

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

Факультет **инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки(специальность) **09.03.03 Прикладная**
Информатика

О Т Ч Е Т
по дисциплине «Проектирование и реализация баз данных»
на тему: Запросы на выборку и модификацию данных, представления и
индексы в PostgreSQL

Обучающийся: Шляхов Д.О., К32402

Преподаватель: Говорова М. М.

Дата 17.04.2023

Санкт-Петербург 2023

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

Практическое задание:

1. 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 :

Вариант 14. БД «Служба заказа такси»

Описание предметной области: Система должна фиксировать все вызовы такси и распределять их между водителями.

Каждому водителю ежедневно начисляется заработная плата в зависимости от количества вызовов и их тарифа (50% от заработанной им суммы). Автомобили могут быть собственностью компании или таксиста.

Заказ принимает дежурный администратор и передает его водителю. В заказе фиксируется тип оплаты – наличными или онлайн.

В системе необходимо хранить график работы водителей.

Ежедневно действуют базовые тарифы на тип предоставляемых авто, но в зависимости от времени суток и ситуации на дорогах, цена может корректироваться.

БД должна содержать следующий минимальный набор сведений: Код сотрудника. ФИО сотрудника. Адрес сотрудника. № телефона сотрудника. Паспортные данные сотрудника. Должность сотрудника. Категория сотрудника. Наименование модели и марки автомобиля. Технические характеристики. Стран-производитель. Стоимость. Код тарифа. Наименование тарифа. Цена за километр. Код автомобиля. Госномер автомобиля. Год выпуска. Пробег. Дата последнего ТО. Дата вызова. Время посадки пассажира. Время высадки пассажира. Номер телефона пассажира. Откуда. Куда. Расстояние. Штраф за время ожидания (в минутах). Оплата (онлайн (при заказе) или наличными). Рекламация клиента на вызов.

Выполнение:

Запросы

Создать хранимые процедуры:

1. Для вывода данных о пассажирах, которые заказывали такси в заданном, как параметр, временном интервале.

```
taxi=# CREATE OR REPLACE FUNCTION check_passenger(start_time DATE, end_time DATE)
taxi=# RETURNS TABLE(client_id INTEGER, full_name TEXT, client_number INTEGER) as $$ BEGIN
taxi=# RETURN QUERY
taxi=# SELECT * from client where client.client_id in(SELECT ordertaxi.client_id from ordertaxi where travel_start BETWEEN start_time and end_time);
taxi=# END;
taxi=# $$ LANGUAGE plpgsql;
CREATE FUNCTION
taxi=# select * from check_passenger('01/01/1000', '24/05/2023');
 client_id |      full_name      | client_number
-----+-----+-----
          8 | ру                  |           77
         11 | ссср                |           23
          6 | спб                 |           55
          7 | точка               |           66
          1 | Шляхов Денис Олегович |        12345
(5 строк)
```

2. Вывести сведения о том, куда был доставлен пассажир по заданному номеру телефона пассажира.

```
taxi=# CREATE OR REPLACE FUNCTION where_passenger(number INTEGER)
taxi=# RETURNS TABLE(place_to_come TEXT) as $$ BEGIN
taxi=# RETURN QUERY
taxi=# SELECT ordertaxi.place_to_come from ordertaxi join client using(client_id) where client_number = number;
taxi=# END;
taxi=# $$ LANGUAGE plpgsql;
CREATE FUNCTION
taxi=# select * from where_passenger(55)
taxi=# ;
 place_to_come
-----
 msc
(1 строка)
```

3. Для вычисления суммарного дохода таксопарка за истекший месяц.

```
taxi=# CREATE OR REPLACE FUNCTION check_increase() RETURNS INTEGER as $$ BEGIN
taxi=# RETURN
taxi=# (SELECT SUM(cost) as sum from ordertaxi where EXTRACT(Month FROM travel_start) = EXTRACT(Month FROM NOW()) - 1 and EXTRACT(YEAR FROM travel_start) = EXTRACT(YEAR FROM NOW()));
taxi=# END;
taxi=# $$ LANGUAGE plpgsql;
CREATE FUNCTION
taxi=# select * from check_increase();
 check_increase
-----
          247072
(1 строка)
```

Создать необходимые триггеры:

Запись в логи при удалении/апдейте/вставке в табл client

```
taxi=# CREATE OR REPLACE FUNCTION add_to_log()
taxi=# RETURNS TRIGGER AS $$
taxi$# DECLARE
taxi$# mstr varchar(30);
taxi$# astr varchar(100);
taxi$# retstr varchar(254);
taxi$# BEGIN
taxi$# IF TG_OP = 'INSERT' THEN
taxi$#   astr = NEW;
taxi$#   mstr := 'Add data ';
taxi$#   retstr := mstr||astr;
taxi$#
taxi$# INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
taxi$# RETURN NEW;
taxi$# ELSIF TG_OP = 'UPDATE' THEN
taxi$#   astr = NEW;
taxi$#   mstr := 'Update data ';
taxi$#   retstr := mstr||astr;
taxi$# INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
taxi$# RETURN NEW;
taxi$# ELSIF TG_OP = 'DELETE' THEN
taxi$#   astr = OLD;
taxi$#   mstr := 'Remove data ';
taxi$#   retstr := mstr || astr;
taxi$# INSERT INTO logs(text, added, table_name) values (retstr, NOW(), TG_TABLE_NAME);
taxi$#
CREATE FUNCTION
```

```
taxi=# CREATE TRIGGER t_client AFTER INSERT OR UPDATE OR DELETE ON client FOR EACH ROW EXECUTE PROCEDURE add_to_log();
CREATE TRIGGER
```

```
taxi=# insert into client (client_id,full_name,client_number) values(15,'АБВ ГД ЙКА',5553535);
INSERT 0 1
```

```
taxi=# insert into client (client_id,full_name,client_number) values(15,'АБВ ГД ЙКА',5553535);
INSERT 0 1
```

```
taxi=# update client set full_name= 'АБВ ГДЖ ЙКА' where client_id = 15;
UPDATE 1
```

```
taxi=# delete from client where client_id = 15;
DELETE 1
```

```
taxi=# select * from logs;
```

text	added	table_name
Add data (15,"АБВ ГД ЙКА",5553535)	2023-05-24	client
Update data (15,"АБВ ГДЖ ЙКА",5553535)	2023-05-24	client
Remove data (15,"АБВ ГДЖ ЙКА",5553535)	2023-05-24	client

(3 строки)

ЗАКЛЮЧЕНИЕ

В данной работе мною были изучены функции и процедуры, а также созданы триггеры для корректного хранения данных.