

**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: Григорьев А.В.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

## Оглавление

Цель работы .....	3
Практическое задание .....	3
Выполнение .....	3
Вывод.....	8

**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)**

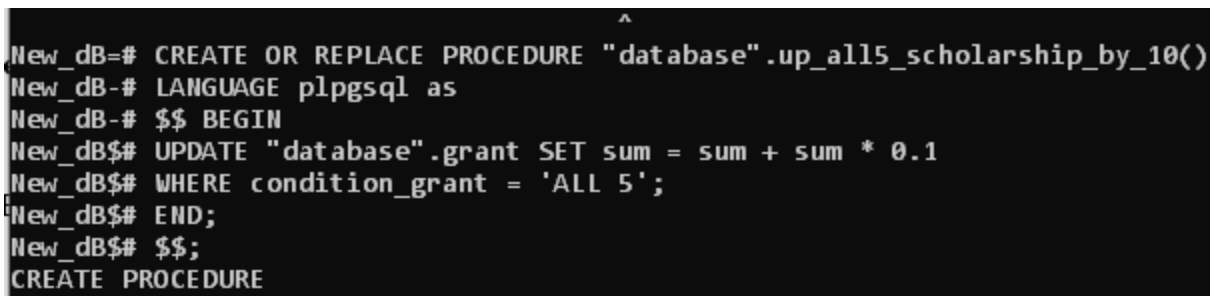
1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
2. Создать авторский триггер по варианту индивидуального задания.

**Выполнение**

**Создайте хранимые процедуры:**

1. Для повышения стипендии отличникам на 10%

```
CREATE OR REPLACE PROCEDURE "database".up_all5_scholarship_by_10()
LANGUAGE plpgsql as
$$ BEGIN
    UPDATE "database".grant SET sum = sum + sum * 0.1
    WHERE condition_grant = 'ALL 5';
END;
$$;
```



```
New_DB=# CREATE OR REPLACE PROCEDURE "database".up_all5_scholarship_by_10()
New_DB=# LANGUAGE plpgsql as
New_DB=# $$ BEGIN
New_DB$$ UPDATE "database".grant SET sum = sum + sum * 0.1
New_DB$$ WHERE condition_grant = 'ALL 5';
New_DB$$ END;
New_DB$$ $$;
CREATE PROCEDURE
```

```
CREATE PROCEDURE
New_dB=# SELECT * FROM "database".grant;
code_grant | form_grant | condition_grant | sum
-----+-----+-----+-----
          3 | UNI        | ALL 5           | 10000
          2 | SOC        | FAMILY 5       | 20000
          1 | GOV        | Science         | 40000
(3 строк)
```

```
New_dB=# CALL "database".up_all5_scholarship_by_10()
New_dB=# ;
CALL
New_dB=# SELECT * FROM "database".grant;
code_grant | form_grant | condition_grant | sum
-----+-----+-----+-----
          2 | SOC        | FAMILY 5       | 20000
          1 | GOV        | Science         | 40000
          3 | UNI        | ALL 5           | 11000
(3 строк)
```

- Для перевода студентов на следующий курс

```
CREATE OR REPLACE PROCEDURE "database".next_year()
```

```
LANGUAGE plpgsql as
```

```
$$ BEGIN
```

```
    UPDATE "database".student SET course = course + 1
```

```
    WHERE course < 4;
```

```
END;
```

```
$$;
```

```
New_dB=# CREATE OR REPLACE PROCEDURE "database".next_year()
```

```
New_dB=# LANGUAGE plpgsql as
```

```
New_dB=# $$ BEGIN
```

```
New_dB$$ UPDATE "database".student SET course = course + 1
```

```
New_dB$$ WHERE course < 4;
```

```
New_dB$$ END;
```

```
New_dB$$ $$;
```

```
CREATE PROCEDURE
```

```
New_dB=#
```

```
New_dB=# SELECT * FROM "database".student;
```

code_student	code_group	status_sudent	course	surname	name	middlename
3	3	REJECT	3	Dmitriev	Dmitriy	Dmitriovich
2	1	AKEDEM	2	Alekseev	Aleksei	Alekseivich
1	3	STUDY	2	Ivanov	Ivan	Ivanovich

```
(3 строки)
```

```
New_dB=# CALL "database".next_year();
```

```
CALL
```

```
New_dB=# SELECT * FROM "database".student;
```

code_student	code_group	status_sudent	course	surname	name	middlename
3	3	REJECT	4	Dmitriev	Dmitriy	Dmitriovich
2	1	AKEDEM	3	Alekseev	Aleksei	Alekseivich
1	3	STUDY	3	Ivanov	Ivan	Ivanovich

```
(3 строки)
```

- Для изменения оценки при успешной пересдаче экзамена.

```
CREATE OR REPLACE PROCEDURE "database".change_mark(new_mark integer, exam_id integer)
LANGUAGE plpgsql as
$$ BEGIN
    UPDATE "Session".attestation_pass SET mark = new_mark
    WHERE attestation_id = exam_id;
END;

$$;
```

```
New_dB=# CREATE OR REPLACE PROCEDURE "database".change_mark(new_mark integer, exam_id integer)
New_dB=# LANGUAGE plpgsql as
New_dB=# $$ BEGIN
New_dB$$ UPDATE "database".exam SET grade_exam = new_mark
New_dB$$ WHERE code_exam = exam_id;
New_dB$$ END;
New_dB$$ $$;
CREATE PROCEDURE
```

```
New_dB=# CALL "database".change_mark(4, 4);
CALL
New_dB=# CALL "database".change_mark(5, 4);
CALL
New_dB=# SELECT * FROM "database".exam;
```

code_subject	code_teacher	number_try	date_exam	code_exam	code_student	grade_exam
3	1	1	2021-01-23	1	1	5
4	1	2	2021-01-11	2	1	4
1	2	3	2021-01-15	3	2	2
2	3	1	2021-01-18	4	3	5

(4 строки)

```
New_dB=# CALL "database".change_mark(4, 4);
CALL
New_dB=# SELECT * FROM "database".exam;
```

code_subject	code_teacher	number_try	date_exam	code_exam	code_student	grade_exam
3	1	1	2021-01-23	1	1	5
4	1	2	2021-01-11	2	1	4
1	2	3	2021-01-15	3	2	2
2	3	1	2021-01-18	4	3	4

(4 строки)

### Создайте необходимые триггеры:

- Триггер для автоматического подсчёта общего количества часов дисциплины после внесения или изменения.

```
CREATE OR REPLACE FUNCTION "database".all_hours_upd() RETURNS TRIGGER AS $$
BEGIN
NEW.all_hours := (NEW.lab_hours + NEW.lec_hours + NEW.prac_hours);
RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER all_hours_update BEFORE INSERT OR UPDATE ON "database".subject
FOR EACH ROW EXECUTE FUNCTION "database".all_hours_upd();
```

```
New_db=# CREATE OR REPLACE FUNCTION "database".all_hours_upd() RETURNS TRIGGER AS $$
New_db$$ BEGIN
New_db$$ NEW.all_hours := (NEW.lab_hours + NEW.lec_hours + NEW.prac_hours);
New_db$$ RETURN NEW;
New_db$$ END;
New_db$$ $$ LANGUAGE plpgsql;
CREATE FUNCTION
New_db=#
New_db=# CREATE OR REPLACE TRIGGER all_hours_update BEFORE INSERT OR UPDATE ON "database".subject
New_db=# FOR EACH ROW EXECUTE FUNCTION "database".all_hours_upd();
CREATE TRIGGER
New_db=#
```

```
New_db=# SELECT * FROM "database".subject;
code_subject | name_subject | form_educ | all_hours | lab_hours | lec_hours | prac_hours | type_exam
-----+-----+-----+-----+-----+-----+-----+-----
          2 | Physic      |          1 |         60 |         10 |         30 |         20 | Exam
          1 | Calculus    |          1 |         50 |          0 |         20 |         30 | Exam
          4 | Biology     |          1 |         50 |         10 |         20 |         20 | Exam
          3 | English     |          0 |         20 |         20 |          0 |          0 | Test
(4 rows)
```

```
New_db=# UPDATE "database".subject SET prac_hours = 20 WHERE code_subject = 3;
UPDATE 1
New_db=# SELECT * FROM "database".subject;
code_subject | name_subject | form_educ | all_hours | lab_hours | lec_hours | prac_hours | type_exam
-----+-----+-----+-----+-----+-----+-----+-----
          2 | Physic      |          1 |         60 |         10 |         30 |         20 | Exam
          1 | Calculus    |          1 |         50 |          0 |         20 |         30 | Exam
          4 | Biology     |          1 |         50 |         10 |         20 |         20 | Exam
          3 | English     |          0 |         40 |         20 |          0 |         20 | Test
(4 rows)
```

## Вывод

В ходе лабораторной работы была освоена работа с процедурами и триггерами.