 громадянин може отримати декілька вакцинацій різними вакцинами. Тому між сутностями Громадянин і Вакцина існує зв'язок 1:N

Для вакцинації кожен раз використовується один громадянин, одна вакцина і один лікар.

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 M:N.

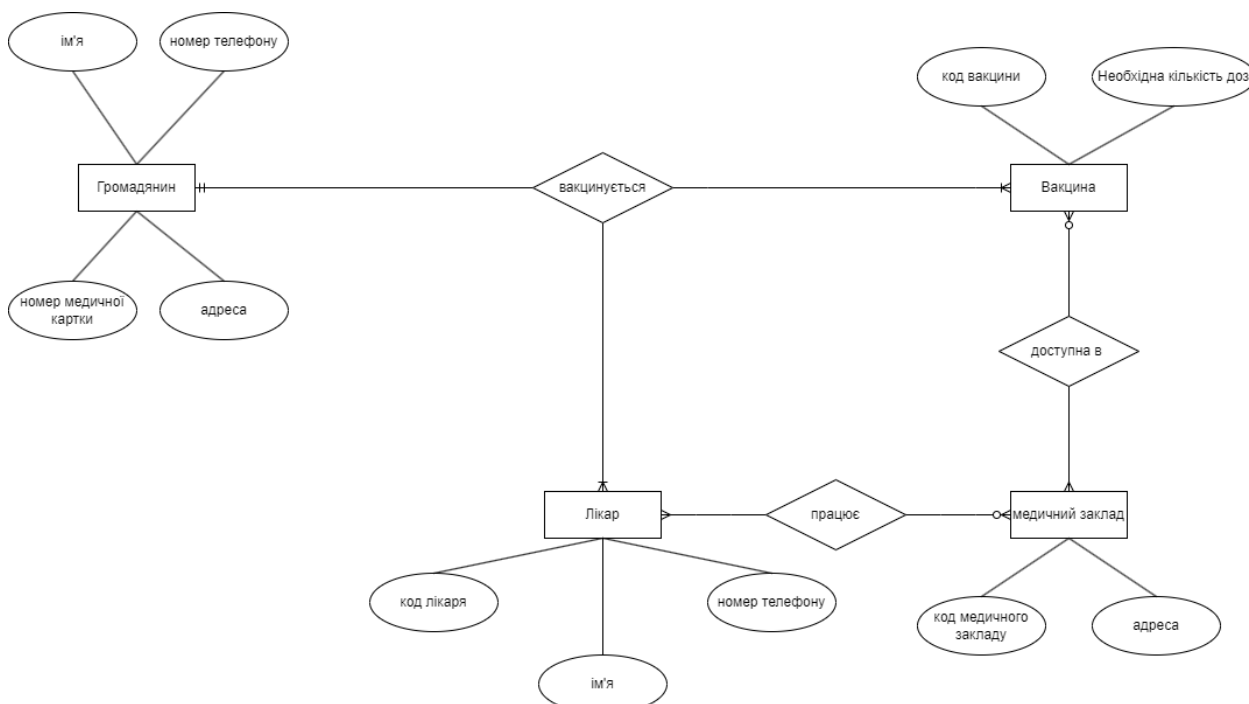


Рисунок 1 - ER-діаграма, побудована за нотацією Чена

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
Citizen – містить дані про громадян	citizen_id – унікальний ідентифікатор того, хто має отримати вакцинацію name – ім'я громадянина phone – номер телефону громадянина address – адреса громадянина	integer (числовий) character varying (рядок) character varying (рядок) character varying (рядок)
Vaccination – зберігає дані про кожну здійснену вакцинацію.	vaccination_id – унікальний ідентифікатор вакцинації citizen_id – унікальний ідентифікатор того, хто отримав вакцинацію doctor_id – унікальний ідентифікатор лікаря, який проводив вакцинацію vaccine_id – унікальний ідентифікатор вакцини, яка була використана	integer (числовий) integer (числовий) integer (числовий) integer (числовий)
Vaccine – містить інформацію про вакцини	vaccine_id – унікальний ідентифікатор вакцини dosage – необхідна кількість вакцинацій цією вакциною	integer (числовий) integer (числовий)
Doctor – містить інформацію про лікарів (тих, хто здійснює/надає вакцинацію)	doctor_id -- унікальний ідентифікатор лікаря name – ім'я лікаря phone – номер телефону лікаря	character varying (рядок) character varying (рядок) character varying (рядок) integer (числовий)
Clinic – містить інформацію про	clinic_id – унікальний ідентифікатор	integer (числовий)

медичні заклади в яких працюють лікарі	медичного закладу address – фізична адреса медичного закладу	character varying (рядок)
Doctor_Clinic – містить інформацію про відповідність лікаря і лікарні, де він працює	id_tab – унікальний ідентифікатор таблиці відповідності doctor_id – унікальний ідентифікатор лікаря hospital_id – унікальний ідентифікатор лікарні	integer (числовий) integer (числовий) integer (числовий)
Vaccine_Clinic – містить інформацію про наявність вакцин у медичних закладах	table_id – унікальний ідентифікатор таблиці vaccine_id – унікальний ідентифікатор вакцини clinic_id – унікальний ідентифікатор медичного закладу	integer (числовий) integer (числовий) integer (числовий)

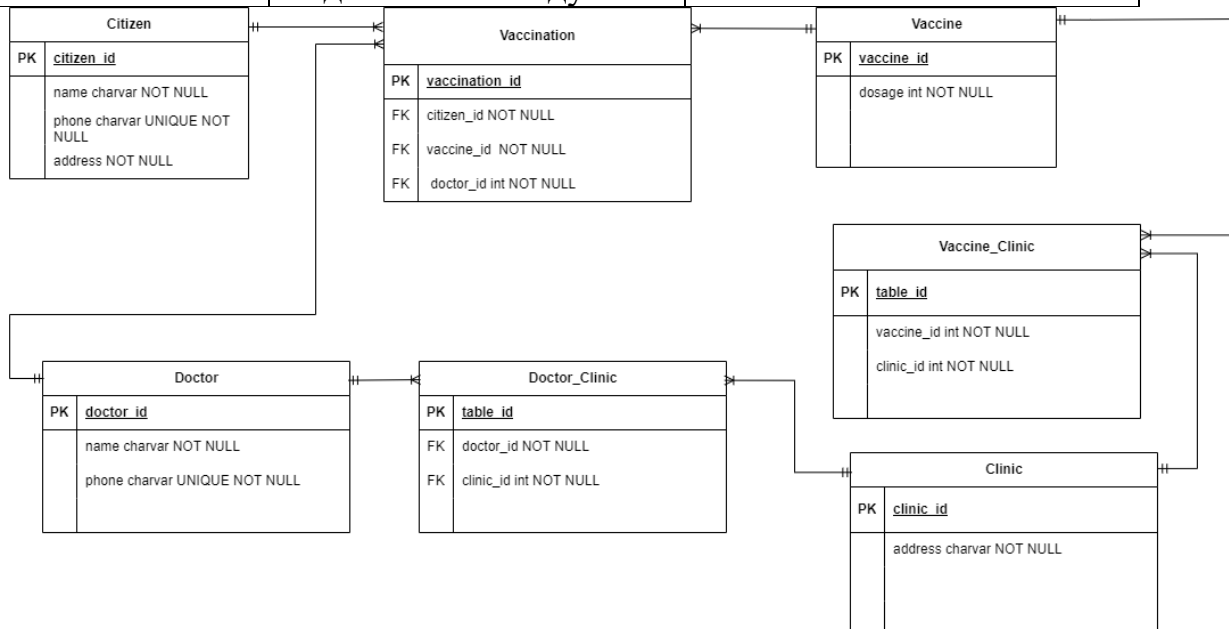


Рисунок 2 – Схема бази даних
Функціональні залежності

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**

citizen_id, vaccine_id, doctor_id → 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

citizen

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	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"/>	

Close

Reset

Save

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

	Name	Columns
	citizen_pk	citizen_id

Close

Reset

Save

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Primary Key Foreign Key Check Unique Exclude

Name	Columns
vaccine_pk	vaccine_id

Close Reset Save

Doctor

doctor

General

Columns

Advanced

Constraints

Parameters







Security



SQL

Inherited from table(s)

Select to inherit from...

Columns

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

Close

Reset

Save

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Name	Columns
<div><div></div><div><div>doctor_pk</div></div></div>	doctor_id

i?

X CloseR ResetS Save

Clinic

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...|v

Columns+

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
⋮	✎	clinic_id	integer v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
⋮	✎	address	character varying v			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Create - Table

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Primary Key



Foreign Key



Check

Unique

Exclude

+

	Name	Columns
 	clinic_pk	clinic_id

Close

Reset

Save

Doctor_Clinic

doctor_clinic

General

Columns

Advanced

Constraints

Parameters

Security

SQL







Inherited from table(s)



Select to inherit from...

| v

Columns

+

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

Close

Reset

Save

doctor_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
 	doctor_id	(doctor_id) -> (doctor_id)	public.doctor

Close

Reset

Save

Vaccine_Clinic

vaccine_clinic

General

Columns

Advanced

Constraints

Parameters

Security
















SQL



Inherited from table(s)

Select to inherit from...

Columns

+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	table_id	integer 					
 	vaccine_id	integer 					
 	clinic_id	integer 					

Close

Reset

Save

Vaccination

vaccination

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	<div>vaccination_id</div>	<div>integer</div>			<div></div>	<div></div>	
	<div>citizen_id</div>	<div>integer</div>			<div></div>	<div></div>	
	<div>doctor_id</div>	<div>integer</div>			<div></div>	<div></div>	
	<div>vaccine_id</div>	<div>integer</div>			<div></div>	<div></div>	
	<div>clinic_id</div>	<div>integer</div>			<div></div>	<div></div>	

Close

Reset

Save

vaccination

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

Name

vaccination_pk

Columns

vaccination_id

Close

Reset

Save

vaccination			
General Columns Advanced Constraints Parameters Security SQL			
Primary Key Foreign Key Check Unique Exclude			
	Name	Columns	Referenced Table
	citizen_id	(citizen_id) -> (citizen_id)	public.citizen
	clinic_id	(clinic_id) -> (clinic_id)	public.clinic
	doctor_id	(doctor_id) -> (doctor_id)	public.doctor
	vaccine_id	(vaccine_id) -> (vaccine_id)	public.vaccine

Таблиці заповнені даними

Doctor

doctor			
Columns (3)			
	doctor_id		
	name		
	phone		
Constraints (1)			
	doctor_pk		
Indexes			
Data Output Messages Notifications			
	doctor_id [PK] integer	name character varying	phone character varying
1	1	Bob	+380123456789
2	2	Jeff	+380111111111

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
1	1	Jeff	Kyiv, Random Street, 1
			phone character varying
			+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
table_id [PK] integer	doctor_id integer	clinic_id integer
1	1	1
2	2	1
3	3	2

Vaccination

vaccination

Columns (5)

vaccination_id

citizen_id

doctor_id

vaccine_id

clinic_id

Constraints (5)

citizen_id

clinic_id

doctor_id

vaccination_pk

vaccine_id

Data Output

Messages

Notifications

SQL

	vaccination_id [PK] integer	citizen_id integer	doctor_id integer	vaccine_id integer	clinic_id integer
1	2	1	2	2	1
2	1	1	1	1	1

Vaccine

vaccine			
Columns (2)			
vaccine_id			
dosage			
Constraints (1)			
vaccine_pk			
Indexes			

Data Output	Messages	Notifications
vaccine_id [PK] integer	dosage integer	
1	1	5
2	2	3

Vaccine_clinic

vaccine_clinic			
Columns (3)			
table_id			
vaccine_id			
clinic_id			
Constraints (3)			
clinic_id			
vaccine_clinic_pk			
vaccine_id			

Data Output	Messages	Notifications
table_id [PK] integer	vaccine_id integer	clinic_id integer
1	1	1
2	2	2
3	3	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