Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5 «ПРОЦЕДУРЫ, ФУНКЦИИ, ТРИГГЕРЫ В POSTGRESQL» по дисциплине «Проектирование и реализация баз данных»

Автор: Шалунов Андрей Ильич

Факультет: ИКТ

Группа: К3240 Преподаватель:

Говорова М.М.



Санкт-Петербург 2023

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

Практическое задание:

Вариант 2 (тах - 8 баллов)

- 1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
- 2.1. Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу).
 - 2.2. Создать авторский триггер по варианту индивидуального задания.

Указание. Работа выполняется в консоли SQL Shell (psql).

Вариант 11. БД «Автомастерская»

Описание предметной области:

Описание предметной области: Сеть автомастерских осуществляет ремонт автомобилей, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы.

Заработная плата мастеров составляет 50% стоимости работы.

С клиентом заключается договор на выполнение авторемонтных и профилактических работ, который сопровождается администратором. В каждом договоре может быть несколько видов услуг. Для выполнения видов работ могут требоваться детали или расходные материалы, которые предоставляет либо клиент, либо автомастерская. Если детали предоставляет автомастерская, то их стоимость включается в смету по договору.

Каждый вид работ могут выполнять разные мастера, в зависимости от их специализации. Распределение мастеров выполняет администратор.

БД должна содержать следующий минимальный набор сведений: Табельный номер сотрудника. ФИО сотрудника. Должность. Разряд мастера. Специализация. Адрес автомастерской. Дата заказа. Гос. Номер автомобиля. Марка. Мощность автомобиля. Год выпуска. Цвет автомобиля. Дата принятия в ремонт. Плановая дата окончания ремонта. Фактическая дата окончания ремонта. Вид ремонта. Стоимость вида ремонта. Название детали. Цена детали. Марка и модель автомобиля. Страна производителя. Госномер автомобиля. ФИО владельца. Номер телефона владельца. Е-mail владельца.

Задание 4. Создать хранимые процедуры:

• **Повышения цены деталей для автомобиля "Ford" на 10 %.** (Сделать таблицу справочник для деталей и марок машин)

CREATE OR REPLACE PROCEDURE increase_ford_detail_prices()
LANGUAGE plpgsql AS \$\$
DECLARE

```
ford car brand character(100) := 'Ford';
BEGIN
      UPDATE public."Detail"
      SET "Price" = "Price" * 1.1
      WHERE "ID_detail" IN (
            SELECT cd."id_detail"
           FROM public."CarDetail" cd
           JOIN public."Automobile" a ON cd."id auto" = a."ID auto"
           JOIN public."Model" m ON a."ID_model" = m."ID_model"
            WHERE m. "Car brand" = ford car brand
     );
END;
$$;
   Data Output Messages Notifications

    □
    √
    □
    √
    □
    √
    □
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓</
                                                           Country_of_manufacturer
                                       Price
                                                                                                     ID detail
             Name
                                                                                                      [PK] bigint
             character
                                        integer
                                                            character
             Wheel Mercede...
                                               36603 Germany
  2
             Запасное коле...
                                               10000 Россия
                                               17289 Германия
             Тормозные ко...
  4
             Масляный фил...
                                               13993
                                                            Китай
                                                                                                                      4
             Свеча зажиган...
                                               40027
                                                            США
                                                                                                                      5
             Воздушный фи...
                                               12493
                                                                                                                      6
  6
                                                            Германия
                                               46395 Россия
                                                                                                                      7
             Моторное мас...
```

CALL increase ford detail prices();

17957 Китай

20618 Германия

28848 Россия

12100 Germany

19397

8

10

11

12

8

10

11

12

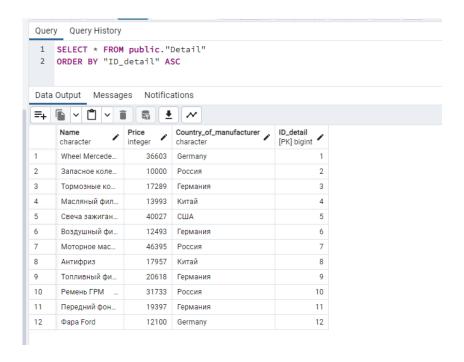
Антифриз

Φapa Ford

Топливный фи...

Ремень ГРМ

Передний фон...



• Для повышения разряда тех мастеров, которые отремонтировали больше 3 автомобилей. (Сделать repaired_cars_count как параметр функции и проверку на максимальный ранк)

```
CREATE OR REPLACE PROCEDURE
increase_mechanic_rank(repaired_cars_count integer)

LANGUAGE plpgsql AS $$

BEGIN

UPDATE public."Employee" as e

SET "rank" = CASE WHEN e."rank" < 5 THEN e."rank" + 1 ELSE e."rank"
END

WHERE e."ID_employee" IN (

SELECT e."ID_employee"

FROM public."Employee" e

JOIN public."Contract" c ON c."ID_employee" = e."ID_employee"

JOIN public."Job_position" jb ON jb."ID_job_position" = e."ID_job_position"
```

WHERE jb. "Name" = 'Мастер-механик'

AND c."Order_status" = 'Completed'

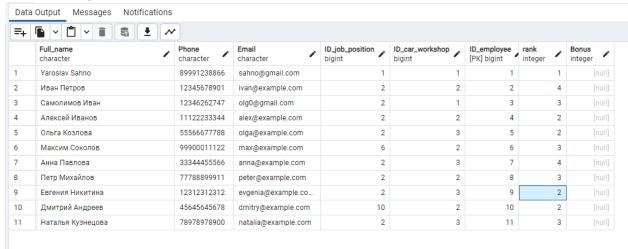
GROUP BY e. "ID_employee"

HAVING COUNT(DISTINCT c. "ID_auto") > repaired_cars_count

);

END;

\$\$;



CALL increase_mechanic_rank(2);

• Сколько автомобилей отремонтировал каждый механик за истекший квартал.

```
CREATE OR REPLACE FUNCTION count_repair_cars()

RETURNS TABLE (
    mechanic_id bigint,
    mechanic_name character(50),
    repaired_count_cars bigint
) AS $$

BEGIN

RETURN QUERY

SELECT

e."ID_employee" AS mechanic_id,
 e."Full_name" AS mechanic_name,
    COUNT(DISTINCT c."ID_auto") AS repaired_count_cars
```

```
FROM
public."Employee" e

JOIN public."Contract" c ON c."ID_employee" = e."ID_employee"

JOIN public."Job_position" jb ON jb."ID_job_position" = e."ID_job_position"

WHERE
jb."Name" = 'Macrep-механик'

AND c."Actual_date_end_of_repair" >= CURRENT_DATE - INTERVAL '3 months'

GROUP BY
e."ID_employee";

END;

$$ LANGUAGE plpgsql;
```

Вызов функции: select * from count repair cars();

```
AutorepairShop=# CREATE OR REPLACE FUNCTION count_repair_cars()
AutorepairShop-# RETURNS TABLE (
AutorepairShop(#
                     mechanic_id bigint,
AutorepairShop(#
                     mechanic_name character(50),
AutorepairShop(#
                     repaired_count_cars bigint
AutorepairShop(# ) AS $$
AutorepairShop$# BEGIN
AutorepairShop$#
                     RETURN QUERY
AutorepairShop$#
                      SELECT
                         e."ID_employee" AS mechanic_id,
AutorepairShop$#
                         e."Full_name" AS mechanic_name,
AutorepairShop$#
AutorepairShop$#
                         COUNT(DISTINCT c."ID_auto") AS repaired_count_cars
AutorepairShop$#
AutorepairShop$#
                         public."Employee" e
                      JOIN public."Contract" c ON c."ID_employee" = e."ID_employee"
JOIN public."Job_position" jb ON jb."ID_job_position" = e."ID_job_position"
AutorepairShop$#
AutorepairShop$#
AutorepairShop$#
AutorepairShop$#
                          jb. "Name" = 'Мастер-механик'
                          AND c."Actual date end of repair" >= CURRENT DATE - INTERVAL '3 months'
AutorepairShop$#
AutorepairShop$#
                      GROUP BY
AutorepairShop$#
                          e."ID_employee";
AutorepairShop$# END;
AutorepairShop$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
AutorepairShop=# select * from count_repair_cars();
mechanic id |
                                   mechanic name
                                                                       repaired count cars
           2 | Иван Петров
             | Самолимов Иван
| Алексой
           4
               Алексей Иванов
                                                                                           68
              Ольга Козлова
                                                                                           56
               Анна Павлова
                                                                                           63
           8
               Петр Михайлов
                                                                                           60
                                                                                           47
               Евгения Никитина
             Наталья Кузнецова
                                                                                           38
8 строк)
```

2.2. Создать авторский триггер по варианту индивидуального задания.

При вставке детали в таблицу Details_from_client, изменяем стоимость ремонта авто в таблице Contract на сумму, равную стоимостью деталей, если поставщик деталей это сам сервис.

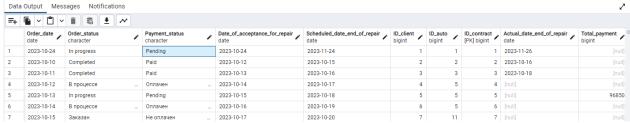
(Убрать в Distribution_of_work id детали от клиентов, и добавить статус от кого детали в Details_from_client, сделать тригтер на инсерт и уведичением total payment в контракте если деталь со статусом от сервиса.)

```
CREATE OR REPLACE FUNCTION update_total_payment_on_detail_insert()
RETURNS TRIGGER AS $$
DECLARE
  total_detail_price INTEGER;
BEGIN
  IF NEW. "supplier" = 'Service' THEN
    total_detail_price := NEW."Amount_of_detail" * (
      SELECT "Price"
      FROM public."Detail"
      WHERE "ID_detail" = NEW."ID_detail"
    );
    UPDATE public. "Contract"
    SET "Total_payment" = "Total_payment" + total_detail_price
    WHERE "ID_contract" IN (
      SELECT "ID_contract"
      FROM public. "Distribution_of_work"
      WHERE "ID_destribution_of_work" = NEW."ID_distribution"
    ) AND "Order_status" = 'In progress' AND "Payment_status" = 'Pending';
  END IF;
```

RETURN NEW:

END;

\$\$ LANGUAGE plpgsql;

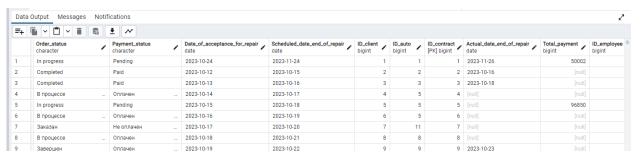


CREATE TRIGGER update_total_payment_on_detail_insert

AFTER INSERT ON public. "Details from client"

FOR EACH ROW

EXECUTE FUNCTION update total payment on detail insert();



Вывод:

В результате выполнения лабораторной работы были закреплены навыки создания и использования процедур, функций и триггеров в PostgreSQL. Освоены принципы их работы, а также применение в конкретных сценариях, связанных с автомастерской и управлением данными в данной предметной области.