Отчет

по лабораторной работе №5 «Процедуры, функции, триггеры в PostgreSQL»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Беломытцев А.И.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.

ЙТМО

Санкт-Петербург 2023

Оглавление

Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql).

Практическое задание:

Вариант 2 (max - 8 баллов)

1. Создать процедуры/функции.

2. Создать авторский триггер.

Выполнение

Создание хранимых процедур:

Процедура добавления канала с преобразованием url в id канала

```sql
create procedure insert_into_channel_by_url(url char(56), lang
char(2), category varchar(50), theme varchar(50), user_name
varchar(50))

language sql

as $$

 insert into channel(channel_id, url, lang, category, theme,
fk_user_id)

 values (substring(url from 33 for 56), url, lang, category,
theme, (select user_id from username where username =
user_name))

$$;

call
insert_into_channel_by_url('https://www.youtube.com/channel/UCP1
itCwGbXmHIsNrCIm1Hsw', 'ru', 'edu', 'all', 'andrei');

select * from channel where channel_id =
'UCP1itCwGbXmHIsNrCIm1Hsw';
```

Процедура добавления канала с преобразованием id канала в url

```sql
create procedure insert_into_channel_by_id(channel_id char(24),
lang char(2), category varchar(50), theme varchar(50), user_name
varchar(50))

language sql

as $$

 insert into channel(channel_id, url, lang, category, theme,
fk_user_id)

 values (channel_id, 'https://www.youtube.com/channel/' ||
channel_id, lang, category, theme, (select user_id from username
where username = user_name))

$$;

call insert_into_channel_by_id('UCWcZz4hOVcrPIcUYNoqsRTQ', 'ru',
'edu', 'all', 'andrei');

select * from channel where url =
'https://www.youtube.com/channel/UCWcZz4hOVcrPIcUYNoqsRTQ';
```

```
1  create procedure insert_into_channel_by_id(channel_id char(24), lang char(2), category varchar(50), theme varchar(50), user_name var
2  language sql
3  as $$
4    insert into channel(channel_id, url, lang, category, theme, fk_user_id)
5    values (channel_id, 'https://www.youtube.com/channel/' || channel_id, lang, category, theme, (select user_id from username where u
6  $$;
7  call insert_into_channel_by_id('UCWcZz4hOVcrPIcUYNoqsRTQ', 'ru', 'edu', 'all', 'andrei');
8  select * from channel where url = 'https://www.youtube.com/channel/UCWcZz4hOVcrPIcUYNoqsRTQ';
```

Data Output   Messages   Notifications

channel_id [PK] character	url character	title character varying (100)	views bigint	subs integer	videos integer	lang character	category character va
1	UCWcZz4hOVcrPIcUYNoqsRTQ	https://www.youtube.com/channel/UCWcZz4hOVcrPIcUYNoqsR...	[null]	[null]	[null]	[null]	ru	edu

Процедура добавления канала с преобразованием url в id канала, а id канала в url

```
create procedure insert_into_channel(channel_id_or_url
varchar(56), lang char(2), category varchar(50), theme
varchar(50), user_name varchar(50))

language plpgsql

as $$ begin

 if length(channel_id_or_url) = 24 then

   insert into channel(channel_id, url, lang, category, theme,
fk_user_id)

   values (channel_id_or_url, 'https://www.youtube.com/channel/'
|| channel_id_or_url, lang, category, theme, (select user_id
from username where username = user_name));

 elsif length(channel_id_or_url) = 56 then

   insert into channel(channel_id, url, lang, category, theme,
fk_user_id)

   values (substring(channel_id_or_url from 33 for 56),
channel_id_or_url, lang, category, theme, (select user_id from
username where username = user_name));

 else

   raise exception 'wrong length of channel id or url';

 end if;
```

```sql
end $$;

call insert_into_channel('https://www.youtube.com/channel/UCz9tNkbNLP8HoZipU6KX7NA', 'en', 'edu', 'it', 'andrei');

select * from channel where channel_id = 'UCz9tNkbNLP8HoZipU6KX7NA';

call insert_into_channel('UCaDZLsDEEBBJESXmmi-9m5w', 'en', 'edu', 'it', 'andrei');

select * from channel where url = 'https://www.youtube.com/channel/UCaDZLsDEEBBJESXmmi-9m5w';

call insert_into_channel('itsmorethanuniversity', 'en', 'edu', 'it', 'andrei');
```

Query   Query History

```sql
1  create procedure insert_into_channel(channel_id_or_url varchar(56), lang char(2), category varchar(50), theme varchar(50), user_name
2  language plpgsql
3  as $$ begin
4    if length(channel_id_or_url) = 24 then
5      insert into channel(channel_id, url, lang, category, theme, fk_user_id)
6      values (channel_id_or_url, 'https://www.youtube.com/channel/' || channel_id_or_url, lang, category, theme, (select user_id from
7    elsif length(channel_id_or_url) = 56 then
8      insert into channel(channel_id, url, lang, category, theme, fk_user_id)
9      values (substring(channel_id_or_url from 33 for 56), channel_id_or_url, lang, category, theme, (select user_id from username whe
10   else
11     raise exception 'wrong length of channel id or url';
12   end if;
13 end $$;
14 call insert_into_channel('https://www.youtube.com/channel/UCz9tNkbNLP8HoZipU6KX7NA', 'en', 'edu', 'it', 'andrei');
15 select * from channel where channel_id = 'UCz9tNkbNLP8HoZipU6KX7NA';
```

Data Output   Messages   Notifications

	channel_id [PK] character	url character	title character varying (100)	views bigint	subs integer	videos integer	lang character	category character va
1	UCz9tNkbNLP8HoZipU6KX7NA	https://www.youtube.com/channel/UCz9tNkbNLP8HoZipU6KX7...	[null]	[null]	[null]	[null]	en	edu

Query   Query History

```sql
1  call insert_into_channel('UCaDZLsDEEBBJESXmmi-9m5w', 'en', 'edu', 'it', 'andrei');
2  select * from channel where url = 'https://www.youtube.com/channel/UCaDZLsDEEBBJESXmmi-9m5w';
```

Data Output   Messages   Notifications

	channel_id [PK] character	url character	title character varying (100)	views bigint	subs integer	videos integer	lang character	category character
1	UCaDZLsDEEBBJESXmmi-9m5w	https://www.youtube.com/channel/UCaDZLsDEEBBJESXmmi-9m5...	[null]	[null]	[null]	[null]	en	edu

Query   Query History

```sql
1  call insert_into_channel('itsmorethanuniversity', 'en', 'edu', 'it', 'andrei');
```

Data Output   Messages   Notifications

```
ERROR:  wrong length of channel id or url
CONTEXT:  PL/pgSQL function insert_into_channel(character varying,character,character varying,character varying,character varying) line 9 at RAISE

SQL state: P0001
```

Процедура бана пользователя по нику. Замена роли на 'ban', удаление браслетов, заказов, статей (не подтверждённых), комментариев, песен (не подтверждённых), каналов (не подтверждённых), сайтов (не подтверждённых) пользователя

```
create procedure ban(user_name varchar(50))
language plpgsql
as $$
declare
 user_id int = (select user_id from username where username =
user_name);
begin
 update username set fk_role_id = (
    select role_id from role where name = 'ban'
 ) where username = user_name;
 delete from mode where fk_user_id = (select band_id from band
where fk_user_id = user_id);
 delete from schedule where fk_band_id = (select band_id from
band where fk_user_id = user_id);
 delete from extension where fk_user_id = (select band_id from
band where fk_user_id = user_id);
 delete from band where fk_user_id = user_id;
 delete from "order" where fk_user_id = user_id;
 delete from article where fk_user_id = user_id and approved is
not true;
 delete from comment where fk_user_id = user_id;
 delete from song where fk_user_id = user_id and approved is not
true;
```

```sql
 delete from channel where fk_user_id = user_id and approved is
not true;

 delete from website where fk_user_id = user_id and approved is
not true;

end;

$$;

call ban('sasha');



select * from band where fk_user_id = (select user_id from
username where username = 'sasha');

select * from comment where fk_user_id = (select user_id from
username where username = 'sasha');

select * from song where fk_user_id = (select user_id from
username where username = 'sasha');

select * from website where fk_user_id = (select user_id from
username where username = 'sasha');
```

Query   Query History

| 1 | select * from band where fk_user_id = (select user_id from username where username = 'sasha'); |

Data Output   Messages   Notifications

	band_id [PK] integer	fk_user_id integer	apikey character	model character varying (10)	version character varying (10)	last_time time without time zone
1	5	5	ICb0OIyAN9x35t8AdM9j	2.2	0.2	[null]

Query   Query History

| 1 | select * from comment where fk_user_id = (select user_id from username where username = 'sasha'); |

Data Output   Messages   Notifications

	comment_id [PK] integer	fk_article_id integer	fk_user_id integer	create_date date	text text
1	4	1	5	2023-12-24	Кринж
2	5	2	5	2023-12-24	Криинж
3	6	3	5	2023-12-25	Кринж
4	7	4	5	2023-12-25	Кринжовый кринж
5	8	5	5	2023-12-26	Кринж
6	9	6	5	2023-12-26	Кринжовый кринж

```sql
1  select * from song where fk_user_id = (select user_id from username where username = 'sasha');
```

Data Output   Messages   Notifications

	song_id [PK] integer	fk_user_id integer	create_date date	approved boolean	fk_author_id integer	song_name character varying (100)	fk_album_id integer	fk_genre_id integer	lang character	year smallint	youtube_linl character va
1	23	5	2023-10-24	true	2	История России в 30 стилях рэпа	[null]	5	ru	2020	https://www
2	24	5	2023-10-24	true	2	Всемирная история в 30 стилях рэпа	[null]	5	ru	2021	https://www

Query   Query History

```sql
1  select * from website where fk_user_id = (select user_id from username where username = 'sasha');
```

Data Output   Messages   Notifications

	website_id [PK] integer	fk_user_id integer	approved boolean	url character varying (100)	description text
1	11	5	true	codecademy.com	Обучение программированию
2	12	5	true	sololearn.com	Обучение программированию
3	13	5	true	quizlet.com	Карточки для изучения
4	14	5	false	randomnerdtutorials.com	Сайт с туториалами по ESP32 и ESP8266
5	15	5	false	readthedocs.io	Документации

Query   Query History

```sql
1   create procedure ban(user_name varchar(50))
2   language plpgsql
3   as $$
4   declare
5     user_id int = (select user_id from username where username = user_name);
6   begin
7     update username set fk_role_id = (
8       select role_id from role where name = 'ban'
9     ) where username = user_name;
10    delete from mode where fk_user_id = (select band_id from band where fk_user_id = user_id);
11    delete from schedule where fk_band_id = (select band_id from band where fk_user_id = user_id);
12    delete from extension where fk_user_id = (select band_id from band where fk_user_id = user_id);
13    delete from band where fk_user_id = user_id;
14    delete from "order" where fk_user_id = user_id;
15    delete from article where fk_user_id = user_id and approved is not true;
16    delete from comment where fk_user_id = user_id;
17    delete from song where fk_user_id = user_id and approved is not true;
18    delete from channel where fk_user_id = user_id and approved is not true;
19    delete from website where fk_user_id = user_id and approved is not true;
20  end;
21  $$;
22  call ban('sasha');
```

Data Output   Messages   Notifications

```
CALL

Query returned successfully in 102 msec.
```

Query   Query History

```sql
1  select * from band where fk_user_id = (select user_id from username where username = 'sasha');
```

Data Output   Messages   Notifications

	band_id [PK] integer	fk_user_id integer	apikey character	model character varying (10)	version character varying (10)	last_time time without time zone

Удалить все неподтверждённые статьи, песни, каналы, сайты

```
create procedure delete_not_approved()

language sql

as $$

 delete from article where not approved;

 delete from song where not approved;

 delete from channel where not approved;

 delete from website where not approved;

$$;

call delete_not_approved();

select * from channel where not approved;
```

```
Query   Query History

1   select * from channel where not approved;
```

Data Output   Messages   Notifications

	channel_id [PK] character	url character	
1	UCP1itCwGbXmHIsNrClm1Hsw	https://www.youtube.com/channel/UCP1itCwGbXmHIsNrClm1Hsw	
2	UCWcZz4hOVcrPIcUYNoqsRTQ	https://www.youtube.com/channel/UCWcZz4hOVcrPIcUYNoqsRTQ	
3	UCz9tNkbNLP8HoZipU6KX7NA	https://www.youtube.com/channel/UCz9tNkbNLP8HoZipU6KX7NA	
4	UCaDZLsDEEBBJESXmmi-9m5w	https://www.youtube.com/channel/UCaDZLsDEEBBJESXmmi-9m5...	

```
Query   Query History

1   create procedure delete_not_approved()
2   language sql
3   as $$
4     delete from article where not approved;
5     delete from song where not approved;
6     delete from channel where not approved;
7     delete from website where not approved;
8   $$;
9   call delete_not_approved();
10  select * from channel where not approved;
```

Data Output   Messages   Notifications

	channel_id [PK] character	url character	title character varying (100)	views bigint

Триггер, который автоматически повышает пользователя до vip, если он набрал 15 баллов. По 1 баллу за одобренный канал, сайт, песню; 5 баллов за статью.

```
create procedure up_user()

language plpgsql

as $$

 update username

 set fk_role_id = (select role_id from role where name = 'vip')

 where user_id = (select t.fk_user_id from

    (

       select fk_user_id, count(*) from channel

       where approved is true

       group by fk_user_id
```

```sql
        union

        select fk_user_id, count(*) from website

        where approved is true

        group by fk_user_id

        union

        select fk_user_id, count(*) from song

        where approved is true

        group by fk_user_id

        union

        select fk_user_id, count(*) * 5 from article

        where approved is true

        group by fk_user_id

    ) as t

    inner join username on t.fk_user_id = username.user_id

    inner join role on username.fk_role_id = role.role_id

    where role.name = 'user'

    group by fk_user_id

    having sum(t.count) >= 15

);

$$;


create trigger up_user_channel

after update on channel

for each row
```

```sql
execute procedure up_user();


create trigger up_user_website

after update on website

execute procedure up_user();


create trigger up_user_song

after update on song

execute procedure up_user();


create trigger up_user_article

after update on article

execute procedure up_user();


select username, name from username inner join role on
username.fk_role_id = role.role_id;


insert into article(fk_user_id, title, create_date, approved,
text)

values (2, 'Процедуры, функции, триггеры в PostgreSQL',
'2024-01-01', true, 'Цель работы: овладеть практическими
создания и использования процедур, функций и триггеров в базе
данных PostgreSQL.\nOборудование: компьютерный
класс.\nПрограммное обеспечение: СУБД PostgreSQL, SQL Shell
(psql).');
```

Query   Query History

```
1  select username, name from username inner join role on username.fk_role_id = role.role_id;
```

Data Output   Messages   Notifications

	username character varying (50) 🔒	name character varying (20) 🔒
1	andrei	admin
2	dima	user
3	alexey	user
4	alisa	user
5	sasha	ban

Query   Query History

```
 1  create function up_user() returns trigger
 2  language plpgsql
 3  as $$ begin
 4    update username
 5    set fk_role_id = (select role_id from role where name = 'vip')
 6    where user_id = (select t.fk_user_id from
 7      (
 8        select fk_user_id, count(*) from channel
 9        where approved is true
10        group by fk_user_id
11        union
12        select fk_user_id, count(*) from website
13        where approved is true
14        group by fk_user_id
15        union
16        select fk_user_id, count(*) from song
17        where approved is true
18        group by fk_user_id
19        union
20        select fk_user_id, count(*) * 5 from article
```

Data Output   Messages   Notifications

	user_id integer	username character varying (50) 🔒	name character varying (20) 🔒
1	1	andrei	admin
2	4	dima	user
3	3	alexey	user
4	2	alisa	vip
5	5	sasha	ban

Вывод

В ходе лабораторной работы освоена работа с процедурами, функциями и триггерами.