

**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: Котовщиков А. Р.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**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. Модифицировать триггер на проверку корректности входа и выхода сотрудника.
3. Создать авторский триггер по варианту индивидуального задания.

### **Выполнение**

#### **Создайте хранимые процедуры:**

- Для получения расписания занятий для групп на определенный день недели.

```
create or replace function lab_3_2.get_group_schedule_by_week_day(  
    name_of_group varchar,  
    day_of_week lab_3_2.schedule_week_day  
)  
  
returns table(  
    subject_name varchar,  
    class_time varchar,  
    address varchar,  
    classroom varchar,  
    group_name varchar,  
    week_day lab_3_2.schedule_week_day,  
    classroom_date date  
)  
  
language plpgsql
```

```

as $$

declare

    nearest_date date;

begin

    nearest_date := (SELECT MIN(s.datetime::date) FROM lab_3_2.schedule AS
s

        WHERE s.week_day = day_of_week

            AND s.group_name = name_of_group

                AND s.datetime::date >= CURRENT_DATE);

    return query (

        SELECT

            subj.name,

            to_char(s.datetime, 'HH24:MI')::varchar AS class_time,

            bs.address,

            c.number,

            s.group_name,

            s.week_day,

            s.datetime::date AS classroom_date

        FROM lab_3_2.schedule AS s

            JOIN lab_3_2.classroom AS c

                ON s.classroom_code = c.code

            JOIN lab_3_2.branch_site AS bs

                ON c.branch_site_id = bs.id

            JOIN lab_3_2.curriculum_subject AS cs

                ON s.curriculum_subject_id = cs.id

            JOIN lab_3_2.subject AS subj

```

```

        ON cs.subject_code = subj.code

        WHERE s.datetime::date = nearest_date

        AND s.group_name = name_of_group

        ORDER BY class_time

    );

end;$$;

```

Query Query History

Scratch Pad

```
1 SELECT * FROM lab_3_2.get_group_schedule_by_week_day('K3239', 'пн');
```

Data Output Messages Notifications

	subject_name character varying	class_time character varying	address character varying	classroom character varying	group_name character varying	week_day lab_3_2.schedule_week_day	classroom_date date
1	Программирование	02:01	Ломоносова д.9 лит.В	1003	K3239	пн	2023-12-27
2	Программирование	02:02	улица Народного Ополчения, 35с7	1004	K3239	пн	2023-12-27
3	Программирование	02:03	улица Народного Ополчения, 35с7	1004	K3239	пн	2023-12-27

✓ Successfully run. Total query runtime: 208 msec. 3 rows affected.
✗

- Для записи на курс слушателя.

```

create or replace procedure lab_3_2.enroll_student(
    student_id int,
    group_name varchar
)
language plpgsql
as $$
declare
    is_student_exists bool;
    is_group_exists bool;
begin
    is_student_exists := (
        SELECT
            CASE WHEN COUNT(*) = 0
                THEN false
            ELSE true
            END
        FROM lab_3_2.student AS s
        WHERE s.id = student_id);

```

```

        IF NOT is_student_exists
            THEN RAISE EXCEPTION 'Студент с идентификатором % не
существует!', student_id;
        END IF;

is_group_exists := (
    SELECT CASE WHEN COUNT(*) = 0
        THEN false
        ELSE true
    END
    FROM lab_3_2.group AS g
    WHERE g.name = group_name);

IF NOT is_group_exists
    THEN RAISE EXCEPTION 'Группа % не существует!', group_name;
END IF;

INSERT INTO lab_3_2.student_in_group
(
    status,
    start_date,
    end_date,
    student_id,
    group_name
)
VALUES
(
    'Учится',
    NOW(),
    (SELECT ic.end_date FROM lab_3_2.implemented_curriculum AS ic
WHERE EXISTS (
        SELECT 1 FROM lab_3_2.group AS g WHERE g.name =
group_name
        AND g.implemented_curriculum_id = ic.id)
    ),
    student_id,
    group_name
);
end; $$;

```



```

        c.type,
        c.capacity,
        bs.address
FROM lab_3_2.classroom AS c
JOIN lab_3_2.branch_site AS bs ON c.branch_site_id = bs.id
WHERE NOT EXISTS (
    SELECT 1 FROM lab_3_2.schedule AS s
    WHERE s.classroom_code = c.code
    AND s.week_day = day_of_week)
);
end;$$;

```

Query

Query History

Scratch Pad

1

SELECT \* FROM lab\_3\_2.get\_free\_classrooms\_or\_throw\_exception('cp');

Data Output

Messages

Notifications

## Модифицировать триггер на проверку корректности входа и выхода сотрудника:

```

create or replace function fn_check_time_punch()
returns trigger
language plpgsql
as $psql$
begin
    -- Валидация входов
    IF NEW.is_out_punch = false THEN
        -- Проверка дата входа >= текущей даты (иначе null)
        IF NEW.punch_time < NOW()
            THEN return null;

        -- Нельзя войти еще раз, пока ты не вышел
    ELSIF (
        SELECT punch_count - out_punch_count FROM (
            SELECT

```

```

        (SELECT COUNT(*) FROM time_punch
          WHERE is_out_punch = false
          AND employee_id = NEW.employee_id
        ) AS punch_count,
        (SELECT COUNT(*) FROM time_punch
          WHERE is_out_punch = true
          AND employee_id = NEW.employee_id
        ) AS out_punch_count
      ) AS table_alias
    ) <> 0
      THEN return null;

    ELSE
      END IF;

  END IF;

-- Валидация выходов
IF NEW.is_out_punch = true THEN
  -- Количество входов должно быть на 1 больше, чем выходов
  IF (
    SELECT punch_count - out_punch_count FROM (
      SELECT DISTINCT
        (SELECT COUNT(*) FROM time_punch
          WHERE is_out_punch = false
          AND employee_id = NEW.employee_id
        ) AS punch_count,
        (SELECT COUNT(*) FROM time_punch
          WHERE is_out_punch = true
          AND employee_id = NEW.employee_id
        ) AS out_punch_count
      ) AS table_alias
    ) <> 1
    THEN return null;

  -- Дата последнего входа должна быть меньше даты выхода
  ELSIF
    (SELECT punch_time FROM time_punch AS tp
     WHERE tp.is_out_punch = false
     AND tp.employee_id = NEW.employee_id
    ORDER BY punch_time DESC
    LIMIT 1) >= NEW.punch_time
    THEN return null;

  ELSE
    END IF;

  END IF;
  return NEW;
end;
$sql$;

CREATE TRIGGER punch_time_checker
BEFORE INSERT

```



```
ON time_punch
FOR EACH ROW EXECUTE FUNCTION fn_check_time_punch();
```

### Создайте необходимые триггеры:

- Триггер, который автоматически генерирует уникальный код для студента в группе.

```
create or replace function generate_code_for_student_in_group()
returns trigger
language plpgsql
AS $$
declare
    student_in_group_code varchar;
begin
    student_in_group_code := (
        SELECT MAX(sig.code::int) + 1
        FROM lab_3_2.student_in_group AS sig);

    IF student_in_group_code IS NULL
        THEN student_in_group_code := 1;
    END IF;

    NEW.code = student_in_group_code;
    return NEW;
end; $$;
```

```
CREATE OR REPLACE TRIGGER generate_code_for_student_in_group_trigger
BEFORE INSERT ON lab_3_2.student_in_group
FOR EACH ROW EXECUTE FUNCTION generate_code_for_student_in_group();
```

Query Query History

```
1 -- CALL lab_3_2.enroll_student(5573, 'K3239');
2
3 SELECT * FROM lab_3_2.student_in_group AS sig
4 WHERE sig.student_id = 5573;
```

Scratch Pad

Data Output Messages Notifications

	code [PK] character varying (15)	status lab_3_2.student_in_group_status	start_date date	end_date date	student_id integer	group_name character varying (20)
1	91644597	Учится	2023-12-22	2026-10-09	5573	K3239

## **Вывод**

В рамках данной лабораторной работы мы познакомились с возможностями процедурного программирования в языке SQL, научились создавать процедуры, функции, а также использовать триггеры.