

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: Якунин А.Д.

Факультет:

ИКТГруппа:

К3241

Преподаватель: Говорова М.М.

Оглавление

Цель работы.....	3
Практическое задание.....	3
Выполнение	3
Вывод	7

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql).

Практическое задание:

Вариант 2 (max - 8 баллов)

1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
2. Создать авторский триггер по варианту индивидуального задания.

Выполнение

Создайте хранимые процедуры:

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

```
CREATE OR REPLACE FUNCTION
    GetPassengersByTimeIntervalProcedure(
start_time TIME,
end_time TIME
)
RETURNS TABLE (
    id_Клиента INTEGER,
    ФИО VARCHAR(255),
    Время_заказа TIME
)
LANGUAGE SQL
AS $$
    SELECT
        Заказ.id_Клиента,
        Клиент.ФИО,
        Заказ.Время_заказа
    FROM
        Заказ
    JOIN
        Клиент ON Заказ.id_Клиента = Клиент.id_клиента
    WHERE
        Заказ.Время_заказа BETWEEN start_time AND end_time;
$$;
SELECT * FROM
    GetPassengersByTimeIntervalProcedure('09:00:00',
        '14:00:00');
```

	id_Клиента integer	ФИО character varying	Время_заказа time without time zone
1	2	Путин Владимир Владимирович	11:55:21
2	2	Путин Владимир Владимирович	09:27:28

В psql

```
[postgres=# SELECT * FROM GetPassengersByTimeIntervalProcedure('09:00:00', '14:00:00');
```

id_Клиента	ФИО	Время_заказа
2	Путин Владимир Владимирович	11:55:21
2	Путин Владимир Владимирович	09:27:28

(2 rows)

- Вывести сведения о том, куда был доставлен пассажир по заданному номеру телефонапассажира.

```
CREATE OR REPLACE FUNCTION GetDeliveryInfoByPhoneNumber(
  IN Номер_телефона VARCHAR(255)
)
RETURNS TABLE (
  Код_заказа INTEGER,
  Имя_пассажира VARCHAR(255),
  Место_доставки VARCHAR(255)
)
LANGUAGE SQL
AS $$
SELECT
  Заказ.Код_заказа,
  Клиент.ФИО AS Имя_пассажира,
  Заказ.Название_улицы_прибытия AS Место_доставки
FROM
  Заказ
JOIN
  Клиент ON Заказ.id_Клиента = Клиент.id_клиента
WHERE
  Клиент.Номер_телефона = Номер_телефона;
$$;
```

```
SELECT * FROM GetDeliveryInfoByPhoneNumber('798193249');
```

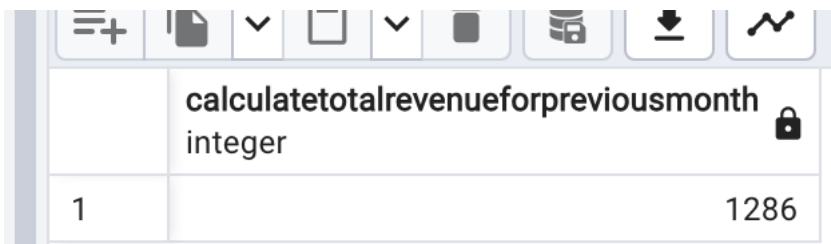
	Код_заказа integer	Имя_пассажира character varying	Место_доставки character varying
1	9	Путин Владимир Владимирович	Ланская
2	6	Путин Владимир Владимирович	Савушкина
3	2	Путин Владимир Владимирович	Лесная
4	8	Джо Байден	Савушкина

В psql

```
postgres=# SELECT * FROM GetDeliveryInfoByPhoneNumber('798193249');
Код_заказа | Имя_пассажира | Место_доставки
-----+-----+-----
          9 | Путин Владимир Владимирович | Ланская
          6 | Путин Владимир Владимирович | Савушкина
          2 | Путин Владимир Владимирович | Лесная
          8 | Джо Байден | Савушкина
```

- Для вычисления суммарного дохода таксопарка за истекший месяц.

```
CREATE OR REPLACE FUNCTION CalculateTotalRevenueForPreviousMonth()
RETURNS INTEGER
LANGUAGE SQL
AS $$
    SELECT
        COALESCE(SUM(Заказ.Сумма), 0)::INTEGER
    FROM
        Заказ
    WHERE
        DATE_TRUNC('MONTH', Заказ.Дата_выполнения) = DATE_TRUNC('MONTH', CURRENT_DATE -
        INTERVAL '1 month');
$$;
SELECT CalculateTotalRevenueForPreviousMonth();
```



	calculatetotalrevenueforpreviousmonth integer
1	1286

```
postgres=# SELECT CalculateTotalRevenueForPreviousMonth();
calculatetotalrevenueforpreviousmonth
-----
1286
```

Триггер:

- Будем вычислять скидку для клиента от его количества поездок

```
CREATE OR REPLACE FUNCTION UpdateDiscount()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS $$
DECLARE
    order_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO order_count
    FROM Заказ
    WHERE id_клиента = NEW.id_клиента;

    UPDATE Клиент
    SET Скидка = CASE
        WHEN order_count = 2 THEN 5
        WHEN order_count >= 3 THEN 8
        ELSE 0
    END
    WHERE id_клиента = NEW.id_клиента;

    RETURN NEW;
END;
```

```
CREATE TRIGGER UpdateDiscountTrigger
AFTER INSERT OR UPDATE ON Заказ
```

Можно заметить что мой Триггер обновляет поле "Скидка" в таблице "Клиент" в зависимости от количества заказов, сделанных клиентом

Если клиент сделал 2 заказа, то устанавливается скидка 5%.

Если клиент сделал 3 или более заказов, то устанавливается скидка 8%.

В остальных случаях скидка устанавливается в 0%.

	id_клиента [PK] integer	Номер_телефона character varying (25)	Рекламация character varying (35)	ФИО character varying (35)	Скидка integer
1	2	798193249	Долгое ожидание	Путин Владимир Владимирович	8
2	3	791143244	Неудобный автомобиль	Джо Байден	5
3	4	798193249	Неприятное обслуживание	Анол Максим Олегович	0
4	5	799423425	Неопытный водитель	Казарян Тигран Андреевич	0

Код_смены integer	номер_карты integer	Код_тарифа integer	id_работного integer	Название_улицы_прибытия character varying	id_Клиента integer	ГОСНОМЕР integer	Время_заказа time without time zone	Сумма integer	Дата_выполнения date
2	8901	2	3	Лесная	2	123890	11:55:21	150	2023-12-01
5	9234	2	6	Савушкина	5	456789	15:58:03	160	2023-12-05
6	9345	3	2	Савушкина	2	123456	18:10:40	211	2023-12-10
7	9456	1	4	Моховая	3	234567	20:28:08	242	2023-12-15
8	9567	2	1	Савушкина	3	123456	21:40:12	123	2023-12-20
9	9678	3	2	Ланская	2	567890	09:27:28	400	2023-12-20

Вывод

В ходе лабораторный я научился работать с триггерами