

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

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

– До применения

Data Output

Messages

Notifications

| | <div>code_content_shipment</div> <div>character varying (10)</div> <div></div> | <div>date_shipment</div> <div>date</div> <div></div> | <div>product_price</div> <div>integer</div> <div></div> | <div>product_amount</div> <div>integer</div> <div></div> |
|---|--|--|---|--|
| 1 | 1 | 2023-08-10 | 6 | 1000 |
| 2 | 2 | 2023-08-11 | 19 | 100 |
| 3 | 3 | 2024-03-05 | 6 | 100 |

– После применения

Query

Query History

1

call

opt.reduce_the_price(current_date - 1);

2

select

pl.code_content_shipment, pu.date_shipment, pl.product_price, pl.product_amount

3

from

opt.content_shipment pl

4

join

opt.shipment pu on pl.code_shipment = pu.code_shipment

5

where

date_shipment <= current_date - 1

Data Output

Messages

Notifications

| | code_content_shipment character varying (10) | date_shipment date | product_price integer | product_amount integer |
|---|---|-----------------------|--------------------------|---------------------------|
| 1 | 1 | 2023-08-10 | 5 | 1000 |
| 2 | 2 | 2023-08-11 | 15 | 100 |
| 3 | 3 | 2024-03-05 | 5 | 100 |

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

– Код

| Query | Query History |
|---|---------------|
| <pre>1 create or replace function opt.calculate_income_for_last_month() 2 returns table(total_income numeric) 3 language plpgsql 4 as \$\$ 5 ▼ begin 6 return query 7 select sum(income) as total_income 8 from(9 select o.date_order as date_day, sum(ol.product_amount * ol.product_price) as income 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 \$\$;</pre> | |

– Примените

The screenshot shows a SQL query editor with a tab labeled "Query" selected. The query is: `1 select * from opt.calculate_income_for_last_month();`. Below the query editor, there are three tabs: "Data Output", "Messages", and "Notifications". The "Data Output" tab is selected, showing a table with one column, "total_income", which is numeric. The table has one row with the value 64010.

| | total_income numeric |
|---|-------------------------|
| 1 | 64010 |

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

```

1  create or replace function opt.check_if_rest_enough()
2      returns trigger
3      language 'plpgsql'
4      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      then
10         raise exception 'The count cannot be greater than rest of shipment';
11     end if;
12
13     return new;
14 end;
15 $BODY$;

```

– Код триггера

```
Query  Query History
1  create or replace trigger check_if_rest_enough_before_insert
2      before insert
3      on opt.order_content
4      for each row
5      execute function opt.check_if_rest_enough();
```

– Применение

Query

Query History

1

insert into opt.order_content(code_order_content, code_order, code_product, code_content_shipment, proc

2

values (1, 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, оператор 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

```

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

```

– Код триггера

Query Query History

```

1 create or replace trigger subtract_count_from_rest
2     after insert
3     on opt.order_content
4     for each row
5     execute function opt.subtract_from_rest();

```

– Применение

Query Query History

```

1 select rest_shipment from opt.content_shipment where code_content_shipment = '1';

```

Data Output Messages Notifications



| rest_shipment integer | |
|-----------------------|---|
| 1 | 9 |

Модифицирование триггера

- Код функции

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

- 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.