

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:

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ:	3
ВЫПОЛНЕНИЕ:	4
1.1 Реализация функции для подсчета общего количества пассажиров	4
1.2 Реализация функции для вывода пассажиров, прибывающих/отправляющихся на станцию «12»	4
1.3 Реализация функции для вывода вместимости состава	5
2.1 Реализация процедуры для добавления информации о пассажире	5
2.2 Реализация процедуры для снижения стоимости билета на 10%, если билет приобретен онлайн	6
3.1 Реализация триггера для добавления информации о месте в таблицу Seat	7
3.2 Реализация триггера для операций над таблицей Seat	8
ВЫВОДЫ:	10

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:

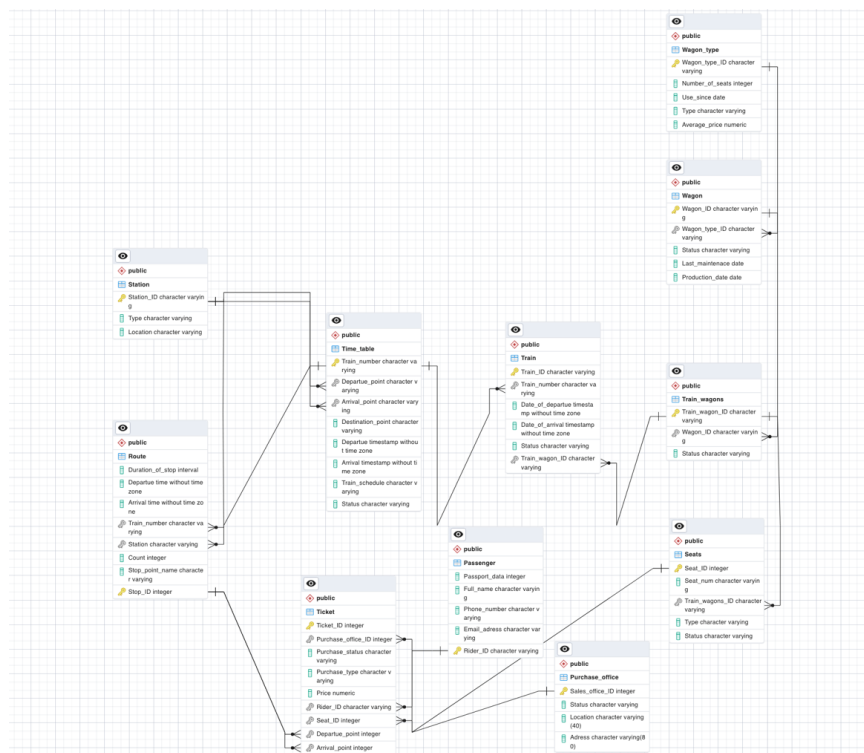
Вариант 1

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

```
1 CREATE OR REPLACE FUNCTION get_passenger_count(status character varying)
2 RETURNS integer
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6     passenger_count INTEGER;
7 BEGIN
8     SELECT COUNT(*) INTO passenger_count
9     FROM public."Passenger"
10    JOIN public."Passenger" p
11    ON public."Ticket" t p."Rider_ID" = t."Rider_ID"
12    WHERE t."Status" = 'выкуплено'
13    RETURN passenger_count;
14 END $$;
```

Результат работы функции:

get_passenger_count integer	
1	12

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

```
1 CREATE OR REPLACE FUNCTION public.get_passengers_by_station(
2     stop_id character varying)
3 RETURNS TABLE(full_name character varying, rider_id character varying)
4 LANGUAGE plpgsql AS $$;
5 BEGIN
6     RETURN QUERY
7     SELECT p."Full_name", p."Rider_ID"
8     FROM public."Ticket" t
9     INNER JOIN public."Passenger" p ON t."Rider_ID" = p."Rider_ID"
10    INNER JOIN public."Route" r ON t."Departue_point" = r."Stop_ID" OR t."Arrival_point" = r."Stop_ID"
11    WHERE r."Stop_ID" = stop_id;
12 END $$;
```

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 Макаров      | Rider08
(5 rows)
```

1.3 Реализация функции для вывода вместимости состава

Query Query History



```
1 CREATE OR REPLACE FUNCTION public.get_wagon_info(  
2     train_wagon_id character varying)  
3     RETURNS TABLE(coach_id character varying, capacity integer)  
4     LANGUAGE plpgsql AS $$  
5 BEGIN  
6     RETURN QUERY  
7     SELECT tw."Train_wagon_ID", wt."Number_of_seats"  
8     FROM public."Train_wagons" tw  
9     INNER JOIN public."Wagon" w ON tw."Wagon_ID" = w."Wagon_ID"  
10    WHERE tw."Train_wagon_ID" = train_wagon_id;  
11 END $$;
```

Результат работы функции:

	coach_id character varying	capacity integer
1	1	54

2.1 Реализация процедуры для добавления информации о пассажире

Query Query History

```
1 CREATE OR REPLACE PROCEDURE public.add_passenger(  
2     IN passport_data integer,  
3     IN full_name character varying,  
4     IN phone_number character varying,  
5     IN email_address character varying,  
6     IN rider_id character varying)  
7     LANGUAGE plpgsql AS $$  
8 BEGIN  
9     INSERT INTO public."Passenger"(  
10        "Passport_data",  
11        "Full_name",  
12        "Phone_number",  
13        "Email_adress",  
14        "Rider_ID")  
15    VALUES (passport_data,  
16        full_name,  
17        phone_number,  
18        email_address,  
19        rider_id);  
20 END $$;
```

Результат работы процедуры:

```
Passenger=# SELECT * FROM public."Passenger";
```

Passport_data	Full_name	Phone_number	Email_address	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_address	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

```
1 CREATE OR REPLACE PROCEDURE public.decrease_ticket_price(  
2 )  
3 LANGUAGE plpgsql AS $$  
4 BEGIN  
5     UPDATE public."Ticket"  
6     SET "Price" = "Price" * 0.9  
7     WHERE "Purchase_type" = 'онлайн';  
8 END $$;
```

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 триггера для добавления информации о месте в таблицу Seat

Query Query History

```
1 CREATE OR REPLACE FUNCTION public.log_seat_insert()  
2 RETURNS trigger  
3 LANGUAGE plpgsql AS $$  
4 BEGIN  
5     INSERT INTO  
6     public."Seat_Log" ("Seat_ID", "Operation", "Timestamp")  
7     VALUES (NEW."Seat_ID", 'INSERT', NOW());  
8     RETURN NEW;  
9 END $$;
```

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

```
1 CREATE OR REPLACE FUNCTION public.changes_in_seats()
2     RETURNS trigger
3     LANGUAGE plpgsql
4 AS $$
5 BEGIN
6 INSERT INTO public."Seats_log" VALUES (
7     TG_NAME,
8     TG_OP,
9     NEW."Seat_ID",
10    NEW."Seat_num",
11    NEW."Train_wagons_ID",
12    NEW."Type",
13    NEW."Status",
14    NOW());
15 RETURN NEW;
16 END $$;
```

Query Query History

```
1 CREATE TRIGGER log_changes_in_seats
2     AFTER INSERT OR DELETE OR UPDATE
3     ON public."Seats"
4     FOR EACH ROW
5     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 | 14 | 6C | 3 | сидячее | забронировано | 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 | 14 | 6C | 3 | сидячее | забронировано | 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.