

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
образования «Национальный исследовательский университет ИТМО»  
Факультет инфокоммуникационных технологий

## **Лабораторная работа №3**

### **Процедуры, функции, триггеры в PostgreSQL**

Выполнил:

студент 2 курса ИКТ группа К32392,  
Елистратов Владимир Дмитриевич

Проверила:

Говорова Марина Михайловна

**Цель работы:**

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

**Вариант 17. БД “Телефонный провайдер ”**

Информационная система служит для хранения информации об абонентах телефонной компании и для учета оплаты всех видов услуг абонентами.

**Практическое задание:**

Вариант 2

- Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
- Создать авторский триггер по варианту индивидуального задания.
- 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.

## Создание процедур\функций

1. Вывести список всех звонков заданного абонента

Текст запроса:

--1--

```
CREATE OR REPLACE FUNCTION getCalls(person_name text)
```

```
RETURNS TABLE(CallID text)
```

```
LANGUAGE SQL
```

```
AS $$
```

```
    (SELECT Call."ID" FROM "Call" as Call
```

```
    Left Join public."Treaty" as Tr ON Call."TreatyID" = Tr."ID"
```

```
    WHERE Tr."FullNameID" = person_name)
```

```
$$;
```

--end--

Создание и результат выполнения функции:

```
Test=# CREATE OR REPLACE FUNCTION getCalls(person_name text)
Test=# RETURNS TABLE(CallID text)
Test=# LANGUAGE SQL
Test=# AS $$
Test$#  (SELECT Call."ID" FROM "Call" as Call
Test$#   Left Join public."Treaty" as Tr ON Call."TreatyID" = Tr."ID"
Test$#   WHERE Tr."FullNameID" = person_name)
Test$#  $$;
CREATE FUNCTION
Test=# SELECT * FROM getCalls('S000000005');
      callid
-----
C0000000012
C0000000006
(2 строки)

Test=#
```

2. Вывести задолженность по оплате для заданного абонента.

Текст запроса:

```
CREATE OR REPLACE FUNCTION getPersonCreditValue(person_name text)
RETURNS TABLE (PersonCreditValue text)
LANGUAGE SQL
AS $$
SELECT CallCredit."sum"+RateCredit."sum" FROM
    ((SELECT sum(Call."Price"::numeric) FROM public."Treaty" as Tr
        LEFT JOIN public."Call" as Call ON Call."TreatyID" = Tr."ID" and
        Call."PaymentState"='false'
        WHERE Tr."FullNameID" = person_name
    ) LIMIT 1) as CallCredit ,
    ((SELECT sum("Rate"."BasePrice"::numeric) FROM public."Treaty" as Tr
        Left Join "Treaty_Rate" as T_R ON T_R."TreatyID" = Tr."ID" and
        T_R."LastMonthPayStatus"='false'
        Left Join "Rate" ON T_R."RateID" = "Rate"."RateID"
        WHERE Tr."FullNameID" = person_name
    ) LIMIT 1) as RateCredit
$$;
```

Создание и результат выполнения функции:

```
Test=# CREATE OR REPLACE FUNCTION getPersonCreditValue(person_name text)
Test=# RETURNS TABLE (PersonCreditValue text)
Test=# LANGUAGE SQL
Test=# AS $$
Test$# SELECT CallCredit."sum"+RateCredit."sum" FROM
Test$# ((SELECT sum(Call."Price"::numeric) FROM public."Treaty" as Tr
Test$# LEFT JOIN public."Call" as Call ON Call."TreatyID" = Tr."ID" and Call."PaymentState"='false'
Test$# WHERE Tr."FullNameID" = person_name
Test$# ) LIMIT 1) as CallCredit ,
Test$# ((SELECT sum("Rate"."BasePrice"::numeric) FROM public."Treaty" as Tr
Test$# Left Join "Treaty_Rate" as T_R ON T_R."TreatyID" = Tr."ID" and T_R."LastMonthPayStatus"='false'
Test$# Left Join "Rate" ON T_R."RateID" = "Rate"."RateID"
Test$# WHERE Tr."FullNameID" = person_name
Test$# ) LIMIT 1) as RateCredit
Test$# $$;
CREATE FUNCTION
Test=# SELECT * FROM getPersonCreditValue('S000000005')
Test=# ;
 personcreditvalue
-----
 99.00
(1 строка)

Test=#
```

3. рассчитать общую стоимость звонков по каждой зоне за истекшую неделю.

Текст запроса:

```
CREATE OR REPLACE FUNCTION getZoneCallPrice()
```

```
RETURNS TABLE (ZoneName text, ZoneCode text, SumPrice numeric)
```

```
LANGUAGE SQL
```

```
AS $$
```

```
    SELECT Distinct ZN."ZoneName", Zn."ZoneCode", sum(Call."Price") FROM "Zone" as Zn
```

```
        LEFT JOIN public."Call" as Call ON Zn."ZoneCodeID" = Call."ZoneCodeID" and
```

```
        Call."StartCallDate" BETWEEN NOW() - interval '7 day' and NOW()
```

```
        Group By Zn."ZoneCodeID"
```

```
    $$;
```

Создание и результат выполнения функции:

```
Test=# CREATE OR REPLACE FUNCTION getZoneCallPrice()
Test=# RETURNS TABLE (ZoneName text, ZoneCode text, SumPrice numeric)
Test=# LANGUAGE SQL
Test=# AS $$
Test$$  SELECT Distinct ZN."ZoneName", Zn."ZoneCode", sum(Call."Price") FROM "Zone" as Zn
Test$$          LEFT JOIN public."Call" as Call ON Zn."ZoneCodeID" = Call."ZoneCodeID" and
Test$$          Call."StartCallDate" BETWEEN NOW() - interval '7 day' and NOW()
Test$$          Group By Zn."ZoneCodeID"
Test$$  $$;
CREATE FUNCTION
Test=# SELECT * FROM getZoneCallPrice();
 zonename | zonecode | sumprice
-----+-----+-----
 Belarus  | +375     | 
 China    | +86      | 
 Korea    | +82      | 
 Russia   | +7       | 
 Russia   | 8        | 
(5 строк)

Test=#
```

## Создание триггера

Создать триггер рассчитывающий стоимость звонка добавленного в БД

Текст запроса:

```
CREATE OR REPLACE FUNCTION SetCallPrice() RETURNS TRIGGER AS $my_table$

BEGIN

UPDATE "Call" SET "Price" =

        (SELECT tb."CallPrice" FROM

                (SELECT "Call"."ID", (((date_part('hour', "Call"."CallDuration") * 60) +
(date_part('minute', "Call"."CallDuration")) + (date_part('second', "Call"."CallDuration") /
60))::numeric * "Zone"."Coefficient" * ("Rate"."MinPrice")::numeric) as "CallPrice"

                        From "Call", "Rate", "Zone"

                                WHERE "Rate"."RateID" = "Call"."RateID" and "Zone"."ZoneCodeID"
= "Call"."ZoneCodeID" and "Call"."Price" is NULL

                                        AND new."ID" = "Call"."ID"

                                                ) AS tb)::numeric::money

        WHERE new."ID" = "Call"."ID" and "Call"."Price" is NULL;

Return NEW;

END;

$my_table$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER CallPriceAndPaymentCalc

AFTER INSERT on "Call"

FOR EACH row

WHEN (NEW."Price" is NULL)

EXECUTE PROCEDURE SetCallPrice();
```

Создание триггера и вспомогательной функции:

```
Test=# CREATE OR REPLACE FUNCTION SetCallPrice() RETURNS TRIGGER AS $my_table$
Test$# BEGIN
Test$# UPDATE "Call" SET "Price" =
Test$# (SELECT tb."CallPrice" FROM
Test$# (SELECT "Call"."ID", (((date_part('hour', "Call"."CallDuration") * 60) + (date
_part('minute', "Call"."CallDuration"))) + (date_part('second', "Call"."CallDuration") / 60))::numeric
* "Zone"."Coefficient" * ("Rate"."MinPrice")::numeric) as "CallPrice"
Test$# From "Call", "Rate", "Zone"
Test$# WHERE "Rate"."RateID" = "Call"."RateID" and "Zone"."ZoneCodeID" = "Cal
l"."ZoneCodeID" and "Call"."Price" is NULL
Test$# AND new."ID" = "Call"."ID"
Test$# ) AS tb)::numeric::money
Test$# WHERE new."ID" = "Call"."ID" and "Call"."Price" is NULL;
Test$# Return NEW;
Test$# END;
Test$# $my_table$ LANGUAGE plpgsql;
CREATE FUNCTION
Test=#
Test=# CREATE OR REPLACE TRIGGER CallPriceAndPaymentCalc
Test=# AFTER INSERT on "Call"
Test=# FOR EACH row
Test=# WHEN (NEW."Price" is NULL)
Test=# EXECUTE PROCEDURE SetCallPrice();
CREATE TRIGGER
Test=#
```

Внесем в таблицу 14 звонков  
Запрос и результат его выполнения без триггера:

```
TEST2=# INSERT INTO public."Call" (
TEST2=# "ID", "ZoneCodeID", "RateID", "TreatyID", "Treaty_Rate_ID", "TargetPnneNumber", "StartCallDate", "CallDuration", "PaymentState", "PaymentDate", "DateOfActualPayment")
TEST2=# VALUES
TEST2-# ('C000000015', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (944) 845-13-99', '2022-03-30 01:01:01', '00:01:12', 'false', '2022-03-30 01:01:01', null),
TEST2-# ('C000000021', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (944) 845-13-99', '2022-04-02 12:00:00', '00:25:11', 'false', '2022-04-02 12:00:00', null),
TEST2-# ('C000000031', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (955) 689-66-18', '2022-04-02 04:00:00', '00:00:30', 'false', '2022-04-02 04:00:00', null),
TEST2-# ('C000000041', '1', 'R000000001', 'T000000004', 'TR000000004', '+7 (955) 689-66-18', '2022-04-05 02:22:22', '00:02:22', 'false', '2022-04-05 02:22:22', null),
TEST2-# ('C000000051', '2', 'R000000001', 'T000000003', 'TR000000001', '8 (944) 845-13-99', '2022-03-06 11:11:11', '00:30:21', 'false', '2022-03-06 11:11:11', null),
TEST2-# ('C000000061', '2', 'R000000002', 'T000000005', 'TR000000002', '8 (955) 689-66-18', '2022-04-06 07:30:33', '00:10:23', 'false', '2022-04-06 07:30:33', null),
TEST2-# ('C000000071', '3', 'R000000005', 'T000000004', 'TR000000005', '+380 (44) 357-14-91', '2022-04-06 09:22:22', '00:01:22', 'false', '2022-04-06 09:22:22', null),
TEST2-# ('C000000081', '3', 'R000000005', 'T000000004', 'TR000000005', '+380 (95) 563-78-90', '2022-04-01 10:22:22', '00:11:15', 'false', '2022-04-01 10:22:22', null),
TEST2-# ('C000000091', '3', 'R000000003', 'T000000002', 'TR000000003', '+380 (67) 917-93-71', '2022-04-07 15:22:22', '00:33:20', 'false', '2022-04-07 15:22:22', null),
TEST2-# ('C000000101', '4', 'R000000003', 'T000000002', 'TR000000003', '+86 (10) 773-17-861', '2022-04-12 12:22:22', '00:05:11', 'false', '2022-04-12 12:22:22', null),
TEST2-# ('C000000111', '4', 'R000000001', 'T000000003', 'TR000000001', '+86 (15) 859-19-101', '2022-05-22 15:22:22', '00:00:45', 'false', '2022-05-22 15:22:22', null),
TEST2-# ('C000000121', '5', 'R000000002', 'T000000005', 'TR000000002', '+82 (10) 418-10-85', '2022-03-11 01:22:22', '00:02:22', 'false', '2022-03-11 01:22:22', null),
TEST2-# ('C000000131', '5', 'R000000004', 'T000000001', 'TR000000004', '+82 (11) 428-41-33', '2022-04-13 02:22:22', '00:10:01', 'false', '2022-04-13 02:22:22', null),
TEST2-# ('C000000141', '5', 'R000000005', 'T000000004', 'TR000000005', '+82 (12) 043-26-78', '2022-05-15 3:22:22', '01:35:53', 'false', '2022-05-15 3:22:22', null);
INSERT 0 14
TEST2=# SELECT * FROM "Call";
```

ID	ZoneCodeID	RateID	TreatyID	Treaty_Rate_ID	TargetPnneNumber	StartCallDate	CallDuration	Price	PaymentState	PaymentDate	DateOfActualPayment
C000000015	1	R000000003	T000000001	TR000000001	+7 (944) 845-13-99	2022-03-30 01:01:01	00:01:12		f	2022-03-30 01:01:01	
C000000021	1	R000000003	T000000001	TR000000001	+7 (944) 845-13-99	2022-04-02 12:00:00	00:25:11		f	2022-04-02 12:00:00	
C000000031	1	R000000003	T000000001	TR000000001	+7 (955) 689-66-18	2022-04-02 04:00:00	00:00:30		f	2022-04-02 04:00:00	
C000000041	1	R000000001	T000000004	TR000000004	+7 (955) 689-66-18	2022-04-05 02:22:22	00:02:22		f	2022-04-05 02:22:22	
C000000051	2	R000000001	T000000003	TR000000001	8 (944) 845-13-99	2022-03-06 11:11:11	00:30:21		f	2022-03-06 11:11:11	
C000000061	2	R000000002	T000000005	TR000000002	8 (955) 689-66-18	2022-04-06 07:30:33	00:10:23		f	2022-04-06 07:30:33	
C000000071	3	R000000005	T000000004	TR000000005	+380 (44) 357-14-91	2022-04-06 09:22:22	00:01:22		f	2022-04-06 09:22:22	
C000000081	3	R000000005	T000000004	TR000000005	+380 (95) 563-78-90	2022-04-01 10:22:22	00:11:15		f	2022-04-01 10:22:22	
C000000091	3	R000000003	T000000002	TR000000003	+380 (67) 917-93-71	2022-04-07 15:22:22	00:33:20		f	2022-04-07 15:22:22	
C000000101	4	R000000003	T000000002	TR000000003	+86 (10) 773-17-861	2022-04-12 12:22:22	00:05:11		f	2022-04-12 12:22:22	
C000000111	4	R000000001	T000000003	TR000000001	+86 (15) 859-19-101	2022-05-22 15:22:22	00:00:45		f	2022-05-22 15:22:22	
C000000121	5	R000000002	T000000005	TR000000002	+82 (10) 418-10-85	2022-03-11 01:22:22	00:02:22		f	2022-03-11 01:22:22	
C000000131	5	R000000004	T000000001	TR000000004	+82 (11) 428-41-33	2022-04-13 02:22:22	00:10:--	Далее --			

## Запрос и результат его выполнения с триггером:

```
Test=# INSERT INTO public."Call"(  
Test=#   "ID", "ZoneCodeID", "RateID", "TreatyID", "Treaty_Rate_ID", "TargetPnOneNumber", "StartCallDate", "CallDuration", "PaymentState", "PaymentDate", "DateOfActualPayment")  
Test=# VALUES  
Test=# ('C000000019', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (944) 845-13-99', '2022-03-30 01:01:01', '00:01:12', 'false', '2022-03-30 01:01:01', null),  
Test=# ('C000000021', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (944) 845-13-99', '2022-04-02 12:00:00', '00:25:11', 'false', '2022-04-02 12:00:00', null),  
Test=# ('C000000031', '1', 'R000000003', 'T000000001', 'TR000000001', '+7 (955) 689-66-18', '2022-04-02 04:00:00', '00:00:30', 'false', '2022-04-02 04:00:00', null),  
Test=# ('C000000041', '1', 'R000000001', 'T000000004', 'TR000000004', '+7 (955) 689-66-18', '2022-04-05 02:22:22', '00:02:22', 'false', '2022-04-05 02:22:22', null),  
Test=# ('C000000051', '2', 'R000000001', 'T000000003', 'TR000000001', '8 (944) 845-13-99', '2022-03-06 11:11:11', '00:30:21', 'false', '2022-03-06 11:11:11', null),  
Test=# ('C000000061', '2', 'R000000002', 'T000000005', 'TR000000002', '8 (955) 689-66-18', '2022-04-06 07:30:33', '00:10:23', 'false', '2022-04-06 07:30:33', null),  
Test=# ('C000000071', '3', 'R000000005', 'T000000004', 'TR000000005', '+380 (44) 357-14-91', '2022-04-06 09:22:22', '00:01:22', 'false', '2022-04-06 09:22:22', null),  
Test=# ('C000000081', '3', 'R000000005', 'T000000004', 'TR000000005', '+380 (95) 563-78-90', '2022-04-01 10:22:22', '00:11:15', 'false', '2022-04-01 10:22:22', null),  
Test=# ('C000000091', '3', 'R000000003', 'T000000002', 'TR000000003', '+380 (67) 917-93-71', '2022-04-07 15:22:22', '00:33:20', 'false', '2022-04-07 15:22:22', null),  
Test=# ('C000000101', '4', 'R000000003', 'T000000002', 'TR000000003', '+86 (10) 773-17-861', '2022-04-12 12:22:22', '00:05:11', 'false', '2022-04-12 12:22:22', null),  
Test=# ('C000000111', '4', 'R000000001', 'T000000003', 'TR000000001', '+86 (15) 859-19-101', '2022-05-22 15:22:22', '00:00:45', 'false', '2022-05-22 15:22:22', null),  
Test=# ('C000000121', '5', 'R000000005', 'T000000005', 'TR000000001', '+82 (10) 418-10-85', '2022-03-11 01:22:22', '00:02:22', 'false', '2022-03-11 01:22:22', null),  
Test=# ('C000000131', '5', 'R000000004', 'T000000001', 'TR000000004', '+82 (11) 428-41-33', '2022-04-13 02:22:22', '00:10:01', 'false', '2022-04-13 02:22:22', null),  
Test=# ('C000000141', '5', 'R000000005', 'T000000004', 'TR000000005', '+82 (12) 043-26-78', '2022-05-15 3:22:22', '01:35:53', 'false', '2022-05-15 3:22:22', null);  
INSERT 0 14  
Test=# SELECT * FROM "Call";  
ID | ZoneCodeID | RateID | TreatyID | Treaty_Rate_ID | TargetPnOneNumber | StartCallDate | CallDuration | Price | PaymentState | PaymentDate | DateOfActualPayment  
-----  
C000000001 | 1 | R000000003 | T000000001 | TR000000001 | +7 (944) 845-13-99 | 2022-03-30 01:01:01 | 00:01:12 | 272,40 ? | f | 2022-03-30 01:01:01 |  
C000000002 | 1 | R000000003 | T000000001 | TR000000001 | +7 (944) 845-13-99 | 2022-04-02 12:00:00 | 00:25:11 | 5a716,62 ? | f | 2022-04-02 12:00:00 |  
C000000003 | 1 | R000000003 | T000000001 | TR000000001 | +7 (955) 689-66-18 | 2022-04-02 04:00:00 | 00:00:30 | 113,50 ? | f | 2022-04-02 04:00:00 |  
C000000004 | 1 | R000000001 | T000000004 | TR000000004 | +7 (955) 689-66-18 | 2022-04-05 02:22:22 | 00:02:22 | 2a364,30 ? | f | 2022-04-05 02:22:22 |  
C000000005 | 2 | R000000001 | T000000003 | TR000000001 | 8 (944) 845-13-99 | 2022-03-06 11:11:11 | 00:30:21 | 30a319,65 ? | f | 2022-03-06 11:11:11 |  
C000000006 | 2 | R000000002 | T000000005 | TR000000002 | 8 (955) 689-66-18 | 2022-04-06 07:30:33 | 00:10:23 | 0,00 ? | f | 2022-04-06 07:30:33 |  
C000000007 | 3 | R000000005 | T000000004 | TR000000005 | +380 (44) 357-14-91 | 2022-04-06 09:22:22 | 00:01:22 | 1,37 ? | f | 2022-04-06 09:22:22 |  
C000000008 | 3 | R000000005 | T000000004 | TR000000005 | +380 (95) 563-78-90 | 2022-04-01 10:22:22 | 00:11:15 | 11,25 ? | f | 2022-04-01 10:22:22 |  
C000000009 | 3 | R000000003 | T000000002 | TR000000003 | +380 (67) 917-93-71 | 2022-04-07 15:22:22 | 00:33:20 | 7a566,67 ? | f | 2022-04-07 15:22:22 |  
C000000010 | 4 | R000000003 | T000000002 | TR000000003 | +86 (10) 773-17-861 | 2022-04-12 12:22:22 | 00:05:11 | 1a011,94 ? | f | 2022-04-12 12:22:22 |  
C000000011 | 4 | R000000001 | T000000003 | TR000000001 | +86 (15) 859-19-101 | 2022-05-22 15:22:22 | 00:00:45 | 899,10 ? | f | 2022-05-22 15:22:22 |  
C000000012 | 5 | R000000002 | T000000005 | TR000000002 | +82 (10) 418-10-85 | 2022-03-11 01:22:22 | 00:02:22 | 0,00 ? | f | 2022-03-11 01:22:22 |  
C000000013 | 5 | R000000004 | T000000001 | TR000000004 | +82 (11) 428-41-33 | 2022-04-13 02:22:22 | 00:10:01 | 100,17 ? | f | 2022-04-13 02:22:22 |  
C000000014 | 5 | R000000005 | T000000004 | TR000000005 | +82 (12) 043-26-78 | 2022-05-15 03:22:22 | 01:35:53 | 191,77 ? | f | 2022-05-15 03:22:22 |  
C000000019 | 1 | R000000003 | T000000001 | TR000000001 | +7 (944) 845-13-99 | 2022-03-30 01:01:01 | 00:01:12 | 272,40 ? | f | 2022-03-30 01:01:01 |  
C000000021 | 1 | R000000003 | T000000001 | TR000000001 | +7 (944) 845-13-99 | 2022-04-02 12:00:00 | 00:25:11 | 5a716,62 ? | f | 2022-04-02 12:00:00 |  
C000000031 | 1 | R000000003 | T000000001 | TR000000001 | +7 (955) 689-66-18 | 2022-04-02 04:00:00 | 00:00:30 | 113,50 ? | f | 2022-04-02 04:00:00 |  
C000000041 | 1 | R000000001 | T000000004 | TR000000004 | +7 (955) 689-66-18 | 2022-04-05 02:22:22 | 00:02:22 | 2a364,30 ? | f | 2022-04-05 02:22:22 |  
C000000051 | 2 | R000000001 | T000000003 | TR000000001 | 8 (944) 845-13-99 | 2022-03-06 11:11:11 | 00:30:21 | 30a319,65 ? | f | 2022-03-06 11:11:11 |  
C000000061 | 2 | R000000002 | T000000005 | TR000000002 | 8 (955) 689-66-18 | 2022-04-06 07:30:33 | 00:10:23 | 0,00 ? | f | 2022-04-06 07:30:33 |  
C000000071 | 3 | R000000005 | T000000004 | TR000000005 | +380 (44) 357-14-91 | 2022-04-06 09:22:22 | 00:01:22 | 1,37 ? | f | 2022-04-06 09:22:22 |  
C000000081 | 3 | R000000005 | T000000004 | TR000000005 | +380 (95) 563-78-90 | 2022-04-01 10:22:22 | 00:11:15 | 11,25 ? | f | 2022-04-01 10:22:22 |  
C000000091 | 3 | R000000003 | T000000002 | TR000000003 | +380 (67) 917-93-71 | 2022-04-07 15:22:22 | 00:33:20 | 7a566,67 ? | f | 2022-04-07 15:22:22 |  
C000000101 | 4 | R000000003 | T000000002 | TR000000003 | +86 (10) 773-17-861 | 2022-04-12 12:22:22 | 00:05:11 | 1a011,94 ? | f | 2022-04-12 12:22:22 |  
C000000111 | 4 | R000000001 | T000000003 | TR000000001 | +86 (15) 859-19-101 | 2022-05-22 15:22:22 | 00:00:45 | 899,10 ? | f | 2022-05-22 15:22:22 |  
C000000121 | 5 | R000000002 | T000000005 | TR000000002 | +82 (10) 418-10-85 | 2022-03-11 01:22:22 | 00:02:22 | 0,00 ? | f | 2022-03-11 01:22:22 |  
C000000131 | 5 | R000000004 | T000000001 | TR000000004 | +82 (11) 428-41-33 | 2022-04-13 02:22:22 | 00:10:01 | 100,17 ? | f | 2022-04-13 02:22:22 |  
C000000141 | 5 | R000000005 | T000000004 | TR000000005 | +82 (12) 043-26-78 | 2022-05-15 03:22:22 | 01:35:53 | 191,77 ? | f | 2022-05-15 03:22:22 |
```

## Модификация триггера из практической работы

```
1 CREATE OR REPLACE FUNCTION fn_check_time_punch() RETURNS TRIGGERS AS $psql$  
2 DECLARE prev tp RECORD;  
3 BEGIN  
4     SELECT punch_time, is_out_punch INTO prev_tp  
5     FROM time_punch  
6     WHERE employee_id = new.employee_id  
7     ORDER BY punch_time DESC  
8     LIMIT 1;  
9     IF prev_tp IS NULL THEN RETURN new;  
10    END IF;  
11    IF prev_tp.is out punch = new.is_out_punch  
12    OR prev_tp.punch time >= new.punch_time  
13    OR new_punch_time > now()  
14    THEN RETURN NULL;  
15    END IF;  
16    RETURN new;  
17 end;  
18 $psql$ language plpgsql;
```

## Выводы:

В процессе выполнения данной лабораторной работы удалось овладеть навыками написания и использования процедур, функций и триггеров в PSQL. Функции и процедуры SQL имеют схожесть с языками программирования, но очень неудобны и ограничены. Простейшие задачи могут вызывать трудности из-за синтаксиса и невозможности выполнить некоторые действия.