

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

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

по дисциплине «Проектирование и реализация баз данных»

Автор: Гусев Н.С.

Факультет: ИКТ

Группа: K3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

Выполнение.....	3
Вывод .....	9

**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

```
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
itCwGbXmHIsNrCImlHsw', 'ru', 'edu', 'all', 'nikita');

select * from channel where channel_id =
'UCP1itCwGbXmHIsNrCImlHsw';
```

QueryQuery History

```

1 create procedure insert_into_channel_by_url(url char(56), lang char(2), category varchar(50), theme varchar(50), user_name varchar(50))
2 language sql
3 as $$
4     insert into channel(channel_id, url, lang, category, theme, fk_user_id)
5     values (substring(url from 33 for 56), url, lang, category, theme, (select user_id from username where username = user_name))
6 $$;
7 call insert_into_channel_by_url('https://www.youtube.com/channel/UCPlitCwGbXmHIsNrCIm1Hsw', 'ru', 'edu', 'all', 'andrei');
8 select * from channel where channel_id = 'UCPlitCwGbXmHIsNrCIm1Hsw';

```

Data OutputMessagesNotifications

	channelId	url	title	views	subs	videos	lang	category
	[PK] character	character	character varying (100)	bigint	integer	integer	character	character varying (50)
1	UCPlitCwGbXmHIsNrCIm1Hsw	https://www.youtube.com/channel/UCPlitCwGbXmHIsNrCIm1Hsw	[null]	[null]	[null]	[null]	ru	edu

```

razvit=# create procedure insert_into_channel_by_url(url char(56), lang char(2), category varchar(50), theme varchar(50), user_name varchar(50))
razvit=# language sql
razvit=# as $$
razvit$# insert into channel(channel_id, url, lang, category, theme, fk_user_id)
razvit$# values (substring(url from 33 for 56), url, lang, category, theme, (select user_id from username where username = user_name))
razvit$# $$;
CREATE PROCEDURE
CALL
razvit=# call insert_into_channel_by_url('https://www.youtube.com/channel/UCPlitCwGbXmHIsNrCIm1Hsw', 'ru', 'edu', 'all', 'andrei');
CALL
razvit=# select * from channel where channel_id = 'UCPlitCwGbXmHIsNrCIm1Hsw';
channel_id | url | title | views | subs | videos | lang | category | theme | avatar_default | avatar_medium | avatar_high | fk_user_id | approved
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
UCPlitCwGbXmHIsNrCIm1Hsw | https://www.youtube.com/channel/UCPlitCwGbXmHIsNrCIm1Hsw |  |  |  |  | ru | edu | all |  |  |  |  | f
(1 row)

```

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 or replace function up_user() returns trigger

language plpgsql

as $$ begin

    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

            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

);

return null;

end; $$;

create trigger up_user_channel
after insert or update on channel
execute procedure up_user();

create trigger up_user_website
after insert or update on website
execute procedure up_user();

```

```
create trigger up_user_song
after insert or update on song
execute procedure up_user();

create trigger up_user_article
after insert or 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.\nОборудование: компьютерный
класс.\nПрограммное обеспечение: СУБД PostgreSQL, SQL Shell
(psql).');

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

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	nikita	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	nikita	admin
2	4	dima	user
3	3	alexey	user
4	2	alisa	vip
5	5	sasha	ban



```

razvit=# create or replace function up_user() returns trigger
razvit=# language plpgsql
razvit=# as $$ begin
razvit=#     update username
razvit=#     set fk_role_id = (select role_id from role where name = 'vip')
razvit=#     where user_id = (select t.fk_user_id from
razvit=#         (
razvit=#             select fk_user_id, count(*) from channel
razvit=#             where approved is true
razvit=#             group by fk_user_id
razvit=#             union
razvit=#             select fk_user_id, count(*) from website
razvit=#             where approved is true
razvit=#             group by fk_user_id
razvit=#             union
razvit=#             select fk_user_id, count(*) from song
razvit=#             where approved is true
razvit=#             group by fk_user_id
razvit=#             union
razvit=#             select fk_user_id, count(*) * 5 from article
razvit=#             where approved is true
razvit=#             group by fk_user_id
razvit=#         ) as t
razvit=#     inner join username on t.fk_user_id = username.user_id
razvit=#     inner join role on username.fk_role_id = role.role_id
razvit=#     where role.name = 'user'
razvit=#     group by fk_user_id
razvit=#     having sum(t.count) >= 15
razvit=# );
razvit=# return null;
razvit=# end; $$;
CREATE FUNCTION
razvit=# create trigger up_user_channel
razvit=# after insert or update on channel
razvit=# execute procedure up_user();
CREATE TRIGGER
razvit=# create trigger up_user_website
razvit=# after insert or update on website
razvit=# execute procedure up_user();
CREATE TRIGGER
razvit=# create trigger up_user_song
razvit=# after insert or update on song
razvit=# execute procedure up_user();
CREATE TRIGGER
razvit=# create trigger up_user_article
razvit=# after insert or update on article
razvit=# execute procedure up_user();
CREATE TRIGGER
razvit=# select username, name from username inner join role on username.fk_role_id = role.role_id;
  username | name
-----+-----
  nikita   | admin
  dima     | user
  alexey   | user
  alisa    | user
  sasha    | user
(5 rows)

razvit=# insert into article(fk_user_id, title, create_date, approved, text)
razvit=# values (2, 'Процедуры, функции, триггеры в PostgreSQL', '2024-01-01', true, 'Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в ба
зе данных PostgreSQL.\nОборудование: компьютерный класс.\nПрограммное обеспечение: СУБД PostgreSQL, SQL Shell (psql).');
INSERT 0 1
razvit=# select username, name from username inner join role on username.fk_role_id = role.role_id;
  username | name
-----+-----
  nikita   | admin
  alisa    | vip
  dima     | user
  alexey   | user
  sasha    | user
(5 rows)

```

## Вывод

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