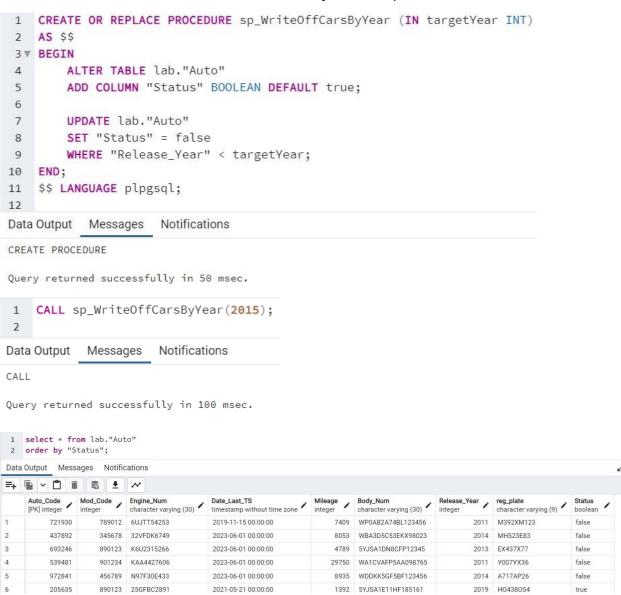
<u>БД «Прокат автомобилей»</u>

- 1. Создание процедур/функций.
- 2. Создание триггера для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL.

Создаём 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
     4
                             discount DECIMAL(10, 2);
      5
                             rentPrice INT;
                             finalPrice INT;
      7 ♥ BEGIN
      8 ₩
                            IF EXISTS (
     9
                                           SELECT 1
   10
                                           FROM lab. "Bonus_Card"
   11
                                           WHERE "Cl_Code" = cl_Code
                      ) THEN
   12
  13
                                           discount := 0.05;
  14
                       ELSE
   15
                                          discount := 0;
                         END IF;
  16
  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")
   30
                              \textit{VALUES} \ ((\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Contr\_Code"}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}("\texttt{Act\_Transf\_C'}), \ \theta) \ + \ 1 \ \texttt{FROM lab."Contract"}), \ (\texttt{SELECT COALESCE}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}), \texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(\texttt{MAX}(
   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.

Скидка применилась!

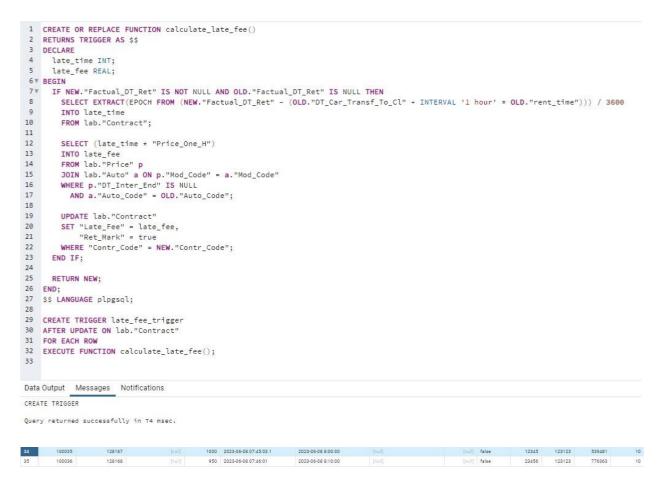
950

• 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 $$
 3 ♥ BEGIN
       SELECT COUNT(*) INTO carCount
 4
        FROM lab. "Auto" a
       JOIN lab. "Model" m USING ("Mod_Code")
 6
 7 W
8 END;
       WHERE m. "Mod_Code" = mod_Code;
 9 $$ LANGUAGE plpgsql;
10
11 DO $$
12 DECLARE
13
        carCount Integer;
14 BEGIN
15
16
         CALL sp_get_car_count_by_model(345678, carCount);
       RAISE NOTICE 'Car Count Result: %', carCount;
17 END $$;
Data Output Messages Notifications
NOTICE: Car Count Result: 7
Query returned successfully in 38 msec.
```

Создаём 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!

Выводы:

Мы ознакомились с понятиями процедур, функций и триггеров. Нами были созданы и проверены на работоспособность несколько процедур, также триггер, который значительно упростил работу с базой данных, автоматизировав вычисление штрафа за просрочку возврата автомобиля.