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: Шаповалов Илья Андреевич

Факультет: ИКТ

Группа: К3239 Преподаватель:

Говорова М.М.

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

Практическое задание:

Вариант 2 (max - 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. E-mail владельца.

Задание 4. Создать хранимые процедуры:

Повышения цены деталей для автомобиля “Ford” на 10 %. (Сделать таблицу справочник для деталей и марок машин)

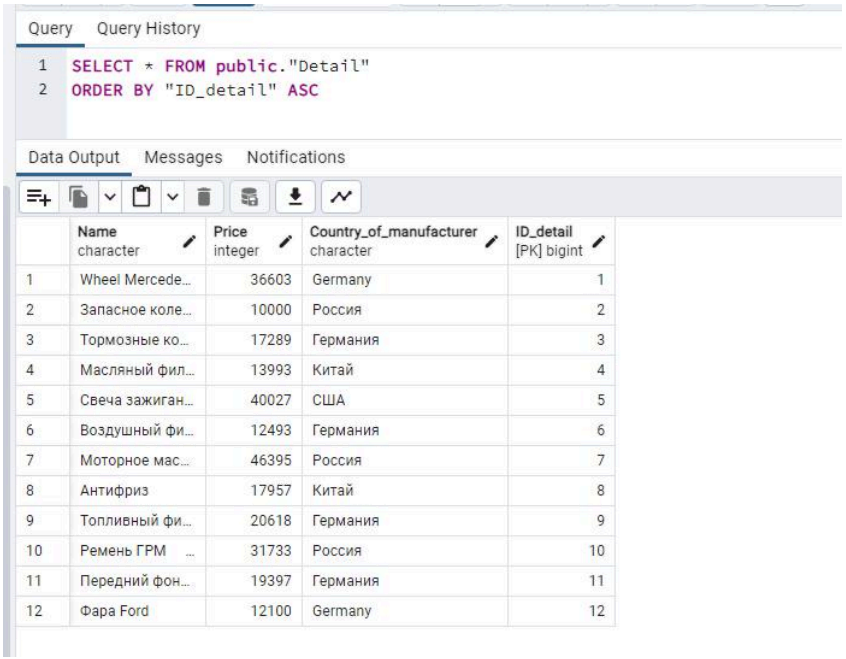
```
CREATE OR REPLACE PROCEDURE increase_ford_detail_prices()
LANGUAGE plpgsql AS $$
DECLARE
```

```
    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

	Name character	Price integer	Country_of_manufacturer character	ID_detail [PK] bigint
1	Wheel Mercede...	36603	Germany	1
2	Запасное коле...	10000	Россия	2
3	Тормозные ко...	17289	Германия	3
4	Масляный фил...	13993	Китай	4
5	Свеча зажиган...	40027	США	5
6	Воздушный фи...	12493	Германия	6
7	Моторное мас...	46395	Россия	7
8	Антифриз	17957	Китай	8
9	Топливный фи...	20618	Германия	9
10	Ремень ГРМ ...	28848	Россия	10
11	Передний фон...	19397	Германия	11
12	Фара Ford	12100	Germany	12

```
CALL increase_ford_detail_prices();
```



	Name character	Price integer	Country_of_manufacturer character	ID_detail [PK] bigint
1	Wheel Mercede...	36603	Germany	1
2	Запасное коле...	10000	Россия	2
3	Тормозные ко...	17289	Германия	3
4	Масляный фил...	13993	Китай	4
5	Свеча зажиган...	40027	США	5
6	Воздушный фи...	12493	Германия	6
7	Моторное мас...	46395	Россия	7
8	Антифриз	17957	Китай	8
9	Топливный фи...	20618	Германия	9
10	Ремень ГРМ ...	31733	Россия	10
11	Передний фон...	19397	Германия	11
12	Фара Ford	12100	Germany	12

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;
$$;
```

Data Output | Messages | Notifications

	Full_name character	Phone character	Email character	ID_job_position bigint	ID_car_workshop bigint	ID_employee [PK] bigint	rank integer	Bonus integer
1	Yaroslav Sahno	89991238866	sahno@gmail.com	1	1	1	1	[null]
2	Иван Петров	12345678901	ivan@example.com	2	2	2	4	[null]
3	Самолимов Иван	12346262747	olg0@gmail.com	2	1	3	3	[null]
4	Алексей Иванов	11122233344	alex@example.com	2	2	4	2	[null]
5	Ольга Козлова	55566677788	olga@example.com	2	3	5	2	[null]
6	Максим Соколов	99900011122	max@example.com	6	2	6	3	[null]
7	Анна Павлова	33344455566	anna@example.com	2	3	7	4	[null]
8	Петр Михайлов	77788899911	peter@example.com	2	2	8	3	[null]
9	Евгения Никитина	12312312312	evgenia@example.co...	2	3	9	2	[null]
10	Дмитрий Андреев	45645645678	dmitry@example.com	10	2	10	2	[null]
11	Наталья Кузнецова	78978978900	natalia@example.com	2	3	11	3	[null]

```
CALL increase_mechanic_rank(2);
```

Сколько автомобилей отремонтировал каждый механик за истекший квартал.

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 $$
AutorepairShop$# BEGIN
AutorepairShop$#     RETURN QUERY
AutorepairShop$#     SELECT
AutorepairShop$#         e."ID_employee" AS mechanic_id,
AutorepairShop$#         e."Full_name" AS mechanic_name,
AutorepairShop$#         COUNT(DISTINCT c."ID_auto") AS repaired_count_cars
AutorepairShop$#     FROM
AutorepairShop$#         public."Employee" e
AutorepairShop$#     JOIN public."Contract" c ON c."ID_employee" = e."ID_employee"
AutorepairShop$#     JOIN public."Job_position" jb ON jb."ID_job_position" = e."ID_job_position"
AutorepairShop$#     WHERE
AutorepairShop$#         jb."Name" = 'Мастер-механик'
AutorepairShop$#         AND c."Actual_date_end_of_repair" >= CURRENT_DATE - INTERVAL '3 months'
AutorepairShop$#     GROUP BY
AutorepairShop$#         e."ID_employee";
AutorepairShop$# END;
AutorepairShop$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
AutorepairShop=#  select * from  count_repair_cars();
 mechanic_id |                   mechanic_name                    | repaired_count_cars
-------------+----------------------------------------------------+---------------------
           2 | Иван Петров                                        |                  51
           3 | Самолимов Иван                                     |                  73
           4 | Алексей Иванов                                     |                  68
           5 | Ольга Козлова                                      |                  56
           7 | Анна Павлова                                       |                  63
```

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 триггер 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;
```

Data Output | Messages | Notifications

	Order_date date	Order_status character	Payment_status character	Date_of_acceptance_for_repair date	Scheduled_date_end_of_repair date	ID_client bigint	ID_auto bigint	ID_contract [PK] bigint	Actual_date_end_of_repair date	Total_payment bigint
1	2023-10-24	In progress	Pending	2023-10-24	2023-11-24	1	1	1	2023-11-26	[null]
2	2023-10-10	Completed	Paid	2023-10-12	2023-10-15	2	2	2	2023-10-16	[null]
3	2023-10-11	Completed	Paid	2023-10-13	2023-10-16	3	3	3	2023-10-18	[null]
4	2023-10-12	В процессе	Оплачен	2023-10-14	2023-10-17	4	5	4	[null]	[null]
5	2023-10-13	In progress	Pending	2023-10-15	2023-10-18	5	5	5	[null]	96850
6	2023-10-14	В процессе	Оплачен	2023-10-16	2023-10-19	6	5	6	[null]	[null]
7	2023-10-15	Заказан	Не оплачен	2023-10-17	2023-10-20	7	11	7	[null]	[null]

```
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();
```

Data Output | Messages | Notifications

	Order_status character	Payment_status character	Date_of_acceptance_for_repair date	Scheduled_date_end_of_repair date	ID_client bigint	ID_auto bigint	ID_contract [PK] bigint	Actual_date_end_of_repair date	Total_payment bigint	ID_employee bigint
1	In progress	Pending	2023-10-24	2023-11-24	1	1	1	2023-11-26	50002	
2	Completed	Paid	2023-10-12	2023-10-15	2	2	2	2023-10-16	[null]	
3	Completed	Paid	2023-10-13	2023-10-16	3	3	3	2023-10-18	[null]	
4	В процессе	Оплачен	2023-10-14	2023-10-17	4	5	4	[null]	[null]	
5	In progress	Pending	2023-10-15	2023-10-18	5	5	5	[null]	96850	
6	В процессе	Оплачен	2023-10-16	2023-10-19	6	5	6	[null]	[null]	
7	Заказан	Не оплачен	2023-10-17	2023-10-20	7	11	7	[null]	[null]	
8	В процессе	Оплачен	2023-10-18	2023-10-21	8	8	8	[null]	[null]	
9	Завершен	Оплачен	2023-10-19	2023-10-22	9	9	9	2023-10-23	[null]	

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.