



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:  
студентка III курсу  
групи KB-94  
Романенко М. В.  
Перевірив:  
Петрашенко А. В.

Київ – 2021

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.

### Завдання 1 – модель «сутність-зв'язок» галузі постів якоїсь соціальної мережі

#### Entities

Post

User

Comments

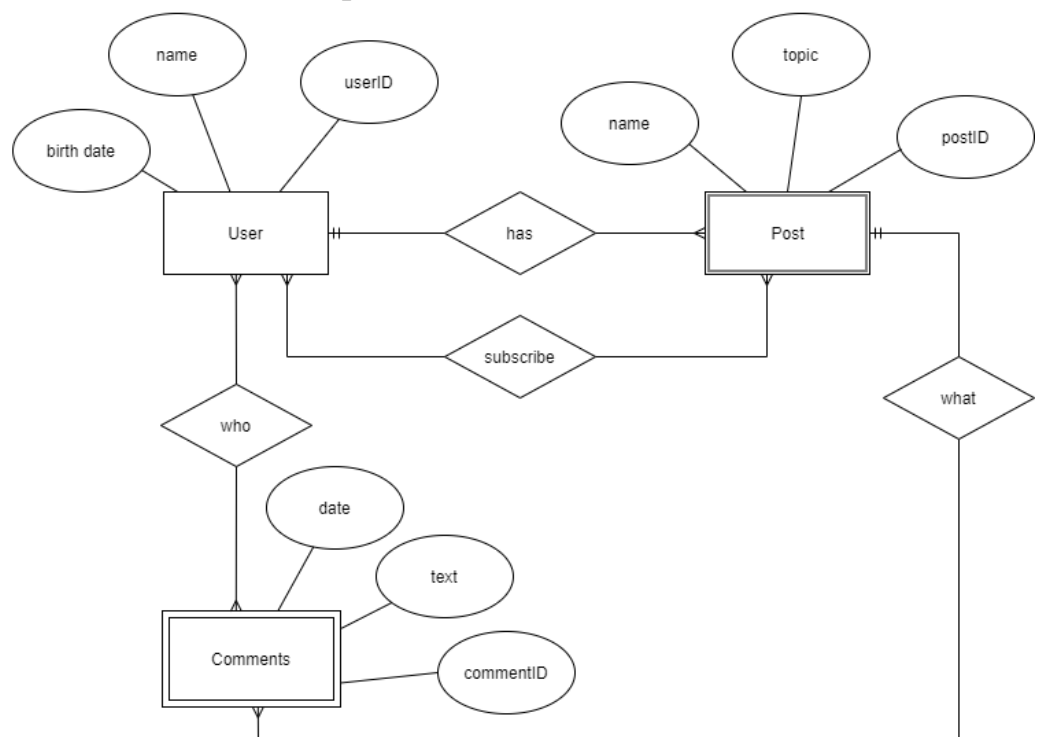


Рисунок 1. ER-діаграма, побудована за нотацією "Пташиної лапки"

#### Сутності, їх призначення та опис зв'язків:

Маємо три сутності: User, Post, Comments. Сутність User описує користувачів соціальної мережі. Кожен користувач має ім'я, дату народження та свій ID. Сутність Post – це пости цих самих користувачів. Тобто кожен окремий користувач може мати скільки завгодно постів в своєму блозі, але скільки завгодно користувачів можуть бути підписаними на ці пости. Кожний пост також має свій ID, назву поста та тему, на яку написано пост. Третя сутність – це Comments. Це коментарі під постами. Коментар має два зв'язки: чий коментар (користувача) та що це за коментар (тобто під яким саме постом).

Багато користувачів можуть написати скільки завгодно коментарів, а також під одним постом може бути багато коментарів. Коментар має свій ID, свою дату та сам текст.

**Завдання 2** – перетворення графічного вигляду бази даних у схему бази даних PostgreSQL

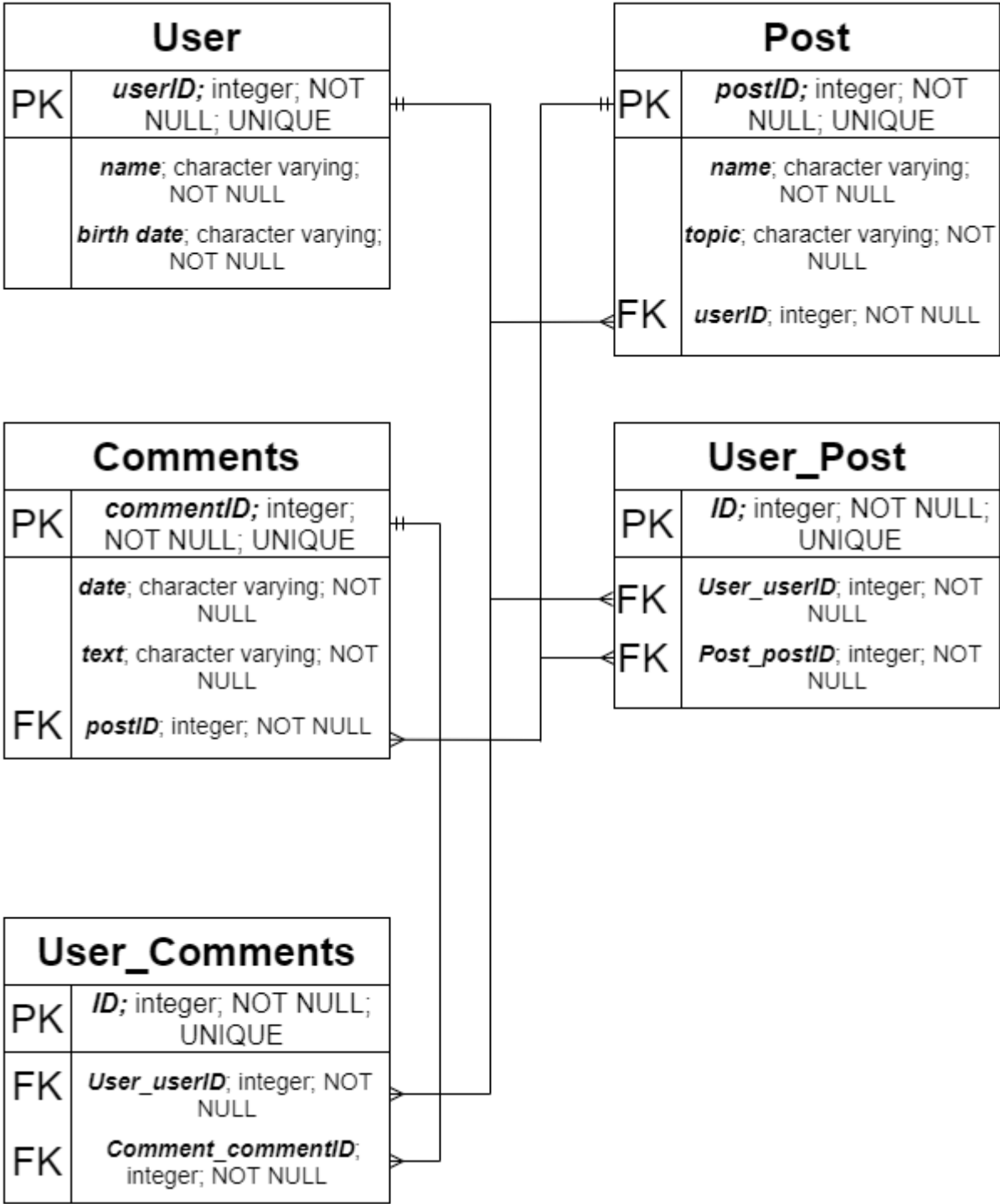


Рисунок 2. Схема бази даних у графічному вигляді

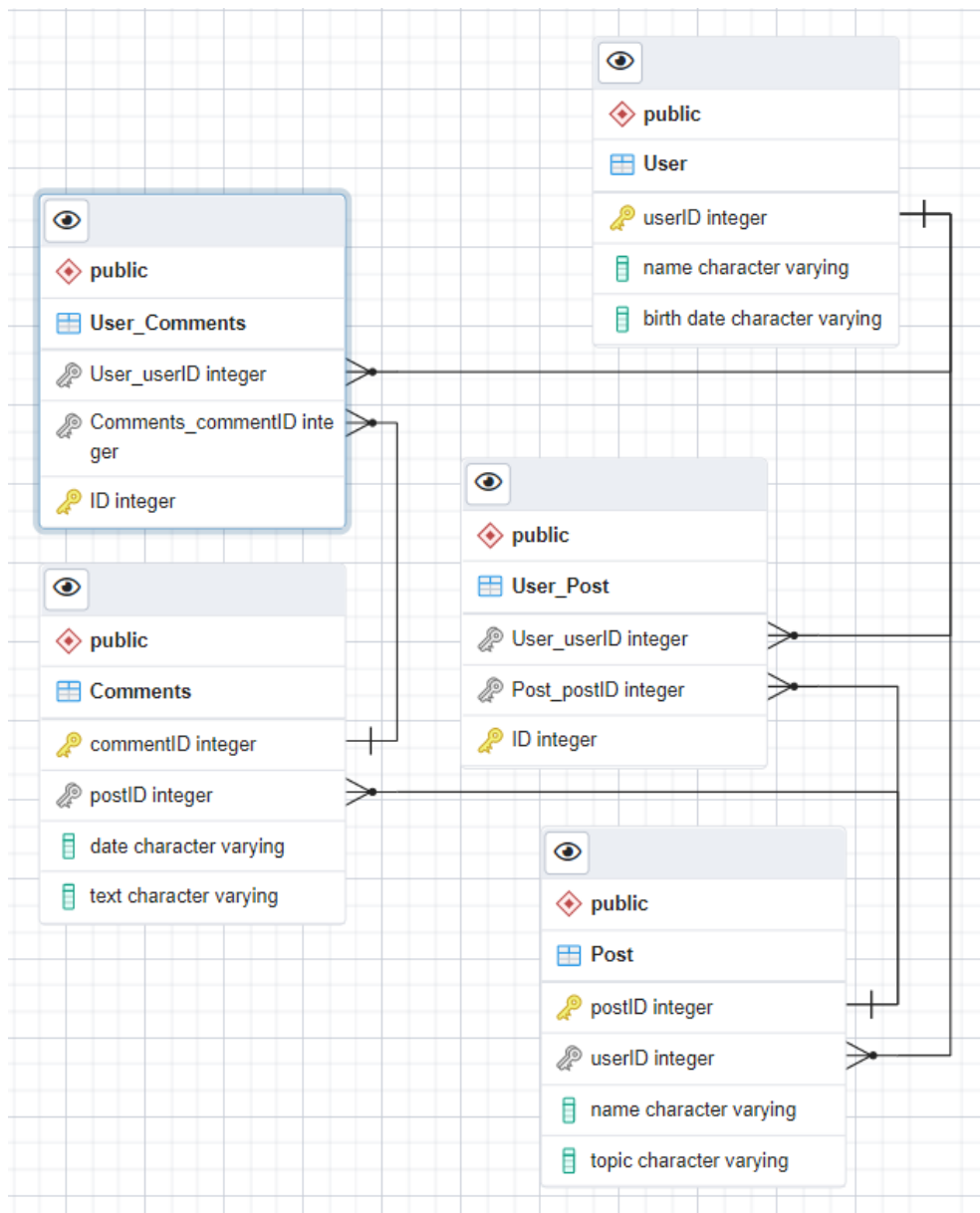


Рисунок 3. Схема бази даних у pgAdmin 4

### Опис процесу перетворення

Сутність User було перетворено в таблицю User. Сутність Post було перетворено в таблицю Post, зв'язок has (1:N) із сутністю User зумовив появу у ній зовнішнього ключа userID. Сутність Comments було перетворено в таблицю Comments, зв'язок what (1:N) із сутністю Post зумовив появу у ній зовнішнього ключа postID. Зв'язок subscribe (N:M) зумовив появу додаткової таблиці User\_Post. Зв'язок who (N:M) зумовив появу додаткової таблиці User\_Comments.

**Завдання 3** - нормалізацію схеми бази даних до третьої нормальної форми

**Функціональні залежності:**

User (userID, name, birth date):

userID → name

userID → birth date

userID → name, birth date

Post (*postID*, userID, name, topic):

postID → userID

postID → name

postID → topic

postID → userID, name, topic

Comments (*commentID*, postID, date, text):

commentID → postID

commentID → date

commentID → text

commentID → postID, date, text

User\_Post (User\_userID, Post\_postID, *ID*):

ID → Post\_postID

ID → User\_userID

ID → Post\_postID, User\_userID

User\_Comments (User\_userID, Comment\_commentID, *ID*):

ID → Comment\_commentID

ID → User\_userID

ID → Comment\_commentID, User\_userID

Схема бази даних відповідає нормальній формі НФ1, тому що всі атрибути таблиці є атомарними, кожна таблиця має primary key та кожна таблиця має мінімальний набір атрибутів.

Схема бази даних відповідає нормальній формі НФ2, тому що вона відповідає нормальній формі НФ1 і кожен неключовий атрибут функціонально залежить від цілого ключа, а не від його частини.

Схема бази даних відповідає нормальній формі НФ3, тому що вона відповідає нормальній формі НФ2 і дані в таблиці залежать від primary key.

#### Завдання 4 – таблиці бази даних у pgAdmin 4

User:



The screenshot shows the pgAdmin 4 interface. On the left, a tree view displays the database structure with 'Post' expanded, showing 'Post\_User' and 'User'. The 'User' table is selected. The main pane shows the table's structure and data. The table has three columns: 'userID' (integer, primary key), 'name' (character varying), and 'birth date' (character varying). The data rows are:

userID	name	birth date
1	Maria	14th of April 2002
2	Gleb	26th of July 2002
3	Sofia	3th of October 2001

User

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Наследуется из: Выберите источник наследия

Столбцы

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	userID	integer			Да	Да
	name	character varying			Да	Нет
	birth date	character varying			Да	Нет

User

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Первичный ключ Внешний ключ Проверка Уникальность Исключить

	Имя	Столбцы
	User_pkey	userID

## Post:

- Post
  - RLS Policies
  - Индексы
  - Ограничения
  - Правила
  - Столбцы
  - Триггеры
- Post\_User
- User
  - RLS Policies
  - Индексы
  - Ограничения
  - Правила
  - Столбцы
  - Триггеры
- User\_Comments

Результат План выполнения Сообщения Notifications

	postID [PK] integer	name character varying	topic character varying	userID integer
1	1	family	about the relationships with family	1
2	2	work	about some projects and other stuff at work	2
3	3	friends	about the closest people after family	3

**Post**

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Наследуется из таблиц(ы)

**Столбцы**

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	postID	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>
	topic	character varying			<input checked="" type="checkbox"/>	<input type="checkbox"/>
	userID	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Post**

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Первичный ключ Внешний ключ Проверка Уникальность Исключить

Имя	Столбцы
Post_pkey	postID

**Post**

Общие Столбцы Дополнительно Ограничения Параметры Безопасность SQL

Первичный ключ Внешний ключ Проверка Уникальность Исключить

Имя	Столбцы	Referenced Table
userID	(userID) -> (userID)	public.User

## Comments:

- > Сторонние таблицы
- > Таблицы (5)
  - > Comments
  - > Post
    - > RLS Policies
    - > Индексы

	commentID [PK] integer	date character varying	text character varying	postID integer
1	1	4th of September 2019	so amasing!	1
2	2	15th of November 2020	awful	2
3	3	26th of April 2021	I like it	3

Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Наследуется из  
таблиц(ы)

Выберите источник насл

Столбцы

+

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ
	commentID	integer			<div>Да</div>	<div>Да</div>
	date	character varying			<div>Да</div>	<div>Нет</div>
	text	character varying			<div>Да</div>	<div>Нет</div>
	postID	integer			<div>Да</div>	<div>Нет</div>

Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы
	Comments_pkey	commentID

Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы	Referenced Table
	postID	(postID) -> (postID)	public.Post

User\_Post:

> Правила

> Столбцы

> Триггеры

> User\_Comments

> User\_Post

	ID [PK] integer	Post_postID integer	User_userID integer
1	1	4	7
2	2	5	8
3	3	6	9



User\_Post

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Наследуется из таблиц(ы)

Выберите источник насл

Столбцы

+

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	ID	integer			Да	Да
	Post_postID	integer			Да	Нет
	User_userID	integer			Да	Нет

User\_Post

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы
	User_Post_pkey	ID

User\_Post

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы	Referenced Table
	Post_postID	(Post_postID) -> (postID)	public.Post
	User_userID	(User_userID) -> (userID)	public.User

User\_Comments:

Ограничения

Правила

Столбцы

Триггеры

User\_Comments

Типы

	ID [PK] integer	User_userID integer	Comment_commentID integer
1	1	3	4
2	2	2	5
3	3	1	6

User\_Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Наследуется из таблиц(ы)

Выберите источник наследия

Столбцы

+

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ
	ID	integer			<div>Да</div>	<div>Да</div>
	User_userID	integer			<div>Да</div>	<div>Нет</div>
	Comment_commentID	integer			<div>Да</div>	<div>Нет</div>

User\_Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы
	User_Comments_pkey	ID

User\_Comments

Общие

Столбцы

Дополнительно

Ограничения

Параметры

Безопасность

SQL

Первичный ключ

Внешний ключ

Проверка

Уникальность

Исключить

+

	Имя	Столбцы	Referenced Table
	Comment_commentID	(Comment_commentID) -> (comment...	public.Comments
	User_userID	(User_userID) -> (userID)	public.User