Министерство науки и высшего образования Российской Федерации федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по Лабораторной Работе № 3

по дисциплине «Проектирование и реализация баз данных»

Вариант 12

Автор: Мазеин Никита Олегович

Факультет: ФИКТ

Группа: К32402

Преподаватель: Говорова Марина Михайловна



Санкт-Петербург 2023

1. Цель работы:

овладеть практическими навыками создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

2. Практическое задание:

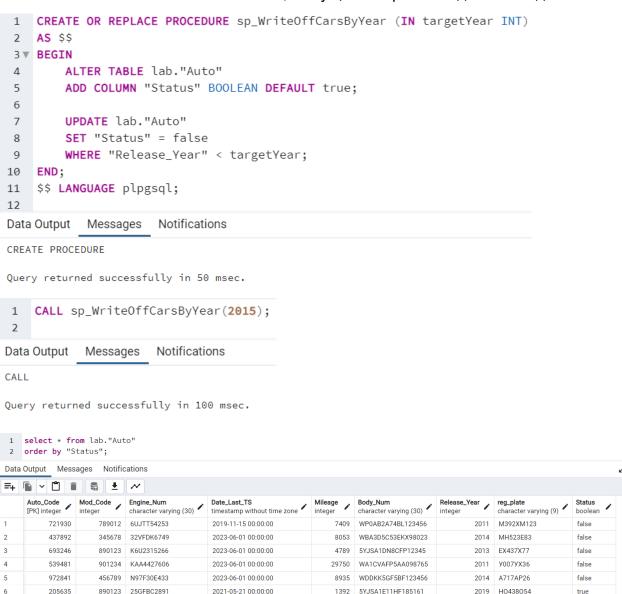
БД «Прокат автомобилей»

- 1. Создать процедуры/функции 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.

3. Выполнение:

Задание 4. Создать хранимые процедуры:

• Выполнить списание автомобилей, выпущенных ранее заданного года.



 Выдачи автомобиля и расчета стоимости с учетом скидки постоянным клиентам.

```
1 CREATE OR REPLACE PROCEDURE sp_IssueCarToCl (IN cl_Code INT, IN stf_Code INT, IN auto_Code INT, IN rent_h INT)
 2 AS $$
 3
     DECLARE
         discount DECIMAL(10, 2);
 5
         rentPrice INT:
 6
         finalPrice INT;
 7 ▼ BEGIN
 8 ₩
         IF EXISTS (
 9
              SELECT 1
10
              FROM lab."Bonus_Card"
 11
              WHERE "Cl_Code" = cl_Code
 12
 13
             discount := 0.05:
14
        FLSE
15
             discount := 0;
16
        END IF;
17
 18
         SELECT
19
20
                  WHEN rent_h < 24 THEN "Price_One_H" * rent_h
21
                  ELSE "Price_Long_Inter" * (rent_h / 24)
 22
             END INTO rentPrice
23
         FROM lab. "Price"
24
         WHERE "Mod_Code" = (SELECT "Mod_Code" FROM lab."Auto" WHERE "Auto_Code" = auto_Code)
25
             AND "DT_Inter_End" IS NULL;
26
 27
         finalPrice := CAST(rentPrice * (1 - discount) AS INT);
28
29
         INSERT INTO lab. "Contract" ("Contr_Code", "Act_Transf_Client", "Act_Transf_Company", "Rent_Price", "DT_Contract")
         VALUES ((SELECT COALESCE(MAX("Contr_Code"), 0) + 1 FROM lab."Contract"), (SELECT COALESCE(MAX("Act_Transf_C
 30
31
 32 END;
33 $$ LANGUAGE plpgsql;
 Data Output Messages Notifications
 CREATE PROCEDURE
Query returned successfully in 73 msec.
INSERT INTO lab."Contract" ("Contr_Code", "Act_Transf_Client", "Act_Transf_Company", "Rent_Price", "DT_Contract",
"DT_Car_Transf_To_Cl", "Factual_DT_Ret", "Late_Fee", "Ret_Mark", "Cl_Code", "Stf_Code", "Auto_Code", "rent_time")
VALUES ((SELECT COALESCE(MAX("Contr_Code"), 0) + 1 FROM lab."Contract"), (SELECT COALESCE(MAX("Act_Transf_Client"), 0) + 1 FROM
 lab."Contract"), NULL, finalPrice, CURRENT_TIMESTAMP, NULL, NULL, NULL, false, cl_Code, stf_Code, auto_Code, rent_h);
```

(То, что не вошло)

Арендуем машины с одинаковой ценой <u>сначала через клиента с картой</u> постоянного клиента, потом через клиента без карты.

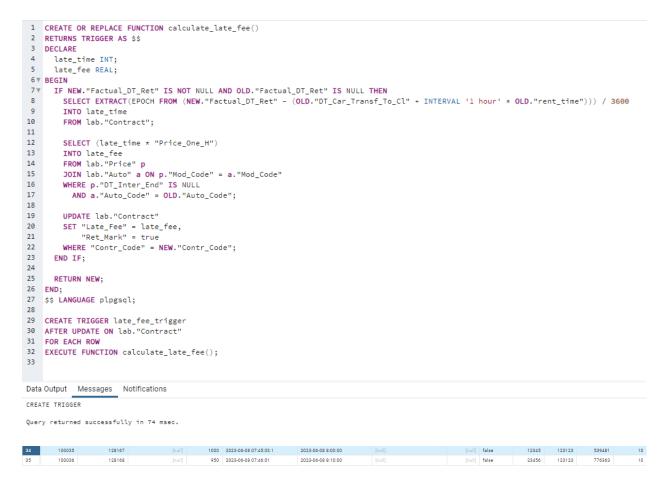
1 CALL sp	_IssueCarT	oCl (1234	5, 12	23123, 539	481, 10);	·					
Data Output	Messages	Notificatio	ns								
CALL											
Query returned	d successful	ly in 75	msec.								
1 CALL sp	_IssueCarT	oCl(2345	6, 1	23123, 776	363, 10);						
Data Output	Messages	Notificatio	ns								
CALL											
Query returne	d successful	lly in 68	msec.								
34 100035	128167		1000	2023-06-08 07:45:03.1			false	12345	123123	539481	10
35 100036	128168		950	2023-06-08 07:46:01			false	23456	123123	776363	10
1000											

• Для вычисления количества автомобилей заданной марки.

```
1 CREATE OR REPLACE PROCEDURE sp_get_car_count_by_model(IN mod_Code Integer, OUT carCount Integer)
 2 AS $$
 3 ▼ BEGIN
        SELECT COUNT(*) INTO carCount
 5
        FROM lab."Auto" a
        JOIN lab. "Model" m USING ("Mod_Code")
 7
       WHERE m. "Mod_Code" = mod_Code;
 8 END;
 9 $$ LANGUAGE plpgsql;
10
11 DO $$
12 DECLARE
        carCount Integer;
13
15
16
        CALL sp_get_car_count_by_model(345678, carCount);
RAISE NOTICE 'Car Count Result: %', carCount;
17 END $$;
Data Output Messages Notifications
NOTICE: Car Count Result: 7
Query returned successfully in 38 msec.
```

Задание 5. Создать триггеры:

Триггер на начисление штрафа после возврата автомобиля с просрочкой



Обновим строки и проверим работу триггера:

34	100035	128167	1000	2023-06-08 07:45:03.1	2023-06-08 08:00:00	2023-06-08 15:30:00		true	12345	123123	539481	10
35	100036	128168	950	2023-06-08 07:46:01	2023-06-08 08:10:00	2023-06-08 19:30:00	100	true	23456	123123	776363	10

Триггер сработал!

4. Выводы:

По итогам выполнения лабораторной работы я ознакомился с понятиями процедур, функций и триггеров. Мною были созданы и проверены на работоспособность несколько процедур, также триггер, работу упростил базой который значительно С данных, штрафа вычисление просрочку возврата автоматизировав за автомобиля.