# 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: Филиппов А.Э.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



# Оглавление

Цель работы:	3
Практическое задание	3
F	
Выполнение	3
Вывол	10

**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 (тах - 8 баллов)

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

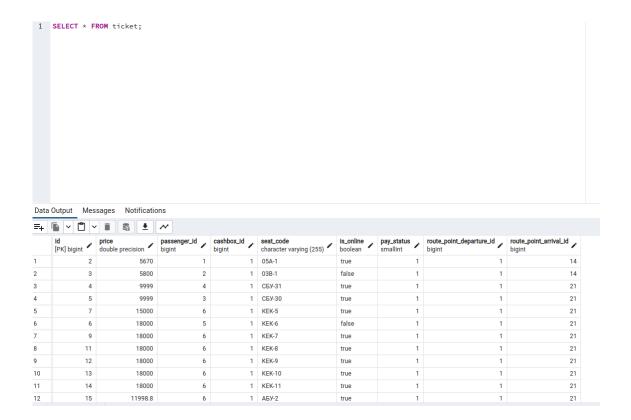
#### Выполнение

### Создайте хранимые процедуры:

• Для повышения цен в пригородные поезда на 20%.

```
CREATE PROCEDURE increase_ticket_prices_by20percent()
LANGUAGE SQL
AS $$

UPDATE ticket SET price = price*1.2 WHERE ticket.id IN (
SELECT ticket.id FROM schedule
INNER JOIN route ON schedule.route_id = route.id
INNER JOIN carriage ON carriage.schedule_id = schedule.id
INNER JOIN seat ON carriage.id = seat.carriage_id
INNER JOIN ticket ON seat.code = ticket.seat_code
WHERE schedule.planned_departure > NOW() AND route.train_type =
'Пригородный')
$$;
```

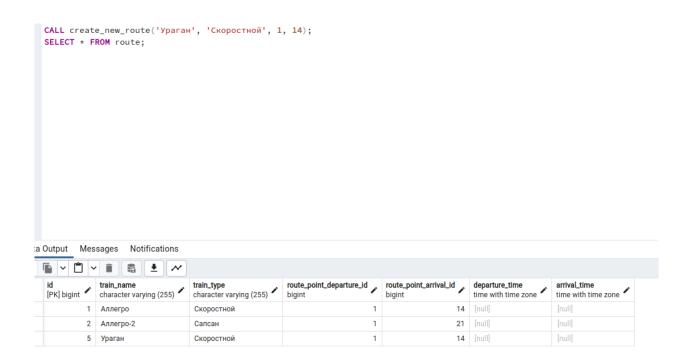


• Для создания нового рейса на поезд.

```
CREATE PROCEDURE create_new_route(train_name varchar, train_type varchar, route_point_departure_id integer, route_point_arrival_id integer)
LANGUAGE SQL
```

### AS \$\$

```
INSERT INTO route("train_name", "train_type", "route_point_departure_id",
"route_point_arrival_id") VALUES(train_name, train_type,
route_point_departure_id, route_point_arrival_id)
$$;
```



• Для формирования общей выручки по продаже билетов за сутки

```
CREATE OR REPLACE PROCEDURE get_last24hr_revenue(
INOUT summ DOUBLE PRECISION DEFAULT 0)

AS $$

SELECT SUM(price) FROM ticket
INNER JOIN seat ON ticket.seat_code = seat.code
INNER JOIN carriage ON carriage.id = seat.carriage_id
INNER JOIN schedule ON carriage.schedule_id = schedule.id
WHERE schedule.planned_departure BETWEEN NOW() - INTERVAL '24 HOURS' and
NOW()

$$ LANGUAGE SQL;
```

```
1 CREATE OR REPLACE PROCEDURE get_last24hr_revenue(
2 INOUT summ DOUBLE PRECISION DEFAULT 0)
3 AS $$
4 SELECT SUM(price) FROM ticket
5 INNER JOIN seat ON ticket.seat_code = seat.code
   INNER JOIN carriage ON carriage.id = seat.carriage_id
   INNER JOIN schedule ON carriage.schedule_id = schedule.id
8 WHERE schedule.planned_departure BETWEEN NOW() - INTERVAL '24 HOURS' and NOW()
   $$ LANGUAGE SQL;
10 CALL get_last24hr_revenue();
Data Output
                     Notifications
           Messages
    summ
    double precision
             19998
```

### Создайте необходимые триггеры:

• Триггер, добавляющий функцию возврата к невозвратным билетам в случае переноса рейса

```
create or replace function fn_enable_refund() returns trigger as $psql$
begin
    update ticket SET pay_status = 2 WHERE id IN
        (SELECT ticket.id FROM schedule INNER JOIN carriage ON schedule.id =
carriage.schedule_id INNER JOIN seat ON seat.carriage_id = carriage.id INNER
JOIN ticket ON ticket.seat_code = seat.code WHERE schedule.id = old.id)
        AND (pay_status = 1);
        return new;
    end;
$psql$ language plpgsql;
CREATE TRIGGER enable_refund
    AFTER UPDATE OF planned_departure ON schedule FOR EACH ROW EXECUTE
PROCEDURE fn_enable_refund();
```



	id [PK] bigint	price double precision	passenger_id /	cashbox_id /	seat_code character varying (255)	is_online boolean	pay_status smallint	route_point_departure_id , bigint	route_point_arrival_id , bigint
1	2	5670	1	1	05A-1	true	1	1	14
2	3	5800	2	1	03B-1	false	1	1	14
3	4	9999	4	1	СБУ-31	true	2	1	21
4	5	9999	3	1	СБУ-30	true	2	1	21
5	6	18000	5	1	KEK-6	false	1	1	21
6	7	15000	6	1	KEK-5	true	1	1	21
7	9	18000	6	1	KEK-7	true	1	1	21
8	11	18000	6	1	KEK-8	true	1	1	21
9	12	18000	6	1	KEK-9	true	1	1	21
10	13	18000	6	1	KEK-10	true	1	1	21
11	14	18000	6	1	KEK-11	true	1	1	21
12	15	11998.8	6		АБУ-2	true	1	1	21

# Вывод

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