**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: Будунов Б. С.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2024

# Оглавление

**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. Создать авторский триггер по варианту индивидуального задания.

**Выполнение**

**Создайте хранимые процедуры:**

1. **Для повышения цен в пригородние поезда на 20%**

```
CREATE OR REPLACE PROCEDURE
passenger_scheme.increase_ticket_prices_for_suburban_trains(IN max_price DECIMAL
DEFAULT null)
LANGUAGE plpgsql AS
$$
BEGIN
   IF max_price IS NOT NULL THEN
      UPDATE passenger_scheme.seat
      SET seat_price = seat_price * 1.2
      WHERE carriage_in_train_id IN
      (SELECT carriage_in_train_id
       FROM passenger_scheme.carriage_in_train
       WHERE train_number IN
       (SELECT train_number
        FROM passenger_scheme.train
        WHERE train_type = 'Пригородный')
       AND seat_price * 1.2 <= max_price);
   ELSE
      UPDATE passenger_scheme.seat
      SET seat_price = seat_price * 1.2
      WHERE carriage_in_train_id IN
      (SELECT carriage_in_train_id
       FROM passenger_scheme.carriage_in_train
       WHERE train_number IN
       (SELECT train_number
        FROM passenger_scheme.train
        WHERE train_type = 'Пригородный'));
   END IF;
   COMMIT;
END;
$$;
```

```
passenger=# CREATE OR REPLACE PROCEDURE passenger_scheme.increase_ticket_prices_for_suburban_trains(IN max_price DECIMAL
 DEFAULT null)
passenger-# LANGUAGE plpgsql AS
passenger-# $$
passenger$# BEGIN
passenger$#     IF max_price IS NOT NULL THEN
passenger$#         UPDATE passenger_scheme.seat
passenger$#         SET seat_price = seat_price * 1.2
passenger$#         WHERE carriage_in_train_id IN
passenger$#         (SELECT carriage_in_train_id
passenger$#          FROM passenger_scheme.carriage_in_train
passenger$#          WHERE train_number IN
passenger$#          (SELECT train_number
passenger$#           FROM passenger_scheme.train
passenger$#           WHERE train_type = 'Пригородный')
passenger$#          AND seat_price * 1.2 <= max_price);
passenger$#     ELSE
passenger$#         UPDATE passenger_scheme.seat
passenger$#         SET seat_price = seat_price * 1.2
passenger$#         WHERE carriage_in_train_id IN
passenger$#         (SELECT carriage_in_train_id
passenger$#          FROM passenger_scheme.carriage_in_train
passenger$#          WHERE train_number IN
passenger$#          (SELECT train_number
passenger$#           FROM passenger_scheme.train
passenger$#           WHERE train_type = 'Пригородный'));
passenger$#     END IF;
passenger$#     COMMIT;
passenger$# END;
passenger$# $$;
CREATE PROCEDURE
```



Таблица После кола (места стоили по 240)

```
passenger=# CALL  passenger_scheme.increase_ticket_prices_for_suburban_trains();
CALL
passenger=# SELECT * FROM  passenger_scheme.seat
passenger-# ;
 seat_number | seat_status | seat_price | seat_id | carriage_in_train_id
-------------+-------------+------------+---------+----------------------
           1 | Куплено     |       4000 |       1 |                    2
           1 | Куплено     |       3000 |       2 |                    3
           1 | Куплено     |       3000 |       3 |                    4
           1 | Куплено     |       4000 |       4 |                    5
           1 | Куплено     |      99999 |       5 |                    6
           2 | Куплено     |       4000 |      15 |                    2
           2 | Свободно    |       3000 |      16 |                    3
           2 | Куплено     |       3000 |      17 |                    4
           2 | Куплено     |       4000 |      18 |                    5
           2 | Куплено     |      99999 |      19 |                    6
           3 | Куплено     |       4000 |      29 |                    2
           3 | Свободно    |       3000 |      30 |                    3
           3 | Куплено     |       3000 |      31 |                    4
           3 | Куплено     |       4000 |      32 |                    5
           4 | Свободно    |       4000 |      41 |                    2
           4 | Свободно    |       3000 |      42 |                    3
           4 | Куплено     |       3000 |      43 |                    4
           4 | Куплено     |       4000 |      44 |                    5
           5 | Свободно    |       3000 |      57 |                    9
           1 | Куплено     |       2500 |      11 |                   14
           1 | Куплено     |       2000 |      12 |                   15
           1 | Куплено     |       2000 |      13 |                   16
           1 | Куплено     |       2500 |      14 |                   17
           2 | Свободно    |       2500 |      25 |                   14
           2 | Свободно    |       2000 |      26 |                   15
           2 | Куплено     |       2000 |      27 |                   16
           2 | Куплено     |       2500 |      28 |                   17
           3 | Куплено     |       2500 |      37 |                   14
           5 | Свободно    |        288 |      65 |                   21
           4 | Свободно    |        288 |      66 |                   21
           3 | Свободно    |        288 |      67 |                   21
           2 | Свободно    |        288 |      68 |                   21
           1 | Свободно    |        288 |      69 |                   21
           3 | Свободно    |       2000 |      38 |                   15
```

- **Для формирования выручки за конкретный день**

```
CREATE OR REPLACE PROCEDURE
passenger_scheme.calculate_revenue_for_special_day(
        IN special_date date,
        INOUT p_total_revenue integer DEFAULT 0)
    LANGUAGE plpgsql
    AS $$
     BEGIN
       SELECT SUM(ticket_price)
       FROM passenger_scheme.ticket t
       WHERE DATE(t.buy_date) = special_date
    INTO p_total_revenue;
    END;
    $$;
```

```
passenger=# CREATE OR REPLACE PROCEDURE passenger_scheme.calculate_revenue_for_special_day(IN special_date DATE, INOUT p_total_revenue INT DEFAULT 0)
passenger-# LANGUAGE plpgsql AS
passenger-# $$ BEGIN
passenger$#     SELECT SUM(ticket_price)
passenger$#     FROM passenger_scheme.ticket t
passenger$#     WHERE DATE(t.buy_date) = special_date
passenger$# INTO p_total_revenue;
passenger$# END;
passenger$# $$;
CREATE PROCEDURE
passenger=# CALL passenger_scheme.calculate_revenue_for_special_day('2024-01-24');
 p_total_revenue
-----------------
          199998
(1 строка)


passenger=#
```

- **Для определения нового рейса на поезд**

```
    CREATE OR REPLACE PROCEDURE
passenger_scheme.update_flight_for_train (IN flightNumber
varchar,IN newTrainNumber INT)
    LANGUAGE plpgsql AS
    $$ BEGIN
       UPDATE passenger_scheme.train
       SET flight_number = flightNumber
       WHERE train_number = newTrainNumber AND
train_status = 'В депо' AND flightNumber IN (
         SELECT flight_number FROM
passneger_scheme.schedule);
    END;
    $$;
```

```
passenger=# CREATE OR REPLACE PROCEDURE passenger_scheme.update_flight_for_train (IN flightNumber varchar,IN newTrainNumber INT)
passenger-# LANGUAGE plpgsql AS
passenger-# $$ BEGIN
passenger$#     UPDATE passenger_scheme.train
passenger$#     SET flight_number = flightNumber
passenger$#     WHERE train_number = newTrainNumber AND train_status = 'В депо' AND flightNumber IN (
passenger$# SELECT flight_number FROM passenger_scheme.schedule);
passenger$# END;
passenger$# $$;
CREATE PROCEDURE
passenger=# CALL passenger_scheme.update_flight_for_train('NEW_FLIGHT', 8)
passenger-# ;
CALL
passenger=# SELECT * FROM passenger_scheme.train;
 train_number |     departure_time     |      arrival_time      |    train_type    | train_status | train_name | flight_number
--------------+------------------------+------------------------+------------------+--------------+------------+---------------
            3 | 2024-01-31 19:00:00+03 | 2024-02-01 05:30:00+03 | Скорый           | В депо       | Ласточка   | TST01
            5 | 2024-01-17 09:30:00+03 | 2024-01-17 18:00:00+03 | Междугородний    | В пути       | Поезд_02   | flt_2
            6 | 2024-01-18 09:30:00+03 | 2024-01-18 18:00:00+03 | Междугородний    | В депо       | Поезд_03   | flt_3
            7 | 2024-01-18 02:00:00+03 | 2024-01-18 15:00:00+03 | Междугородний    | В депо       | Поезд_04   | flt_4
            4 | 2024-01-17 02:00:00+03 | 2024-01-17 15:00:00+03 | Междугородний    | Прибыл       | Поезд_01   | flt_1
            9 | 2024-02-22 14:00:00+03 | 2024-02-22 15:00:00+03 | Пригородный      | В депо       | Ласточка   | prig_1
            8 | 2024-01-25 17:45:00+03 | 2024-01-26 05:00:00+03 | Спец. назначения | В депо       | Дрозд      | NEW_FLIGHT
(7 строк)
```

**Создайте необходимые триггеры:**

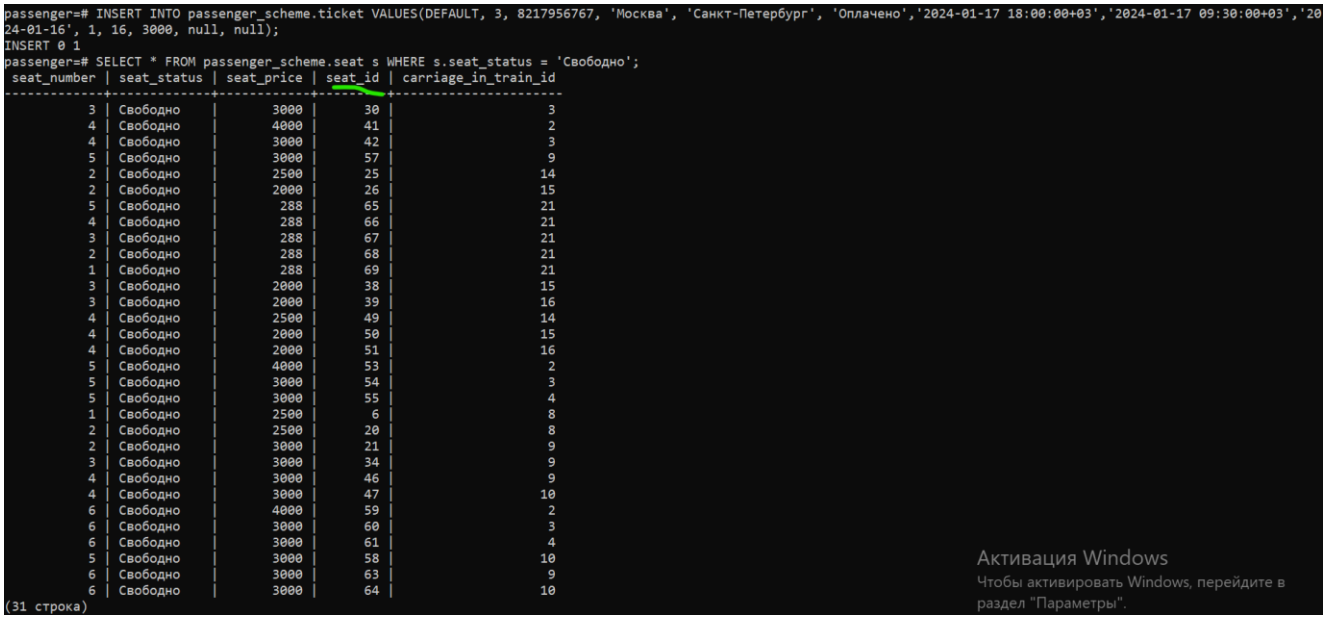
- Триггер для смены статуса места в соответствии со статусом оплаты билета

```
CREATE OR REPLACE FUNCTION passenger_scheme.update_seat_status()
RETURNS TRIGGER AS $$
BEGIN
   IF TG_OP = 'INSERT' AND NEW. payment_status = 'Оплачен' THEN
      UPDATE passenger_scheme.seat
      SET seat_status = 'Куплено'
      WHERE seat_id = NEW.seat_id AND seat_status = 'Свободно';
   ELSIF TG_OP = 'UPDATE' AND NEW.payment_status = 'Оплачен' THEN
      UPDATE passenger_scheme.seat
      SET seat_status = 'Куплено'
      WHERE seat_id = NEW.seat_id AND seat_status = 'Свободно';
   ELSIF TG_OP = 'UPDATE' AND NEW. payment_status = 'Возврат' THEN
      UPDATE passenger_scheme.seat
      SET seat_status = 'Свободно'
      WHERE seat_id = NEW.seat_id AND seat_status = 'Куплено';
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER after_insert_update_ticket
AFTER INSERT OR UPDATE ON passenger_scheme.ticket
FOR EACH ROW
EXECUTE FUNCTION passenger_scheme.update_seat_status();
```

```
passenger=# CREATE OR REPLACE FUNCTION passenger_scheme.update_seat_status()
passenger-# RETURNS TRIGGER AS $$
passenger$# BEGIN
passenger$#     IF TG_OP = 'INSERT' AND NEW.status = 'Оплачен' THEN
passenger$#         UPDATE passenger_scheme.seat
passenger$#         SET seat_status = 'Куплено'
passenger$#         WHERE seat_id = NEW.seat_id AND seat_status = 'Свободно';
passenger$#     ELSIF TG_OP = 'UPDATE' AND NEW.status = 'Оплачен' THEN
passenger$#         UPDATE passenger_scheme.seat
passenger$#         SET seat_status = 'Куплено'
passenger$#         WHERE seat_id = NEW.seat_id AND seat_status = 'Свободно';
passenger$#     ELSIF TG_OP = 'UPDATE' AND NEW.status = 'Возврат' THEN
passenger$#         UPDATE passenger_scheme.seat
passenger$#         SET seat_status = 'Свободно'
passenger$#         WHERE seat_id = NEW.seat_id AND seat_status = 'Куплено';
passenger$#     END IF;
passenger$#     RETURN NEW;
passenger$# END;
passenger$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
passenger=#
passenger=# CREATE TRIGGER after_insert_update_ticket
passenger-# AFTER INSERT OR UPDATE ON passenger_scheme.ticket
passenger-# FOR EACH ROW
passenger-# EXECUTE FUNCTION passenger_scheme.update_seat_status();
CREATE TRIGGER
passenger=#
```

```
assenger=# SELECT * FROM passenger_scheme.seat s WHERE s.seat_status = 'Свободно';
 seat_number | seat_status | seat_price | seat_id | carriage_in_train_id
-------------+-------------+------------+---------+----------------------
           2 | Свободно    |       3000 |      16 |                    3
           3 | Свободно    |       3000 |      30 |                    3
           4 | Свободно    |       4000 |      41 |                    2
           4 | Свободно    |       3000 |      42 |                    3
           5 | Свободно    |       3000 |      57 |                    9
           2 | Свободно    |       2500 |      25 |                   14
           2 | Свободно    |       2000 |      26 |                   15
           5 | Свободно    |        288 |      65 |                   21
           4 | Свободно    |        288 |      66 |                   21
           3 | Свободно    |        288 |      67 |                   21
           2 | Свободно    |        288 |      68 |                   21
           1 | Свободно    |        288 |      69 |                   21
           3 | Свободно    |       2000 |      38 |                   15
           3 | Свободно    |       2000 |      39 |                   16
           4 | Свободно    |       2500 |      49 |                   14
           4 | Свободно    |       2000 |      50 |                   15
           4 | Свободно    |       2000 |      51 |                   16
           5 | Свободно    |       4000 |      53 |                    2
           5 | Свободно    |       3000 |      54 |                    3
           5 | Свободно    |       3000 |      55 |                    4
           1 | Свободно    |       2500 |       6 |                    8
```



```
passenger=# INSERT INTO passenger_scheme.ticket VALUES(DEFAULT, 3, 8217956767, 'Москва', 'Санкт-Петербург', 'Оплачено','2024-01-17 18:00:00+03','2024-01-17 09:30:00+03','20
24-01-16', 1, 16, 3000, null, null);
INSERT 0 1
passenger=# SELECT * FROM passenger_scheme.seat s WHERE s.seat_status = 'Свободно';
 seat_number | seat_status | seat_price | seat_id | carriage_in_train_id
-------------+-------------+------------+---------+----------------------
           3 | Свободно    |       3000 |      30 |                    3
           4 | Свободно    |       4000 |      41 |                    2
           4 | Свободно    |       3000 |      42 |                    3
           5 | Свободно    |       3000 |      57 |                    9
           2 | Свободно    |       2500 |      25 |                   14
           2 | Свободно    |       2000 |      26 |                   15
           5 | Свободно    |        288 |      65 |                   21
           4 | Свободно    |        288 |      66 |                   21
           3 | Свободно    |        288 |      67 |                   21
           2 | Свободно    |        288 |      68 |                   21
           1 | Свободно    |        288 |      69 |                   21
           3 | Свободно    |       2000 |      38 |                   15
           3 | Свободно    |       2000 |      39 |                   16
           4 | Свободно    |       2500 |      49 |                   14
           4 | Свободно    |       2000 |      50 |                   15
           4 | Свободно    |       2000 |      51 |                   16
           5 | Свободно    |       4000 |      53 |                    2
           5 | Свободно    |       3000 |      54 |                    3
           5 | Свободно    |       3000 |      55 |                    4
           1 | Свободно    |       2500 |       6 |                    8
           2 | Свободно    |       2500 |      20 |                    8
           2 | Свободно    |       3000 |      21 |                    9
           3 | Свободно    |       3000 |      34 |                    9
           4 | Свободно    |       3000 |      46 |                    9
           4 | Свободно    |       3000 |      47 |                   10
           6 | Свободно    |       4000 |      59 |                    2
           6 | Свободно    |       3000 |      60 |                    3
           6 | Свободно    |       3000 |      61 |                    4
           5 | Свободно    |       3000 |      58 |                   10
           6 | Свободно    |       3000 |      63 |                    9
           6 | Свободно    |       3000 |      64 |                   10
(31 строка)
```



- Триггер для определения стоимости билета по стоимости места и размера скидки

```
CREATE OR REPLACE FUNCTION passenger_scheme.calculate_ticket_price()
RETURNS TRIGGER AS $$
BEGIN
   DECLARE
      discount_amount DECIMAL;
      start_seat_price DECIMAL;
      discounted_price DECIMAL;
      startDate DATE;
      endDate DATE;
   BEGIN
      SELECT amount_of_discount, DATE(begin_date), DATE(end_date) INTO
discount_amount, startDate, endDate FROM passenger_scheme.discount WHERE
id_discount = NEW.id_discount;

      SELECT seat_price INTO start_seat_price FROM passenger_scheme.seat WHERE
seat_id = NEW.seat_id;

      IF CURRENT_DATE BETWEEN startDate AND endDate THEN
         discounted_price := start_seat_price * (1 - discount_amount/100);
      ELSE
         discounted_price := start_seat_price;
      END IF;

      NEW.ticket_price := discounted_price;
      RETURN NEW;
   END;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER calculate_ticket_price_trigger
BEFORE INSERT ON passenger_scheme.ticket
FOR EACH ROW
EXECUTE FUNCTION passenger_scheme.calculate_ticket_price();
```

```
passenger=# CREATE OR REPLACE FUNCTION passenger_scheme.calculate_ticket_price()
passenger-# RETURNS TRIGGER AS $$
passenger$# BEGIN
passenger$#     DECLARE
passenger$#         discount_amount DECIMAL;
passenger$#         seat_price DECIMAL;
passenger$#         discounted_price DECIMAL;
passenger$#     BEGIN
passenger$#         SELECT amount_of_discount INTO discount_amount FROM passenger_scheme.discount WHERE id_discount = NE
W.id_discount;
passenger$#
passenger$#         SELECT seat_price INTO seat_price FROM passenger_scheme.seat WHERE seat_id = NEW.seat_id;
passenger$#
passenger$#         IF CURRENT_DATE BETWEEN passenger_scheme.discount.start_date AND passenger_scheme.discount.end_date
THEN
passenger$#             discounted_price := seat_price * (1 - discount_amount);
passenger$#         ELSE
passenger$#             discounted_price := seat_price;
passenger$#         END IF;
passenger$#
passenger$#         NEW.ticket_price := discounted_price;
passenger$#         RETURN NEW;
passenger$#     END;
passenger$# END;
passenger$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
passenger=#
passenger=# CREATE TRIGGER calculate_ticket_price_trigger
passenger-# BEFORE INSERT ON passenger_scheme.ticket
passenger-# FOR EACH ROW
passenger-# EXECUTE FUNCTION passenger_scheme.calculate_ticket_price();
CREATE TRIGGER
```



```
passenger=# INSERT INTO passenger_scheme.ticket VALUES(DEFAULT, 3, 8217956767, 'Москва', 'Санкт-Петербург', 'Оплачен', '2024-01-17 18:00:00', '2024-01-17 09:30:00', '2024-
01-11', 2, 30);
INSERT 0 1
passenger=# SELECT s.seat_id, s.seat_price, sch.arrival_point, sch.departure_point, sch.arrival_time, sch.departure_time
passenger-# FROM passenger_scheme.seat s, passenger_scheme.carriage_in_train cit,
passenger-#  passenger_scheme.train t, passenger_scheme.schedule sch
passenger-# WHERE s.seat_id = 30 AND s.carriage_in_train_id = cit.carriage_in_train_id AND
passenger-# cit.train_number = t.train_number AND t.flight_number = sch.flight_number;
 seat_id | seat_price | arrival_point | departure_point |      arrival_time      |     departure_time
---------+------------+---------------+-----------------+------------------------+------------------------
      30 |       3000 | Москва        | Санкт-Петербург | 2024-01-17 18:00:00+03 | 2024-01-17 09:30:00+03
(1 строка)


passenger=# SELECT * FROM passenger_scheme.ticket WHERE seat_id = 30;
 ticket_number | cashbox_number | passport_number | arrival_point | departure_point | payment_status |      arrival_time      |     departure_time     |        buy_date
    | id_discount | seat_id | ticket_price | arrival_point_id | departure_point_id
---------------+----------------+-----------------+---------------+-----------------+----------------+------------------------+------------------------+----------------
----+-------------+---------+--------------+------------------+--------------------
            47 |              3 |      8217956767 | Москва        | Санкт-Петербург | Оплачен        | 2024-01-17 18:00:00+03 | 2024-01-17 09:30:00+03 | 2024-01-11 00:00:00
+03 |           2 |      30 |         2400 |                  |
(1 строка)


passenger=# SELECT ticket_price FROM passenger_scheme.ticket WHERE seat_id = 30;
 ticket_price
--------------
         2400
(1 строка)


passenger=#
```



**Вывод**

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