

## **Бази даних та інформаційні системи**

### **ЛАБОРАТОРНА РОБОТА №7**

#### **Користувачькі функції на мові запитів SQL**

Виконав:

Студент Процьків Назарій

Група ПМі-21

Оцінка - \_\_\_\_\_

Перевірила:

доц. Малець Р.Б.

**Тема:** Користувацькі функції на мові запитів SQL.

**Мета роботи:** Ознайомлення з поняттям користувацьких функцій на мові запитів SQL, їх створенням та застосуванням.

### Завдання (Варіант 17):

Розробити базу даних для сайту соціальної мережі. Соціальна мережа підтримує реєстрацію користувачів зі збереженням усіх їхніх деталей (ім'я, прізвище, дата народження, місце проживання, телефони, сайти/е-mail/skype іт.д.), місця і періоди перебування, місця і періоди навчання, роботи, служби, приєднані файли (зображення, фільми, аудіо), які можна пов'язувати із місцями з деталей. Крім того, кожен користувач має можливість розміщувати свої повідомлення на власній сторінці, отримувати на повідомлення «лайки» та коментарі, а також додавати інших користувачів в друзі, або в «чорний список». Додатково користувач повинен мати змогу шукати нових друзів за довільними критеріями.

### Хід роботи

1. Опрацював теоретичний матеріал.
2. Відповідно до свого завдання написав 4 користувацькі функції на мові запитів SQL. Виконання функцій і результати проілюструвати скрінами, а також поясненнями роботи функції. Функції мають бути різних типів - з базовими типами чи складними типами, які повертають стрічку таблиці чи всю таблицю (TABLE), і т.п.

#### Функція №1 (Повертає стрічку):

Функція приймає два параметри: ім'я користувача та дату. Повертає стрічку, в якій міститься інформація про те, де користувач з таким іменем служив станом на цю дату.

```
Query  Query History
1  create function get_service_place_from_date(firstname_of_user text, date_to_check date)
2      returns text
3      language plpgsql
4  as
5  $$
6  declare
7      to_return text;
8  begin
9      select service into to_return
10     from serviceplaces where serviceid =
11         (select "servicePlace"
12          from users
13          where firstname = firstname_of_user) and (startdate < date_to_check and date_to_check < enddate);
14     return to_return;
15 end;
16 $$;
```

#### Пояснення:

get\_service\_place\_from\_date – назва функції. Після неї в дужках написані параметри, які вона буде приймати. Слово returns вказує на те, що буде повертати функція. Слово declare створює змінну яку присвоїться результат query. Між begin та end я написав query, яка буде дізнаватись де саме служив користувач з таким іменем на таку дату.

Результат роботи функції:

Query

Query History

1

select get\_service\_place\_from\_date('Nazarii', '2020-08-21')

Data Output

Messages

Notifications

get\_service\_place\_from\_date

text

1

LvivService

Користувач з іменем Nazarii станом на 21 серпня 2020 року служив у Львові.

Функція №2 (Повертає таблицю):

Функція не приймає параметрів. Повертає таблицю, в якій міститься інформація про те, скільки сумарно лайків отримало кожне повідомлення в соціальній мережі.

Query	Query History
1	<code>create function get_likes_on_messages()</code>
2	<code>returns table (</code>
3	<code>msg_id messageid_check,</code>
4	<code>likes_count bigint</code>
5	<code>)</code>
6	<code>language plpgsql</code>
7	<code>as</code>
8	<code>\$\$</code>
9	<code>begin</code>
10	<code>return query</code>
11	<code>select message_id, count(*) as count_of_likes</code>
12	<code>from likes</code>
13	<code>group by message_id;</code>
14	<code>end;</code>
15	<code>\$\$;</code>

Пояснення:

get\_likes\_on\_messages – назва функції. Після неї в дужках нічого не написано, бо вона не приймає параметрів. Слово returns вказує на те, що буде повертати функція. В даному випадку це є дві колонки. Між begin та end я написав query, яка буде дізнаватись сумарну кількість лайків на кожне повідомлення.

## Результат роботи функції:

Query	Query History																		
<pre>1 select get_likes_on_messages()</pre>																			
Data Output	Messages																		
<div>get_likes_on_messages record</div> <table><tr><td>1</td><td>(#31,3)</td></tr><tr><td>2</td><td>(#21,2)</td></tr><tr><td>3</td><td>(#22,2)</td></tr><tr><td>4</td><td>(#51,3)</td></tr><tr><td>5</td><td>(#55,1)</td></tr><tr><td>6</td><td>(#44,1)</td></tr><tr><td>7</td><td>(#11,2)</td></tr><tr><td>8</td><td>(#41,2)</td></tr><tr><td>9</td><td>(#33,1)</td></tr></table>		1	(#31,3)	2	(#21,2)	3	(#22,2)	4	(#51,3)	5	(#55,1)	6	(#44,1)	7	(#11,2)	8	(#41,2)	9	(#33,1)
1	(#31,3)																		
2	(#21,2)																		
3	(#22,2)																		
4	(#51,3)																		
5	(#55,1)																		
6	(#44,1)																		
7	(#11,2)																		
8	(#41,2)																		
9	(#33,1)																		

Повідомлення з id #31 отримало 3 лайки. Повідомлення з id #21 отримало 2 лайки. І так далі.

## Функція №3 (Нічого не повертає):

Функція приймає два параметри. Оновлює ім'я користувача зі старого на нове.

Query	Query History
<pre>1 create function change_firstname(user_id_to_change user_id, new_firstname text) 2 returns void as 3 \$\$ 4 update users 5 set firstname = new_firstname 6 where id = user_id_to_change; 7 \$\$ 8 language sql;</pre>	

## Пояснення:

change\_firstname – назва функції. Після неї в дужках user\_id\_to\_change – параметр типу домену id користувача, ім'я якого потрібно замінити. Другий параметр new\_firstname – нове ім'я, яке потрібно записати замість старого. returns void вказує на те, що буде функція нічого повертати не буде.

## Результат роботи функції:

### До застосування функції:

Query

Query History

1

select \* from users

Data Output

Messages

Notifications

	id [PK] bigint	firstname text	lastname text	birthdate date	phonenumber text	password text	address text	workPlace text	studyPlace text	servicePlace text	beingPlace text
1	1	Nazarii	Protskiv	2003-12-29	+380963566929	qwerty123456	Lviv, Chervonoi Kalyny 58B	work1	study1	service1	being1
2	2	Oleksandr	Zhenchenko	2003-12-28	+380683838806	sanya123456	Lviv, Hnata Hotkevycha 1	work2	study2	service2	being2
3	3	Olena	Hatala	2004-07-23	+380993740427	olena123456	Velykyi Lubin 1	work3	study3	service3	being3
4	4	Sofia	Hoshko	2005-12-03	+380685435516	sofia123456	Dubliany 1	work4	study4	service4	being4
5	5	Yarema	Tymchyslyn	2004-08-20	+380959388072	yarema123456	Lutsk 1	work5	study5	service5	being5
6	6	bad	user	2002-10-21	+380954358972	baduser123456	Lviv, Shevchenka 1	work6	study6	service6	being6
7	7	7	7	2001-08-01	+380421505772	password7	address7	work7	study7	service7	being7
8	8	8	8	2007-01-14	+380625648721	password8	address8	work8	study8	service8	being8
9	9	9	9	2001-08-10	+380761049381	password9	address9	work9	study9	service9	being9
10	10	10	10	2001-12-16	+380716615141	password10	address10	work10	study10	service10	being10

Уявімо, що Назарій захотів змінити ім'я.

Query

Query History

1

select change\_firstname(1, 'PABLO');

2

select \* from users

Data Output

Messages

Notifications

+

📄

▼

📁

🗑️

🗄️

⬇️

📈

	id [PK] bigint	firstname text	lastname text	birthdate date	phonenumber text	password text	address text	workPlace text	studyPlace text	servicePlace text	beingPlace text
1	1	PABLO	Protskiv	2003-12-29	+380963566929	qwerty123456	Lviv, Chervonoi Kalyny 58B	work1	study1	service1	being1
2	2	Oleksandr	Zhenchenko	2003-12-28	+380683838806	sanya123456	Lviv, Hnata Hotkevycha 1	work2	study2	service2	being2
3	3	Olena	Hatala	2004-07-23	+380993740427	olena123456	Velykyi Lubin 1	work3	study3	service3	being3
4	4	Sofiia	Hoshko	2005-12-03	+380685435516	sofia123456	Dubliany 1	work4	study4	service4	being4
5	5	Yarema	Tymchyslyn	2004-08-20	+380959388072	yarema123456	Lutsik 1	work5	study5	service5	being5
6	6	bad	user	2002-10-21	+380954358972	baduser123456	Lviv, Shevchenka 1	work6	study6	service6	being6
7	7	7	7	2001-08-01	+380421505772	password7	address7	work7	study7	service7	being7

Функція приймає два параметри. Видаляє користувача з чорного списку іншого користувача.

```
Query    Query History
1  create function remove_from_blacklist(first_user_id user_id, second_user_id user_id)
2  returns void as
3  $$
4  delete from blacklist
5  where userid = first_user_id and enemyid = second_user_id
6  $$
7  language sql;
```

`remove_from_blacklist` — название функции. После неё в круглых скобках `first, user_id` — параметры

By  $\tilde{\gamma} = 1 - \gamma$  and  $\tilde{\beta} = 1 - \beta$ , we have

До застосування функції:

Query Query History

	Query
1	<code>select * from blacklist</code>

Query

Query History

1

select \* from blacklist

Data Output

Messages

Notifications

+

📄

▼

📋

🗑️

🗄️

⬇️

✍️

	userid bigint	🔒	enemyid bigint	🔒
1		1		6
2		2		6
3		3		6
4		4		6
5		5		6
6		6		1
7		6		2
8		6		3
9		6		4
10		6		5
11		3		5

Користувач з id 1 додав до чорного списку користувача з id 6.  
Користувач з id 2 додав до чорного списку користувача з id 6.  
І так далі.

Після застосування функції:

Наприклад користувач з id 1 захотів розблокувати користувача з id 6.

Query

Query History

1

select remove\_from\_blacklist(1, 6)

2


select \* from blacklist


Data Output


Messages


Notifications


≡+


▼













	userid bigint		enemyid bigint	
1		2		6
2		3		6
3		4		6
4		5		6
5		6		1
6		6		2
7		6		3
8		6		4
9		6		5
10		3		5
11		3		4

Висновок: ознайомився з поняттям користувацьких функцій на мові запитів SQL, їх створенням та застосуванням. Написав власні 4 функції, які можуть використовуватись в реальних базах даних соціальних мереж.