

Санкт-Петербургский Политехнический Университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

## Базы данных

Отчет по лабораторной работе №6  
Sql программирование: Триггеры, вызовы процедур

**Работу**  
**выполнил:**  
Еременко Д.Ю.  
Группа: 33531/2  
**Преподаватель:**  
Мяснов А.В.

Санкт-Петербург  
2019

# Содержание

<b>1. Цель работы</b>	<b>2</b>
<b>2. Программа работы</b>	<b>2</b>
<b>3. Ход выполнения работы</b>	<b>2</b>
3.1. Общие ХП . . . . .	2
3.2. Индивидуальные ХП . . . . .	3
<b>4. Выводы</b>	<b>4</b>

# 1. Цель работы

Познакомить студентов с возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур и триггеров.

# 2. Программа работы

1. Создание двух триггеров: один триггер для автоматического заполнения ключевого поля, второй триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице.
2. Создание триггера в соответствии с индивидуальным заданием, полученным у преподавателя.
3. Создание триггера в соответствии с индивидуальным заданием, вызывающего хранимую процедуру.
4. Выкладывание скрипта с созданными сущностями в GitLab.
5. Демонстрация результатов преподавателю.

# 3. Ход выполнения работы

```
1  /*Триггер
2   для автоматического заполнения ключевого поля.
3  */
4
5  drop function if exists new_id() cascade;
6  create function new_id() returns trigger
7  as
8  $$
9  begin
10 select max(id) + 1 into new.id from refrigerator;
11 return new;
12 end;
13 $$ language plpgsql;
14
15 /*TODO maybe someday
16 select coalesce(min(nullif(r1.id - 1, r2.id)), max(r2.id) + 1) as min_free_id
17 from refrigerator r1
18 left join (select id from refrigerator) r2 on r1.id - 1 = r2.id;
19 */
20
21 create trigger new_id_trig
22 before insert
23 on refrigerator
24 for each row
25 execute procedure new_id();
26
27 /*Триггер
28 для контроля целостности данных в подчиненной таблице при удалении/изменении записей в
   ↪ главной таблице
29 */
30
31 drop function if exists delete_market_name() cascade;
```

```

32 create function delete_market_name() returns trigger
33 as
34 $$
35 declare
36 market_name_id_ integer;
37 begin
38 select id into market_name_id_ from market_name where name = old.name;
39 delete from refrigerator where market_name_id = market_name_id_;
40 return old;
41 end;
42 $$ language plpgsql;
43
44 create trigger check_delete
45 before update or delete
46 on market_name
47 for each row
48 execute procedure delete_market_name();
49
50
51 /*Автоматически
52   заполнять дату покупки текущей() при добавлении записей в холодильник.
53 */
54
55 drop function if exists fix_buying_date() cascade;
56
57 create function fix_buying_date() returns trigger
58 as
59 $$
60 declare
61 begin
62 new.buying_date = current_date;
63 return new;
64 end;
65 $$ language plpgsql;
66
67
68 create trigger fix_byuing_date_trig
69 before insert or update
70 on refrigerator
71 for each row
72 execute procedure fix_buying_date();
73
74
75 /*Проверять
76   дубли при добавлении продуктов в рецепт. При дублировании выбрасывать исключение.
77 */
78 drop function if exists check_repeat_in_recipe_product() cascade;
79 create function check_repeat_in_recipe_product() returns trigger
80 as
81 $$
82 begin
83 if new.product_id in (select product_id from recipe_product where recipe_id =
84   ↪ new.recipe_id)
85 then
86 RAISE EXCEPTION 'product_%_already_exist_in_recipe%',new.product_id,new.
87   ↪ recipe_id;
88 end if;
89 return null;
90 end;
91 $$ language plpgsql;

```

```
90|
91| create trigger check_repeat_in_recipe_product_trig
92| before insert or update
93| on recipe_product
94| for each row
95| execute procedure check_repeat_in_recipe_product();
```

## 4. Выводы

В ходе выполнения данной лабораторной работы был изучен язык создания триггеров и их связь с хранимыми процедурами.