

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

ОТЧЕТ

по лабораторной работе «Процедуры, функции, триггеры в PostgreSQL»

по дисциплине «Проектирование и реализация баз данных»

Отчет выполнил:

Студент: Никоноров Максим

Группа: K32422

Факультет: ИКТ

Преподаватель: Говорова Марина Михайловна



Санкт-Петербург 2023

Цель работы

Овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

Практическое задание

Часть I

Запрос 1. Вывести список свободных аудиторий для проведения практических занятий в заданное время.

```
CREATE OR REPLACE FUNCTION auditory_gr(year_num
INTEGER, week_num INTEGER, week_day INTEGER, b_time
TIME) RETURNS
```

```
SETOF record LANGUAGE SQL AS $$
```

```
SELECT
```

```
    auditory.auditory_no
```

```
FROM
```

```
    structure.auditory
```

```
WHERE
```

```
    auditory.auditory_no NOT IN
```

```
(
```

```
    SELECT
```

```
        auditory_no
```

```
    FROM
```

```
        structure.schedule
```

```
    WHERE
```

```
        year = year_num AND week_no = week_num
AND day_of_week_no = week_day AND begin_time =
b_time
```

```
)
```

```
AND auditory_type = 'laboratory'

$$;

SELECT auditory_gr(2022, 45, 1, '10:00:00');
```

Query

Query History

1

CREATE OR REPLACE FUNCTION auditory_gr(year_num INTEGER, week_num INTEGER, week_day INTEGER, b_time TIME)

2

RETURNS SETOF record LANGUAGE SQL AS \$\$

3

SELECT

4

auditory.auditory_no

5

FROM

6

structure.auditory

7

WHERE

8

auditory.auditory_no NOT IN

9

(

10

SELECT

11

auditory_no

12

FROM

13

structure.schedule

14

WHERE

15

year = year_num AND week_no = week_num AND day_of_week_no = week_day AND begin_time = b_time

16

)

Messages

Explain

×

Notifications

Data Output

+

+

+

+

+

+

+

+

auditory_gr

record

🔒

1

(11420)

2

(12122)

3

(12210)

4

(13125)

Запрос 2. Вывести расписание занятий для заданного преподавателя.

```
CREATE OR REPLACE FUNCTION aud_tutor (tutor_num
INTEGER, year_num INTEGER, week_num INTEGER) RETURNS
TABLE(day_of_week_no INTEGER, fixed_day DATE,
begin_time TIME WITHOUT TIME ZONE, discipline_code
CHAR, auditory_no INT) LANGUAGE SQL AS $$

SELECT

    day_of_week_no, fixed_day, begin_time,
discipline_code, auditory_no

FROM

    structure.schedule

WHERE

    tutor_isu_no = tutor_num AND year = year_num
AND week_no = week_num

$$;

SELECT *

FROM aud_tutor(233876, 2022, 46)
```

The screenshot shows a PostgreSQL IDE window with the title 'unikitchen/postgres@PostgreSQL 14'. The 'Query' tab is active, displaying a SQL script. The script defines a function 'aud_tutor' and then calls it with parameters (233876, 2022, 46). The 'Data Output' tab is also visible, showing the results of the function call as a table with 5 columns and 3 rows.

```

1 CREATE OR REPLACE FUNCTION aud_tutor (tutor_num INTEGER, year_num INTEGER, week_num INTEGER) RETURNS TABLE(day_o
2 SELECT
3     day_of_week_no, fixed_day, begin_time, discipline_code, auditory_no
4 FROM
5     structure.schedule
6 WHERE
7     tutor_isu_no = tutor_num AND year = year_num AND week_no = week_num
8 $$;
9
10 SELECT * FROM aud_tutor(233876, 2022, 46)
11
12
13
14

```

	day_of_week_no integer	fixed_day date	begin_time time without time zone	discipline_code character	auditory_no integer
1	5	[null]	08:20:00	PS101	24307
2	5	[null]	10:00:00	PS101	21401
3	5	[null]	11:40:00	PS101	21401

Запрос 3. Вывести список аудиторий, в которых может разместиться заданная группа.

```

CREATE OR REPLACE FUNCTION
auditory_list_for_gr(group_num VARCHAR(7)) RETURNS
SETOF record LANGUAGE SQL AS $$
SELECT
    auditory_no
FROM
    structure.auditory
WHERE
    value_of_seats >=
    (
        SELECT
            COUNT(student_isu_no)
        FROM
            structure.student_status
        WHERE

```

```

status = 'study' AND group_no =
group_num

);

$$;

```

```

SELECT auditory_list_for_gr('K32422')

```

The screenshot shows the UniKitchen PostgreSQL IDE interface. The top menu bar includes Dashboard, Properties, SQL, Dependencies, Dependents, and Processes. The main window displays a SQL query in the editor, which is a CREATE OR REPLACE FUNCTION for 'auditory_list_for_gr'. The query defines a function that returns a set of records based on a group number and a status filter. The function body includes a SELECT statement that joins the 'structure.auditory' table with a subquery that counts student statuses. The results of the function call for group 'K32422' are displayed in the 'Data Output' pane on the right, showing 15 rows of data.

Query Editor Content:

```

1 CREATE OR REPLACE FUNCTION auditory_list_for_gr(group_num VARCHAR(7)) RETURNS SETOF record LANGUAGE SQL
2     SELECT
3         auditory_no
4     FROM
5         structure.auditory
6     WHERE
7         value_of_seats >=
8         (
9             SELECT
10                COUNT(student_isu_no)
11            FROM
12                structure.student_status
13            WHERE
14                status = 'study' AND group_no = group_num
15        );
16 $$;
17
18 SELECT auditory_list_for_gr('K32422')
19
20
21
22

```

Data Output:

auditory_list_for_gr record
1 (11404)
2 (11420)
3 (12122)
4 (12210)
5 (12306)
6 (13329)
7 (14204)
8 (14210)
9 (14315)
10 (14321)
11 (14414)
12 (14424)
13 (21309)
14 (21325)
15 (21329)

Часть II

Создать функцию и триггер для таблицы со студентами, которая бы вела логи операций удаления, вставки и обновления.

```
CREATE OR REPLACE FUNCTION add_log() RETURNS TRIGGER
AS $$
DECLARE
    mstr varchar(50);
    astr varchar(100);
    retstr varchar(254);
BEGIN
    IF
        TG_OP = 'INSERT' THEN
            astr = NEW.name;
            mstr := 'Добавлен новый студент ';
            retstr := mstr||astr;
            INSERT INTO structure.logs(text,added) values
(retstr,NOW());
            RETURN NEW;
        ELIF
            TG_OP = 'UPDATE' THEN
                astr = NEW.name;
                mstr := 'Изменение информации о студенте ';
                retstr := mstr||astr;
                INSERT INTO structure.logs(text,added) values
(retstr,NOW());
                RETURN NEW;
            ELIF
                TG_OP = 'DELETE' THEN
```

```

        astr = OLD.name;
        mstr := 'Удален студент ';
        retstr := mstr || astr;
        INSERT INTO structure.logs(text,added) values
(retstr,NOW());
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE PLPGSQL;

```

```

CREATE OR REPLACE TRIGGER ch_students AFTER
INSERT
OR
UPDATE
OR
DELETE ON structure.student
FOR EACH ROW EXECUTE PROCEDURE add_log ();

```

```

INSERT INTO structure.student
VALUES ('+79633150363', 455500, 2021, 'Замятин
Акакий');

```

```

UPDATE structure.student
SET phone_no = '+77056019649'
WHERE student_isu_no = 455500;

```

```

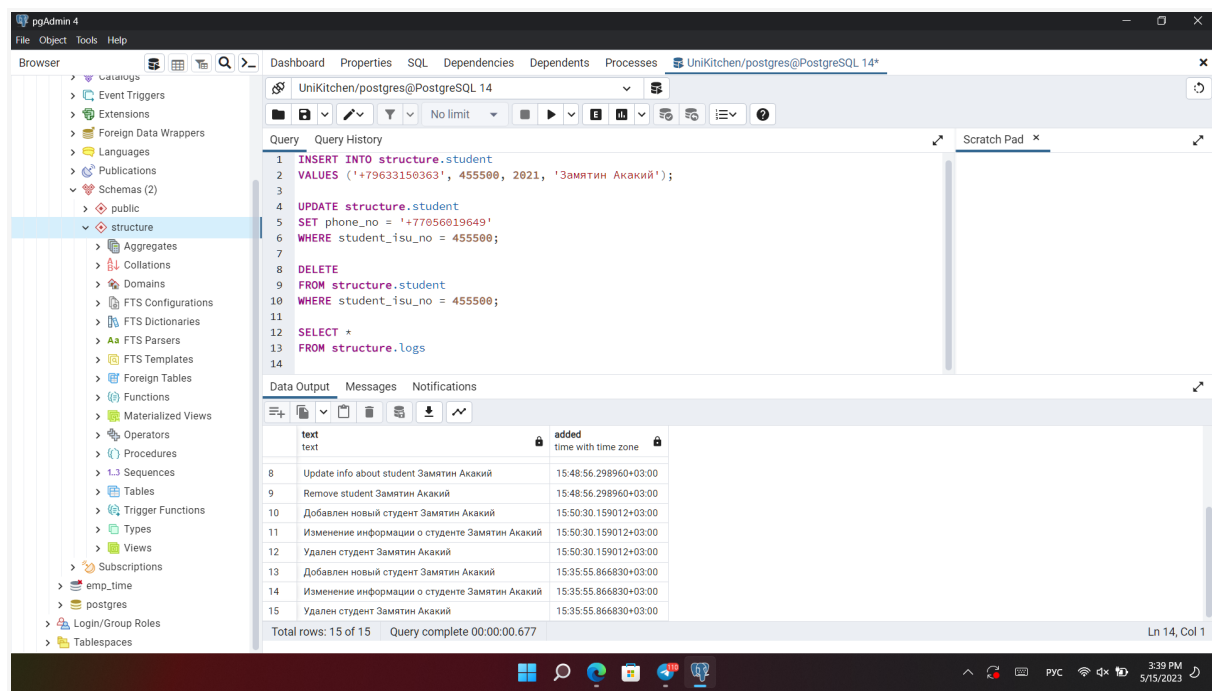
DELETE
FROM structure.student
WHERE student_isu_no = 455500;

```

```

SELECT *
FROM structure.logs

```

Выводы

В третьей лабораторной работе я познакомился со структурой и синтаксисом функций и триггеров в PostgreSQL, научился объявлять переменные внутри функций и вставлять скриншоты в Google Docs, выучил русский язык и научился основам оформления отчетов к лабораторным работам.