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:

Дущенко Д.А. 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

- 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

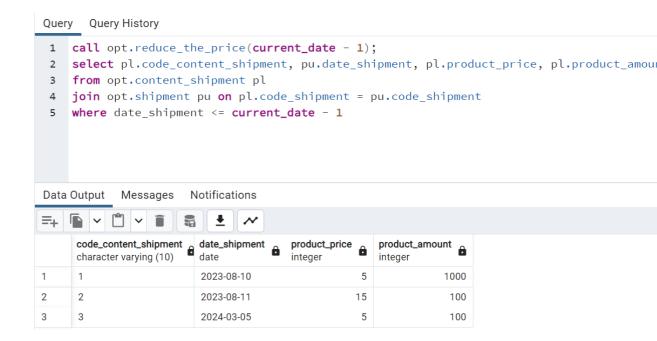
• Для снижения цены на заданный процент для товаров, у которых срок пребывания на складе превысил заданный норматив.

- Код Query Query History create or replace procedure opt.reduce_the_price(time_var TIMESTAMP) 1 2 language sql as \$\$ 3 4 update opt.content_shipment pl 5 set product_price = product_price * 0.8 from opt.shipment pu 6 7 where pu.code_shipment = pl.code_shipment and pu.date_shipment <= time_var</pre> 8 \$\$ 9

- До применения

Notifications Data Output Messages =+ code_content_shipment date_shipment product_price product_amount date character varying (10) integer integer 1 1 2023-08-10 6 1000 2 2 2023-08-11 19 100 3 3 100 2024-03-05 6

- После применения

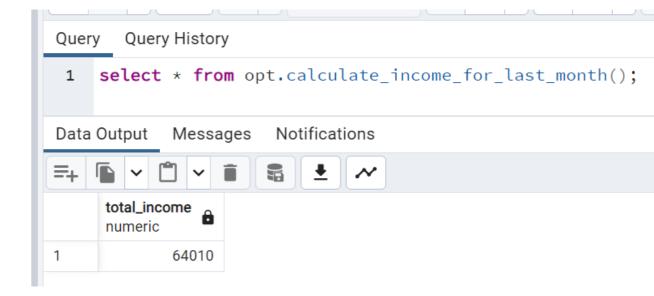


• Для расчета стоимости всех партий товаров, проданных за прошедшие сутки.

- Код

```
Query Query History
1 create or replace function opt.calculate_income_for_last_month()
   returns table(total_income numeric)
3
   language plpgsql
4
    as $$
5 ▼ begin
6
    return query
7
        select sum(income) as total_income
8
        from(
            select o.date_order as date_day, sum(ol.product_amount * ol.product_price) as income
9
10
            from opt.order o
11
            join opt.order_content ol on o.code_order = ol.code_order
12
            where o.date_order between current_date - interval '1 year' and current_date
13
            group by o.date_order
14
        );
15 end;
   $$;
16
```

- Применине



Триггеры

• Триггер проверяет достаточно ли остатков для конкретного товара на складе, для того чтобы использовать во вставке к покупке конкретного товара со склада

- Код функции

```
Query Query History
 1
   create or replace function opt.check_if_rest_enough()
2
       returns trigger
3
        language 'plpgsql'
        cost 100
 5
        volatile not leakproof
 6 as $BODY$
7 ▼ begin
8 ₹
        if (select rest_shipment from opt.content_shipment where code_content_shipment = new.code_content_s
9
            raise exception 'The count cannot be greater than rest of shipment';
10
        end if;
11
12
        return new;
14 end;
   $BODY$;
15
```

- Код триггера

- Применение

Query Query History

1 insert into opt.order_content(code_order_content, code_order, code_product, code_content_shipment, proc values (1, 1, 1, 10, 50)

Data Output Messages Notifications

ERROR: The count cannot be greater than rest of shipment
CONTEXT: функция PL/pgSQL opt.check_if_rest_enough(), строка 5, oneparop RAISE

ОШИБКА: The count cannot be greater than rest of shipment
SQL state: P0001

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

```
Query Query History
```

```
create or replace function opt.subtract_from_rest()
1
2
        returns trigger
        language 'plpgsql'
3
        cost 100
4
        volatile not leakproof
5
    as $BODY$
6
7 ▼
    begin
        update opt.content_shipment
8
        set rest_shipment = rest_shipment - new.product_amount
9
        where code_content_shipment = new.code_content_shipment;
10
11
12
        return new;
    end:
13
    $BODY$;
14
```

- Код триггера

1

```
Query Query History

create or replace trigger subtract_count_from_rest
after insert
on opt.order_content
for each row
execute function opt.subtract_from_rest();
```

Модифицирование триггера

• Код функции

```
Query
       Query History
    create or replace function fn_check_time_punch() returns trigger as $psq1
1
2
        begin
 3 ₩
            if new.punch_time > now()
4
            or new.is_out_punch = (
5
                 select tps.is_out_punch
6
                 from time_punch tps
7
8
                 where tps.employee_id = new.employee_id
                 order by tps.id desc limit 1
9
10
            or new.punch_time <= (</pre>
11
                 select tps.punch_time from time_punch tps
12
                 where tps.employee_id = new.employee_id
13
14
                 order by tps.id desc limit 1
15
            then
16
```

• Код триггера

```
Query Query History

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

• Применение

```
Query Query History

1 insert into time_punch(employee_id, is_out_punch, punch_time)
2 values (1, false, '2022-01-01 02:00:00'), (1, true, '2022-01-01 01:00:00')

Data Output Messages Notifications

INSERT 0 0
```

Вывод

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