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 реалізаії 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 \_\_ІП-13\_\_ Семестр \_\_\_3\_\_

**З А В Д А Н Н Я**

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Романюк Діані Олексіївні\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(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 | 05.15.2022 |  |
| 10 | Захист курсової роботи | 16.01.2022 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Студент \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_**Романюк Д. О.

(підпис ) (прізвище та ініціали)

**Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**Ліщук О. В.

(підпис ) (прізвище та ініціали)

ЗМІСТ

[ВСТУП 5](#_Toc124947789)

[1. ОПИС ПРЕДМЕТНОГО СЕРИДОВИЩА 6](#_Toc124947790)

[2. ПОСТАНОВКА ЗАДАЧІ 7](#_Toc124947791)

[3. ПРОЕКТУВАННЯ БАЗИ ДАНИХ 8](#_Toc124947792)

[4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ 10](#_Toc124947793)

[5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ 11](#_Toc124947794)

[6. РОБОТА З БАЗОЮ ДАНИХ 16](#_Toc124947795)

[ВИСНОВОК 31](#_Toc124947796)

# 

# 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.

# 

# ОПИС ПРЕДМЕТНОГО СЕРИДОВИЩА

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

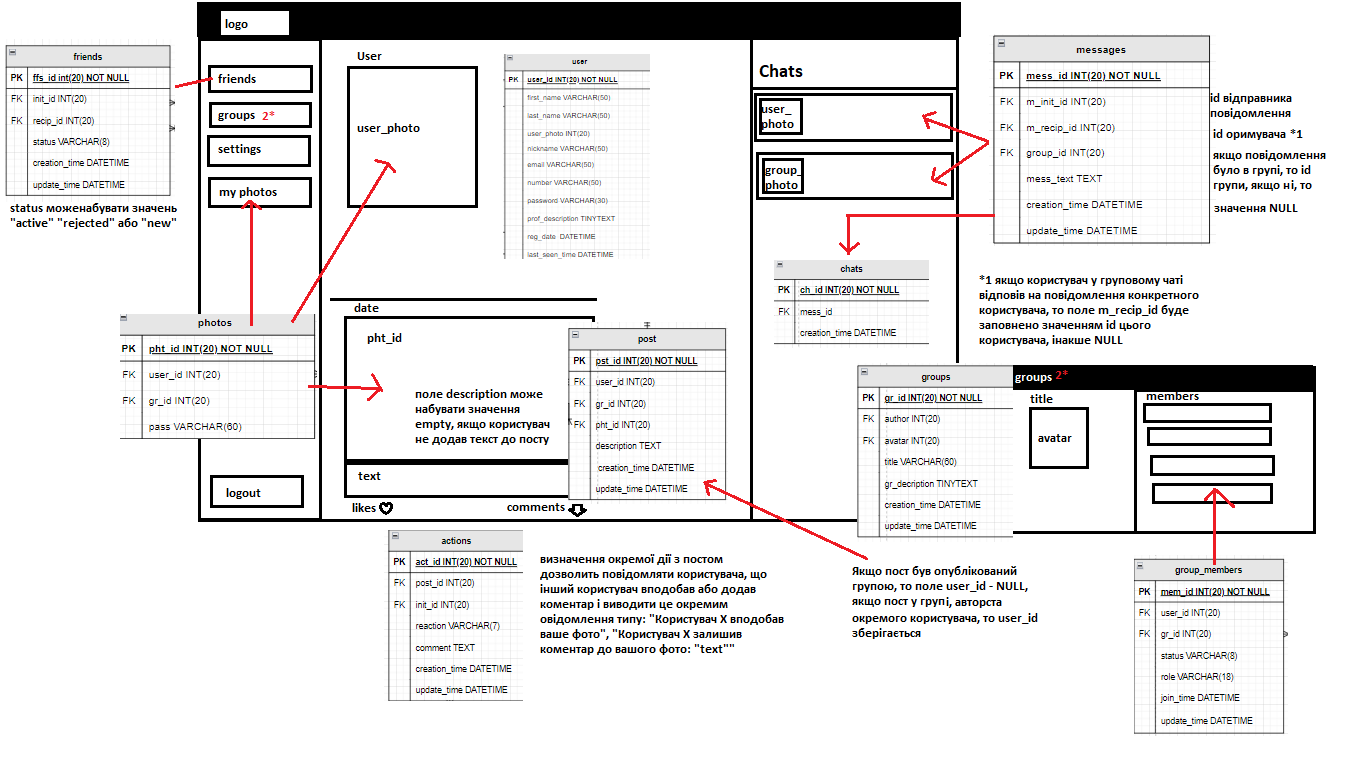


Рис 1.1 Схема сторінки соціальної мережі

# 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.

# 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, часник групи, пост, чат та дія не можуть повторюватись у своїй категорії та набувати нульового значенння. Імена та назви груп визначенні буквенно. Номери телефону, назви файлів аватарів користувачів та номери телефонів визначенні чисельно.

ER-модель:

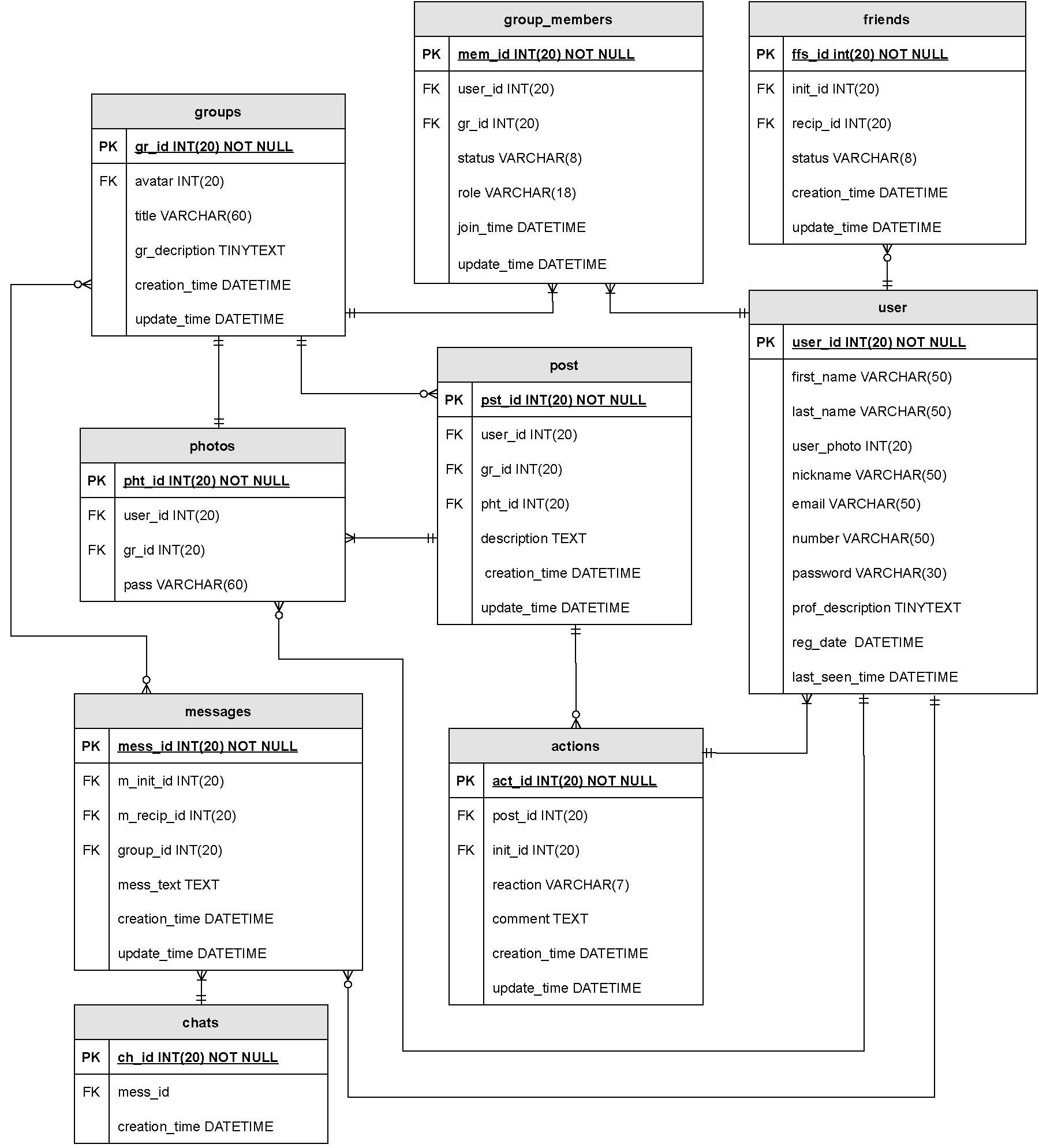


Рис. 3.1 ER-модель бази даних

# РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

Реляційна модель:

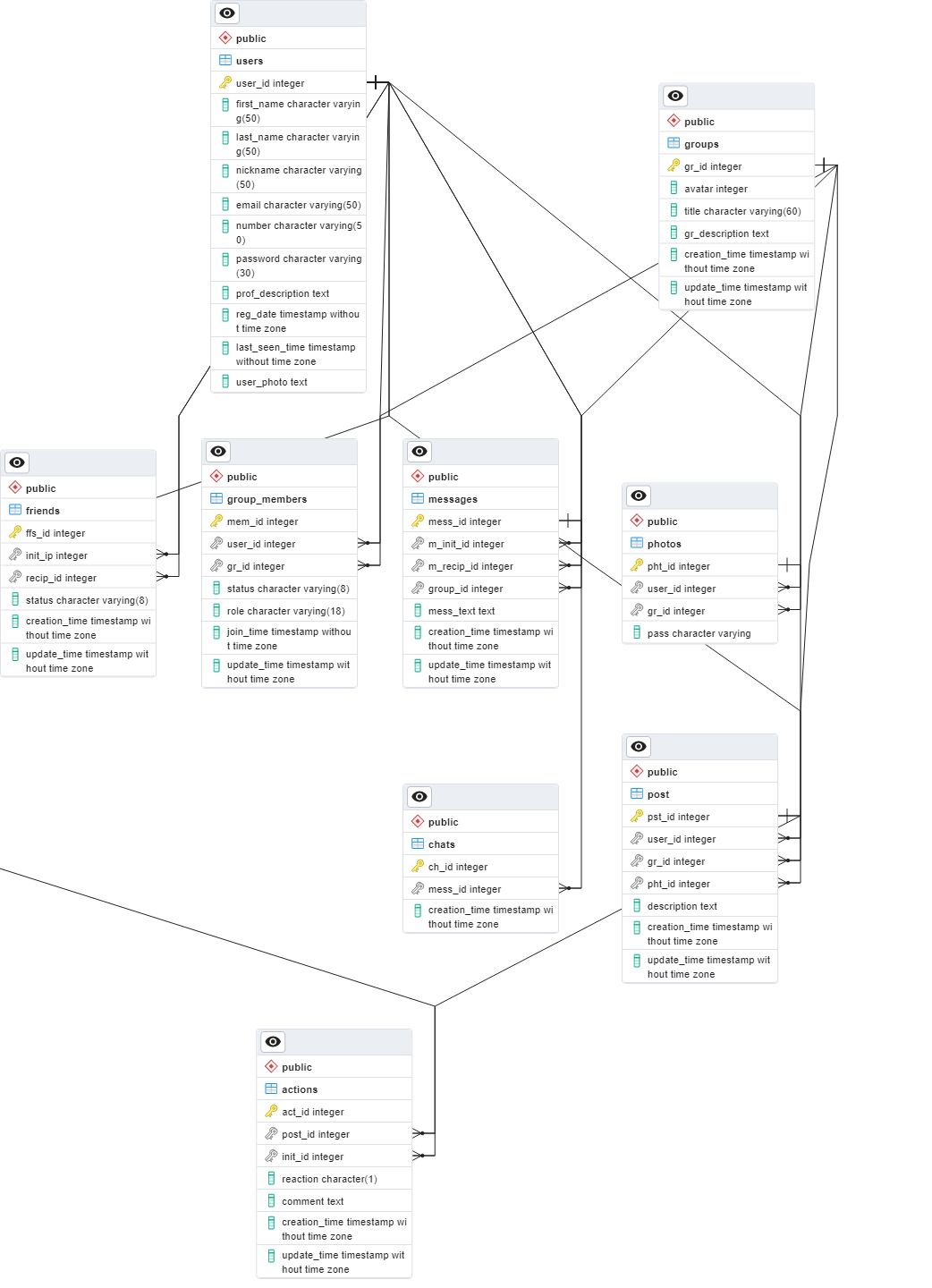


Рис. 4.1 Реляційна модель бази даних

# 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

# РОБОТА З БАЗОЮ ДАНИХ

СТВОРЮЄМО КОРИСТУВАЧІВ

create user Mary with password 'MaryMary';

create user adinistrator with password 'AdminAdmin';

create user moderator with password 'AdminAdmin';

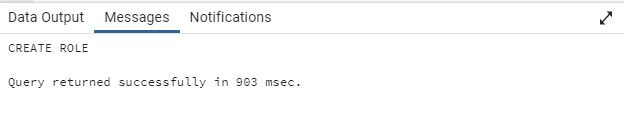


Рис 6.1

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;

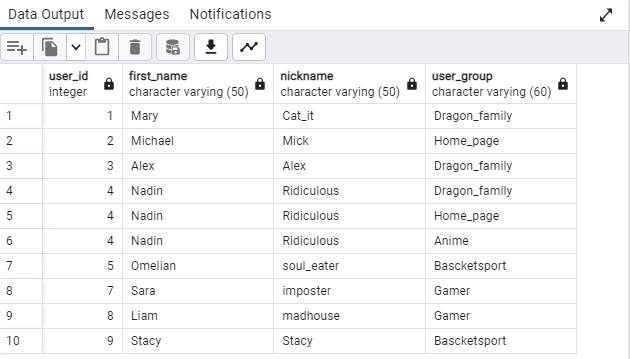


Рис 6.2

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

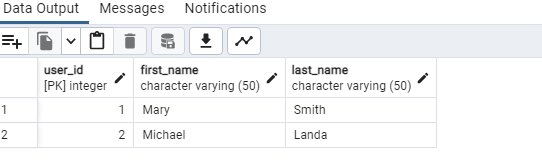
AND group\_members.gr\_id BETWEEN 1 AND 4;  


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;

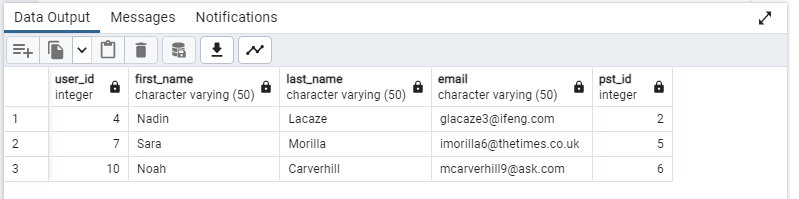


Рис 6.4

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;

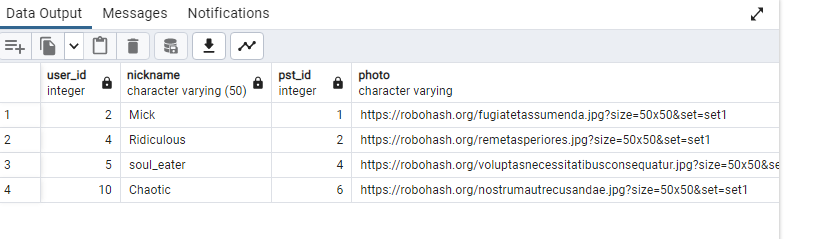


Рис 6.5

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';

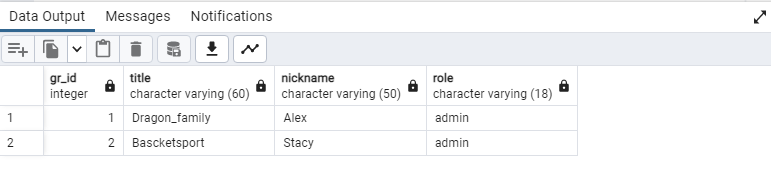


Рис 6.6

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;

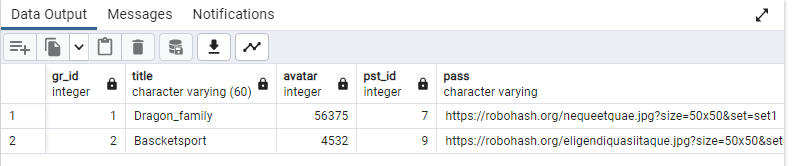


Рис 6.7

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’єдання

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;

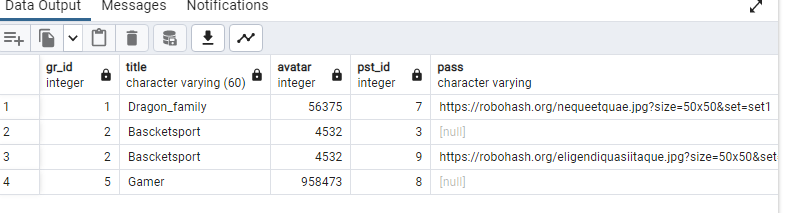


Рис 6.8

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’єдання, 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;

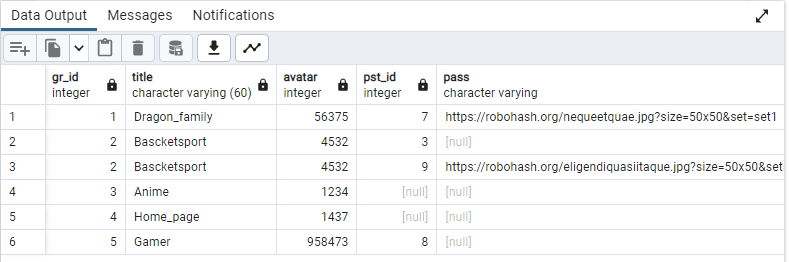


Рис 6.9

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;

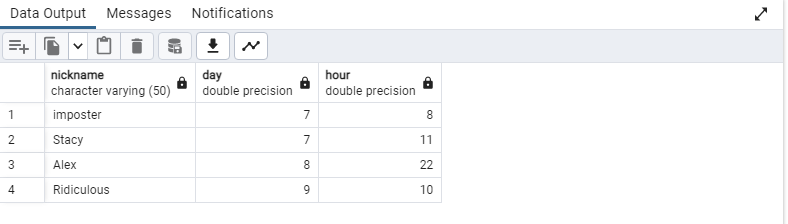


Рис 6.1.1

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;

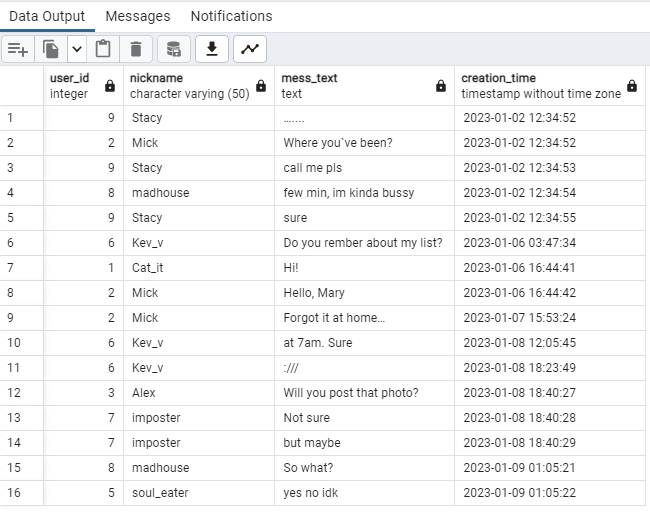


Рис 6.1.2

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;

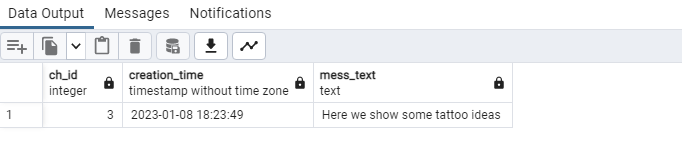


Рис 6.1.3

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.init\_id = users.user\_id;



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';

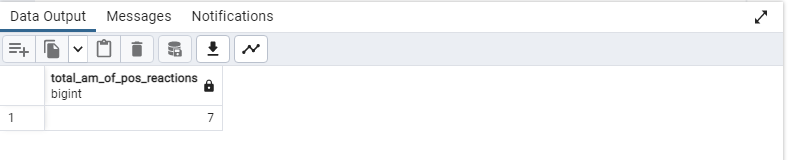


Рис 6.1.5

Загальну кількість коментарів

SELECT COUNT(comment) AS total\_am\_of\_pos\_reactions FROM actions

WHERE comment IS NOT NULL;

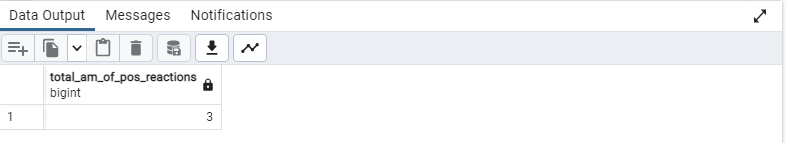


Рис 6.1.6

15) Обираємо інформацію про користувачів, що не є учасниками груп взагалі

SELECT \* FROM users WHERE user\_id NOT IN

(SELECT user\_id FROM group\_members);

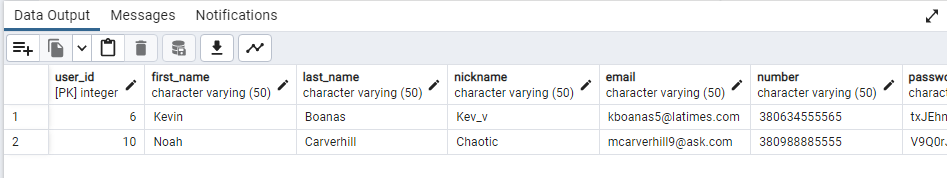


Рис 6.1.7

16)Відображаємо повідомлення що були оримані користувачами з 1 ночі по 5 ранку

SELECT messages.mess\_text, users.nickname AS recip, messages.creation\_time

FROM messages, users

WHERE messages.m\_recip\_id = users.user\_id

AND DATE\_PART('hour', messages.creation\_time) BETWEEN 1 AND 5;

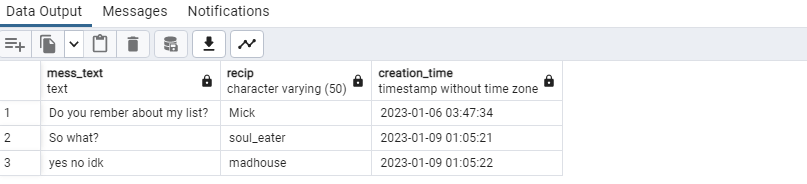


Рис 6.1.8

17)Відображаємо текстові пости груп

SELECT post.description, groups.title FROM post, groups

WHERE post.gr\_id = groups.gr\_id

AND description is NOT NULL;

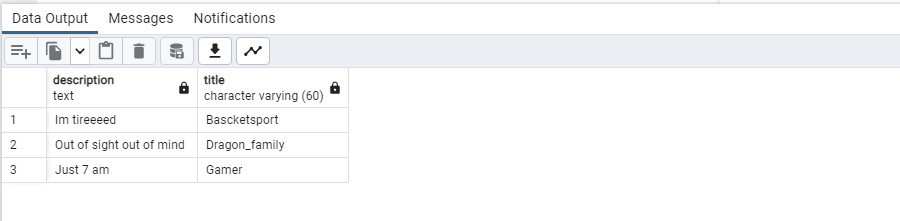


Рис 6.1.9

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;

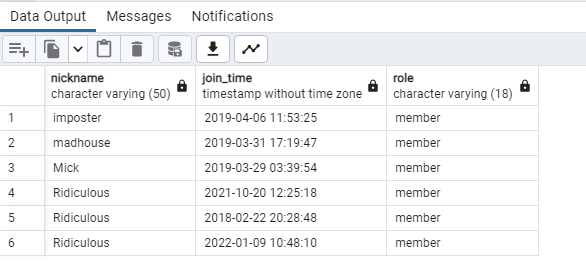


Рис 6.1.1.1

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

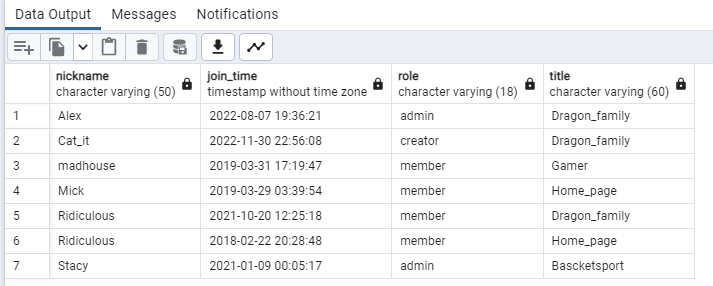


Рис 6.1.1.2

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;



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

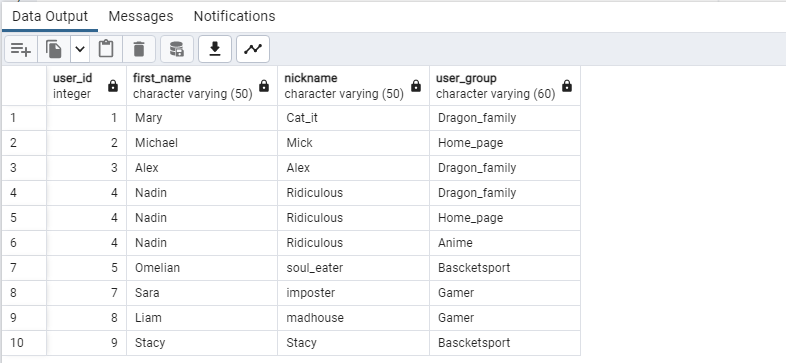


Рис 6.2.2

1. 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;

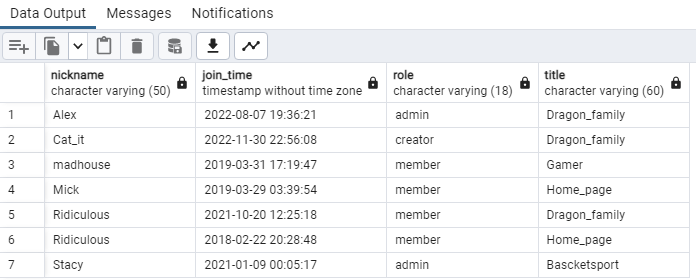


Рис 6.2.3

1. 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;

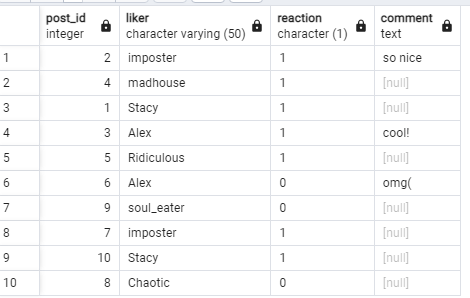


Рис 6.2.4

1. 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;

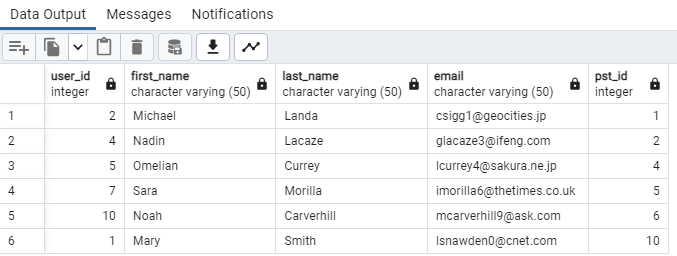


Рис 6.2.5

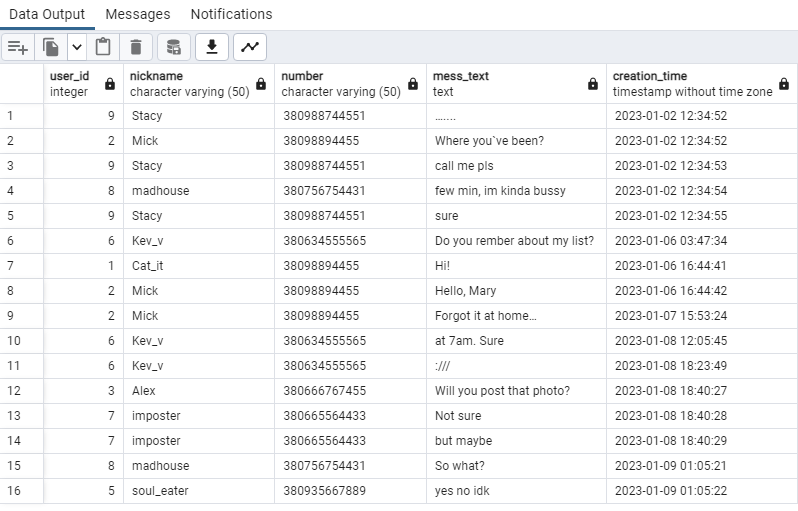
5)Створюємо подання для відображення останніх повідомлень користувачів

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;

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*();

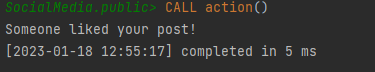


Рис 6.3.3.1

1. 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*();

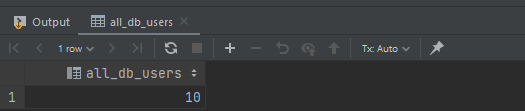


Рис 6.3.3.2

1. Створюємо керувальну структуру для обходу записів в базі даних

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

1. 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;

1. 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*();

1. 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;

1. 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*();

1. 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;

1. 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

