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: Пеньков Г.Д.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М. М.



Санкт-Петербург 2023

Цель работы

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

Вариант 3. БД «Библиотека»

Практическое задание:

Вариант 2 (тах - 8 баллов)

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

Выполнение

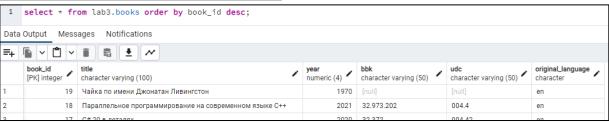
- 1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).
 - 1.1. Для проверки наличия экземпляров заданной книги в библиотеке (процедура должна возвращать количество экземпляров книги).

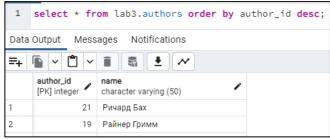
```
create or replace function lab3.check_book_availability(_book_id int)
 2
      returns int
3
      language plpgsql
4 as
 5
   ŚŚ
 6
     declare num_of_available_copies int;
7
 8 ₩
       if _book_id not in (select book_id from lab3.books) then
9
        raise exception 'Книга с id % не найдена', _book_id;
10
     end if;
11
     SELECT COUNT(*) INTO NUM_OF_AVAILABLE_COPIES
12
       FROM LAB3.PUBLICATION_COPIES AS PC
13
14
         (SELECT PC.INVENTORY_NUMBER
15
         FROM LAB3.PUBLICATION_COPIES AS PC
16
        WHERE PC.WRITE_OFF_ACT IS NULL
17
         EXCEPT SELECT BOOK
18
         FROM LAB3.BOOK_LENDING AS BL
19
         WHERE REAL_RETURN_DATE IS NULL) AS AVAILABLE_COPIES
20
      ON AVAILABLE_COPIES.INVENTORY_NUMBER = PC.INVENTORY_NUMBER
21
       JOIN LAB3. PUBLICATIONS AS PUB ON PC. ISBN = PUB. ISBN
22
       JOIN LAB3.BOOKS AS B ON B.BOOK_ID = PUB.BOOK
23
       WHERE B.BOOK_ID = _book_id;
24
       return NUM_OF_AVAILABLE_COPIES;
25
     end;
26 $$;
27  select * from lab3.check_book_availability(1)
Data Output Messages Notifications
check_book_availability
    integer
                    4
```

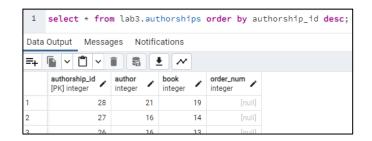
1.2. Для ввода в базу данных новой книги.

```
CREATE OR REPLACE PROCEDURE lab3.add_book(
 2
      title VARCHAR(100),
 3
      original_lang_code varchar(2),
 4
      author varchar(50),
 5
      year numeric(4) default null,
     bbk varchar(50) default null,
 7
     udk varchar(50) default null
 8
9 language plpgsql
10 as $$
11 declare
12
     _author_id int;
13
      _book_id int;
14▼ begin
15 ₹
      if author not in (select name from lab3.authors) then
16
          insert into lab3.authors (name) values (author) returning author_id into _author_id;
17
      else
18
        select author_id into _author_id from lab3.authors where name = author;
19
        end if:
20
21 ₹
      if original_lang_code not in (select lang_code from lab3.book_original_languages) then
22
        insert into lab3.book_original_languages (lang_code) values (original_lang_code);
23
        end if:
24
25
      insert into lab3.books (title, year, bbk, udc, original_language)
26
      values (title, year, bbk, udk, original_lang_code) returning book_id into _book_id;
27
28
      insert into lab3.authorships (author, book) values (_author_id, _book_id);
29 end;
30
    $$;
31
1 call lab3.add book(
2
        'Чайка по имени Джонатан Ливингстон',
```

```
1 call lab3.add_book(
2 'Чайка по имени Джонатан Ливингстон',
3 'en',
4 'Ричард Бах',
5 1970
6 );
```







1.3. Для ввода нового читателя (необходимо проверить наличие читателя в картотеке, чтобы не назначить ему номер вторично).

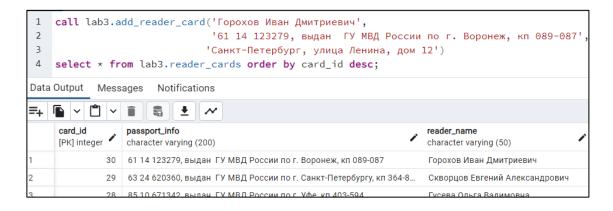
```
create or replace procedure lab3.add_reader_card(
       _name varchar(50),
3
       _passport_info varchar(200),
4
       address varchar(100).
5
       _phone_num varchar(20) default null,
6
       _email varchar(50) default null,
       _education varchar(50) default null
8
9
  language plpgsql
10 as $$
11⊽ begin
12 ₹
       if _passport_info in (select passport_info from lab3.reader_cards) then
13
           raise exception 'Данный читатель уже существует';
14
       end if:
       insert into lab3.reader_cards (passport_info, reader_name, phone_number, address, email, education)
16
       values (_passport_info, _name, _phone_num, _address, _email, _education);
  end; $$;
```

```
1 call lab3.add_reader_card('Куприянов Александр Александрович',
2 '60 13 099279, выдан ГУ МВД России по г. Саратову, кп 088-049',
3 'Санкт-Петербург, улица Ленина, дом 10')

Data Output Messages Notifications

ERROR: Данный читатель уже существует
CONTEXT: функция PL/pgSQL lab3.add_reader_card(character varying,character varying,character varying,character varying.

ОШИБКА: Данный читатель уже существует
SQL state: P0001
```



2. Модификация триггера из практического задания

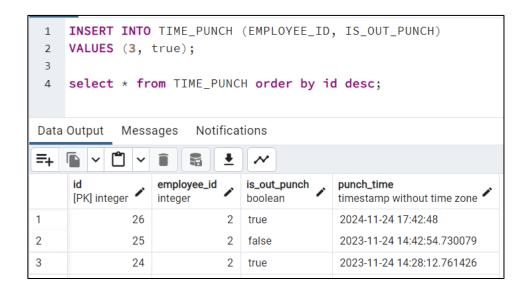
Для создания триггерной функции мы создали вспомогательную функцию, которая возвращает последнее время выхода (или входа) для заданного работника:

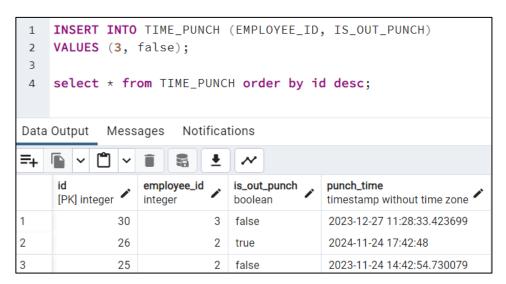


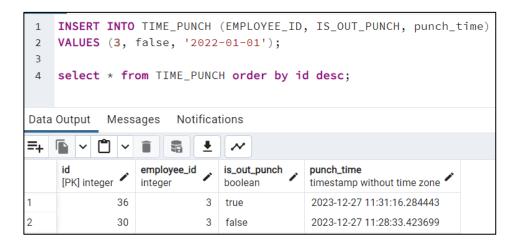
Триггерная функция и создание триггера:

```
create or replace function fn_check_time_punch() returns trigger as $psql$
        declare
2
3
            _last_punch_time timestamp;
4
            _last_is_out_punch bool;
5
        begin
           select is_out_punch into _last_is_out_punch from get_last_punch(new.employee_id);
6
           select last_punch_time into _last_punch_time from get_last_punch(new.employee_id);
7
8
            -- нельзя выйти/войти 2 раза подряд
9
10
            if new.is_out_punch = _last_is_out_punch then return null;
11
            end if;
12
            -- выход не может быть первой записью (сначала нужно войти)
13
           if new.is_out_punch = true and _last_is_out_punch is null then return null;
14
           end if;
15
            -- новый выход/вход должен быть позже последнего
16
            if new.punch_time < _last_punch_time then return null;</pre>
            end if;
17
18
            return new;
19
        end:
20 $psql$ language plpgsql;
21
   create trigger check_time_punch before insert on time_punch
22
23
   for each row execute procedure fn_check_time_punch();
```

Примеры выполнения:







3. Создание триггера по варианту индивидуального задания

```
1 CREATE OR REPLACE FUNCTION FN_CHECK_BOOK_LENDING() RETURNS TRIGGER LANGUAGE PLPGSQL AS $$
2 ▼ begin
3
        -- нельзя выдать книгу, которая уже выдана
4 ₹
       if new.book in
5
           (select book from lab3.book_lending where real_return_date is null) then
6
               return null;
7
       end if;
8
9
        -- нельзя выдать книгу, которая списана с учета
10 ₹
       if new.book in
           (select inventory_number from lab3.publication_copies where write_off_act is not null) then
11
12
               return null;
13
       end if:
14
15
       -- нельзя выдать книгу, если у человека более чем 3 задолженных книги
16 ₹
       if (select num_of_fined_books from lab3.debtors where card_id = new.reader_card) > 3 then
17
           return null;
18
       end if;
19
       return new;
20 end; $$;
21
22 create trigger check_book_lending before insert on lab3.book_lending
for each row execute procedure fn_check_book_lending();
```

Примеры выполнения:

```
1 insert into lab3.book_lending (reader_card, book, employee)
2 values (24, 89, 4)
3 -- книга с инвентарным номером 89 уже взята

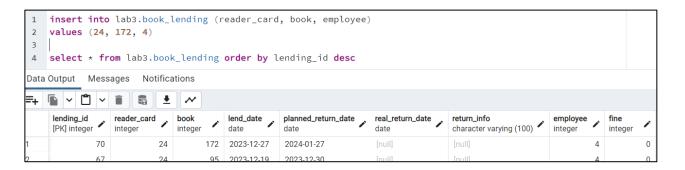
Data Output Messages Notifications

INSERT 0 0
```

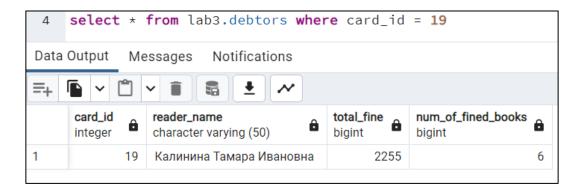
```
1 insert into lab3.book_lending (reader_card, book, employee)
2 values (24, 73, 4)
3 -- книга с инвентарным номером 73 снята с учета

Data Output Messages Notifications

INSERT 0 0
```







Вывод

В данной лабораторной работе мы написали несколько процедур, функций и триггеров. Тем самым, мы познакомились с основами процедурного языка PL/pgSQL.