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

**Отчет**

по лабораторной работе «Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL»

по дисциплине «**Базы данных**»

Автор: Пронина Мария Владимировна

Факультет: ИКТ

Группа: K32392

Преподаватель: Говорова М. М.

Дата: 18.05.2023

Санкт-Петербург 2023

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

**Практическое задание:**
Вариант 1:

Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).

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.

**Выполнение работы:**
Предметная область – Система для продажи железнодорожных билетов (вариант 6).
Наименование БД – Passenger.

**Создание процедур:**

1. Для повышения цен в пригородные поезда на 20%.

```
create or replace procedure increase_prices ()
LANGUAGE SQL
AS $$
      update seat
      set price = price * 1.2
      where seat_code in (select seat.seat_code from seat, carriage,
train, schedule
                                  where seat.carriage_number =
carriage.carriage_number and carriage.train_number = train.train_number
                                  and train.route_number =
schedule.route_number and schedule.train_type = 'shuttle'
                                  and train.departure_time > now() and
seat.if_occupied = false);
$$
```

Запрос select для вывода цен на пригородные поезда:

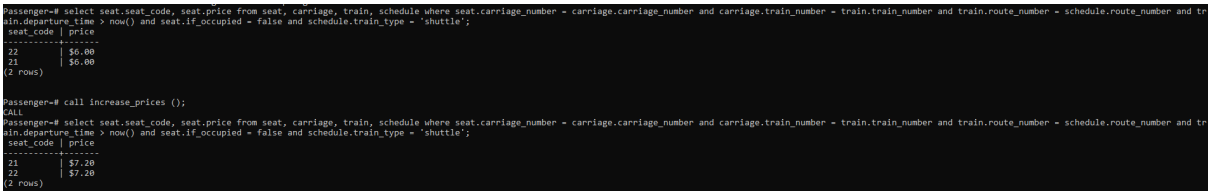
```
select seat.seat_code, seat.price from seat, carriage, train, schedule
where seat.carriage_number = carriage.carriage_number and
carriage.train_number = train.train_number and train.route_number =
schedule.route_number and train.departure_time > now() and
seat.if_occupied = false and schedule.train_type = 'shuttle';
```

Результат выполнения запроса до запуска процедуры:

```
 seat_code | price
-----------+-------
 22        | $6.00
 21        | $6.00
(2 rows)
```

Результат выполнения запроса после запуска процедуры:

```
 seat_code | price
-----------+-------
 21        | $7.20
 22        | $7.20
(2 rows)
```




2. Для создания нового рейса на поезд.

```
create or replace procedure add_train_for_route(p_route_number
integer, p_day_of_train date)
language sql
as $$
declare
    v_departure_time timestamp without time zone;
    v_arrival_time timestamp without time zone;
    v_status character varying(20);
    v_frequency character varying(20);
begin
    select p_day_of_train + departure_time
    into v_departure_time
    from schedule
    where route_number = p_route_number;

    select v_departure_time + duration
    into v_arrival_time
    from schedule
    where route_number = p_route_number;
```

```
    if v_arrival_time < now() then
        v_status := 'completed';
    elsif v_departure_time > now() then
        v_status := 'planned';
    else
        v_status := 'in progress';
    end if;

    select days_of_route
    into v_frequency
    from schedule
    where route_number = p_route_number;

    insert into train (route_number, status, frequency,
departure_time, arrival_time)
    values (p_route_number, v_status, v_frequency,
v_departure_time, v_arrival_time);
end;
$$;
```

Пример работы процедуры:

```
Passenger=# select * from train;
 train_number | route_number |    status    | frequency |   departure_time    |    arrival_time
--------------+--------------+--------------+-----------+---------------------+---------------------
            2 |            4 | в процессе   | ежедневно | 2023-02-04 00:00:00 | 2023-02-04 00:00:00
            3 |            4 | в процессе   | ежедневно | 2023-03-04 00:00:00 | 2023-03-04 00:00:00
            4 |            3 | запланирован | ежедневно | 2023-06-04 00:00:00 | 2023-06-05 00:00:00
            5 |            3 | отменен      | ежедневно | 2022-02-24 00:00:00 | 2022-02-24 00:00:00
            7 |            2 | завершен     | ежедневно | 2023-05-02 05:00:00 | 2023-05-02 08:00:00
            6 |            4 | запланирован | ежедневно | 2023-05-05 05:00:00 | 2023-05-04 08:00:00
           10 |            2 | запланирован | ежедневно | 2023-05-09 05:00:00 | 2023-05-09 08:00:00
            8 |            2 | запланирован | ежедневно | 2023-05-08 22:00:00 | 2023-05-09 05:00:00
            9 |            3 | завершен     | ежедневно | 2023-05-07 00:00:00 | 2023-05-08 00:00:00
           11 |            6 | запланирован | ежедневно | 2023-06-17 13:00:00 | 2023-06-17 15:00:00
           15 |            6 | planned      | daily     | 2023-05-20 13:00:00 | 2023-05-20 15:00:00
(11 rows)


Passenger=# call add_train_for_route (6, '2023-06-01');
CALL
Passenger=# select * from train;
 train_number | route_number |    status    | frequency |   departure_time    |    arrival_time
--------------+--------------+--------------+-----------+---------------------+---------------------
            2 |            4 | в процессе   | ежедневно | 2023-02-04 00:00:00 | 2023-02-04 00:00:00
            3 |            4 | в процессе   | ежедневно | 2023-03-04 00:00:00 | 2023-03-04 00:00:00
            4 |            3 | запланирован | ежедневно | 2023-06-04 00:00:00 | 2023-06-05 00:00:00
            5 |            3 | отменен      | ежедневно | 2022-02-24 00:00:00 | 2022-02-24 00:00:00
            7 |            2 | завершен     | ежедневно | 2023-05-02 05:00:00 | 2023-05-02 08:00:00
            6 |            4 | запланирован | ежедневно | 2023-05-05 05:00:00 | 2023-05-04 08:00:00
           10 |            2 | запланирован | ежедневно | 2023-05-09 05:00:00 | 2023-05-09 08:00:00
            8 |            2 | запланирован | ежедневно | 2023-05-08 22:00:00 | 2023-05-09 05:00:00
            9 |            3 | завершен     | ежедневно | 2023-05-07 00:00:00 | 2023-05-08 00:00:00
           11 |            6 | запланирован | ежедневно | 2023-06-17 13:00:00 | 2023-06-17 15:00:00
           15 |            6 | planned      | daily     | 2023-05-20 13:00:00 | 2023-05-20 15:00:00
           18 |            6 | planned      | daily     | 2023-06-01 13:00:00 | 2023-06-01 15:00:00
(12 rows)
```

3. Для формирования общей выручки по продаже билетов за сутки.

```
create or replace function count_income(p_date date)
returns money as $$
declare v_income money;
begin
    select sum(seat.price) + sum(seat.additional_service_price)
    into v_income from seat, ticket
    where ticket.seat_code = seat.seat_code and cast(ticket.sale_time as
date) = p_date and ticket.status = 'paid';

    return v_income;
end$$

language plpgsql;
```

Пример использования процедуры:

```
Passenger=# select count_income('2023-03-25');
 count_income
--------------
      $100.00
(1 row)
```

**Создание необходимых триггеров:**

1. Триггер гарантирует, что в вагоне не будет добавлено больше мест, чем допустимо для его типа

```
create or replace function check_seat_count()
returns trigger as $$
declare
    max_seats INTEGER;
    current_seats INTEGER;
begin
    select number_of_seats into max_seats from carriage_type
    where type_id = (select id_carriage_type from carriage where
carriage_number = new.carriage_number);

    select count(*) into current_seats from seat
    where carriage_number = new.carriage_number;

    if current_seats + 1 > max_seats then
        raise exception 'Cannot add an extra seat. Exceeded maximum
```

```
number of seats for the carriage type.';
    end if;

    return new;
end;
$$
language plpgsql;

create trigger check_seat_count_trigger before insert on seat
for each row execute function check_seat_count();
```

Пример работы триггера:

```
Passenger=# select * from seat;
 seat_code | carriage_number | index_number | price  | additional_service_price | if_occupied
-----------+-----------------+--------------+--------+--------------------------+-------------
1          |               2 |            1 | $30.00 |                    $5.00 | f
2          |               3 |            1 | $60.00 |                    $5.00 | f
5          |               2 |            2 | $30.00 |                    $5.00 | f
10         |              10 |            1 | $40.00 |                    $5.00 | f
11         |              10 |            2 | $40.00 |                    $5.00 | f
12         |               8 |            2 | $50.00 |                    $5.00 | f
13         |               8 |            3 | $50.00 |                    $5.00 | f
14         |              11 |            1 | $30.00 |                    $5.00 | f
16         |              12 |            1 | $50.00 |                    $5.00 | f
17         |               4 |            2 | $30.00 |                    $5.00 | f
20         |              13 |            2 | $30.00 |                    $5.00 | f
15         |              11 |            2 | $30.00 |                    $5.00 | t
18         |               5 |            2 | $60.00 |                    $5.00 | t
19         |              13 |            1 | $30.00 |                    $5.00 | t
3          |               4 |            1 | $30.00 |                    $5.00 | t
4          |               5 |            1 | $60.00 |                    $5.00 | t
6          |               6 |            1 | $40.00 |                    $5.00 | t
7          |               8 |            1 | $50.00 |                    $5.00 | t
8          |               9 |            1 | $70.00 |                    $0.00 | t
9          |               7 |            1 | $40.00 |                    $5.00 | t
21         |              14 |            1 |  $7.20 |                    $0.00 | f
22         |              14 |            2 |  $7.20 |                    $0.00 | f
23         |              12 |            2 | $50.00 |                    $5.00 | f
24         |              12 |            3 | $50.00 |                    $5.00 | f
25         |              12 |            4 | $50.00 |                    $5.00 | f
26         |              12 |            5 | $50.00 |                    $5.00 | f
27         |              12 |            6 | $50.00 |                    $5.00 | f
28         |              12 |            7 | $50.00 |                    $5.00 | f
29         |              12 |            8 | $50.00 |                    $5.00 | f
30         |              12 |            9 | $50.00 |                    $5.00 | f
31         |              12 |           10 | $50.00 |                    $5.00 | f
32         |              12 |           11 | $50.00 |                    $5.00 | f
33         |              12 |           12 | $50.00 |                    $5.00 | f
34         |              12 |           13 | $50.00 |                    $5.00 | f
35         |              12 |           14 | $50.00 |                    $5.00 | f
36         |              12 |           15 | $50.00 |                    $5.00 | f
37         |              12 |           16 | $50.00 |                    $5.00 | f
38         |              12 |           17 | $50.00 |                    $5.00 | f
39         |              12 |           18 | $50.00 |                    $5.00 | f
(39 rows)


Passenger=# insert into seat (seat_code, carriage_number, index_number, price, additional_service_price, if_occupied) values ('40', 12, 19, 50, 5, false);
ERROR:  Cannot add an extra seat. Exceeded maximum number of seats for the carriage type.
CONTEXT:  PL/pgSQL function check_seat_count() line 24 at RAISE
```

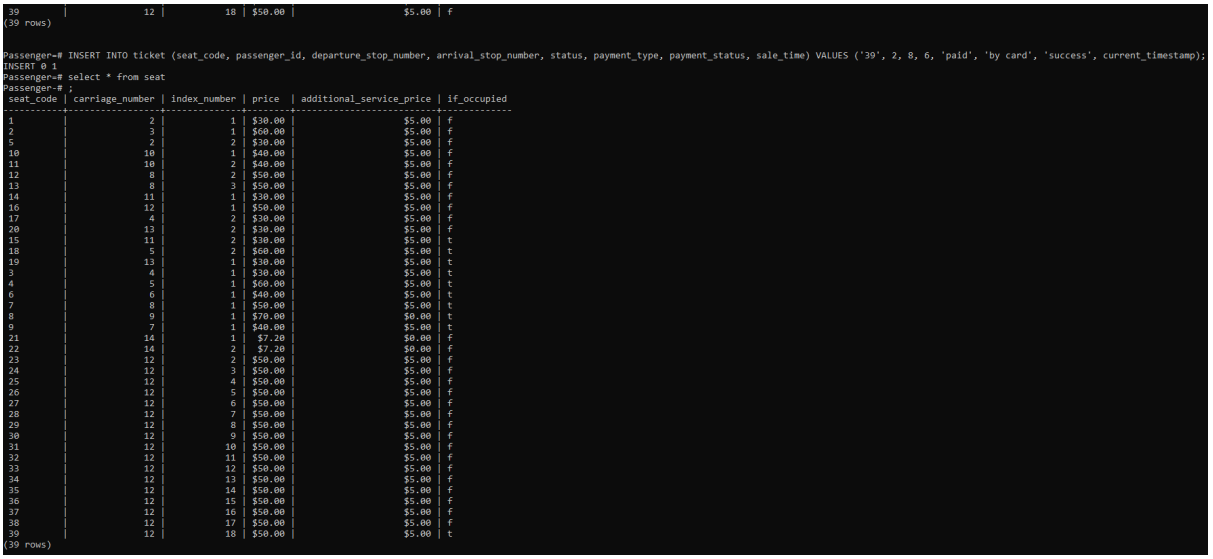
2. При добавлении билета, триггер автоматически обновляет статус занятости места.

```
create or replace function update_seat_occupied_status()
returns trigger as $$
begin
    if TG_OP = 'INSERT' then
        update seat
        set if_occupied = true
        where seat_code = NEW.seat_code;
    end if;

    return new;
end;
$$ language plpgsql;

create trigger update_seat_occupied_status_trigger
after insert on ticket
for each row
execute function update_seat_occupied_status();
```

Пример работы триггера:



```
 39        |            12 |           18 | $50.00 |                $5.00 | f
(39 rows)


Passenger-# INSERT INTO ticket (seat_code, passenger_id, departure_stop_number, arrival_stop_number, status, payment_type, payment_status, sale_time) VALUES ('39', 2, 8, 6, 'paid', 'by card', 'success', current_timestamp);
INSERT 0 1
Passenger-# select * from seat
Passenger-# ;
 seat_code | carriage_number | index_number | price  | additional_service_price | if_occupied
-----------+-----------------+--------------+--------+--------------------------+-------------
 1         |               2 |            1 | $30.00 |                $5.00 | f
 2         |               3 |            1 | $60.00 |                $5.00 | f
 5         |               2 |            2 | $30.00 |                $5.00 | f
 10        |              10 |            1 | $40.00 |                $5.00 | f
 11        |              10 |            2 | $40.00 |                $5.00 | f
 12        |               8 |            2 | $50.00 |                $5.00 | f
 13        |               8 |            3 | $50.00 |                $5.00 | f
 14        |              11 |            1 | $30.00 |                $5.00 | f
 16        |              12 |            1 | $50.00 |                $5.00 | f
 17        |               4 |            2 | $30.00 |                $5.00 | f
 20        |              13 |            2 | $30.00 |                $5.00 | f
 15        |              11 |            2 | $30.00 |                $5.00 | t
 18        |               5 |            2 | $60.00 |                $5.00 | t
 19        |              13 |            1 | $30.00 |                $5.00 | t
 3         |               4 |            1 | $30.00 |                $5.00 | t
 4         |               5 |            1 | $60.00 |                $5.00 | t
 6         |               6 |            1 | $40.00 |                $5.00 | t
 7         |               8 |            1 | $50.00 |                $5.00 | t
 8         |               9 |            1 | $70.00 |                $0.00 | t
 9         |               7 |            1 | $40.00 |                $5.00 | t
 21        |              14 |            1 |  $7.20 |                $0.00 | f
 22        |              14 |            2 |  $7.20 |                $0.00 | f
 23        |              12 |            2 | $50.00 |                $5.00 | f
 24        |              12 |            3 | $50.00 |                $5.00 | f
 25        |              12 |            4 | $50.00 |                $5.00 | f
 26        |              12 |            5 | $50.00 |                $5.00 | f
 27        |              12 |            6 | $50.00 |                $5.00 | f
 28        |              12 |            7 | $50.00 |                $5.00 | f
 29        |              12 |            8 | $50.00 |                $5.00 | f
 30        |              12 |            9 | $50.00 |                $5.00 | f
 31        |              12 |           10 | $50.00 |                $5.00 | f
 32        |              12 |           11 | $50.00 |                $5.00 | f
 33        |              12 |           12 | $50.00 |                $5.00 | f
 34        |              12 |           13 | $50.00 |                $5.00 | f
 35        |              12 |           14 | $50.00 |                $5.00 | f
 36        |              12 |           15 | $50.00 |                $5.00 | f
 37        |              12 |           16 | $50.00 |                $5.00 | f
 38        |              12 |           17 | $50.00 |                $5.00 | f
 39        |              12 |           18 | $50.00 |                $5.00 | t
(39 rows)
```

**Выводы:**

При выполнении данной лабораторной работы были приобретены практические навыки по работе с таблицами в базе данных PostgreSQL, созданию и выполнению процедур и функций, а также написанию необходимых для базы данных триггеров.