

Министерство науки и высшего образования Российской
Федерации

федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №3

«ПРОЦЕДУРЫ, ФУНКЦИИ, ТРИГГЕРЫ В PostgreSQL»

Выполнил:

Чухонин Иван Андреевич

Группа К32421

Преподаватель:

Говорова Мария Михайловна

Санкт-Петербург
2023

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

Практическое задание:

2. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).
- 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.

Индивидуальное задание - Вариант 3. БД «Библиотека»

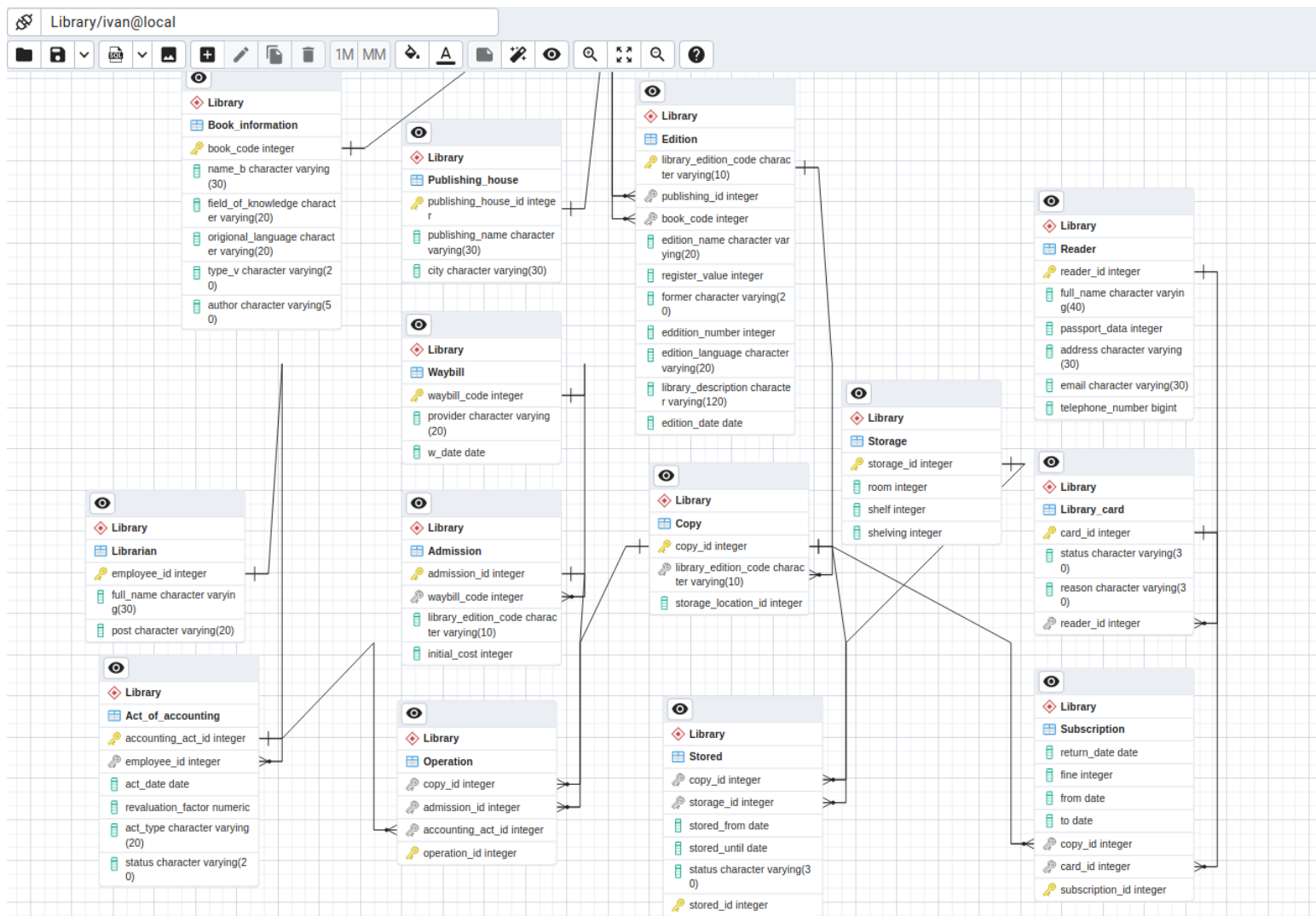


рисунок №1 - схема библиотеки

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:

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

	ok_code (integer)	name_b character varying (30)	field_of_knowledge character varying (20)	original_language character varying (20)	type_v character varying (20)	author character varying (50)	book_code integer	total bigint	book_code integer	given bigint
1	3	Физика	образовательная	русский	учебник	д. Джанколи	3	1	3	1
2	5	Преступление и наказание	худ. литература	русский	роман	Ф.М. Достоевский	5	1	5	1
3	4	Мартин Иден	худ. литература	английский	роман	Джек Лондон	4	1	4	1
4	6	Бесы	худ. литература	русский	роман	Ф.М. Достоевский	6	1	6	1
5	2	Заводной апельсин	худ. литература	русский	антиутопия	Энтони Берджесс	2	1	2	1
6	1	Игрок	худ. литература	русский	роман	Ф.М. Достоевский	1	1	1	1

рисунок №2 - таблица с экземплярами, которые находятся в библиотеке

Query Query History

```
1 CREATE OR REPLACE PROCEDURE "Library".count_copi(IN BOOK_COD integer, INOUT copy_count bigint) LANGUAGE PLPGSQL AS $$
2 BEGIN
3     SELECT total INTO copy_count
4     FROM "Library"."Book_information"
5 INNER JOIN
6     (SELECT "Book_information".book_code,
7      count(copy_id) AS total
8     FROM "Library"."Book_information"
9 INNER JOIN "Library"."Edition"
10    ON "Book_information".book_code="Edition".book_code
11 INNER JOIN "Library"."Copy"
12    ON "Edition".library_edition_code="Copy".library_edition_code
13 GROUP BY "Book_information".book_code ) ALL_BOOKS
14 ON ALL_BOOKS.book_code="Book_information".book_code
15 LEFT JOIN
16     (SELECT "Book_information".book_code,
17      COUNT(subscription_id) AS given
18     FROM "Library"."Book_information"
19 INNER JOIN "Library"."Edition"
20    ON "Book_information".book_code="Edition".book_code
21 INNER JOIN "Library"."Copy"
22    ON "Edition".library_edition_code="Copy".library_edition_code
23 INNER JOIN "Library"."Subscription"
24    ON "Copy".copy_id="Subscription".copy_id
25 WHERE "Subscription".return_date IS NOT NULL
26 GROUP BY "Book_information".book_code) GIVEN_BOOKS
27 ON ALL_BOOKS.book_code=GIVEN_BOOKS.book_code
28 WHERE total=coalesce(given, 0) and "Book_information".book_code = BOOK_COD LIMIT 1;
29 END;
30 $$;
```

Messages Notifications

CREATE PROCEDURE
Query returned successfully in 177 msec.

Data Output

No data output. Execute a query to get output.

рисунок №3 - реализация процедуры

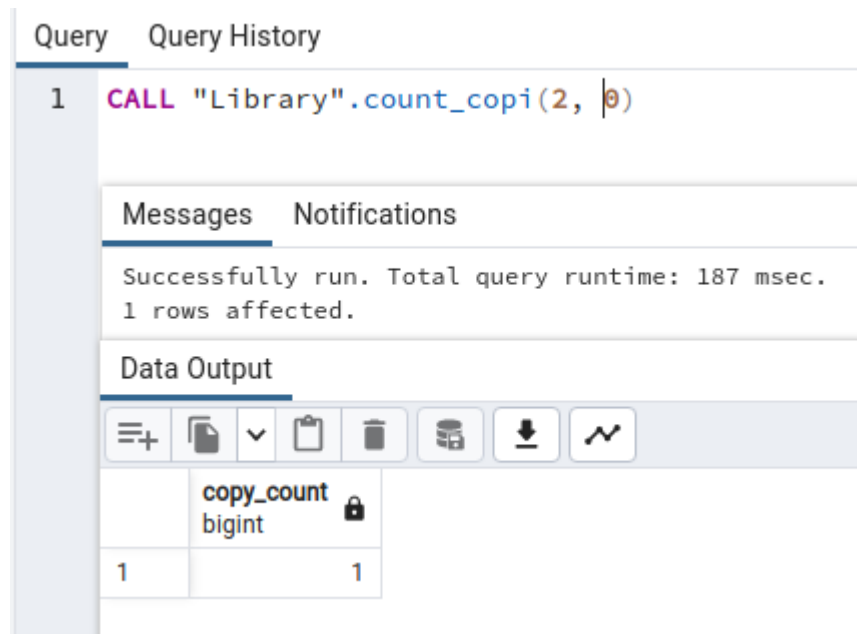


рисунок №4 - вызов процедуры

Процедура получает на вход id книги и её суть заключается в проверке на количество экземпляров заданной по id книги

1.2 Для ввода в базу данных новой книги.

	book_code [PK] integer	name_b character varying (30)	field_of_knowledge character varying (20)	original_language character varying (20)	type_v character varying (20)	author character varying (50)
1	1	Игрок	худ. литература	русский	роман	Ф.М. Достоевский
2	2	Заводной апельсин	худ. литература	русский	антиутопия	Энтони Бёрджесс
3	3	Физика	образовательная	русский	учебник	д. Джанколи
4	4	Мартин Иден	худ. литература	английский	роман	Джек Лондон
5	5	Преступление и наказание	худ. литература	русский	роман	Ф.М. Достоевский
6	6	Бесы	худ. литература	русский	роман	Ф.М. Достоевский
7	7	Превращение	худ. литература	английский	повесть	Ф. Кафка
8	8	Три товарища	худ. литература	английский	роман	Э.М. Ремарк

рисунок №5 - таблица содержащая книги

Query	Query History
1	CREATE OR REPLACE PROCEDURE "Library".add_book(2 IN book_code integer, 3 IN name_b character varying(30), 4 IN field_of_knowledge character varying(20), 5 IN origional_language character varying(20), 6 IN type_v character varying(20), 7 IN author character varying(50)) 8 LANGUAGE plpgsql 9 AS \$\$ 10 BEGIN 11 INSERT INTO "Library"."Book_information"(book_code, 12 name_b, 13 field_of_knowledge, 14 origional_language, 15 type_v, 16 author) 17 values (book_code, 18 name_b, 19 field_of_knowledge, 20 origional_language, 21 type_v, 22 author); 23 END ; 24 \$\$;

Data Output	Messages	Notifications
CREATE PROCEDURE		
Query returned successfully in 125 msec.		

рисунок №6 - реализация процедуры

Query	Query History
1	CALL "Library".add_book(2 9, 3 'Фиеста', 4 'худ. литература', 5 'английский', 6 'роман', 7 'Э. Хемингуэй') 8 9

Data Output	Messages	Notifications
CALL		
Query returned successfully in 143 msec.		

рисунок №7 - вызов процедуры

	book_code [PK] integer	name_b character varying (30)	field_of_knowledge character varying (20)	original_language character varying (20)	type_v character varying (20)	author character varying (50)
1	1	Игрок	худ. литература	русский	роман	Ф.М. Достоевский
2	2	Заводной апельсин	худ. литература	русский	антиутопия	Энтони Бёрджесс
3	3	Физика	образовательная	русский	учебник	д. Джанколи
4	4	Мартин Иден	худ. литература	английский	роман	Джек Лондон
5	5	Преступление и наказание	худ. литература	русский	роман	Ф.М. Достоевский
6	6	Бесы	худ. литература	русский	роман	Ф.М. Достоевский
7	7	Превращение	худ. литература	английский	повесть	Ф. Кафка
8	8	Три товарища	худ. литература	английский	роман	Э.М. Ремарк
9	9	Фиеста	худ. литература	английский	роман	Э. Хемингуэй

рисунок №8 - добавлена новая книга

Процедура помогает реализовать универсальный метод вставки новых произведений в базу данных

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

Data Output Messages Notifications							
	reader_id [PK] integer	full_name character varying (40)	passport_data integer	address character varying (30)	email character varying (30)	telephone_number bigint	knowledge_lvl integer
1	1	Борисов Юрий Александрович	78787878	набережная Фонтанки 28	ry@gmail.com	89112345678	2
2	2	Кирилл	50560633	улица Яхтенная 16	banshi@gmail.com	89117456758	1
3	3	Ира	30347455	улица Туристская	lobster@gmail.com	89113563435	2
4	4	Женя	20230322	улица Яхтенная 15	whale@gmail.com	89112442323	2
5	5	Максим	78343445	ул. Серебристая 57	tree@gmail.com	89112564830	1
6	6	Алла Владимировна	78343876	пер. Горохов 17	realzone@gmail.com	89117744830	3
7	7	Лодыгин Юра	78325645	ул. Спортивная 97	fatcatlife@gmail.com	89112534450	2
8	8	Армасова Аня	78378045	пер. Книжный 44	plainwear@gmail.com	89112564007	3
9	9	Кузьяев Семён	78341195	ул. Дачная 55	criptosqad@gmail.com	89112506830	1

рисунок №9 - база читателей

```

4
5 CREATE OR REPLACE PROCEDURE "Library".check_reader_id(
6     IN full_name character varying(40),
7     IN passport integer,
8     IN address character varying(30),
9     IN email character varying(30),
10    IN telephone_number bigint,
11    IN knowledge_lvl integer,
12    INOUT rid integer)
13 LANGUAGE 'plpgsql'
14 AS $BODY$
15 BEGIN
16     SELECT reader_id INTO rid
17     FROM "Library"."Reader"
18     WHERE "Reader".passport_data = passport LIMIT 1;
19 IF coalesce(rid,0) = 0 THEN
20     SELECT MAX("Reader".reader_id) INTO rid
21     FROM "Library"."Reader";
22     RAISE NOTICE 'rid %', rid;
23     INSERT INTO "Library"."Reader"(reader_id, full_name, passport_data, address, email, telephone_number, knowledge_lvl)
24         VALUES
25             (rid + 1,
26              full_name,
27              passport,
28              address,
29              email,
30              telephone_number,
31              knowledge_lvl);
32     END IF;
33 END;
34 $BODY$;
35 ALTER PROCEDURE "Library".check_reader_id(integer, integer)
36     OWNER TO ivan;
37

```

рисунк №10 - код составления процедуры

Query	Query History
<pre> 1 CALL "Library".check_reader_id(2 'Елена Зеленцова', 3 59960633, 4 'ул. Глинки', 5 'wear@gmail.com', 6 89213749156, 7 2, 8 0 9) </pre>	

Messages	Notifications	Data Output
NOTICE: rid 9		
Successfully run. Total query runtime: 141 msec.		
1 rows affected.		

Data Output		
	rid	
	integer	
1		9

рисунк №11 - отображение результата

Данная процедура получает на вход паспортные данные, проверяет нет ли такого читателя в базе данных и на выход даёт его id или если такого нет, то null и вставляет данные в таблицу Reader.

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

```
5 CREATE TABLE IF NOT EXISTS public.reader_logger
6 (
7     operation character varying(50) COLLATE pg_catalog."default",
8     reader_id integer NOT NULL DEFAULT nextval('reader_logger_r_id_seq'::regclass),
9     full_name character varying(50) COLLATE pg_catalog."default",
10    passport_data character varying(8) COLLATE pg_catalog."default",
11    address character varying(50) COLLATE pg_catalog."default",
12    email character varying(50) COLLATE pg_catalog."default",
13    telephone_number bigint,
14    knowledge_lvl integer,
15    data_log timestamp without time zone NOT NULL
16 )
17
18 TABLESPACE pg_default;
19
20 ALTER TABLE IF EXISTS public.reader_logger
21     OWNER to ivan;
```

рисунок №12 - создаем таблицу для хранения данных reader_logger

Query	Query History
5	CREATE OR REPLACE FUNCTION "Library".log_reader()
6	RETURNS trigger
7	LANGUAGE 'plpgsql'
8	COST 100
9	VOLATILE NOT LEAKPROOF
10	AS \$BODY\$
11	DECLARE
12	action TEXT;
13	BEGIN
14	IF TG_OP = 'DELETE' THEN
15	action := 'DELETE';
16	INSERT INTO public.reader_logger
17	(operation,reader_id,full_name,passport_data,address,email,telephone_number,knowledge_lvl,data_log)
18	VALUES (action, OLD."reader_id", OLD."full_name", OLD."passport_data", OLD."address", OLD."email", OLD."telephone_number",
19	OLD."knowledge_lvl",
20	NOW());
21	RETURN OLD;
22	ELSE
23	IF TG_OP = 'INSERT' THEN
24	action := 'INSERT';
25	ELSIF TG_OP = 'UPDATE' THEN
26	action := 'UPDATE';
27	END IF;
28	INSERT INTO public.reader_logger
29	(operation,reader_id,full_name,passport_data,address,email,telephone_number,knowledge_lvl,data_log)
30	VALUES (action, NEW."reader_id", NEW."full_name", NEW."passport_data", NEW."address", NEW."email",
31	NEW."telephone_number",
32	NEW."knowledge_lvl",
33	NOW());
34	RETURN NEW;
35	END IF;
36	END;
37	\$BODY\$;

рисунок №13 - функция

В функции определяется выполняемая операция (UPDATE, INSERT, DELETE) и записываются данные в таблицу reader_logger



рисунок №15 - реализация триггера

Триггер будет вызываться после каждого события на вставку, обновление и удаление данных и записывать данные в таблицу reader_logger

The screenshot shows a database query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL script to select all data from the reader_logger table:

```

1 SELECT * FROM public.reader_logger
2

```

Below the query editor, there is a 'Messages' tab and a 'Data Output' tab. The 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: operation, reader_id, full_name, passport_data, address, email, and telephone_number. The table contains 4 rows of data.

	operation character varying (50)	reader_id integer	full_name character varying (50)	passport_data character varying (8)	address character varying (50)	email character varying (50)	telephone_number bigint
1	INSERT	10	Елена Зеленцова	59960633	ул. Глинки	wear@gmail.com	89213749156
2	UPDATE	10	Елена Зеленцова	59960633	ул. Глинки	wear@gmail.com	89213749156
3	DELETE	10	Елена Зеленцова	59960633	ул. Глинки	wear@gmail.com	89213749156
4	UPDATE	9	Кузьяев Семён	78341195	ул. Дачная 55	criptosqad@gmail.com	89112506830

рисунок №16 - данные таблицы регистрации

Вывод: в ходе выполнения лабораторной работы были созданы хранимые процедуры и триггеры к PostgreSQL согласно индивидуальному заданию часть 4. Выполнены все шаги указанные в практическом задании. Хранимые процедуры полезны своей многофункциональностью и в значительной степени помогают автоматизировать и ускорить процесс.