Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет: **Инфокоммуникационных технологий**
Образовательная программа: **Интеллектуальные системы в гуманитарной сфере**
Направление подготовки: **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Лабораторная работа №2
«Работа с триггерами и функциями в PostgreSQL»

по дисциплине:
«Базы данных»

Выполнила:
Редькина Любовь Александровна,
группа **К32422**
Преподаватель:
Говорова Марина Михайловна

Санкт-Петербург
2023

СОДЕРЖАНИЕ:

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4, PSQL Shell

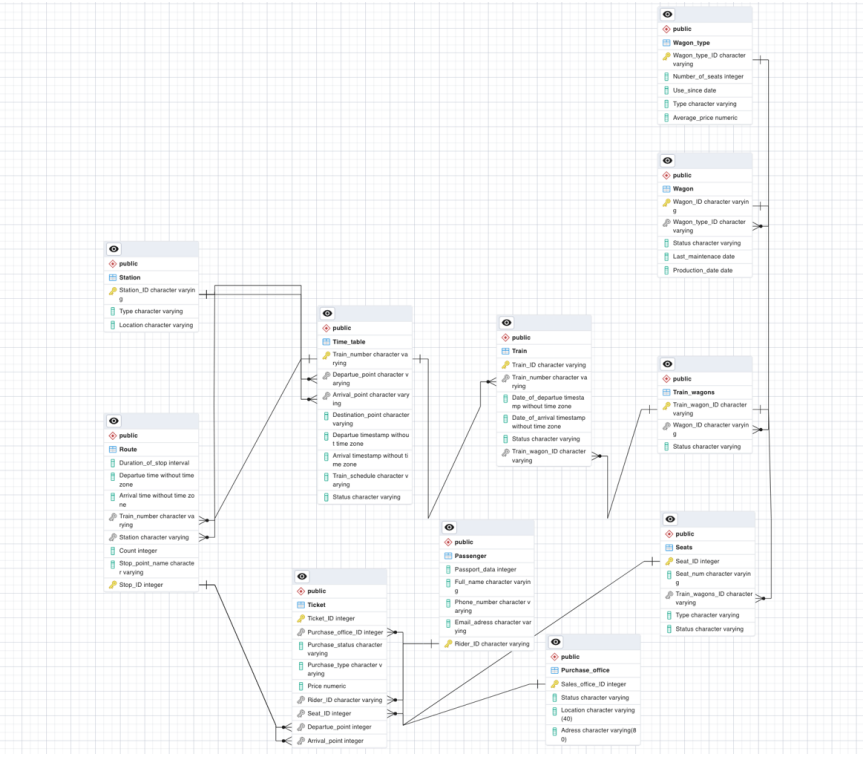
Практическое задание:

Вариант 1

1. 2.Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
2. 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.

Индивидуальное задание:
Используя данные предыдущей лабораторной работы (Создание базы данных для варианта 6 «Пассажир»), реализовать различные процедуры, функции и триггеры к БД.

Рис. 1 Графическое представление реализованной ранее БД



Выполнение:

1.1 Реализация функции для подсчета общего количества пассажиров

Query Query History

```
1  CREATE OR REPLACE FUNCTION public.get_passenger_count(
2      )
3      RETURNS integer
4      LANGUAGE plpgsql AS $$
5  DECLARE
6      passenger_count INTEGER;
7  BEGIN
8      SELECT COUNT(*) INTO passenger_count FROM public."Passenger";
9
10     RETURN passenger_count;
11 END $$;
```

Результат работы функции:

```
Passenger=# SELECT * FROM get_passenger_count();
 get_passenger_count
---------------------
                  11
(1 row)
```

1.2 Реализация функции для вывода пассажиров, прибывающих/отправляющихся на станцию «12»

Query Query History

```
1  CREATE OR REPLACE FUNCTION public.get_passengers_by_station(
2      )
3      RETURNS TABLE(full_name character varying,
4                    rider_id character varying)
5      LANGUAGE plpgsql AS $$
6  BEGIN
7      RETURN QUERY
8      SELECT p."Full_name", p."Rider_ID"
9      FROM public."Ticket" t
10     INNER JOIN public."Passenger" p ON t."Rider_ID" = p."Rider_ID"
11     INNER JOIN public."Route" r ON t."Departue_point" = r."Stop_ID"
12                                 OR t."Arrival_point" = r."Stop_ID"
13     WHERE r."Stop_ID" = '12';
14 END $$;
```

Результат работы функции:

```
Passenger=# SELECT * FROM get_passengers_by_station();
        full_name         | rider_id
--------------------------+----------
 Иванов Иван Иванович     | Rider02
 Сидоров Сидор Сидорович  | Rider04
 Иванов Иван Иванович     | Rider05
 Ульянова Ульяна          | Rider07
 Макаров Макар            | Rider08
(5 rows)
```

1.3 Реализация функции для вывода вместимости состава

Query | Query History

```sql
CREATE OR REPLACE FUNCTION public.get_train_wagon_info(
    train_wagon_id character varying)
    RETURNS TABLE(coach_id character varying,
                  wagon_type character varying,
                  capacity integer)
    LANGUAGE plpgsql AS $$
BEGIN
    RETURN QUERY
    SELECT tw."Train_wagon_ID", wt."Wagon_type_ID", wt."Number_of_seats"
    FROM public."Train_wagons" tw
    INNER JOIN public."Wagon" w
        ON tw."Wagon_ID" = w."Wagon_ID"
    INNER JOIN public."Wagon_type" wt
        ON w."Wagon_type_ID" = wt."Wagon_type_ID"
    WHERE tw."Train_wagon_ID" = train_wagon_id;
END $$;
```

Результат работы функции:

```
Passenger=# SELECT * FROM get_train_wagon_info('0');
 coach_id | wagon_type | capacity
----------+------------+----------
 0        | WT10       |       34
(1 row)

Passenger=# SELECT * FROM get_train_wagon_info('1');
 coach_id | wagon_type | capacity
----------+------------+----------
 1        | WT01       |       54
(1 row)

Passenger=# SELECT * FROM get_train_wagon_info('2');
 coach_id | wagon_type | capacity
----------+------------+----------
 2        | WT02       |       48
(1 row)
```

2.1 Реализация процедуры для добавления информации о пассажире

Query Query History

```
CREATE OR REPLACE PROCEDURE public.add_passenger(
    IN passport_data integer,
    IN full_name character varying,
    IN phone_number character varying,
    IN email_address character varying,
    IN rider_id character varying)
LANGUAGE plpgsql AS $$
BEGIN
    INSERT INTO public."Passenger"(
        "Passport_data",
        "Full_name",
        "Phone_number",
        "Email_adress",
        "Rider_ID")
    VALUES (passport_data,
            full_name,
            phone_number,
            email_address,
            rider_id);
END $$;
```

Результат работы процедуры:

```
Passenger=# SELECT * FROM public."Passenger";
 Passport_data |        Full_name         |   Phone_number   |      Email_adress        | Rider_ID
---------------+--------------------------+------------------+--------------------------+----------
      12891631 | Королева Екатерина       | +700000000       | queenmary@example.com    | Rider01
     123456789 | Иванов Иван Иванович     | +123456789       | ivan@example.com         | Rider02
     987654321 | Петров Петр Петрович     | +987654321       | petr@example.com         | Rider03
     456789123 | Сидоров Сидор Сидорович  | +456789123       | sidor@example.com        | Rider04
    1234567890 | Иванов Иван Иванович     | 1234567890       | ivanov@example.com       | Rider05
     678901234 | Зайцев Лев               | 89118448299      | leva@example.com         | Rider06
       3456789 | Ульянова Ульяна          | +79118448266     | ulala@example.com        | Rider07
      67896789 | Макаров Макар            | +79633276263     | makarkarkar@example.com  | Rider08
       4435672 | Валерьева Валерия        | 88129110203      | lero4ka@example.com      | Rider09
       5567412 | Екупчик Марина           | +7999999999      | marinka@example.com      | Rider10
             0 | Кира Кирилловна          | +7-911-354-44-44 | kirka@example.com        | Rider11
(11 rows)

Passenger=# CALL add_passenger(2, 'Говоров Антон', '89715643412', 'govorokk@example.
com', 'Rider12');
CALL
Passenger=# SELECT * FROM public."Passenger";
 Passport_data |        Full_name         |   Phone_number   |      Email_adress        | Rider_ID
---------------+--------------------------+------------------+--------------------------+----------
      12891631 | Королева Екатерина       | +700000000       | queenmary@example.com    | Rider01
     123456789 | Иванов Иван Иванович     | +123456789       | ivan@example.com         | Rider02
     987654321 | Петров Петр Петрович     | +987654321       | petr@example.com         | Rider03
     456789123 | Сидоров Сидор Сидорович  | +456789123       | sidor@example.com        | Rider04
    1234567890 | Иванов Иван Иванович     | 1234567890       | ivanov@example.com       | Rider05
     678901234 | Зайцев Лев               | 89118448299      | leva@example.com         | Rider06
       3456789 | Ульянова Ульяна          | +79118448266     | ulala@example.com        | Rider07
      67896789 | Макаров Макар            | +79633276263     | makarkarkar@example.com  | Rider08
       4435672 | Валерьева Валерия        | 88129110203      | lero4ka@example.com      | Rider09
       5567412 | Екупчик Марина           | +7999999999      | marinka@example.com      | Rider10
             0 | Кира Кирилловна          | +7-911-354-44-44 | kirka@example.com        | Rider11
             2 | Говоров Антон            | 89715643412      | govorokk@example.com     | Rider12
(12 rows)
```

2.2 Реализация процедуры для снижения стоимости билета на 10%, если билет приобретен онлайн

Query | Query History

```
CREATE OR REPLACE PROCEDURE public.decrease_ticket_price(
    )
LANGUAGE plpgsql AS $$
BEGIN
    UPDATE public."Ticket"
    SET "Price" = "Price" * 0.9
    WHERE "Purchase_type" = 'онлайн';
END $$;
```

Результат работы процедуры:

```
Passenger=# SELECT * FROM public."Ticket";
 Ticket_ID | Purchase_office_ID | Purchase_status | Purchase_type | Price | Rider_ID | Seat_ID | Departue_point | Arrival_point
-----------+--------------------+-----------------+---------------+-------+----------+---------+----------------+---------------
         9 |                  2 | забронировано   | касса         |  8000 | Rider09  |       8 |              7 |            11
         1 |                  1 | выкуплено       | онлайн        |  2300 | Rider02  |       1 |              7 |             8
         2 |                  2 | забронировано   | онлайн        |  1250 | Rider02  |       2 |              7 |            12
         4 |                  8 | выкуплено       | онлайн        |   899 | Rider04  |       4 |             12 |            11
         6 |                  2 | забронировано   | онлайн        |  2570 | Rider01  |       6 |              7 |            11
         7 |                 10 | выкуплено       | онлайн        |   340 | Rider07  |       7 |              8 |            12
         8 |                  6 | выкуплено       | онлайн        |  1900 | Rider08  |       8 |              8 |            12
         3 |                  7 | выкуплено       | касса         |  4000 | Rider05  |       3 |             13 |            10
         5 |                  3 | забронировано   | касса         |  1250 | Rider05  |       5 |             12 |            13
(9 rows)

Passenger=# CALL decrease_ticket_price();
CALL
Passenger=# SELECT * FROM public."Ticket";
 Ticket_ID | Purchase_office_ID | Purchase_status | Purchase_type | Price  | Rider_ID | Seat_ID | Departue_point | Arrival_point
-----------+--------------------+-----------------+---------------+--------+----------+---------+----------------+---------------
         9 |                  2 | забронировано   | касса         |   8000 | Rider09  |       8 |              7 |            11
         1 |                  1 | выкуплено       | онлайн        | 2070.0 | Rider02  |       1 |              7 |             8
         2 |                  2 | забронировано   | онлайн        | 1125.0 | Rider02  |       2 |              7 |            12
         4 |                  8 | выкуплено       | онлайн        |  809.1 | Rider04  |       4 |             12 |            11
         6 |                  2 | забронировано   | онлайн        | 2313.0 | Rider01  |       6 |              7 |            11
         7 |                 10 | выкуплено       | онлайн        |  306.0 | Rider07  |       7 |              8 |            12
         8 |                  6 | выкуплено       | онлайн        | 1710.0 | Rider08  |       8 |              8 |            12
         3 |                  7 | выкуплено       | касса         |   4000 | Rider05  |       3 |             13 |            10
         5 |                  3 | забронировано   | касса         |   1250 | Rider05  |       5 |             12 |            13
(9 rows)
```

3.1 Реализация тгриггера для добавления информации о месте в таблицу Seat

Query | Query History

```
CREATE OR REPLACE FUNCTION public.log_seat_insert()
    RETURNS trigger
    LANGUAGE plpgsql AS $$
BEGIN
    INSERT INTO
        public."Seat_Log" ("Seat_ID", "Operation", "Timestamp")
    VALUES (NEW."Seat_ID", 'INSERT', NOW());
    RETURN NEW;
END $$;
```

Результат работы триггера:

```
Passenger=# SELECT * FROM public."Seat_Log";
 Seat_ID | Operation | Timestamp
---------+-----------+-----------
(0 rows)

Passenger=# INSERT INTO public."Seats" ("Seat_ID", "Seat_num", "Train_wagons_ID", "Type", "Status")
VALUES (14, '6C', '3', 'сидячее', 'забронировано');
INSERT 0 1
Passenger=# SELECT * FROM public."Seat_Log";
 Seat_ID | Operation |    Timestamp
---------+-----------+-----------------
      14 | INSERT    | 21:10:12.372481
(1 row)
```

3.2 Реализация триггера для операций над таблицей Seat

Query Query History

```
CREATE OR REPLACE FUNCTION public.changes_in_seats()
    RETURNS trigger
    LANGUAGE plpgsql
AS $$
BEGIN
INSERT INTO public."Seats_log" VALUES (
    TG_NAME,
    TG_OP,
    NEW."Seat_ID",
    NEW."Seat_num",
    NEW."Train_wagons_ID",
    NEW."Type",
    NEW."Status",
    NOW());
RETURN NEW;
END $$;
```

Query Query History

```
CREATE TRIGGER log_changes_in_seats
    AFTER INSERT OR DELETE OR UPDATE
    ON public."Seats"
    FOR EACH ROW
    EXECUTE FUNCTION public.changes_in_seats();
```

Результат работы триггера:

```
Passenger=# SELECT * FROM public."Seats_log";
 Target | Operation | Seat_ID | Seat_num | Train_wagons_ID | Type | Status | Date
--------+-----------+---------+----------+-----------------+------+--------+------
(0 rows)

Passenger=# SELECT * FROM public."Seats";
 Seat_ID | Seat_num | Train_wagons_ID |   Type    |     Status
---------+----------+-----------------+-----------+---------------
       1 | 1A       | 1               | сидячее   | выкуплено
       2 | 2B       | 2               | сидячее   | забронировано
       3 | 3C       | 3               | спальное  | свободно
       5 | 5A       | 5               | спальное  | выкуплено
       6 | 1A       | 6               | сидячее   | забронировано
       8 | 3A       | 8               | спальное  | свободно
       4 | 4D       | 4               | спальное  | выкуплено
       7 | 1A       | 7               | сидячее   | выкуплено
       9 | 5A       | 9               | сидячее   | забронировано
      10 | 6B       | 9               | спальное  | выкуплено
      11 | 5A       | 9               | сидячее   | забронировано
      12 | 12A      | 4               | спальное  | выкуплено
      13 | 3C       | 7               | сидячее   | выкуплено
      14 | 6C       | 3               | сидячее   | забронировано
(14 rows)

Passenger=# INSERT INTO public."Seats" ("Seat_ID", "Seat_num", "Train_wagons_ID", "Type", "Status")
VALUES (15, '8B', '2', 'сидячее', 'выкуплено');
INSERT 0 1
Passenger=# SELECT * FROM public."Seats_log";
        Target        | Operation | Seat_ID | Seat_num | Train_wagons_ID |  Type   |  Status   |            Date
----------------------+-----------+---------+----------+-----------------+---------+-----------+----------------------------
 log_changes_in_seats | INSERT    |      15 | 8B       | 2               | сидячее | выкуплено | 2023-05-20 21:17:08.793551
(1 row)

Passenger=# DELETE FROM public."Seats" WHERE "Seat_ID" = 14;
DELETE 1
Passenger=# SELECT * FROM public."Seats_log";
        Target        | Operation | Seat_ID | Seat_num | Train_wagons_ID |  Type   |  Status   |            Date
----------------------+-----------+---------+----------+-----------------+---------+-----------+----------------------------
 log_changes_in_seats | INSERT    |      15 | 8B       | 2               | сидячее | выкуплено | 2023-05-20 21:17:08.793551
 log_changes_in_seats | DELETE    |         |          |                 |         |           | 2023-05-20 21:18:59.547815
(2 rows)

Passenger=# UPDATE public."Seats" SET "Seat_num" = '8A' WHERE "Seat_ID" = '13';
UPDATE 1
Passenger=# SELECT * FROM public."Seats_log";
        Target        | Operation | Seat_ID | Seat_num | Train_wagons_ID |  Type   |  Status   |            Date
----------------------+-----------+---------+----------+-----------------+---------+-----------+----------------------------
 log_changes_in_seats | INSERT    |      15 | 8B       | 2               | сидячее | выкуплено | 2023-05-20 21:17:08.793551
 log_changes_in_seats | DELETE    |         |          |                 |         |           | 2023-05-20 21:18:59.547815
 log_changes_in_seats | UPDATE    |      13 | 8A       | 7               | сидячее | выкуплено | 2023-05-20 21:28:05.630079
(3 rows)

Passenger=# SELECT * FROM public."Seats";
 Seat_ID | Seat_num | Train_wagons_ID |   Type    |     Status
---------+----------+-----------------+-----------+---------------
       1 | 1A       | 1               | сидячее   | выкуплено
       2 | 2B       | 2               | сидячее   | забронировано
       3 | 3C       | 3               | спальное  | свободно
       5 | 5A       | 5               | спальное  | выкуплено
       6 | 1A       | 6               | сидячее   | забронировано
       8 | 3A       | 8               | спальное  | свободно
       4 | 4D       | 4               | спальное  | выкуплено
       7 | 1A       | 7               | сидячее   | выкуплено
       9 | 5A       | 9               | сидячее   | забронировано
      10 | 6B       | 9               | спальное  | выкуплено
      11 | 5A       | 9               | сидячее   | забронировано
      12 | 12A      | 4               | спальное  | выкуплено
      15 | 8B       | 2               | сидячее   | выкуплено
      13 | 8A       | 7               | сидячее   | выкуплено
(14 rows)
```

ВЫВОДЫ:

В рамках лабораторной работы были созданы процедуры, функции и триггеры к базе данных PostgreSQL, согласно индивидуальному заданию, часть 2 и 3. Работа была выполнена используя PSQL Tool, которая является внутренней утилитой pgadmin4.