

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение  
высшего образования  
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

## **Лабораторная работа №3**

# **«Процедуры, функции, триггеры в PostgreSQL»**

по дисциплине:

**«Проектирование и реализация баз данных»**

**Выполнила:**

студентка III курса ИКТ  
группы К33402  
Самчук Анита Алексеевна

**Проверила:**

Говорова Марина Михайловна

Санкт-Петербург  
2020

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

### Практическое задание:

1. Создать процедуры/функции согласно индивидуальному заданию часть 4.
2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

### Схема базы данных:



**Задание 1.** Создайте хранимые процедуры

1. Для повышения стипендии отличникам на 10%

До:

Query Editor

Query History

1

```
select * from "exams schema".scholarship
```

Data Output

Explain

Messages

Notifications

	id_scholarship [PK] integer	type_scholarship character varying	payment_start date	payment_end date	size_scholarship integer	id_student integer
1	3	ПГАС	2022-09-01	2022-12-31	10000	264920
2	4	ГСС	2022-09-01	2022-12-31	3000	308535
3	2	ГАС	2022-09-01	2022-12-31	2000	309670
4	1	ГАС отлично успевающего студента	2022-09-01	2022-12-31	4510	313264

Создание:

Query Editor

Query History

1

2

3

4

5

6

7

8

9

```
create procedure "exams schema".raise_scholarship()  
language sql  
as $$  
update "exams schema".scholarship  
set size_scholarship = size_scholarship * 1.1 where id_scholarship in  
(select id_scholarship from "exams schema".scholarship  
where type_scholarship = 'ГАС отлично успевающего студента' and  
current_timestamp between payment_start and payment_end);  
$$;
```

Data Output

Explain

Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 32 msec.

После:

Query Editor

Query History

1

2

```
call "exams schema".raise_scholarship();  
select * from "exams schema".scholarship;
```

Data Output

Explain

Messages

Notifications

	id_scholarship [PK] integer	type_scholarship character varying	payment_start date	payment_end date	size_scholarship integer	id_student integer
1	3	ПГАС	2022-09-01	2022-12-31	10000	264920
2	4	ГСС	2022-09-01	2022-12-31	3000	308535
3	2	ГАС	2022-09-01	2022-12-31	2000	309670
4	1	ГАС отлично успевающего студента	2022-09-01	2022-12-31	4961	313264

2. Для перевода студентов на следующий курс

До:

1

SELECT \* FROM "exams schema".study\_group

2

ORDER BY id\_group ASC

	id_group [PK] integer	number_group character varying	course integer	study_year character varying	id_direction integer
1	1	K33401	3	2022/2023	1
2	2	K3243	2	2022/2023	2
3	3	M3112	1	2022/2023	3
4	4	P34684	4	2022/2023	4
5	5	K4142	1	2022/2023	1

После:

Query Editor

Query History

1

create procedure "exams schema".update\_course()

2

language sql

3

as \$\$

4

update "exams schema".study\_group

5

set course = course + 1,

6

study\_year = concat(cast(substring(study\_year, 1, 4) as int)+1 , '/', cast(substring(study\_year, 6, 4) as int) + 1),

7

number\_group = concat(substring(number\_group, 1, 2),

8

cast(substring(number\_group, 3, 1) as int) + 1,

9

substring(number\_group, 4, 4))

10

where course < 4;

11

\$\$;

12

13

call "exams schema".update\_course();

14

select \* from "exams schema".study\_group;

	id_group [PK] integer	number_group character varying	course integer	study_year character varying	id_direction integer
1	4	P34684	4	2022/2023	4
2	1	K34401	4	2023/2024	1
3	2	K3343	3	2023/2024	2
4	3	M3212	2	2023/2024	3
5	5	K4242	2	2023/2024	1

### 3. Для изменения оценки при успешной пересдаче экзамена

До:

Query Editor Query History

```
1 SELECT * FROM "exams schema".passing_the_exam
2 ORDER BY id_exam ASC
```

Data Output Explain Messages Notifications

	id_exam [PK] integer	attempt_number integer	id_schedule integer	id_student integer	points integer	mark character varying	record_book_number integer
1	1	1	1	313264	20	5	5
2	2	1	2	309670	20	5	4
3	3	1	4	313264	20	5	5
4	4	1	3	282928	0	Незачет	2
5	5	1	5	264920	16	Зачет	1
6	6	1	6	247891	12	Зачет	8
7	7	1	4	282928	0	2	2
8	8	1	2	313264	18	4	5
9	9	1	3	264920	16	Зачет	1
10	10	1	1	309670	20	5	4
11	11	1	1	282928	0	2	2
12	12	3	7	282928	0	2	2
13	13	1	3	308535	20	Зачет	3

После:

Query Editor Query History

```
1 create procedure "exams schema".retake(schedule_id integer, points integer, mark integer, number_record_book integer)
2 language sql
3 as $$
4 insert into "exams schema".passing_the_exam(id_exam, attempt_number, id_student, id_schedule, points, mark, record_book_number)
5 values((select max(id_exam) from "exams schema".passing_the_exam) + 1,
6        (select max(attempt_number) from "exams schema".passing_the_exam
7         where id_schedule = schedule_id and record_book_number = number_record_book) + 1,
8        (select id_student from "exams schema".studying_student
9         where record_book_number = number_record_book),
10       schedule_id, points, mark, number_record_book);
11 $$;
12
13 call "exams schema".retake(4, 14, 3, 2);
14 select * from "exams schema".passing_the_exam;
```

Data Output Explain Messages Notifications

	id_exam [PK] integer	attempt_number integer	id_schedule integer	id_student integer	points integer	mark character varying	record_book_number integer
1	1	1	1	313264	20	5	5
2	2	1	2	309670	20	5	4
3	3	1	4	313264	20	5	5
4	4	1	3	282928	0	Незачет	2
5	5	1	5	264920	16	Зачет	1
6	6	1	6	247891	12	Зачет	8
7	7	1	4	282928	0	2	2
8	8	1	2	313264	18	4	5
9	9	1	3	264920	16	Зачет	1
10	10	1	1	309670	20	5	4
11	11	1	1	282928	0	2	2
12	12	3	7	282928	0	2	2
13	13	1	3	308535	20	Зачет	3
14	14	2	4	282928	14	3	2

## Задание 2. Создать триггер

Query Editor Query History

```
1 create or replace function "exams schema".add_to_log() returns trigger as $$
2 declare
3     mstr varchar(30);
4     astr varchar(100);
5     retstr varchar(254);
6 begin
7     if TG_OP = 'INSERT' then
8         astr = new;
9         mstr := 'Add data';
10        retstr := mstr || astr;
11        insert into "exams schema".logs(text, added, table_name)
12        values(retstr, now(), TG_TABLE_NAME());
13        return new;
14    elsif TG_OP = 'UPDATE' then
15        astr = new;
16        mstr := 'Update data';
17        retstr := mstr || astr;
18        insert into "exams schema".logs(text, added, table_name)
19        values(retstr, now(), TG_TABLE_NAME());
20        return new;
21    elsif TG_OP = 'DELETE' then
22        astr = old;
23        mstr := 'Remove data';
24        retstr := mstr || astr;
25        insert into "exams schema".logs(text, added, table_name)
26        values(retstr, now(), TG_TABLE_NAME());
27        return old;
28    end if;
29 end;
30 $$ language plpgsql;
```

Data Output Explain Messages Notifications

CREATE FUNCTION

Query returned successfully in 56 msec.

Query Editor Query History

```
1 create trigger t_pass_exam after insert or update or delete on "exams schema".passing_the_exam
2 for each row execute procedure "exams schema".add_to_log();
```

Data Output Explain Messages Notifications

CREATE TRIGGER

Query returned successfully in 48 msec.

### Проблема:

Query Editor Query History

```
1 insert into "exams schema".passing_the_exam(
2 id_exam, attempt_number, id_schedule, id_student, points, mark, record_book_number)
3 values (15, 1, 3, 282928, 12, 'Зачет', 2);
4
5 update "exams schema".passing_the_exam
6 set id_exam = 16
7 where id_exam = 15;
8
9 delete from "exams schema".passing_the_exam
10 where id_exam = 16;
11
12 select * from "exams schema".logs
```

Data Output Explain **Messages** Notifications

```

ERROR:  relation "exams schema.logs" does not exist
LINE 1: insert into "exams schema".logs(text, added, table_name)
          ^
QUERY:  insert into "exams schema".logs(text, added, table_name)
          values(retstr, now(), TG_TABLE_NAME())
CONTEXT:  PL/pgSQL function "exams schema".add_to_log() line 11 at SQL statement
SQL state: 42P01

```