

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**  
по лабораторной работе 3  
«процедуры, функции, триггеры в PostgreSQL»  
по дисциплине **«Проектирование и реализация баз данных»**  
Вариант 3

Автор: Мосин З. И.

Факультет: Инфокоммуникационных технологий (ИКТ)

Группа: K32402

Преподаватель: Говорова М. М.

Дата: 10.08.2023

**ИТМО**

Санкт-Петербург 2022

**Цель работы:** овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

**Практическое задание:**

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5). Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

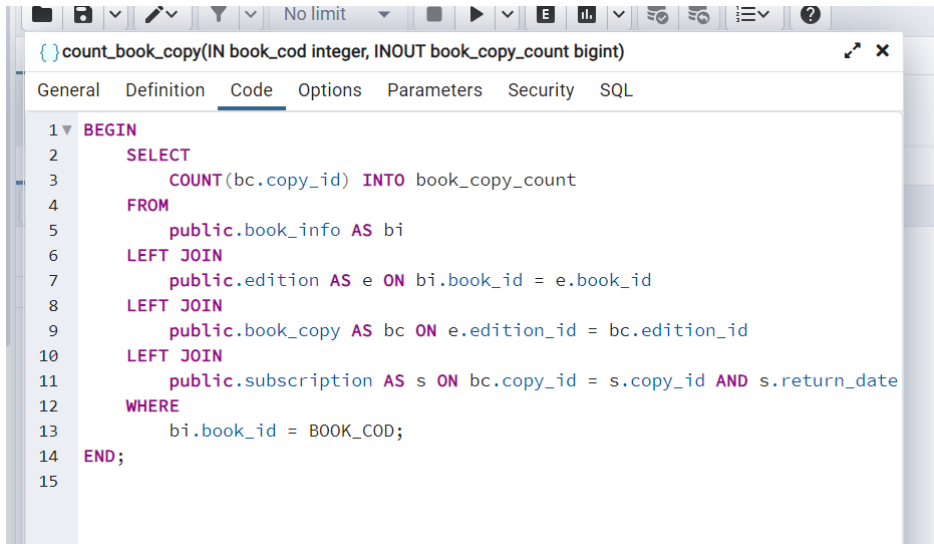
**Задание 1.** Создать хранимые процедуры:

- Для проверки наличия экземпляров заданной книги в библиотеке (процедура должна возвращать количество экземпляров книги).
- Для ввода в базу данных новой книги.
- Для ввода нового читателя (необходимо проверить наличие читателя в картотеке, чтобы не назначить ему номер вторично).

## Выполнение:

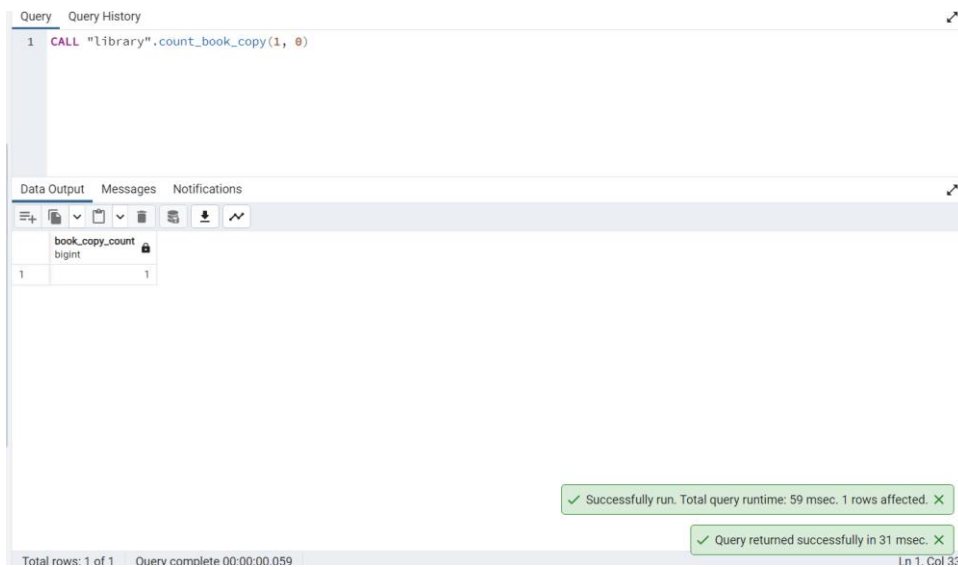
1. Для проверки наличия экземпляров заданной книги в библиотеке (процедура должна возвращать количество экземпляров книги).

Реализация процедуры:



```
{ } count_book_copy(IN book_cod integer, INOUT book_copy_count bigint)
General Definition Code Options Parameters Security SQL
1 BEGIN
2     SELECT
3         COUNT(bc.copy_id) INTO book_copy_count
4     FROM
5         public.book_info AS bi
6     LEFT JOIN
7         public.edition AS e ON bi.book_id = e.book_id
8     LEFT JOIN
9         public.book_copy AS bc ON e.edition_id = bc.edition_id
10    LEFT JOIN
11        public.subscription AS s ON bc.copy_id = s.copy_id AND s.return_date
12    WHERE
13        bi.book_id = BOOK_COD;
14 END;
15
```

Результат вызова процедуры:



```
Query Query History
1 CALL "library".count_book_copy(1, 0)
```

Data Output	
book_copy_count	bigint
1	1

Successfully run. Total query runtime: 59 msec. 1 rows affected. X

Query returned successfully in 31 msec. X

Total rows: 1 of 1 Query complete 00:00:00.059 In 1. Col 33

## 2. Процедура для ввода в базу данных новой книги

Реализация процедуры:

```
Properties  SQL  Untitled*  library/postgres@PostgreSQL 15*  libr
library/postgres@PostgreSQL 15
Query  Query History
1 CREATE OR REPLACE PROCEDURE "library".add_book (
2     IN book_id integer,
3     IN book_name character varying(30),
4     IN field_of_knowledge character varying(20),
5     IN original_language character varying(20),
6     IN book_type character varying(20),
7     IN book_author character varying(50))
8 LANGUAGE plpgsql
9 AS $$
10 BEGIN
11     INSERT INTO "library". "book_info" (book_id,
12         book_name,
13         field_of_knowledge,
14         original_language,
15         book_type,
16         book_author)
17     VALUES (book_id,
18         book_name,
19         field_of_knowledge,
20         original_language,
21         book_type,
22         book_author);
23 END;
24 $$;
Data Output  Messages  Notifications
CREATE PROCEDURE
Query returned successfully in 52 msec.
Total rows: 0 of 0  Query complete 00:00:00.052
```

Выполнение процедуры:

```
Query  Query History  Execute/Refresh (F5)
1 CALL "library".add_book(
2     8,
3     'Над пропастью во ржи',
4     'художественная литература',
5     'английский',
6     'роман',
7     'Сэлинджер Джером Д.')
```

Data Output Messages Notifications

CALL

Query returned successfully in 32 msec.

✓ Query returned successfully in 32 msec. X

Total rows: 0 of 0 Query complete 00:00:00.032 Ln 7, Col 24

Результат:

Data Output  Messages  Notifications						
	book_id [PK] integer	book_name character varying (40)	field_of_knowledge character varying (40)	original_language character varying (20)	book_author character varying (40)	book_type character varying (20)
1	1	Преступление и наказание	художественная литература	русский	Ф.М. Достоевский	роман
2	2	Колыбель для кошки	художественная литература	английский	Курт Воннегут	научная фантастика
3	3	Три товарища	художественная литература	русский	Эрих Мария Ремарк	роман
4	4	Физика 7	образовательная литература	русский	Перышкин	учебник
5	5	Степной Волк	художественная литература	английский	Герман Гессе	роман
6	6	Скотный двор	художественная литература	английский	Джордж Оруэлл	повесть
7	7	Программирование на C#	образовательная литература	русский	Казанцев Д. В	учебник
8	8	Над пропастью во ржи	художественная литература	английский	Сэлинджер Джером Д.	роман

✓ Successfully run. Total query runtime: 120 msec. 8 rows affected. X

mas > library > Tables > book\_info 00:00:00.120 Ln 1, Col 1

### 3. Процедура для ввода нового читателя (необходимо проверить наличие читателя в картотеке, чтобы не назначить ему номер вторично).

Реализация процедуры:

```
Query Query History
1 CREATE OR REPLACE PROCEDURE "library".check_reader_id(
2   IN full_name character varying(40),
3   IN passport character varying(11),
4   IN address character varying(40),
5   IN email character varying(40),
6   IN phone character varying(18),
7   IN education_level integer,
8   INOUT rid integer)
9   LANGUAGE 'plpgsql'
10  AS $$
11  BEGIN
12    SELECT reader_id INTO rid
13    FROM "library"."reader"
14    WHERE "reader".passport_details = passport LIMIT 1;
15    IF coalesce(rid, 0) = 0 THEN
16      SELECT MAX("reader".reader_id) INTO rid
17      FROM "library"."reader";
18      RAISE NOTICE 'rid %', rid;
19      INSERT INTO "library"."reader" (reader_id, full_name, passport_details, address, email, phone, education_level)
20      VALUES
21      (rid + 1,
22      full_name,
23      passport,
24      address,
25      email,
26      phone,
27      education_level);
28    END IF;
29  END;
30  $$;
```

Вызов процедуры:

Query Query History

```
1 CALL "library".check_reader_id(
2   'Аресний Кириллов',
3   '2448 133769',
4   'Москва, Полежаевская 11',
5   'asn@gmail.com',
6   '+7 (999) 111-47-59',
7   2,
8   1
9 )
```

Data Output Messages Notifications

	rid	integer
1		3

✓ Successfully run. Total query runtime: 36 msec. 1 rows affected. ✕

## Задание №2 - Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL.

Создание таблицы для хранения reader\_logger:

```
Query    Query History
1 CREATE TABLE IF NOT EXISTS library.reader_logger
2
3 operation character varying(50) COLLATE pg_catalog."default",
4 reader_id integer PRIMARY KEY GENERATED BY DEFAULT AS IDENTITY,
5 full_name character varying (50) COLLATE pg_catalog."default",
6 passport_details character varying (8) COLLATE pg_catalog."default",
7 address character varying(50) COLLATE pg_catalog."default",
8 email character varying(50) COLLATE pg_catalog."default",
9 phone bigint,
10 education_level integer,
11 data_log timestamp without time zone NOT NULL
12
Data Output Messages Notifications
ЗАМЕЧАНИЕ: отношение "reader_logger" уже существует, пропускается
CREATE TABLE
Query returned successfully in 29 msec.
```

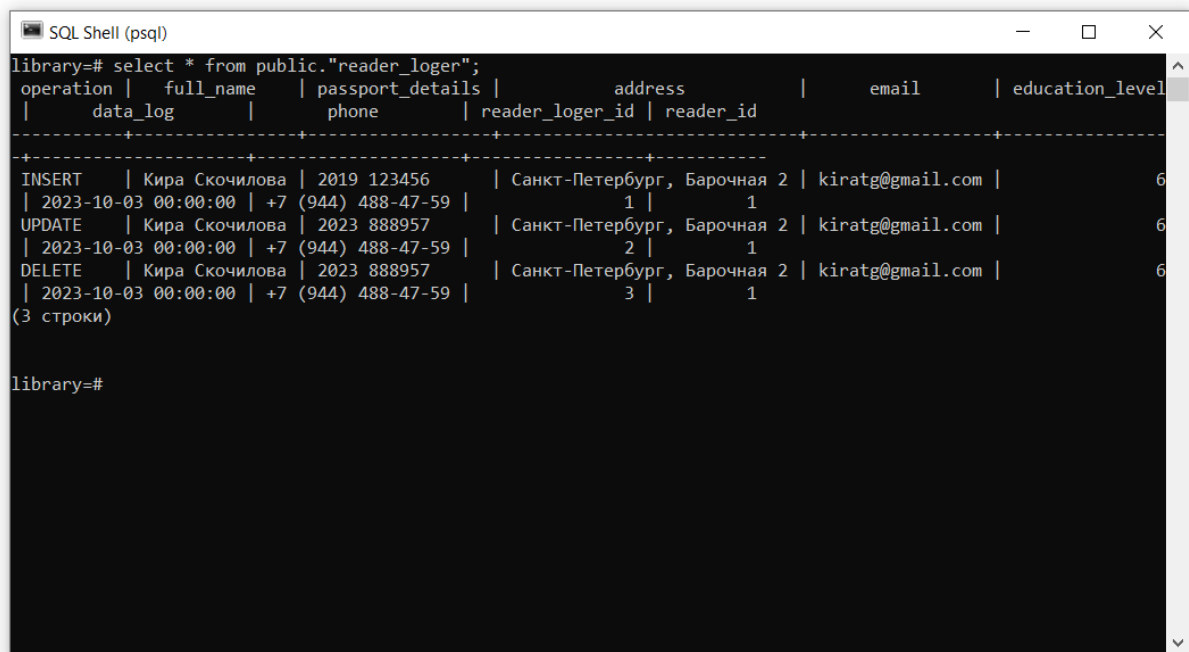
## Создание процедуры:

```
Query    Query History
1 CREATE OR REPLACE FUNCTION "library".log_reader()
2 RETURNS trigger
3 LANGUAGE 'plpgsql'
4 COST 100
5 VOLATILE NOT LEAKPROOF
6 AS $BODY$
7 DECLARE
8     action TEXT;
9 BEGIN
10 IF TG_OP = 'DELETE' THEN
11     action:='DELETE';
12     INSERT INTO public.reader_logger
13         (operation, reader_id, full_name, passport_details, address, email, phone, education_level, data_log)
14         VALUES (action, OLD. "reader_id", OLD. "full_name", OLD. "passport_details", OLD. "address", OLD. "email", OLD. "p
15             NOW());
16     RETURN OLD;
17 ELSE
18 IF TG_OP = 'INSERT' THEN
19     action = 'INSERT';
20 ELSEIF TG_OP= 'UPDATE' THEN
21     action='UPDATE';
22 END IF;
23 INSERT INTO library.reader_logger
24     (operation, reader_id, full_name, passport_details, address, email, phone, education_level, data_log)
25     VALUES (action, NEW."reader_id", NEW."full_name", NEW."passport_details", NEW."address", NEW."email",
26         NEW."phone", NEW."education_level",
27         NOW());
28 RETURN NEW;
29 END IF;
30 END;
31
Total rows: 0 of 0    Query complete 00:00:00.034    Ln 26, Col 18
```

## Реализация триггера:

```
Properties SQL Untitled* library/postgre... library/postgre... library/postgre... library/postgre... library/postgres@PostgreSQL 15*
library/postgres@PostgreSQL 15
Query    Query History
1 CREATE OR REPLACE TRIGGER add_r
2 AFTER INSERT OR UPDATE OR DELETE ON library.reader_logger
3 FOR EACH ROW
4 EXECUTE FUNCTION "library".log_reader();
Execute/Refresh (F5)
Query returned successfully in 31 msec.
Total rows: 0 of 0    Query complete 00:00:00.031    Ln 4, Col 49
```

Пример работы триггера:



```
library=# select * from public."reader_logger";
operation | full_name | passport_details | address | email | education_level
-----+-----+-----+-----+-----+-----
| data_log | phone | reader_logger_id | reader_id | kiratg@gmail.com | 6
-----+-----+-----+-----+-----+-----
INSERT | Кира Скочилова | 2019 123456 | Санкт-Петербург, Барочная 2 | kiratg@gmail.com | 6
| 2023-10-03 00:00:00 | +7 (944) 488-47-59 | 1 | 1 | kiratg@gmail.com | 6
UPDATE | Кира Скочилова | 2023 888957 | Санкт-Петербург, Барочная 2 | kiratg@gmail.com | 6
| 2023-10-03 00:00:00 | +7 (944) 488-47-59 | 2 | 1 | kiratg@gmail.com | 6
DELETE | Кира Скочилова | 2023 888957 | Санкт-Петербург, Барочная 2 | kiratg@gmail.com | 6
| 2023-10-03 00:00:00 | +7 (944) 488-47-59 | 3 | 1 | kiratg@gmail.com | 6
(3 строки)

library=#
```

### Выводы:

В ходе выполнения лабораторной работы были разработаны хранимые процедуры и триггеры для PostgreSQL в соответствии с индивидуальным заданием, представленным в части 4. Мы успешно выполнили все этапы, указанные в практическом задании. Хранимые процедуры являются многофункциональными и значительно способствуют автоматизации и ускорению процессов.