

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 №3
«процедуры, функции, триггеры в PostgreSQL»

по дисциплине:
«ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ»

Выполнила:
Чагина Вероника Александровна,
группа К32422
Преподаватель:
Говорова Марина Михайловна

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

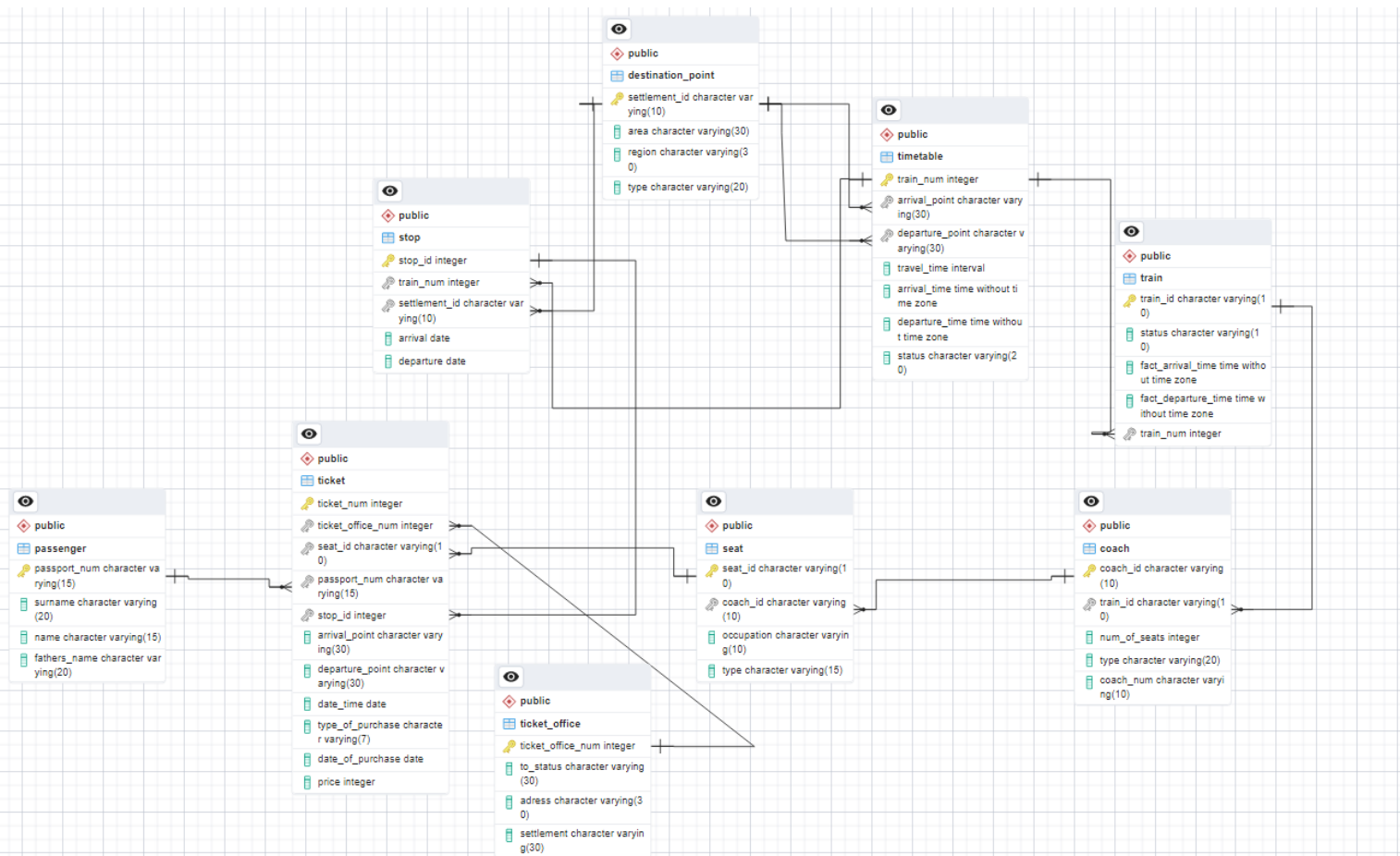
Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Выполнение:

1. Название: БД «Пассажир»
2. Схема логической модели базы данных, сгенерированная в Generate ERD

Рисунок 1



Выполнение:

1. Создать хранимые процедуры

1.1. Для повышения цен в купейные типы вагонов на 20%.

Запрос История запросов

```
1 CREATE OR REPLACE PROCEDURE raise_prices_coupes()
2 LANGUAGE SQL
3 AS $$
4     UPDATE public.coach
5     SET standard_price = standard_price * 1.2
6     WHERE coach_type = '7740' OR coach_type = '7745' OR coach_type = '7750'
7 $$|
```

Data Output Сообщения Notifications

CREATE PROCEDURE

Запрос завершён успешно, время выполнения: 46 msec.

train_log=# SELECT * FROM public.coach;

coach_id	train_id	coach_num	standard_price	coach_type
010	HS1	9901	1500	6670
011	HS1	9911	1500	6660
020	INT1	9902	2500	7745
030	IC1	9903	2000	9965
021	INT1	9912	2500	7740
031	IC1	9913	2000	9955
022	INT1	9922	2500	7750
032	IC1	9923	2000	9960
033	IC1	9933	2000	9950

(9 строк)

train_log=# CALL raise_prices_coupes();

CALL

train_log=# SELECT * FROM public.coach;

coach_id	train_id	coach_num	standard_price	coach_type
010	HS1	9901	1500	6670
011	HS1	9911	1500	6660
030	IC1	9903	2000	9965
031	IC1	9913	2000	9955
032	IC1	9923	2000	9960
033	IC1	9933	2000	9950
020	INT1	9902	3000	7745
021	INT1	9912	3000	7740
022	INT1	9922	3000	7750

(9 строк)

1.2. Для создания нового рейса на поезд

Запрос История запросов

```
1 CREATE OR REPLACE PROCEDURE new_route(IN train_num integer,
2                                     IN arrival_point character varying,
3                                     IN departure_point character varying,
4                                     IN travel_time interval,
5                                     IN arrival_time time without time zone,
6                                     IN departure_time time without time zone,
7                                     IN departure_date date)
8 LANGUAGE SQL
9 AS $$
10     INSERT INTO public.timetable(train_num, arrival_point,
11                                departure_point, travel_time,
12                                arrival_time, departure_time, departure_date)
13     VALUES (train_num, arrival_point,
14            departure_point, travel_time,
15            arrival_time, departure_time, departure_date)
16 $$
```

Data Output Сообщения Notifications

CREATE PROCEDURE

train_log=# SELECT * FROM public.timetable;

train_num	arrival_point	departure_point	travel_time	arrival_time	departure_time	departure_date
101	MSC	SPB	05:00:00	11:30:00	06:30:00	2023-12-12
111	MSC	SPB	05:00:00	06:30:00	01:30:00	2023-12-12
201	RAH	SPB	16:00:00	04:30:00	12:30:00	2023-12-12
202	RAH	SPB	15:30:00	16:00:00	01:30:00	2023-12-12
301	SPB	MSC	08:00:00	08:30:00	00:30:00	2023-12-12

(5 строк)

train_log=# CALL new_route(203, 'RAH', 'SPB', '15:00:00', '18:30:00', '03:30:00', '2023-01-01');

CALL

train_log=# SELECT * FROM public.timetable;

train_num	arrival_point	departure_point	travel_time	arrival_time	departure_time	departure_date
101	MSC	SPB	05:00:00	11:30:00	06:30:00	2023-12-12
111	MSC	SPB	05:00:00	06:30:00	01:30:00	2023-12-12
201	RAH	SPB	16:00:00	04:30:00	12:30:00	2023-12-12
202	RAH	SPB	15:30:00	16:00:00	01:30:00	2023-12-12
301	SPB	MSC	08:00:00	08:30:00	00:30:00	2023-12-12
203	RAH	SPB	15:00:00	18:30:00	03:30:00	2023-01-01

(6 строк)

1.3. Функция для формирования общей выручки по продаже билетов.

Запрос История запросов

```
1 CREATE OR REPLACE FUNCTION all_revenue()
2 RETURNS INTEGER
3 AS $$
4 DECLARE
5     total_revenue INTEGER;
6 BEGIN
7     SELECT SUM(price) INTO total_revenue FROM ticket;
8     RETURN total_revenue;
9 END;
10 $$ LANGUAGE plpgsql;
```

Data Output Сообщения Notifications

CREATE FUNCTION

train_log=# SELECT all_revenue();

all_revenue

16100

(1 строка)

2. Создать необходимые триггеры

2.1. Для добавления информации о месте

Создана таблица logs

Запрос	История запросов
<pre>1 CREATE OR REPLACE FUNCTION public.new_seat() 2 RETURNS trigger 3 LANGUAGE plpgsql 4 AS \$\$ 5 BEGIN 6 INSERT INTO public."logs" 7 VALUES (NEW."seat_id", 'INSERT', NOW()); 8 RETURN NEW; 9 END \$\$; 10 CREATE TRIGGER new_log 11 AFTER INSERT OR DELETE OR UPDATE 12 ON seat 13 FOR EACH ROW 14 EXECUTE FUNCTION new_seat();</pre>	
Data Output	Сообщения
	<pre>CREATE TRIGGER Запрос завершён успешно, время выполнения: 42 msec. train_log=# SELECT * FROM logs; seat_id operation time_of_operation -----+-----+----- (0 строк) train_log=# INSERT INTO public.seat("seat_id", "coach_id", "occupation") VALUES ('34', '020', 'Sold'); INSERT 0 1 train_log=# SELECT * FROM logs; seat_id operation time_of_operation -----+-----+----- 34 INSERT 12:15:39.273589 (1 строка)</pre>

Вывод:

Созданы функции и триггеры на запрос, вставку, удаление и редактирование данных.