

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»

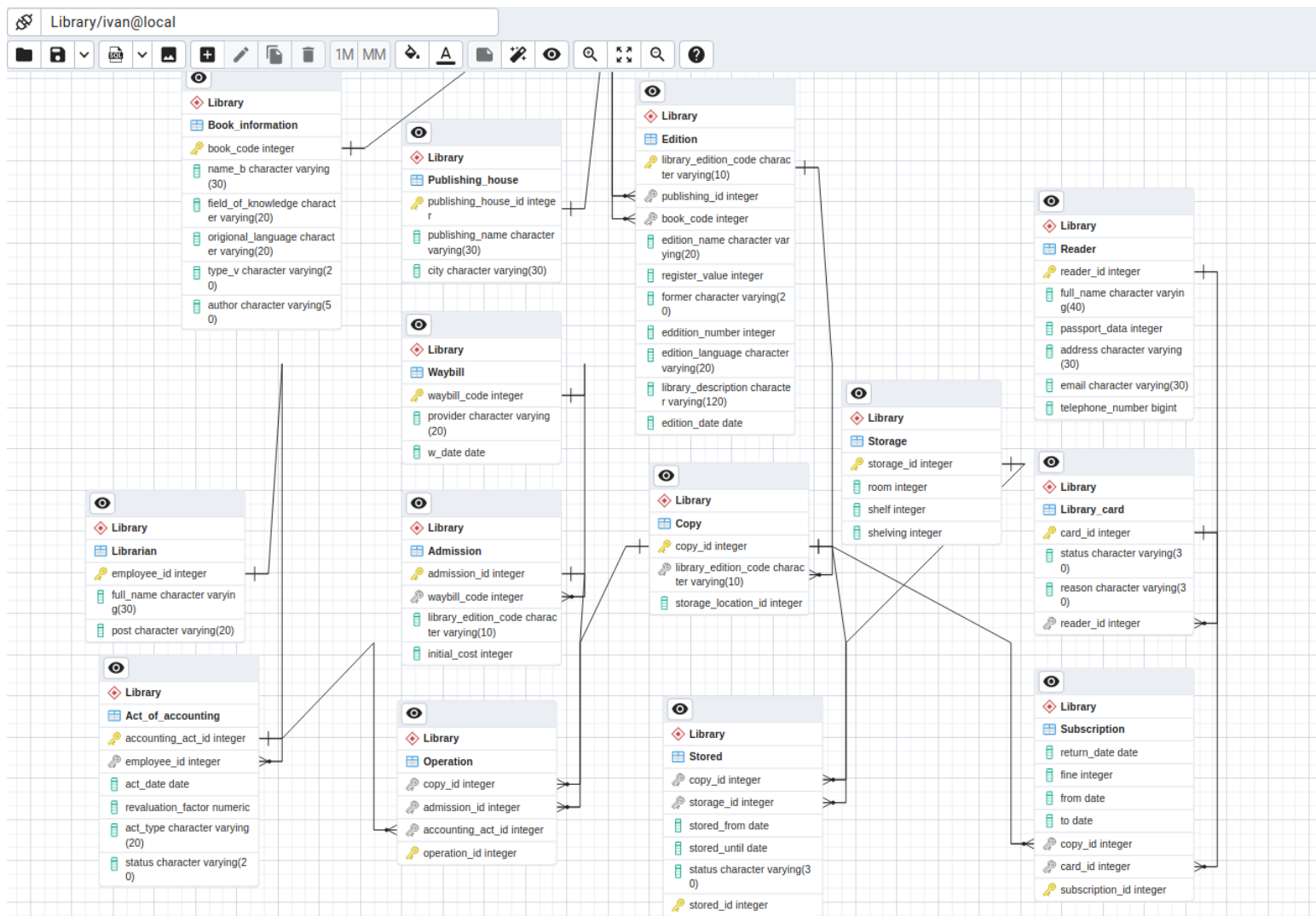


рисунок №1 - схема библиотеки

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

рисунки №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.

рисунки №3 - реализация процедуры

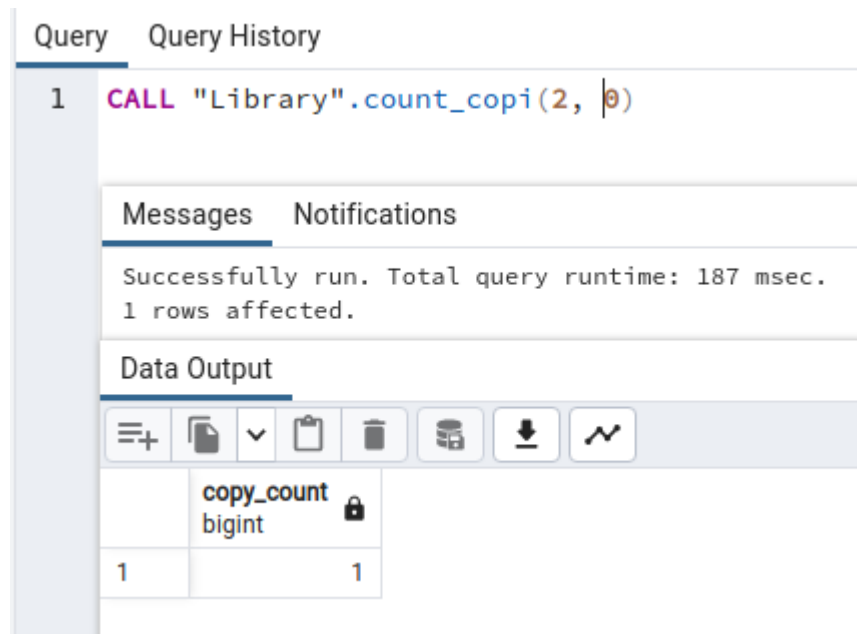


рисунок №4 - вызов процедуры

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

The screenshot shows a database IDE interface with a toolbar at the top containing icons for file operations, filters, and execution. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL script for creating a procedure. The script is as follows:

```
1 CREATE OR REPLACE PROCEDURE "Library".check_reader_id(  
2     IN passport integer,  
3     INOUT rid integer)  
4 LANGUAGE plpgsql  
5 AS $$  
6 BEGIN  
7     SELECT reader_id INTO rid  
8     FROM "Library"."Reader"  
9     WHERE passport_data = passport LIMIT 1;  
10 END;  
11 $$;  
12
```

Below the script, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the following message:

```
CREATE PROCEDURE  
  
Query returned successfully in 211 msec.
```

рисунок №10 - код составления процедуры

The screenshot shows the same database IDE interface. The 'Query' tab is active, displaying a SQL script for calling the procedure. The script is as follows:

```
1 CALL "Library".check_reader_id(  
2     50560633,  
3     0  
4 )  
5  
6
```

Below the script, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following data:

	rid	integer
1		2

рисунок №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.

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

Query	Query History
1	CREATE TABLE reader_log(2 r_id SERIAL PRIMARY KEY , 3 full_name VARCHAR(50) , 4 passport_data VARCHAR(8) , 5 address VARCHAR(50) , 6 email VARCHAR(50) , 7 telephone_number bigint , 8 knowledge_lvl integer) 9
Messages	Notifications
CREATE TABLE	
Query returned successfully in 203 msec.	

рисунок №12 - создаем таблицу для хранения данных

Query	Query History
1	CREATE OR REPLACE FUNCTION "Library".log_reader()
2	RETURNS TRIGGER
3	LANGUAGE PLPGSQL AS \$\$
4	DECLARE
5	action TEXT;
6	BEGIN
7	IF operation = 'INSERT' THEN
8	action := 'INSERT';
9	ELSIF operation = 'UPDATE' THEN
10	action := 'UPDATE';
11	ELSIF operation = 'DELETE' THEN
12	action := 'DELETE';
13	END IF;
14	INSERT INTO public.reader_logger VALUES (
15	action,
16	NEW."reader_id",
17	NEW."full_name",
18	NEW."passport_data",
19	NEW."address",
20	NEW."email",
21	NEW."telephone_number",
22	NEW."knowledge_lvl",
23	NOW());
24	RETURN NEW;
25	END;
26	\$\$;
27	

Messages	Notifications
CREATE FUNCTION	
Query returned successfully in 158 msec.	

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

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.