

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАФЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
(назва дисципліни)

на тему: База даних для реалізації соціальної мережі

Студента (ки) 2 курсу ІП-13 групи
спеціальності 121 «Інженерія програмного
забезпечення»

Романюк Діани Олексіївни
(прізвище та ініціали)

Керівник Ліщук О. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії

(підпис) _____
(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) _____
(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) _____
(вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2023 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-13 Семестр 3

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Романюк Діані Олексіївни _____
(прізвище, ім'я, по батькові)

1. Тема роботи База даних для реалізації соціальної мережі

керівник роботи Ліщук О. В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 15.01.2022

3. Вихідні дані до роботи _____
завдання на розробку бази даних для внесення та зберігання
інформації усіх необхідних аспектів у соціальній мережі для подальшого
використання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

- 1) Аналіз предметного середовища
 - 2) Побудова ER-моделі
 - 3) Побудова реляційної схеми з ER-моделі
 - 4) Створення бази даних, у форматі обраної системи управління базою даних
 - 5) Створення користувачів бази даних
 - 6) Імпорт даних з використанням засобів СУБД в створену базу даних
 - 7) Створення мовою SQL запитів
 - 8) Оптимізація роботи запитів
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Дата видачі завдання 30.10.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметного середовища	25.11.2022	
2	Побудова ER-моделі	30.11.2022	
3	Побудова реляційної схеми з ER-моделі	05.12.2022	
4	Створення бази даних, у форматі обраної системи управління базою даних	15.12.2022	
5	Створення користувачів бази даних	16.12.2022	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	20.12.2022	
7	Створення мовою SQL запитів	25.12.2022	
8	Оптимізація роботи запитів	27.12.2022	
9	Оформлення пояснювальної записки	05.01.2023	
10	Захист курсової роботи	16.01.2023	

Студент

_____ Романюк Д. О.
(підпис) (прізвище та ініціали)

Керівник роботи

_____ Ліщук О. В.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП.....	5
1. ОПИС ПРЕДМЕТНОГО СЕРИДОВИЩА.....	6
2. ПОСТАНОВКА ЗАДАЧІ.....	7
3. ПРОЕКТУВАННЯ БАЗИ ДАНИХ	8
4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ	10
5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ	11
6. РОБОТА З БАЗОЮ ДАНИХ.....	16
ВИСНОВОК	31

ВСТУП

Поняття соціальної мережі у сучасному інформаційному просторі є широко поширеним і насамперед сприймається будь-якою людиною як «велике сховище користувацьких даних». Якщо розглядати цю структуру зсередини то можна зауважити, що так і є. Ми не можемо уявити своє життя без онлайн-спілкування. Роки тому ми б чекали тижнями, щоб отримати лист у поштовій скриньці. Однак сьогодні навіть не починаємо ранок, поки не перевіримо стрічки новин у Facebook та Instagram та не прочитаємо повідомлення на Messenger, WhatsApp, Snapchat та LinkedIn. Найпопулярніші соціальні мережі світу налічують мільйони користувачів. Початкова ідея соціальної мережі це зберігання та обмін певними даними. Тож можемо зробити висновок – будь-яка соціальна мережа не може існувати без побудованої під неї бази даних. Вищезгадана ідея і є представленням даної роботи.

1. ОПИС ПРЕДМЕТНОГО СЕРИДОВИЩА

На даній схемі представлений опис предметного середовища та його необхідні складові. Звичний користувачькому оку вигляд соціальної мережі представлений схемою та пояснення до кожного елементу. Збільшену схему можна розглянути в додатку А рис. 1.1.

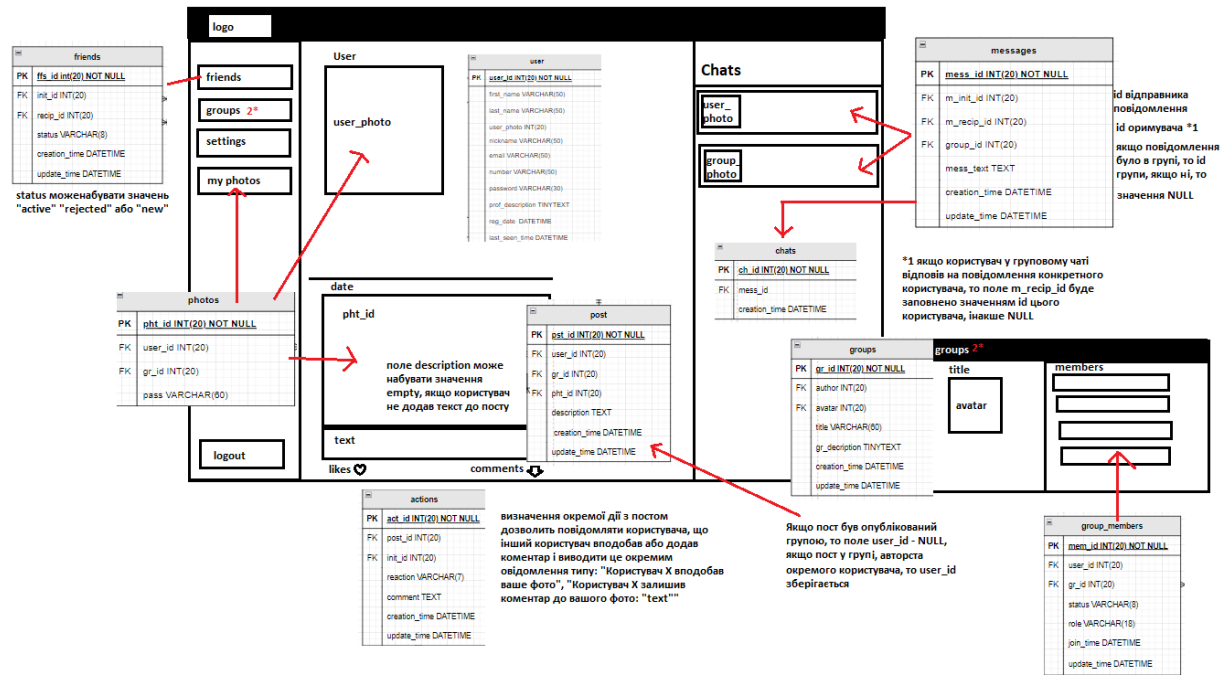


Рис 1.1 Схема сторінки соціальної мережі

2. ПОСТАНОВКА ЗАДАЧІ

Мета даної роботи – створення бази даних для реалізації соціальної мережі. Необхідно побудувати чітку структуру доцільного збереження даних для їх зручного зв'язку між собою для подальшого використання.

Основні задачі модератора та адміністратора соціальної мережі полягають у контролі контенту, що надається користувачем або групою, вирішення технічних проблем та ін. Їм необхідно мати доступ до повної інформації кожного користувача, групи, альбому, посту та виконаних користувачами або групами дій. Єдине структура (для дотримання політики конфіденційності) до якої немає доступу адміністратор це приватні повідомлення та чати.

Користувачам доступні сторінки, альбоми та пости інших користувачів за умови, що другі самостійно не обмежили доступ перших в межах своєї сторінки. Крім того користувачі, що мають роль адміністратора певної групи мають також доступ до редагування та перегляду вмісту груп та групових чатів, постів тощо. Також вони мають змогу обмінюватись повідомленнями, реакціями та коментарями і об'єднуватись в окремі від чатів групи.

Система фіксує дату та час реєстрації користувача в соціальній мережі, дату останнього відвідування, викладення посту, створення та оновлення повідомлення, виконання певної дії над постом («вподобання» та коментарі).

Для більш доцільного зберігання зображень у базі даних зберігаються лише шляхи до них, а самі фотографії лежать на умовному сервері для зменшення перенавантаження бази.

3. ПРОЕКТУВАННЯ БАЗИ ДАНИХ

Представлені сутності:

- Користувачі (users)
- Групи (groups)
- Фото (photos)
- Повідомлення (messages)
- Друзі (friends)
- Учасники груп (group_members)
- Пости (post)
- Чати (chats)
- Дії (actions)

Обмеження:

Усі id за якими визначається користувач, група, фото, повідомлення, друг, учасник групи, пост, чат та дія не можуть повторюватись у своїй категорії та набувати нульового значення. Імена та назви груп визначенні буквенно. Номери телефону, назви файлів аватарів користувачів та номери телефонів визначенні чисельно.

ER-модель:

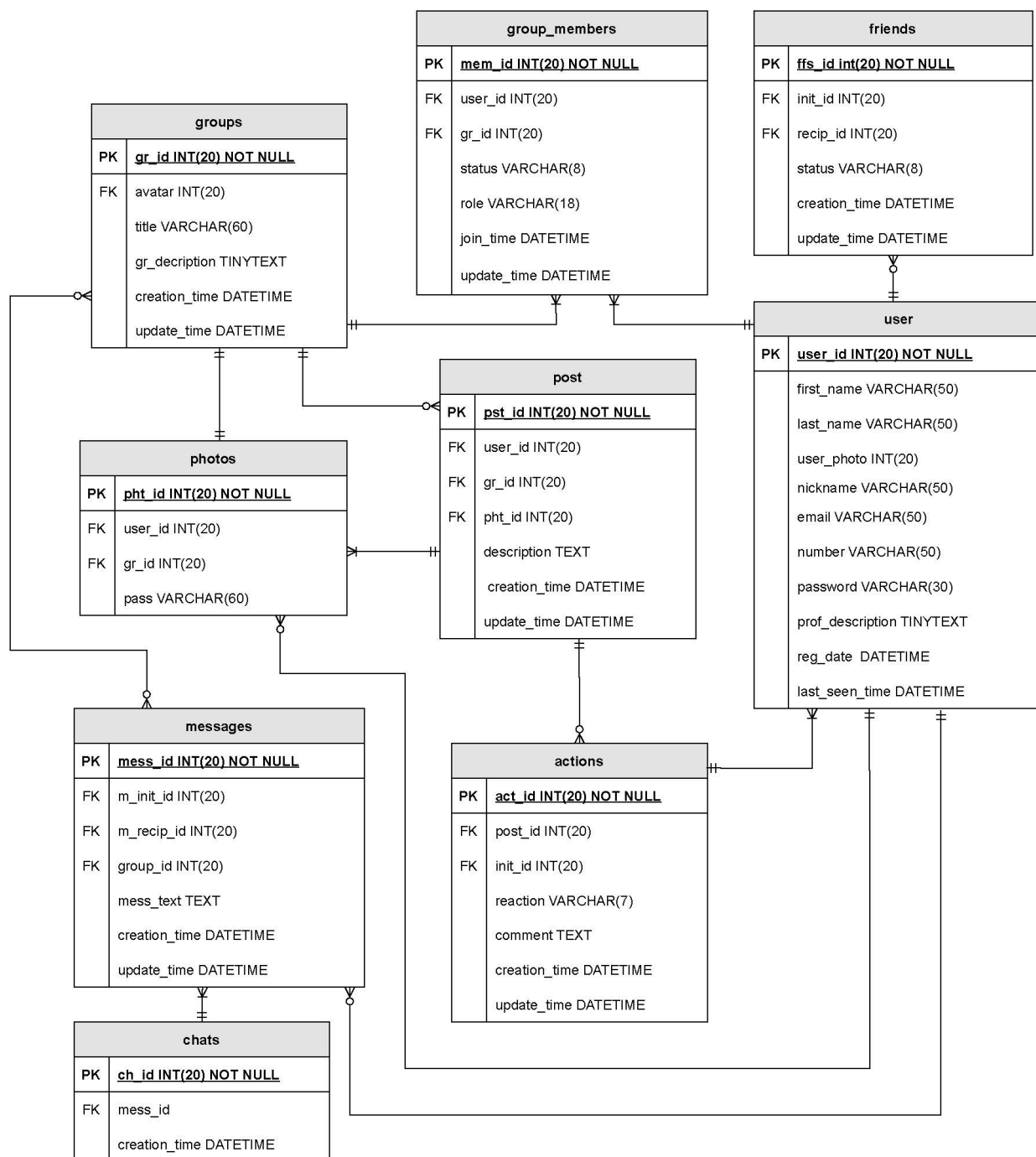


Рис. 3.1 ER-модель базы данных

4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

Реляційна модель:

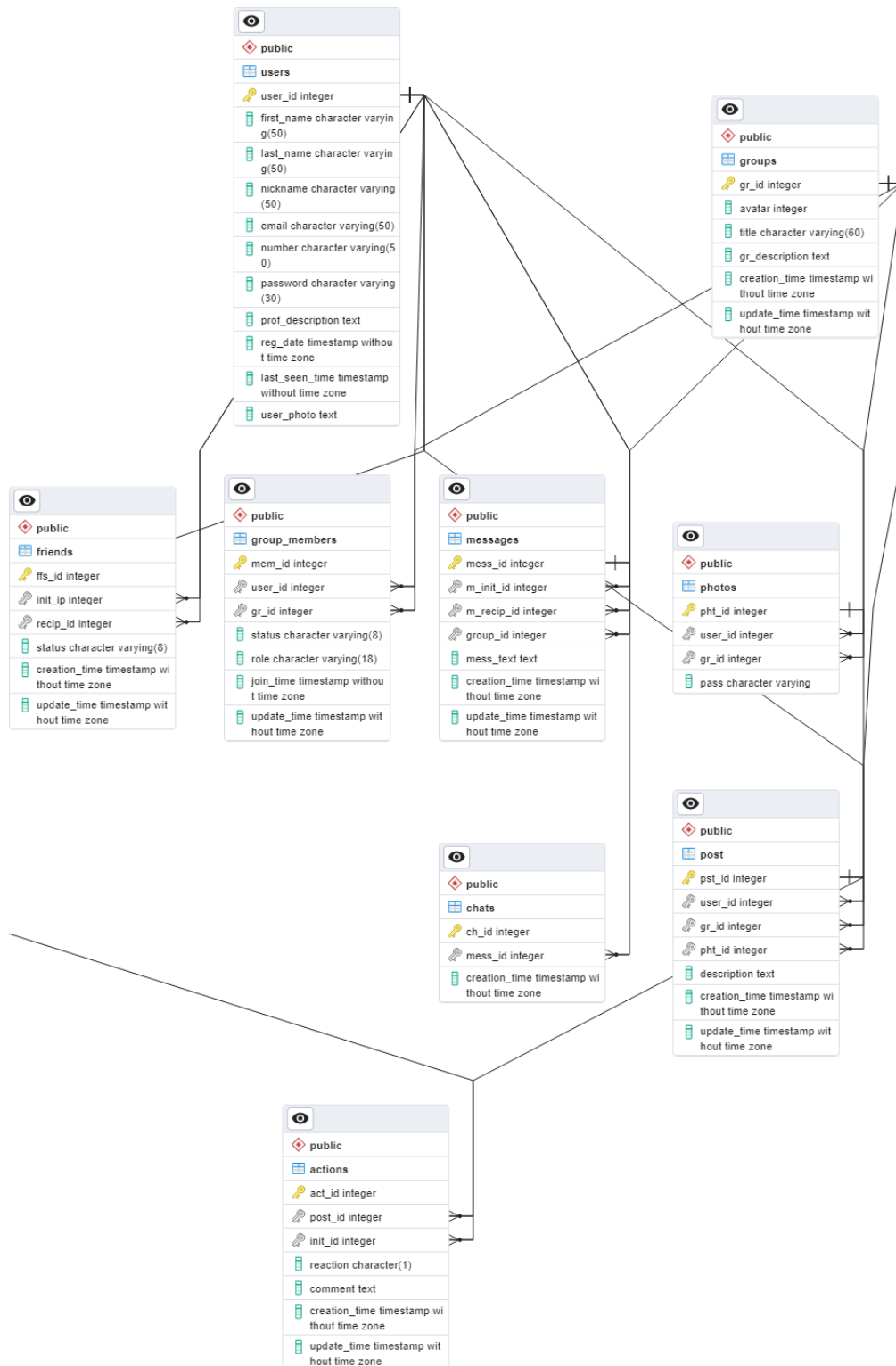


Рис. 4.1 Реляційна модель бази даних

5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

Створюємо базу даних

```
create table users (
```

```
user_id integer primary key,
```

```
firts_name varchar(50),
```

```
last_name varchar(50),
```

```
nickname varchar(50),
```

```
email varchar(50),
```

```
number varchar(50),
```

```
password varchar(30),
```

```
prof_description text,
```

```
reg_date timestamp,
```

```
last_seen_time timestamp,
```

```
user_photo text);
```

```
copy users(user_id, firts_name, last_name, nickname,email, number, password,
```

```
prof_description,reg_date, last_seen_time, user_photo)
```

```
from 'C:\Users\Asus\Desktop\DB\users.csv'
```

```
delimiter ';'
```

```
csv header
```

```
create table friends(ffs_id integer primary key,
```

```
init_ip integer references users(user_id),
```

recip_id integer references users(user_id),

status VARCHAR(8),

creation_time timestamp,

update_time timestamp);

copy friends(ffs_id, init_ip, recip_id, status, creation_time, update_time)

from 'C:\Users\Asus\Desktop\DB\friendship.csv'

delimiter ';'

csv header

create table **groups**(gr_id integer primary key,

avatar integer,

title varchar(60),

gr_description text,

creation_time timestamp,

update_time timestamp);

copy groups(gr_id, avatar, title, gr_description, creation_time, update_time)

from 'C:\Users\Asus\Desktop\DB\groups.csv'

delimiter ';'

csv header

```
create table photos(  
pht_id integer primary key,  
user_id integer references users(user_id),  
gr_id integer references groups(gr_id),  
pass VARCHAR);
```

```
copy photos(pht_id, user_id, gr_id, pass)  
from 'C:\Users\Asus\Desktop\DB\photos.csv'  
delimiter ';'   
csv header
```

```
create table group_members(mem_id integer primary key,  
user_id integer references users(user_id),  
gr_id integer references groups(gr_id),  
status varchar(8),  
role varchar(18),  
join_time timestamp,  
update_time timestamp);
```

```
copy group_members(mem_id, user_id, gr_id, status, role, join_time, update_time)  
from 'C:\Users\Asus\Desktop\DB\group_members.csv'  
delimiter ';'   
csv header
```

```
create table post(pst_id integer primary key,  
user_id integer references users(user_id),  
gr_id integer references groups(gr_id),  
pht_id integer references photos(pht_id),  
description text,  
creation_time timestamp,  
update_time timestamp);
```

```
copy post(pst_id,user_id, gr_id, pht_id, description, creation_time, update_time)  
from 'C:\Users\Asus\Desktop\DB\post.csv'  
delimiter ';'   
csv header
```

```
create table actions(act_id integer primary key,  
post_id integer references post(pst_id),  
init_id integer references users(user_id),  
reaction char,  
comment text,  
creation_time timestamp,  
update_time timestamp);
```

```
copy actions(act_id,post_id, init_id, reaction, comment, creation_time,  
update_time)
```

```
from 'C:\Users\Asus\Desktop\DB\actions.csv'
```

```
delimiter ';' 
```

```
csv header
```

```
create table messages(mess_id integer primary key,
```

```
m_init_id integer references users(user_id),
```

```
m_recip_id integer references users(user_id),
```

```
group_id integer references groups(gr_id),
```

```
mess_text text,
```

```
creation_time timestamp, update_time timestamp);
```

```
copy messages(mess_id, m_init_id, m_recip_id, group_id, mess_text,
```

```
creation_time, update_time)
```

```
from 'C:\Users\Asus\Desktop\DB\messages.csv'
```

```
delimiter ';' 
```

```
csv header
```

6. РОБОТА З БАЗОЮ ДАНИХ

СТВОРЮЄМО КОРИСТУВАЧІВ

```
create user Mary with password 'MaryMary';  
create user administrator with password 'AdminAdmin';  
create user moderator with password 'AdminAdmin';
```

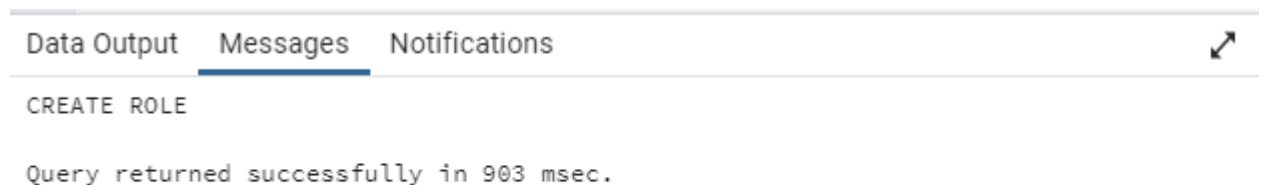


Рис 6.1

ЗАПИТИ

1)Вибірка користувачів соціальної мережі та груп, до яких вони належать

```
SELECT users.user_id, users.first_name, users.nickname, groups.title AS  
user_group FROM users, groups, group_members WHERE  
users.user_id = group_members.user_id AND group_members.gr_id =  
groups.gr_id ORDER BY users.user_id;
```

The screenshot shows a database management interface with three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with 10 rows of user data. The table has five columns: 'user_id', 'first_name', 'nickname', and 'user_group'. The 'user_id' column is labeled 'integer' and has a lock icon. The 'first_name' column is labeled 'character varying (50)' and has a lock icon. The 'nickname' column is labeled 'character varying (50)' and has a lock icon. The 'user_group' column is labeled 'character varying (60)' and has a lock icon. The table contains 10 rows of data, numbered 1 to 10 in the first column.

	user_id integer	first_name character varying (50)	nickname character varying (50)	user_group character varying (60)
1	1	Mary	Cat_it	Dragon_family
2	2	Michael	Mick	Home_page
3	3	Alex	Alex	Dragon_family
4	4	Nadin	Ridiculous	Dragon_family
5	4	Nadin	Ridiculous	Home_page
6	4	Nadin	Ridiculous	Anime
7	5	Omelian	soul_eater	Bascketsport
8	7	Sara	imposter	Gamer
9	8	Liam	madhouse	Gamer
10	9	Stacy	Stacy	Bascketsport

Рис 6.2

2) Шукаємо користувачів ім'я яких починається на літеру М та вони є учасниками припустимо одних з 4х найпопулярніших груп у соц.мережі


```
SELECT users.user_id, users.first_name, users.last_name
```

```
FROM users, group_members WHERE first_name
```

```
SIMILAR TO 'M%'='t' AND users.user_id = group_members.user_id
```

```
AND group_members.gr_id BETWEEN 1 AND 4;
```

Data Output Messages Notifications



	user_id [PK] integer	first_name character varying (50)	last_name character varying (50)
1	1	Mary	Smith
2	2	Michael	Landa

Рис 6.3

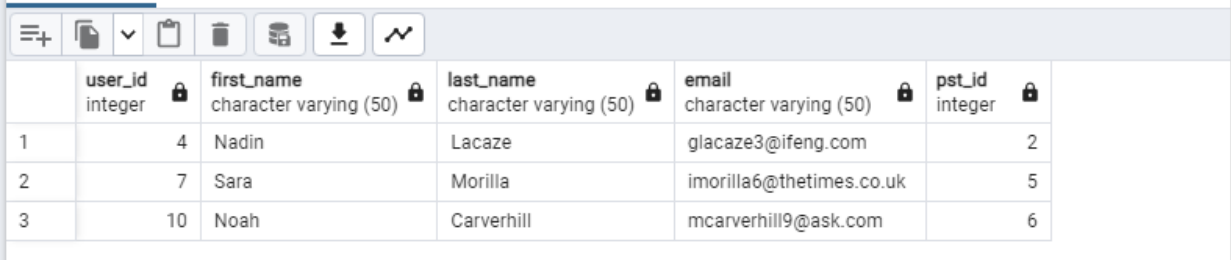
3) шукаємо користувачів року приєднання пізніше 2017 та відображаємо id постів які вони з того часу виклали

```
SELECT users.user_id, users.first_name, users.last_name, users.email, post.pst_id
```

```
FROM users, post WHERE DATE_PART('year', reg_date) > 2017
```

```
AND users.user_id = post.user_id;
```

Data Output Messages Notifications



	user_id integer	first_name character varying (50)	last_name character varying (50)	email character varying (50)	pst_id integer
1	4	Nadin	Lacaze	glacaze3@ifeng.com	2
2	7	Sara	Morilla	imorilla6@thetimes.co.uk	5
3	10	Noah	Carverhill	mcarverhill9@ask.com	6

Рис 6.4

4) Відображаємо користувачів, що викладали пости з фото а також шлях доступу до фото, що лежить в загальному альбомі бази даних

```
SELECT users.user_id, users.nickname, post.pst_id, photos.pass AS photo
```

```
FROM users, post, photos WHERE users.user_id = post.user_id
```

```
AND post.pht_id is NOT NULL
```

```
AND post.pht_id = photos.pht_id;
```

Data Output Messages Notifications				
	user_id integer	nickname character varying (50)	pst_id integer	photo character varying
1	2	Mick	1	https://robohash.org/fugiatetassumenda.jpg?size=50x50&set=set1
2	4	Ridiculous	2	https://robohash.org/remetasperiores.jpg?size=50x50&set=set1
3	5	soul_eater	4	https://robohash.org/voluptasneccitatibusconsequatur.jpg?size=50x50&set=set1
4	10	Chaotic	6	https://robohash.org/nostrumautrecusandae.jpg?size=50x50&set=set1

Рис 6.5

5) Відображаємо групу з назвою і її адміністратора використовуючи таблиці, що зберігають групи, членів групи та таблицю користувачів:

```
SELECT groups.gr_id, groups.title, users.nickname, group_members.role
```

```
FROM groups, users, group_members WHERE groups.gr_id =  
group_members.gr_id
```

```
AND group_members.user_id = users.user_id
```

```
AND group_members.role = 'admin';
```

Data Output Messages Notifications				
	gr_id integer	title character varying (60)	nickname character varying (50)	role character varying (18)
1	1	Dragon_family	Alex	admin
2	2	Bascketsport	Stacy	admin

Рис 6.6

6) Відображаємо групи, файли їх зображень сторінки, пости з фотографіями та відображаємо шлях що зазначений в загальному альбомі «photos»

```
SELECT groups.gr_id, groups.title, groups.avatar, post.pst_id, photos.pass
```

```
FROM groups, post, photos WHERE groups.gr_id = post.gr_id
```

```
AND post.pht_id = photos.pht_id;
```

Data Output Messages Notifications					
	gr_id integer	title character varying (60)	avatar integer	pst_id integer	pass character varying
1	1	Dragon_family	56375	7	https://robohash.org/nequeetquae.jpg?size=50x50&set=set1
2	2	Bascketsport	4532	9	https://robohash.org/eligendiquasiitaque.jpg?size=50x50&set=set1

Рис 6.7

7)Відображаємо групи, файли їх зображень сторінки, пости та шлях до фото якщо він присутній, якщо ні – відображаємо нулі (для реалізації використовуємо праве зовнішнє з’єднання

```
SELECT groups.gr_id, groups.title, groups.avatar, post.pst_id, photos.pass
```

```
FROM groups, photos
```

```
RIGHT OUTER JOIN post ON photos.pht_id = post.pht_id
```

```
WHERE groups.gr_id = post.gr_id ORDER BY groups.gr_id;
```

Data Output Messages Notifications						
	gr_id integer	title character varying (60)	avatar integer	pst_id integer	pass character varying	
1	1	Dragon_family	56375	7	https://robohash.org/nequeetquae.jpg?size=50x50&set=set1	
2	2	Bascketsport	4532	3	[null]	
3	2	Bascketsport	4532	9	https://robohash.org/eligendiquasiitaque.jpg?size=50x50&set	
4	5	Gamer	958473	8	[null]	

Рис 6.8

8) Відображаємо групи, файли їх зображень сторінки, пости та шлях до фото якщо він присутній, якщо ні – відображаємо нулі (для реалізації використовуємо праве зовнішнє з’єднання, також відображаємо групи які не мають пости, запит має збіжності із запитом 7 вказаним вище, але також додаємо ліве зовнішнє з’єднання для доцільної реалізації

```
SELECT groups.gr_id, groups.title, groups.avatar, post.pst_id, photos.pass
```

```
FROM photos
```

```
RIGHT OUTER JOIN post ON photos.pht_id = post.pht_id
```

```
RIGHT OUTER JOIN groups ON post.gr_id = groups.gr_id ORDER BY  
groups.gr_id;
```

Data Output Messages Notifications						
	gr_id integer	title character varying (60)	avatar integer	pst_id integer	pass character varying	
1	1	Dragon_family	56375	7	https://robohash.org/nequeetquae.jpg?size=50x50&set=set1	
2	2	Bascketsport	4532	3	[null]	
3	2	Bascketsport	4532	9	https://robohash.org/eligendiquasiitaque.jpg?size=50x50&set	
4	3	Anime	1234	[null]	[null]	
5	4	Home_page	1437	[null]	[null]	
6	5	Gamer	958473	8	[null]	

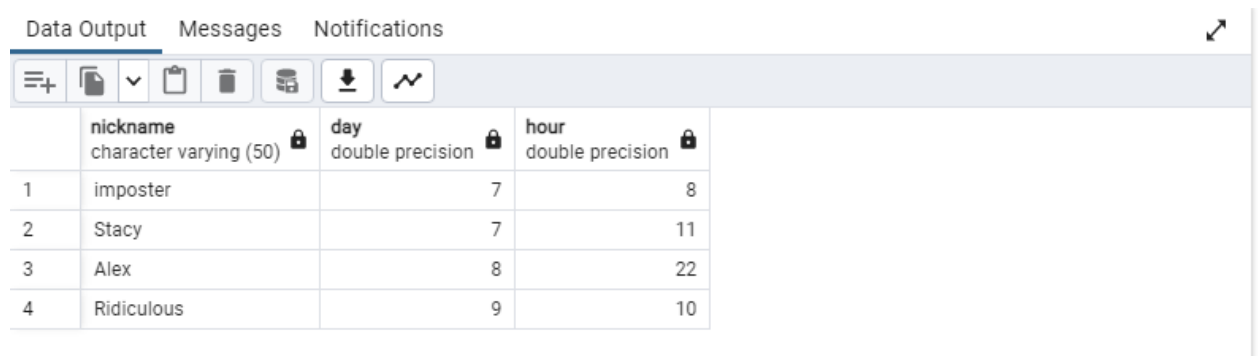
Рис 6.9

9) Визначаємо загальну кількість поточних користувачів даної мережі

```
SELECT COUNT(user_id) AS total_am_of_users FROM users;
```

10) Визначаємо користувачів, що були останній раз в мережі після 3го числа у проміжку з 1 до 23 години

```
SELECT nickname, DATE_PART('day', last_seen_time) AS day,  
DATE_PART('hour', last_seen_time) AS hour  
FROM users WHERE DATE_PART('day', last_seen_time) > 03  
AND DATE_PART('hour', last_seen_time) BETWEEN 1 AND 23  
ORDER BY day;
```



The screenshot shows a database query result interface. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with icons for various actions. The main area displays a table with the following data:

	nickname character varying (50)	day double precision	hour double precision
1	imposter	7	8
2	Stacy	7	11
3	Alex	8	22
4	Ridiculous	9	10

Рис 6.1.1

11)Список користувачів та текст повідомлень, що вони відправляли сортований за часом створення повідомлення

```
SELECT users.user_id, users.nickname, messages.mess_text,  
messages.creation_time  
FROM users, messages WHERE users.user_id = messages.m_init_id  
ORDER BY messages.creation_time;
```

Data Output Messages Notifications				
	user_id integer	nickname character varying (50)	mess_text text	creation_time timestamp without time zone
1	9	Stacy	2023-01-02 12:34:52
2	2	Mick	Where you've been?	2023-01-02 12:34:52
3	9	Stacy	call me pls	2023-01-02 12:34:53
4	8	madhouse	few min, im kinda bussy	2023-01-02 12:34:54
5	9	Stacy	sure	2023-01-02 12:34:55
6	6	Kev_v	Do you rember about my list?	2023-01-06 03:47:34
7	1	Cat_it	Hi!	2023-01-06 16:44:41
8	2	Mick	Hello, Mary	2023-01-06 16:44:42
9	2	Mick	Forgot it at home...	2023-01-07 15:53:24
10	6	Kev_v	at 7am. Sure	2023-01-08 12:05:45
11	6	Kev_v	:///	2023-01-08 18:23:49
12	3	Alex	Will you post that photo?	2023-01-08 18:40:27
13	7	imposter	Not sure	2023-01-08 18:40:28
14	7	imposter	but maybe	2023-01-08 18:40:29
15	8	madhouse	So what?	2023-01-09 01:05:21
16	5	soul_eater	yes no idk	2023-01-09 01:05:22

Рис 6.1.2

12)Відображаємо повідомлення в чатах створені за 8 число
SELECT chats.ch_id, chats.creation_time, messages.mess_text
FROM chats, messages
WHERE chats.mess_id = messages.mess_id
AND DATE_PART('day', chats.creation_time) = 8;


Data Output Messages Notifications			
	ch_id integer	creation_time timestamp without time zone	mess_text text
1	3	2023-01-08 18:23:49	Here we show some tattoo ideas

Рис 6.1.3

13)Відображаємо пости та людей що виконували певні дії над ним: наявність лайку – 1, коментарі – відображення тексту

```
SELECT actions.post_id, users.nickname AS liker, actions.reaction,  
actions.comment
```

```
FROM actions, users WHERE actions.post_id = users.user_id;
```



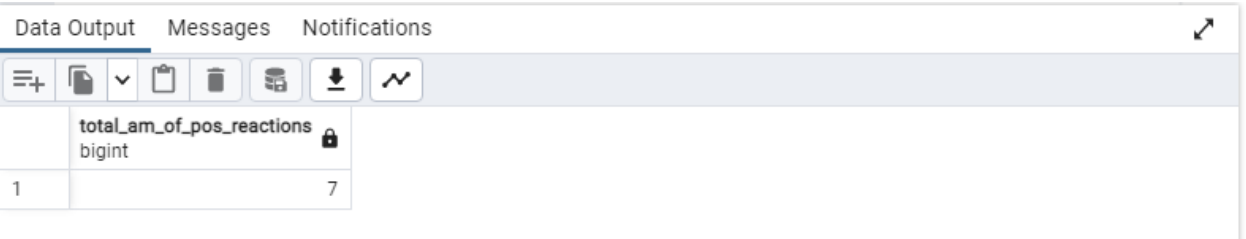
The screenshot shows a database interface with tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with icons for various actions. The main area displays a table with the following data:

	post_id integer	liker character varying (50)	reaction character (1)	comment text
1	2	imposter	1	so nice
2	4	madhouse	1	[null]
3	1	Stacy	1	[null]
4	3	Alex	1	cool!
5	5	Ridiculous	1	[null]
6	6	Alex	0	omg(
7	9	soul_eater	0	[null]
8	7	imposter	1	[null]
9	10	Stacy	1	[null]
10	8	Chaotic	0	[null]

Рис 6.1.4

14)Відображаємо загальну кількість позитивних реакцій у мережі

```
SELECT COUNT(reaction) AS total_am_of_pos_reactions FROM actions  
WHERE reaction = '1';
```



The screenshot shows a database interface with tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with icons for various actions. The main area displays a table with the following data:

	total_am_of_pos_reactions bigint
1	7

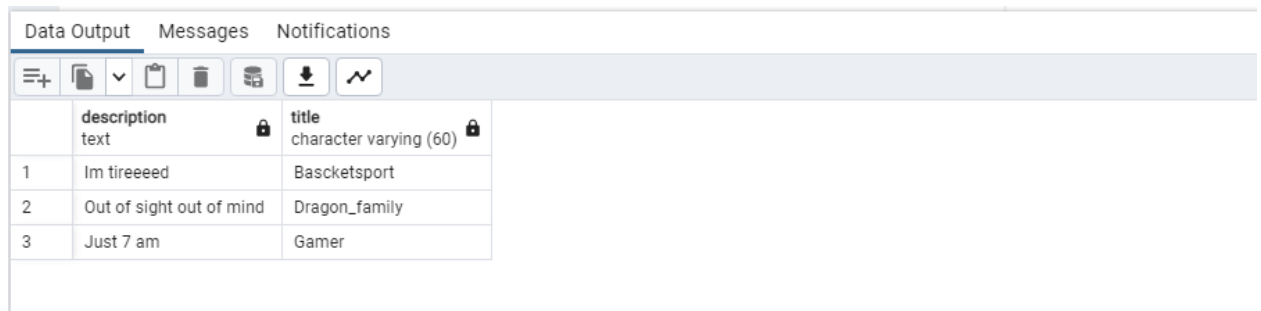
Рис 6.1.5

Загальну кількість коментарів

```
SELECT COUNT(comment) AS total_am_of_pos_reactions FROM actions  
WHERE comment IS NOT NULL;
```


WHERE post.gr_id = groups.gr_id

AND description is NOT NULL;



The screenshot shows a database interface with tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with two columns: 'description' (text) and 'title' (character varying (60)). The table contains three rows of data.

	description text	title character varying (60)
1	Im tireeed	Bascketsport
2	Out of sight out of mind	Dragon_family
3	Just 7 am	Gamer

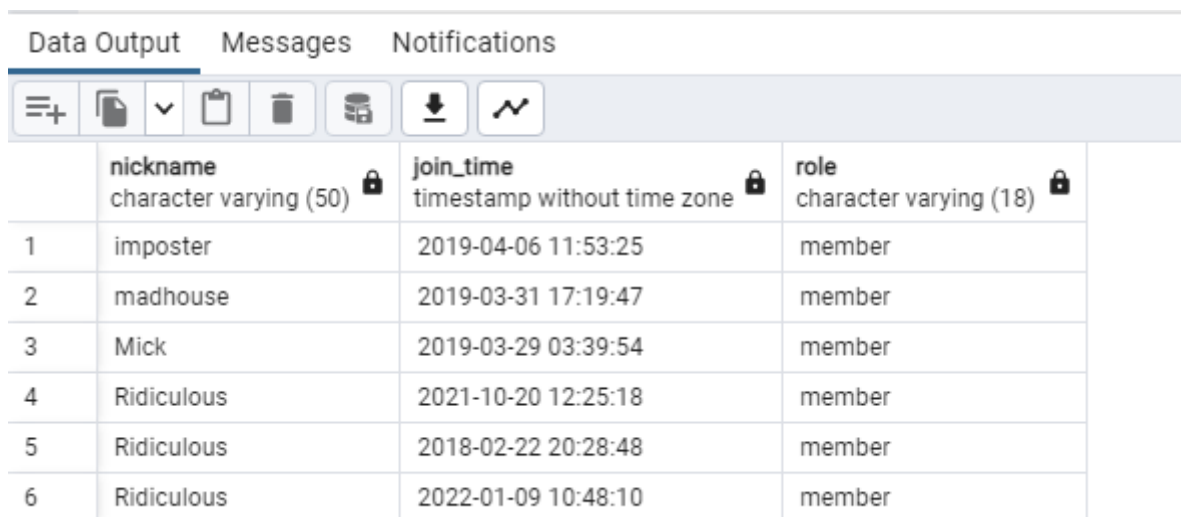
Рис 6.1.9

18)Відображаємо дату та час приєднання користувачів до певних груп

SELECT users.nickname, group_members.join_time, group_members.role

FROM users, group_members WHERE users.user_id = group_members.user_id

AND group_members.role = 'member' ORDER BY users.nickname;



The screenshot shows a database interface with tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with three columns: 'nickname' (character varying (50)), 'join_time' (timestamp without time zone), and 'role' (character varying (18)). The table contains six rows of data.

	nickname character varying (50)	join_time timestamp without time zone	role character varying (18)
1	imposter	2019-04-06 11:53:25	member
2	madhouse	2019-03-31 17:19:47	member
3	Mick	2019-03-29 03:39:54	member
4	Ridiculous	2021-10-20 12:25:18	member
5	Ridiculous	2018-02-22 20:28:48	member
6	Ridiculous	2022-01-09 10:48:10	member

Рис 6.1.1.1

19)Відображаємо нікнейм користувача, дату та час приєднання та, безпосередньо, групу в якій він займає певну роль. Тільки ті користувачі, яким надано статус «active»

SELECT users.nickname, group_members.join_time, group_members.role,
groups.title

FROM users, group_members, groups WHERE users.user_id =
group_members.user_id

AND group_members.gr_id = groups.gr_id

AND group_members.status = 'active'

ORDER BY users.nickname;

Data Output Messages Notifications				
	nickname character varying (50)	join_time timestamp without time zone	role character varying (18)	title character varying (60)
1	Alex	2022-08-07 19:36:21	admin	Dragon_family
2	Cat_it	2022-11-30 22:56:08	creator	Dragon_family
3	madhouse	2019-03-31 17:19:47	member	Gamer
4	Mick	2019-03-29 03:39:54	member	Home_page
5	Ridiculous	2021-10-20 12:25:18	member	Dragon_family
6	Ridiculous	2018-02-22 20:28:48	member	Home_page
7	Stacy	2021-01-09 00:05:17	admin	Bascketsport

Рис 6.1.1.2

20)Відображаємо користувачів, що змінювали повідомлення та час редагування

```
SELECT users.first_name, users.nickname, messages.mess_text,  
messages.update_time  
FROM users, messages WHERE users.user_id = messages.m_init_id  
AND messages.update_time IS NOT NULL ORDER BY messages.update_time;
```

Data Output Messages Notifications				
	first_name character varying (50)	nickname character varying (50)	mess_text text	update_time timestamp without time zone
1	Mary	Cat_it	Hi!	2023-01-06 16:44:00

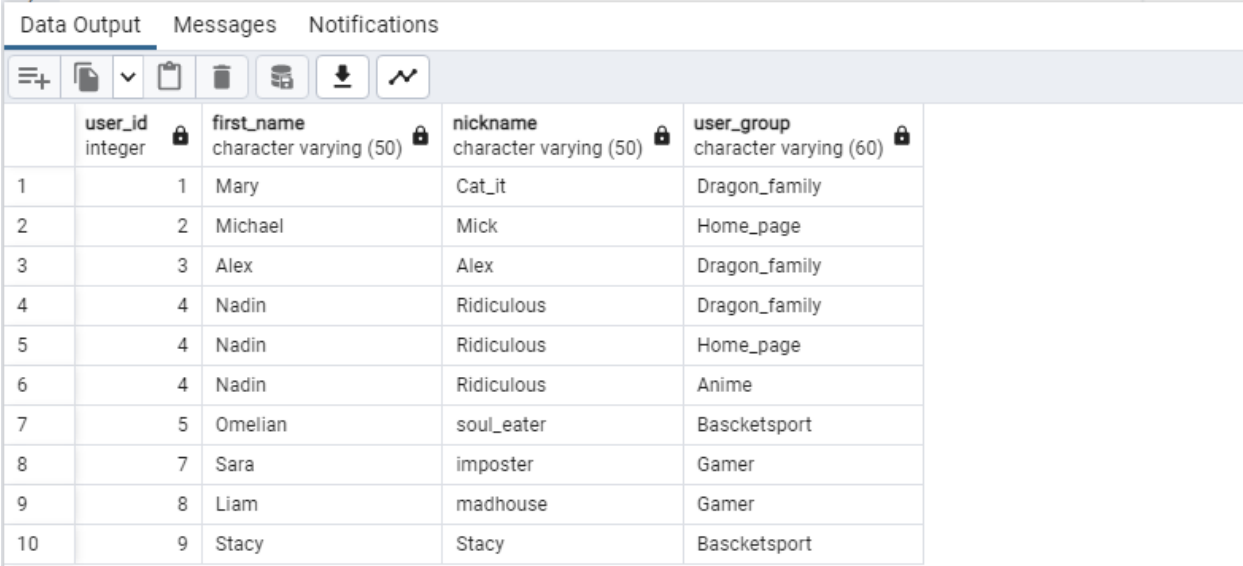
Рис 6.1.1.3

ПОДАННЯ

- 1) Створюємо подання для відображення користувачів, що належать певним групам

```
CREATE VIEW groups_by_users AS
```

```
SELECT users.user_id, users.first_name, users.nickname, groups.title AS  
user_group FROM users, groups, group_members WHERE  
users.user_id = group_members.user_id AND group_members.gr_id =  
groups.gr_id ORDER BY users.user_id
```



	user_id integer	first_name character varying (50)	nickname character varying (50)	user_group character varying (60)
1	1	Mary	Cat_it	Dragon_family
2	2	Michael	Mick	Home_page
3	3	Alex	Alex	Dragon_family
4	4	Nadin	Ridiculous	Dragon_family
5	4	Nadin	Ridiculous	Home_page
6	4	Nadin	Ridiculous	Anime
7	5	Omelian	soul_eater	Bascketsport
8	7	Sara	imposter	Gamer
9	8	Liam	madhouse	Gamer
10	9	Stacy	Stacy	Bascketsport

Рис 6.2.2

- 2) Створюємо подання для відображення дати приєднання та певної ролі користувача в групі

```
CREATE VIEW reg_in_group AS
```

```
SELECT users.nickname, group_members.join_time, group_members.role,  
groups.title  
FROM users, group_members, groups WHERE users.user_id =  
group_members.user_id  
AND group_members.gr_id = groups.gr_id  
AND group_members.status = 'active'  
ORDER BY users.nickname;
```

Data Output Messages Notifications				
	nickname character varying (50)	join_time timestamp without time zone	role character varying (18)	title character varying (60)
1	Alex	2022-08-07 19:36:21	admin	Dragon_family
2	Cat_it	2022-11-30 22:56:08	creator	Dragon_family
3	madhouse	2019-03-31 17:19:47	member	Gamer
4	Mick	2019-03-29 03:39:54	member	Home_page
5	Ridiculous	2021-10-20 12:25:18	member	Dragon_family
6	Ridiculous	2018-02-22 20:28:48	member	Home_page
7	Stacy	2021-01-09 00:05:17	admin	Bascketsport

Рис 6.2.3

3) Створюємо подання для відображення дій користувачів

CREATE VIEW users_actions AS

SELECT actions.post_id, users.nickname AS liker, actions.reaction,
actions.comment

FROM actions, users WHERE actions.init_id = users.user_id;

	post_id integer	liker character varying (50)	reaction character (1)	comment text
1	2	imposter	1	so nice
2	4	madhouse	1	[null]
3	1	Stacy	1	[null]
4	3	Alex	1	cool!
5	5	Ridiculous	1	[null]
6	6	Alex	0	omg(
7	9	soul_eater	0	[null]
8	7	imposter	1	[null]
9	10	Stacy	1	[null]
10	8	Chaotic	0	[null]

Рис 6.2.4

4) Відображаємо користувачів та пости, що за ними закріплені

CREATE VIEW users_posts AS

SELECT users.user_id, users.first_name, users.last_name, users.email,
post.pst_id

FROM users, post WHERE users.user_id = post.user_id;

Data Output

Messages

Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

📈

	user_id integer	first_name character varying (50)	last_name character varying (50)	email character varying (50)	pst_id integer
1	2	Michael	Landa	csigg1@geocities.jp	1
2	4	Nadin	Lacaze	glacaze3@ifeng.com	2
3	5	Omelian	Currey	lcurrey4@sakura.ne.jp	4
4	7	Sara	Morilla	imorilla6@thetimes.co.uk	5
5	10	Noah	Carverhill	mcarverhill9@ask.com	6
6	1	Mary	Smith	lsnawden0@cnet.com	10

Рис 6.2.5

5) Створюємо подання для відображення останніх повідомлень користувачів

CREATE VIEW last_mess AS

SELECT users.user_id, users.nickname, users.number, messages.mess_text,
messages.creation_time FROM users, messages WHERE users.user_id =
messages.m_init_id

ORDER BY messages.creation_time;

Data Output

Messages

Notifications

	user_id integer	nickname character varying (50)	number character varying (50)	mess_text text	creation_time timestamp without time zone
1	9	Stacy	380988744551	2023-01-02 12:34:52
2	2	Mick	38098894455	Where you've been?	2023-01-02 12:34:52
3	9	Stacy	380988744551	call me pls	2023-01-02 12:34:53
4	8	madhouse	380756754431	few min, im kinda bussy	2023-01-02 12:34:54
5	9	Stacy	380988744551	sure	2023-01-02 12:34:55
6	6	Kev_v	380634555565	Do you rember about my list?	2023-01-06 03:47:34
7	1	Cat_it	38098894455	Hi!	2023-01-06 16:44:41
8	2	Mick	38098894455	Hello, Mary	2023-01-06 16:44:42
9	2	Mick	38098894455	Forgot it at home...	2023-01-07 15:53:24
10	6	Kev_v	380634555565	at 7am. Sure	2023-01-08 12:05:45
11	6	Kev_v	380634555565	:///	2023-01-08 18:23:49
12	3	Alex	380666767455	Will you post that photo?	2023-01-08 18:40:27
13	7	imposter	380665564433	Not sure	2023-01-08 18:40:28
14	7	imposter	380665564433	but maybe	2023-01-08 18:40:29
15	8	madhouse	380756754431	So what?	2023-01-09 01:05:21
16	5	soul_eater	380935667889	yes no idk	2023-01-09 01:05:22

Рис 6.2.6

ПРОЦЕДУРИ, ФУНКЦІЇ, КУРСОРИ, ТРИГЕРИ

- 1) Процедура, що надсилає сповіщення, коли певний користувач ставить лайк над певним постом іншого користувача

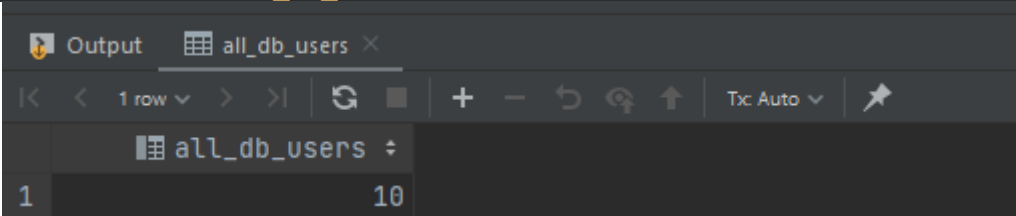
```
CREATE OR REPLACE PROCEDURE action() AS $$
BEGIN
    IF (SELECT actions.reaction FROM actions WHERE actions.post_id =
'2'
                                AND
actions.init_id = '7')
    THEN RAISE NOTICE 'Someone liked your post!';
END IF;
END;
$$language plpgsql;
CALL action();

SocialMedia.public> CALL action()
Someone liked your post!
[2023-01-18 12:55:17] completed in 5 ms
```

Рис 6.3.3.1

- 2) Функція, що рахує усіх користувачів бази даних

```
CREATE OR REPLACE FUNCTION all_db_users() returns numeric AS $$
BEGIN
    RETURN COUNT(user_id) FROM users;
END;
$$language plpgsql;
SELECT * FROM all_db_users();
```



all_db_users	
1	10

Рис 6.3.3.2

- 3) Створюємо керувальну структуру для обходу записів в базі даних

```
DO
$$
DECLARE
    CUR CURSOR FOR SELECT user_id, first_name, number, email FROM users;
    REC RECORD;
BEGIN
    OPEN CUR;
    loop
        FETCH CUR INTO REC;
        EXIT WHEN NOT FOUND;
        RAISE NOTICE 'id: %, name: %, number: %, email: %',rec.user_id,
rec.first_name,
        rec.number, rec.email;
    end loop;
    CLOSE CUR;
END;
$$;
```

4) Створюємо функцію для тригера оновлення

```
CREATE OR REPLACE FUNCTION check_group_title_update() RETURNS TRIGGER AS
$$
BEGIN
    raise notice 'you gona update % title', new;
    return new;
END;
$$language plpgsql;
```

5) Створюємо тригер на оновлення назви групи

```
CREATE OR REPLACE TRIGGER check_update
BEFORE UPDATE ON groups
FOR EACH ROW
WHEN ( old.title != new.title )
EXECUTE FUNCTION check_group_title_update();
```

6) Створюємо функцію для тригера видалення

```
CREATE OR REPLACE FUNCTION check_user_deliting() RETURNS TRIGGER AS
$$
BEGIN
    raise notice 'you gona delete % user', old;
    return old;
END;
$$language plpgsql;
```

7) Створюємо тригер на видалення користувача

```
CREATE OR REPLACE TRIGGER check_del
BEFORE DELETE ON users
FOR EACH ROW
WHEN (OLD.nickname = NULL)
EXECUTE FUNCTION check_user_deliting();
```

8) Створення функції для тригера вставлення даних

```
CREATE OR REPLACE FUNCTION check_inserting() RETURNS TRIGGER AS
$$
BEGIN
    raise notice 'you gona add new user';
    return notice;
END;
$$language plpgsql;
```

9) Створення тригера на додавання даних в таблицю користувачів

```
CREATE OR REPLACE TRIGGER check_insert
BEFORE INSERT ON users
FOR EACH ROW
EXECUTE FUNCTION check_inserting();
```

ВИСНОВОК

У даній роботі представлена база даних для реалізації соціальної мережі. Найпопулярніші соціальні мережі світу налічують мільйони користувачів. Початкова ідея соціальної мережі це зберігання та обмін певними даними. Тож можемо зробити висновок – будь-яка соціальна мережа не може існувати без побудованої під неї бази даних. Тож дана робота є початковим ключем для створення доцільної та зручної користувачу мережі. Також у цій роботі представлені варіанти використання даної бази даних. Для проектування було використано СУБД PostgreSQL, pgAdmin4, середовище розробки скриптів DataGrip. Представлений код створення таблиць, управління запитами, представленнями, створення процедур функцій, курсорів та тригерів. В результаті детально представлені всі дії та операції у даній роботі.

ДОДАТОК А

Рис 1

