

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №5

процедуры, функции, триггеры в PostgreSQL
по дисциплине:
«Проектирование и реализация баз данных»

Выполнил студент:

Алексеев Павел Алексеевич

Группа № К3243

Проверила:

Говорова Марина Михайловна

Санкт-Петербург
2022

ЦЕЛЬ РАБОТЫ:

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

Оборудование: компьютерный класс.

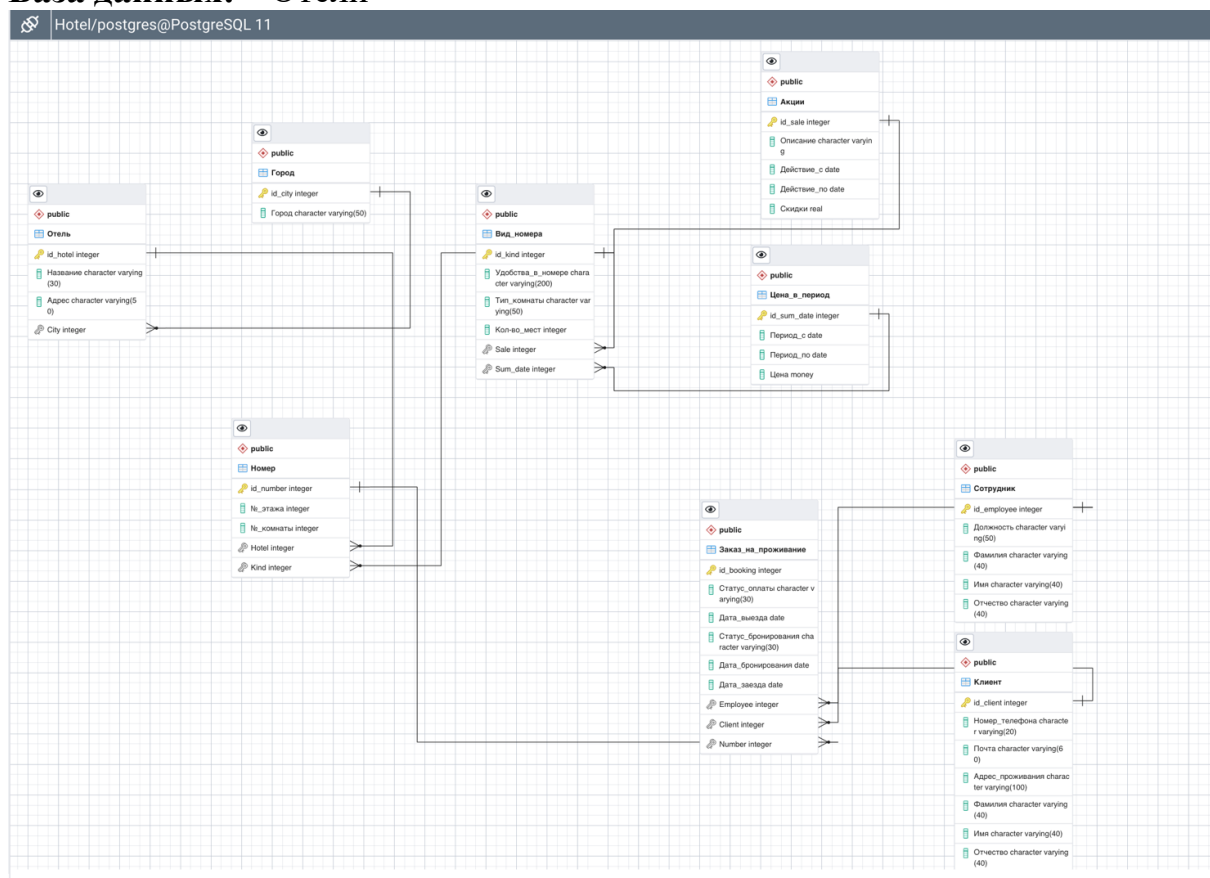
Программное обеспечение: СУБД PostgreSQL, SQL Shell (psql).

Вариант 1

Практическое задание:

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

База данных: “Отели”



Выполнение:

Задание 1. Создайте хранимые процедуры.

1) Для увеличения цены всех номеров на 5 %

Query EditorИстория запросов

```
1 create or replace function edit_price_5_proc() RETURNS integer
2 AS
3 $$
4 begin
5 UPDATE "Цена_в_период" SET "Цена" = "Цена" * 1.05;
6 RAISE NOTICE 'PRICE UPDATE!';
7 return 1;|
8 end;
9 $$ LANGUAGE plpgsql VOLATILE;
```

РезультатПлан выполненияСообщенияNotifications

CREATE FUNCTION

Запрос завершён успешно, время выполнения: 27 msec.

РезультатПлан выполненияСообщенияNotifications

	id_sum_date [PK] integer	Период_c date	Период_по date	Цена money	number_kind integer
1	17	2022-05-01	2022-09-30	\$630.00	4
2	18	2022-05-01	2022-09-30	\$420.00	5
3	19	2022-05-01	2022-09-30	\$2,152.50	6
4	20	2022-05-01	2022-09-30	\$1,260.00	7
5	21	2022-05-01	2022-09-30	\$367.50	8
6	22	2022-05-01	2022-09-30	\$2,625.00	9
7	23	2022-05-01	2022-09-30	\$1,050.00	10
8	16	2022-05-01	2022-09-30	\$882.00	3

РезультатПлан выполненияСообщенияNotifications

	id_sum_date [PK] integer	Период_c date	Период_по date	Цена money	number_kind integer
1	17	2022-05-01	2022-09-30	\$661.50	4
2	18	2022-05-01	2022-09-30	\$441.00	5
3	19	2022-05-01	2022-09-30	\$2,260.12	6
4	20	2022-05-01	2022-09-30	\$1,323.00	7
5	21	2022-05-01	2022-09-30	\$385.88	8
6	22	2022-05-01	2022-09-30	\$2,756.25	9
7	23	2022-05-01	2022-09-30	\$1,102.50	10
8	16	2022-05-01	2022-09-30	\$926.10	3

2) Для получения информации о свободных номерах отеля на завтрашний день;

Query Editor

История запросов

```
1 CREATE OR REPLACE function count_of_free_rooms(id integer)
2     returns table
3     (
4         total bigint
5     )
6     language plpgsql
7 as
8 $$
9 begin
10 return query (
11 SELECT COUNT(*) FROM "Номер"
12 LEFT JOIN "Заказ_на_проживание" ON "Номер"."id_number" = "Заказ_на_проживание"."Number"
13 WHERE "Номер"."Hotel" = id AND ("Заказ_на_проживание"."Дата_заезда" IS NULL
14 OR current_date + interval '1 day' NOT BETWEEN "Заказ_на_проживание"."Дата_заезда" AND "Заказ_на_проживание"."Дата_выезда"));
15 end;
16 $$;
```

Query Editor

История запросов

```
1 SELECT count_of_free_rooms(3);
2
```

Результат

План выполнения

Сообщения

Notifications

	count_of_free_rooms	
	bigint	
1	16	

3) Бронирования в гостинице на заданную дату и количество дней проживания;

Query Editor

История запросов

```
1 CREATE OR REPLACE function add_new_booking(INOUT "Статус_оплаты" VARCHAR(50), INOUT "Дата_выезда" DATE,
2     INOUT "Дата_заезда" DATE, INOUT "Employee" INTEGER, INOUT "Client" INTEGER, INOUT "Number" INTEGER)
3 as
4 $$
5 begin
6     INSERT INTO "Заказ_на_проживание"("Статус_оплаты", "Дата_выезда", "Статус_бронирования", "Дата_бронирования", "Дата_заезда", "Employee", "Client", "Number")
7     VALUES("Статус_оплаты", "Дата_выезда", 'Забронировано', current_date, "Дата_заезда", "Employee", "Client", "Number");
8     RAISE NOTICE 'DONE!';
9 end;
10 $$ language plpgsql VOLATILE;
```

Query Editor

История запросов

```
1 SELECT add_new_booking('Оплачено', '2022-05-10', '2022-05-05', 5, 1, 8);
```

Результат

План выполнения

Сообщения

Notifications

	add_new_booking	
	record	
1	(Оплачено,2022-05-10,2022-05-05,5,1,8)	

	id_booking [PK] integer	Статус_оплаты character varying (30)	Дата_выезда date	Статус_бронирования character varying (30)	Дата_бронирования date	Дата_заезда date	Employee integer	Client integer	Number integer	
1		10 Оплачено	2022-03-23	Выехал	2022-03-14	2022-03-16		4	2	9
2		15 Оплачено	2022-04-06	Выехал	2022-03-21	2022-03-27		5	6	17
3		17 Оплачено	2022-03-27	Выехал	2022-03-11	2022-03-20		5	7	18
4		18 Оплачено	2022-03-27	Выехал	2022-03-13	2022-03-20		6	7	23
5		19 Оплачено	2022-03-25	Выехал	2022-03-12	2022-03-19		6	8	24
6		20 Оплачено	2022-03-26	Выехал	2022-03-15	2022-03-17		6	9	27
7		21 Оплачено	2022-03-28	Выехал	2022-03-10	2022-03-20		7	10	30
8		13 Оплачено	2022-04-05	Забронировано	2022-03-21	2022-03-27		5	4	13
9		14 Оплачено	2022-04-06	Забронировано	2022-03-21	2022-03-27		5	5	15
10		23 Оплачено	2022-04-05	Забронировано	2022-03-20	2022-04-03		7	10	30
11		25 Оплачено	2022-04-03	Забронировано	2022-03-25	2022-03-25		7	1	32
12		9 Оплачено	2022-03-20	Забронировано	2022-03-10	2022-03-13		4	1	7
13		24 Оплачено	2022-04-03	Забронировано	2022-03-27	2022-04-01		7	1	31
14		27 Оплачено	2022-03-28	Забронировано	2022-03-21	2022-03-24		5	2	13
15		34 Оплачено	2022-05-10	Забронировано	2022-05-04	2022-05-05		5	1	7
16		35 Оплачено	2022-05-10	Забронировано	2022-05-01	2022-05-05		5	1	8

Задание 2. Создать триггер для логирования событий вставки, удаления, редактирования

```

Query Editor  История запросов
1  CREATE OR REPLACE FUNCTION add_to_log() RETURNS TRIGGER AS $$
2  DECLARE
3      mstr varchar(30);
4      astr varchar(100);
5      retstr varchar(254);
6  BEGIN
7      IF TG_OP = 'INSERT' THEN
8          astr = NEW;
9          mstr := 'Add new data';
10         retstr := mstr || astr;
11         INSERT INTO logs(text,added,table_name) values (retstr,NOW(), TG_TABLE_NAME);
12         RETURN NEW;
13     ELSEIF TG_OP = 'UPDATE' THEN
14         astr = NEW;
15         mstr := 'Update data';
16         retstr := mstr || astr;
17         INSERT INTO logs(text,added,table_name) values (retstr,NOW(), TG_TABLE_NAME);
18         RETURN NEW;
19     ELSEIF TG_OP = 'DELETE' THEN
20         astr = OLD;
21         mstr := 'Remove data ';
22         retstr := mstr || astr;
23         INSERT INTO logs(text,added,table_name) values (retstr,NOW(), TG_TABLE_NAME);
24         RETURN OLD;
25     END IF;
26 END;
27 $$ LANGUAGE plpgsql;
28

```

Триггер был выполнен для таблицы employee.

1	SELECT * FROM logs
---	--------------------

text character varying (200)	added date	table_name character varying (200)
1 Add new data(12,Менеджер,Петров,Дмитрий,Васильевич)	2022-05-01	Сотрудник
2 Update data(12,Портье,Петров,Дмитрий,Васильевич)	2022-05-01	Сотрудник
3 Remove data (12,Портье,Петров,Дмитрий,Васильевич)	2022-05-01	Сотрудник

Вывод:

В данной работе мною были изучены функции и процедуры, а также триггеры для корректного хранения данных.