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: Волжева М. И.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы 3
Практическое задание 3
Вариант 19. БД «Пассажир» 3
Выполнение 3
Вывод 9

Цель работы

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

Практическое задание

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

Вариант 19. БД «Пассажир»

Описание предметной области:

Информационная система служит для продажи железнодорожных билетов. Билеты могут продаваться на текущие сутки или предварительно (не более чем за 45 суток). Цена билета при предварительной продаже снижается на 5%. Билет может быть приобретен в кассе или онлайн. Если билет приобретен в кассе, необходимо знать, в какой. Для каждой кассы известны номер и адрес. Кассы могут располагаться в различных населенных пунктах.

Поезда курсируют по расписанию, но могут назначаться дополнительные поезда на заданный период или определенные даты.

По всем промежуточным остановкам на маршруте известны название, тип населенного пункта, время прибытия, отправления, время стоянки.

Необходимо учитывать, что местом посадки и высадки пассажира могут быть промежуточные пункты по маршруту.

БД должна содержать следующий минимальный набор сведений: Номер поезда. Название поезда. Тип поезда. Пункт назначения. Пункт назначения для проданного билета. Номер вагона. Тип вагона. Количество мест в вагоне. Цена билета. Дата отправления. Дата прибытия. Дата прибытия для пункта назначения проданного билета. Время отправления. Номер вагона в поезде. Номер билета. Место. Тип места. Фамилия пассажира. Имя пассажира. Отчество пассажира. Паспортные данные.

Задание 4. Создать хранимые процедуры:

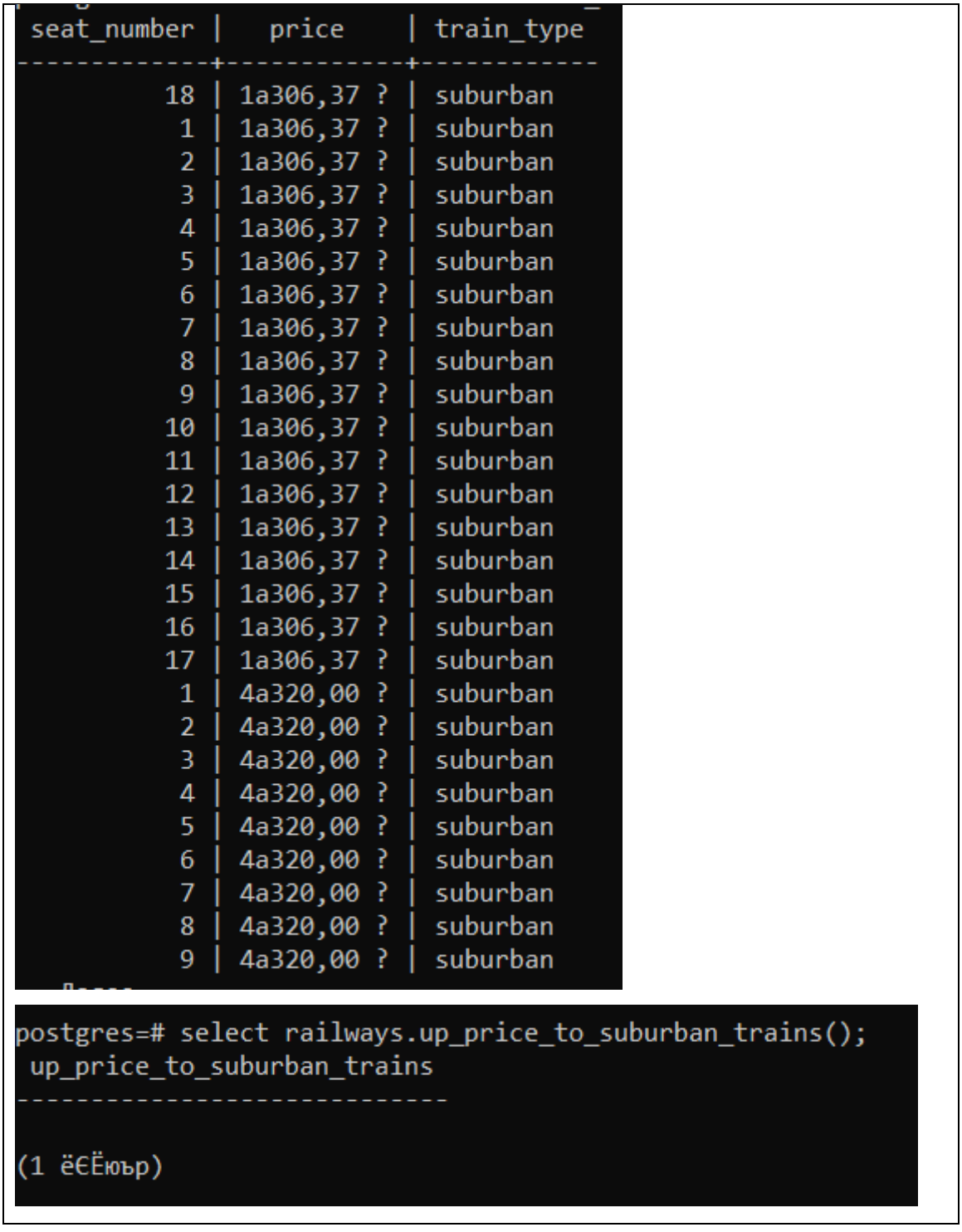
1. Для повышения цен в пригородные поезда на 20%.
2. Для создания нового рейса на поезд.
3. Для формирования общей выручки по продаже билетов за сутки.

Задание 5. Создать необходимые триггеры

Выполнение

Создание хранимых процедур:

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



```
 seat_number |   price    | train_type
-------------+------------+------------
          18 | 1a306,37 ? | suburban
           1 | 1a306,37 ? | suburban
           2 | 1a306,37 ? | suburban
           3 | 1a306,37 ? | suburban
           4 | 1a306,37 ? | suburban
           5 | 1a306,37 ? | suburban
           6 | 1a306,37 ? | suburban
           7 | 1a306,37 ? | suburban
           8 | 1a306,37 ? | suburban
           9 | 1a306,37 ? | suburban
          10 | 1a306,37 ? | suburban
          11 | 1a306,37 ? | suburban
          12 | 1a306,37 ? | suburban
          13 | 1a306,37 ? | suburban
          14 | 1a306,37 ? | suburban
          15 | 1a306,37 ? | suburban
          16 | 1a306,37 ? | suburban
          17 | 1a306,37 ? | suburban
           1 | 4a320,00 ? | suburban
           2 | 4a320,00 ? | suburban
           3 | 4a320,00 ? | suburban
           4 | 4a320,00 ? | suburban
           5 | 4a320,00 ? | suburban
           6 | 4a320,00 ? | suburban
           7 | 4a320,00 ? | suburban
           8 | 4a320,00 ? | suburban
           9 | 4a320,00 ? | suburban
```

```
postgres=# select railways.up_price_to_suburban_trains();
 up_price_to_suburban_trains
-----------------------------

(1 ёЄЁюър)
```

seat_number	price	train_type
18	1a567,64 ?	suburban
1	1a567,64 ?	suburban
2	1a567,64 ?	suburban
3	1a567,64 ?	suburban
4	1a567,64 ?	suburban
5	1a567,64 ?	suburban
6	1a567,64 ?	suburban
7	1a567,64 ?	suburban
8	1a567,64 ?	suburban
9	1a567,64 ?	suburban
10	1a567,64 ?	suburban
11	1a567,64 ?	suburban
12	1a567,64 ?	suburban
13	1a567,64 ?	suburban
14	1a567,64 ?	suburban
15	1a567,64 ?	suburban
16	1a567,64 ?	suburban
17	1a567,64 ?	suburban
1	5a184,00 ?	suburban
2	5a184,00 ?	suburban
3	5a184,00 ?	suburban
4	5a184,00 ?	suburban
5	5a184,00 ?	suburban
6	5a184,00 ?	suburban
7	5a184,00 ?	suburban
8	5a184,00 ?	suburban
9	5a184,00 ?	suburban
10	5a184,00 ?	suburban
11	5a184,00 ?	suburban
12	5a184,00 ?	suburban
13	5a184,00 ?	suburban
14	5a184,00 ?	suburban
15	5a184,00 ?	suburban
16	5a184,00 ?	suburban
17	5a184,00 ?	suburban
18	5a184,00 ?	suburban
19	5a184,00 ?	suburban
20	5a184,00 ?	suburban
21	5a184,00 ?	suburban
22	5a184,00 ?	suburban
23	5a184,00 ?	suburban
24	5a184,00 ?	suburban
25	5a184,00 ?	suburban

```
postgres=# create or replace function railways.up_price_to_suburban_trains(
postgres(# ) returns void as $up_price_to_suburban_trains$
postgres$# begin
postgres$#             UPDATE railways.seats
postgres$#             SET price = price * 1.2
postgres$#             WHERE scheduled_train_carriage_id in
postgres$#             (
postgres$#                 select
postgres$#                     scheduled_train_carriages.scheduled_train_carriage_id
postgres$#                 from
postgres$#                     railways.trains,
postgres$#                     railways.timetable,
postgres$#                     railways.scheduled_train_carriages,
postgres$#                     railways.seats
postgres$#
postgres$#                 where
postgres$#                         train_type = 'suburban'
postgres$#                   and scheduled_train_carriages.scheduled_train_carriage_id = seats.scheduled_train_carriage_id
postgres$#                   and seats.is_empty = true
postgres$#                   and trains.train_id = timetable.train_id
postgres$#                   and railways.scheduled_train_carriages.scheduled_train_id = railways.timetable.scheduled_train_id
postgres$#
postgres$#             );
postgres$# end;
postgres$# $up_price_to_suburban_trains$ language plpgsql;
CREATE FUNCTION
```

2. Для создания нового рейса на поезд.

```
postgres=# select *
postgres-# from
postgres-#     railways.timetable
postgres-# where
postgres-#     timetable.train_id = 1;
 timetable_id | train_id |   status   |   departure_time    |    arrival_time     | scheduled_tra
--------------+----------+------------+---------------------+---------------------+--------------
            2 |        1 | departured | 2023-11-27 04:05:06 | 2023-11-27 08:05:00 |
           16 |        1 | scheduled  | 2023-11-27 04:05:06 | 2023-11-26 08:05:00 |
           14 |        1 | scheduled  | 2023-11-25 04:05:06 | 2023-11-26 08:05:06 |
            0 |        1 | departured | 2023-11-26 04:05:06 | 2023-11-26 08:05:00 |
         1003 |        1 | scheduled  | 2023-12-26 04:05:06 | 2023-12-26 08:05:00 |
         1006 |        1 | scheduled  | 2023-12-26 04:05:06 | 2023-12-26 08:05:00 |
            3 |        1 | scheduled  | 2023-10-31 04:05:06 | 2023-01-31 08:05:00 |
(7 строк)
```

```
postgres=# create or replace function railways.create_scheduled_train(
postgres(#     id_train        int,
postgres(#     _price          money,
postgres(#     _departure_time  timestamp,
postgres(#     _arrival_time    timestamp
postgres(# ) returns integer as $create_scheduled_train$
postgres$# declare
postgres$#     number_of_seats_1           integer;
postgres$#     number_of_seats_2           integer;
postgres$#     id_scheduled_train_carriage_1 integer;
postgres$#     id_scheduled_train_carriage_2 integer;
postgres$#     id__scheduled_train  integer;
postgres$# begin
postgres$#     id_scheduled_train_carriage_1 = random()*(16);
postgres$#     number_of_seats_1 = (
postgres$#         select
postgres$#             carriages_types.number_of_seats
postgres$#         from
postgres$#             railways.carriages_types,
postgres$#             railways.carriages,
postgres$#             railways.scheduled_train_carriages
postgres$#         where
postgres$#                 scheduled_train_carriages.scheduled_train_carriage_id = id_scheduled_train_carriage_1
postgres$#           and  scheduled_train_carriages.carriage_id = carriages.carriage_id
postgres$#           and  carriages.carriage_type_id = carriages_types.carriage_type_id
postgres$#     );
postgres$#     id_scheduled_train_carriage_2 = random()*(16);
postgres$#     number_of_seats_2 = (
postgres$#         select
postgres$#             carriages_types.number_of_seats
postgres$#         from
postgres$#             railways.carriages_types,
postgres$#             railways.carriages,
postgres$#             railways.scheduled_train_carriages
postgres$#         where
postgres$#                 scheduled_train_carriages.scheduled_train_carriage_id = id_scheduled_train_carriage_2
postgres$#           and  scheduled_train_carriages.carriage_id = carriages.carriage_id
postgres$#           and  carriages.carriage_type_id = carriages_types.carriage_type_id
postgres$#     );
postgres$#
postgres$#     if number_of_seats_1 is null then number_of_seats_1 = 0;
postgres$#     end if;
postgres$#     if number_of_seats_2 is null then number_of_seats_2 = 0;
postgres$#     end if;
postgres$#
postgres$#     id__scheduled_train = nextval('railways.scheduled_train_id_seq'::regclass);
postgres$#     insert into railways.scheduled_trains(scheduled_train_id)
```

```
postgres$#
postgres$#     insert into railways.timetable(timetable_id, train_id, status, departure_time, arrival_time, scheduled_train_id)
postgres$#     values (nextval('railways.scheduled_train_carriage_id_seq'::regclass), id_train, 'scheduled', _departure_time, _arrival_time, id__scheduled_train);
postgres$#
postgres$#
postgres$#     insert into railways.scheduled_train_carriages(scheduled_train_carriage_id, carriage_id, scheduled_train_id, carriage_order_number)
postgres$#     values (nextval('railways.scheduled_train_carriage_id_seq'::regclass), id_scheduled_train_carriage_1, id__scheduled_train, 1);
postgres$#
postgres$#     insert into railways.scheduled_train_carriages(scheduled_train_carriage_id, carriage_id, scheduled_train_id, carriage_order_number)
postgres$#     values (nextval('railways.scheduled_train_carriage_id_seq'::regclass), id_scheduled_train_carriage_2, id__scheduled_train, 2);
postgres$#
postgres$#     for i in 1..number_of_seats_1 loop
postgres$#             insert into railways.seats(seat_id, seat_number, is_empty, price, scheduled_train_carriage_id)
postgres$#             values (nextval('railways.seat_id_seq'::regclass), i, true, _price, id_scheduled_train_carriage_1);
postgres$#         end loop;
postgres$#
postgres$#     for i in 1..number_of_seats_2 loop
postgres$#             insert into railways.seats(seat_id, seat_number, is_empty, price, scheduled_train_carriage_id)
postgres$#             values (nextval('railways.seat_id_seq'::regclass), i, true, _price, id_scheduled_train_carriage_2);
postgres$#         end loop;
postgres$#
postgres$#     return number_of_seats_1 + number_of_seats_2;
postgres$# end;
postgres$# $create_scheduled_train$ language plpgsql;
CREATE FUNCTION
```

```
postgres=# select railways.create_scheduled_train(1, 3000::money, '2023-12-27 04:05:06'::timestamp, '2023-12-2
'::timestamp);
 create_scheduled_train
------------------------
                     40
(1 строка)
```

```
postgres=# select *
postgres-# from
postgres-#     railways.timetable
postgres-# where
postgres-#     timetable.train_id = 1;
 timetable_id | train_id |   status   |   departure_time    |    arrival_time     | scheduled_trai
--------------+----------+------------+---------------------+---------------------+---------------
            2 |        1 | departured | 2023-11-27 04:05:06 | 2023-11-27 08:05:00 |
           16 |        1 | scheduled  | 2023-11-27 04:05:06 | 2023-11-26 08:05:00 |              1
           14 |        1 | scheduled  | 2023-11-25 04:05:06 | 2023-11-26 08:05:06 |
            0 |        1 | departured | 2023-11-26 04:05:06 | 2023-11-26 08:05:00 |
         1003 |        1 | scheduled  | 2023-12-26 04:05:06 | 2023-12-26 08:05:00 |              1
         1006 |        1 | scheduled  | 2023-12-26 04:05:06 | 2023-12-26 08:05:00 |              1
         1009 |        1 | scheduled  | 2023-12-27 04:05:06 | 2023-12-27 08:05:00 |              1
            3 |        1 | scheduled  | 2023-10-31 04:05:06 | 2023-01-31 08:05:00 |
(8 строк)
```

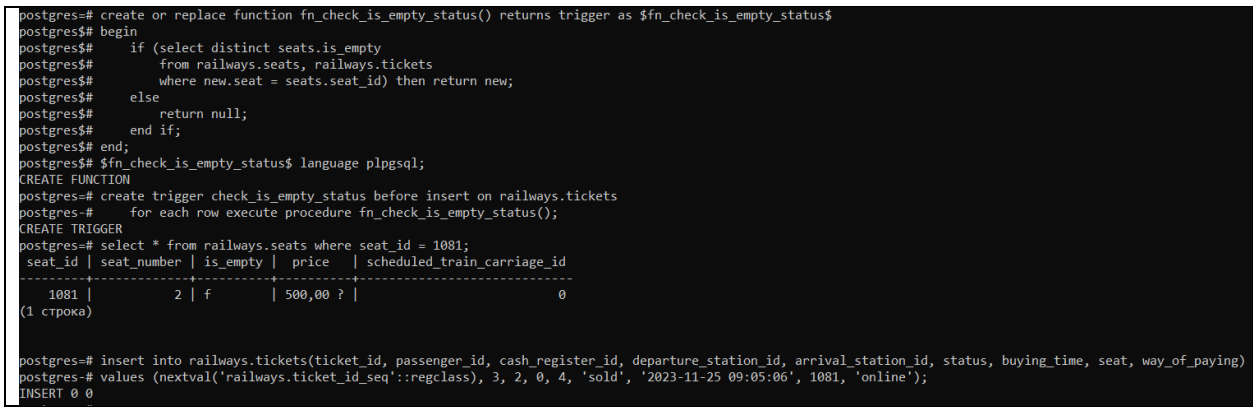
3. Для формирования общей выручки по продаже билетов за сутки.

```
postgres=# create or replace function railways.get_money_for_day(
postgres(#     data timestamp
postgres(# ) returns money as $get_money_for_day$
postgres$# declare
postgres$#     get_money_plus  money;
postgres$#     get_money_minus  money;
postgres$#     get_money  money;
postgres$# begin
postgres$#     get_money_plus = (
postgres$#         select sum(seats.price)
postgres$#         from railways.seats, railways.tickets
postgres$#         where
postgres$#             tickets.buying_time <= date_trunc('day', data + interval '1 day')
postgres$#             and tickets.buying_time >= date_trunc('day', data)
postgres$#             and tickets.seat = seats.seat_id
postgres$#             and tickets.status = 'sold'
postgres$#     );
postgres$#     get_money_minus = (
postgres$#         select sum(seats.price)
postgres$#         from railways.seats, railways.tickets
postgres$#         where
postgres$#                 tickets.buying_time <= date_trunc('day', data + interval '1 day')
postgres$#           and tickets.buying_time >= date_trunc('day', data)
postgres$#           and tickets.seat = seats.seat_id
postgres$#           and tickets.status = 'returned'
postgres$#     );
postgres$#     if get_money_plus::numeric is null then get_money_plus = 0::money;
postgres$#     end if;
postgres$#     if get_money_minus::numeric is null then get_money_minus = 0::money;
postgres$#     end if;
postgres$#     get_money = (get_money_plus::numeric - get_money_minus::numeric);
postgres$#     return get_money;
postgres$# end;
postgres$# $get_money_for_day$ language plpgsql;
CREATE FUNCTION
```

```
postgres=# select railways.get_money_for_day('2023-11-25 09:05:06'::timest
 get_money_for_day
-------------------
        1 000,00 ?
(1 строка)


postgres=# select tickets.buying_time, tickets.seat, seats.price
postgres-# from railways.tickets, railways.seats
postgres-# where tickets.seat = seats.seat_id;
     buying_time      | seat |  price
----------------------+------+----------
 2023-11-25 09:05:06 | 1080 | 500,00 ?
 2023-11-25 09:05:06 | 1081 | 500,00 ?
(2 строки)
```

Создание необходимых триггеров:



Вывод

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