

Национальный исследовательский Университет ИТМО  
Мегафакультет информационных и трансляционных технологий  
Факультет мобильных и сетевых технологий

# Проектирование и реализация баз данных

Лабораторная работа №5

Процедуры, функции, триггеры в PostgreSQL

**Работу выполнил:**

Фадеев Д.А.

Группа: К3239

**Преподаватель:**

Говорова М.М.

Санкт-Петербург  
2023

## Цель Работы

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

## Практическое задание

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

## Выполнение

### Процедуры и функции

- Для снижения цены на заданный процент для товаров, у которых срок пребывания на складе превысил заданный норматив.

– Код

```
Query Query History
1 CREATE OR REPLACE PROCEDURE wholesale_base.reduce_the_price_of_purchase_list(time_var TIME
2 LANGUAGE SQL
3 AS $$
4     UPDATE wholesale_base.purchase_list pl
5     SET price = price * 0.8
6     FROM wholesale_base.purchase pu
7     WHERE pu.purchase_id = pl.purchase_id
8     AND pu.date_of_purchase <= time_var;
9 $$
```

– До применения

```
wholesale_base=# SELECT pl.purchase_list_id, pu.date_of_purchase, pl.price, pl.c
ount
FROM wholesale_base.purchase_list pl
JOIN wholesale_base.purchase pu ON pl.purchase_id = pu.purchase_id
WHERE date_of_purchase <= CURRENT_DATE - INTERVAL '4 year';
 purchase_list_id | date_of_purchase | price | count
-----+-----+-----+-----
6 | 2019-05-15 | 251 | 83
36 | 2019-05-15 | 152 | 49
75 | 2019-06-20 | 495 | 83
80 | 2019-08-30 | 117 | 64
2 | 2019-08-30 | 577 | 51
17 | 2019-10-10 | 555 | 74
35 | 2019-07-25 | 315 | 21
16 | 2019-10-10 | 275 | 55
27 | 2019-07-25 | 579 | 27
30 | 2019-09-05 | 171 | 24
42 | 2019-08-30 | 366 | 75
69 | 2019-10-10 | 481 | 69
83 | 2019-07-25 | 392 | 37
(13 строк)
```

– После применения





- Триггер автоматически вычитает количество купленного товара из его остатков на складе

– Код функции

```
CREATE OR REPLACE FUNCTION wholesale_base.subtract_from_leftovers()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
  UPDATE wholesale_base.purchase_list
  SET leftovers = leftovers - NEW.count
  WHERE purchase_list_id = NEW.purchase_list_id;

  RETURN NEW;
END;
$BODY$;

ALTER FUNCTION wholesale_base.subtract_from_leftovers()
  OWNER TO postgres;
```

– Код триггера

```
CREATE OR REPLACE TRIGGER subtract_count_from_leftovers
  AFTER INSERT
  ON wholesale_base.order_list
  FOR EACH ROW
  EXECUTE FUNCTION wholesale_base.subtract_from_leftovers();
```

– Применение

```
wholesale_base=# SELECT leftovers FROM wholesale_base.purchase_list
WHERE purchase_list_id = 32;
 leftovers
-----
          79
(1 строка)

wholesale_base=# INSERT INTO wholesale_base.order_list(price,
count, order_id, purchase_list_id)
VALUES (258,29,31,32);
INSERT 0 1
wholesale_base=# SELECT leftovers FROM wholesale_base.purchase_list
WHERE purchase_list_id = 32;
 leftovers
-----
          50
(1 строка)

wholesale_base=# █
```

## Модифицирование триггера

- Код функции

```

create or replace function fn_check_time_punch() returns trigger as $psql$
begin
    if new.punch_time > now()

    or new.is_out_punch = (
        select tps.is_out_punch
        from time_punch tps
        where tps.employee_id = new.employee_id
        order by tps.id desc limit 1
    )
    or new.punch_time <= (
        select tps.punch_time from time_punch tps
        where tps.employee_id = new.employee_id
        order by tps.id desc limit 1
    )
    then
        return null;
    end if;
    return new;
end;
$psql$ language plpgsql;

```

- Код триггера

```

drop trigger if exists check_time_punch on time_punch;
create trigger check_time_punch before insert on time_punch
for each row execute procedure fn_check_time_punch();

```

- Применение

```

emp_time=# INSERT INTO time_punch(employee_id, is_out_punch, punch_time)
VALUES
(3, false, '2022-01-01 00:00:00'),
[(3, true, '2022-01-01 02:00:00');
INSERT 0 2
emp_time=# --Выход раньше входа--
emp_time=# INSERT INTO time_punch(employee_id, is_out_punch, punch_time)
VALUES
(3, false, '2022-01-01 04:00:00'),
[(3, true, '2022-01-01 03:00:00');
INSERT 0 1
emp_time=# --Второй выход к ряду--
emp_time=# INSERT INTO time_punch(employee_id, is_out_punch, punch_time)
VALUES
[(3, false, '2022-01-01 05:00:00');
INSERT 0 0
emp_time=# █

```

## Вывод

В ходе выполнения лабораторной работы были созданы и протестированны процедуры, функции и триггеры.