

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

Отчет

по лабораторной работе «Запросы на выборку и модификацию данных,
представления и индексы в PostgreSQL»
по дисциплине «**Базы данных**»

Автор: Плеханова Дарья Денисовна

Факультет: ИКТ

Группа: К32422

Преподаватель: Говорова М. М.

Дата: 25.05.2023

ИТМО

Санкт-Петербург 2023

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

Практическое задание:

Вариант 1:

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.

Выполнение работы:

Предметная область – Отель (вариант 1)

Наименование БД – Hotel

Создание процедур:

- для увеличения цены всех номеров на 5 %, если в отеле нет свободных номеров;

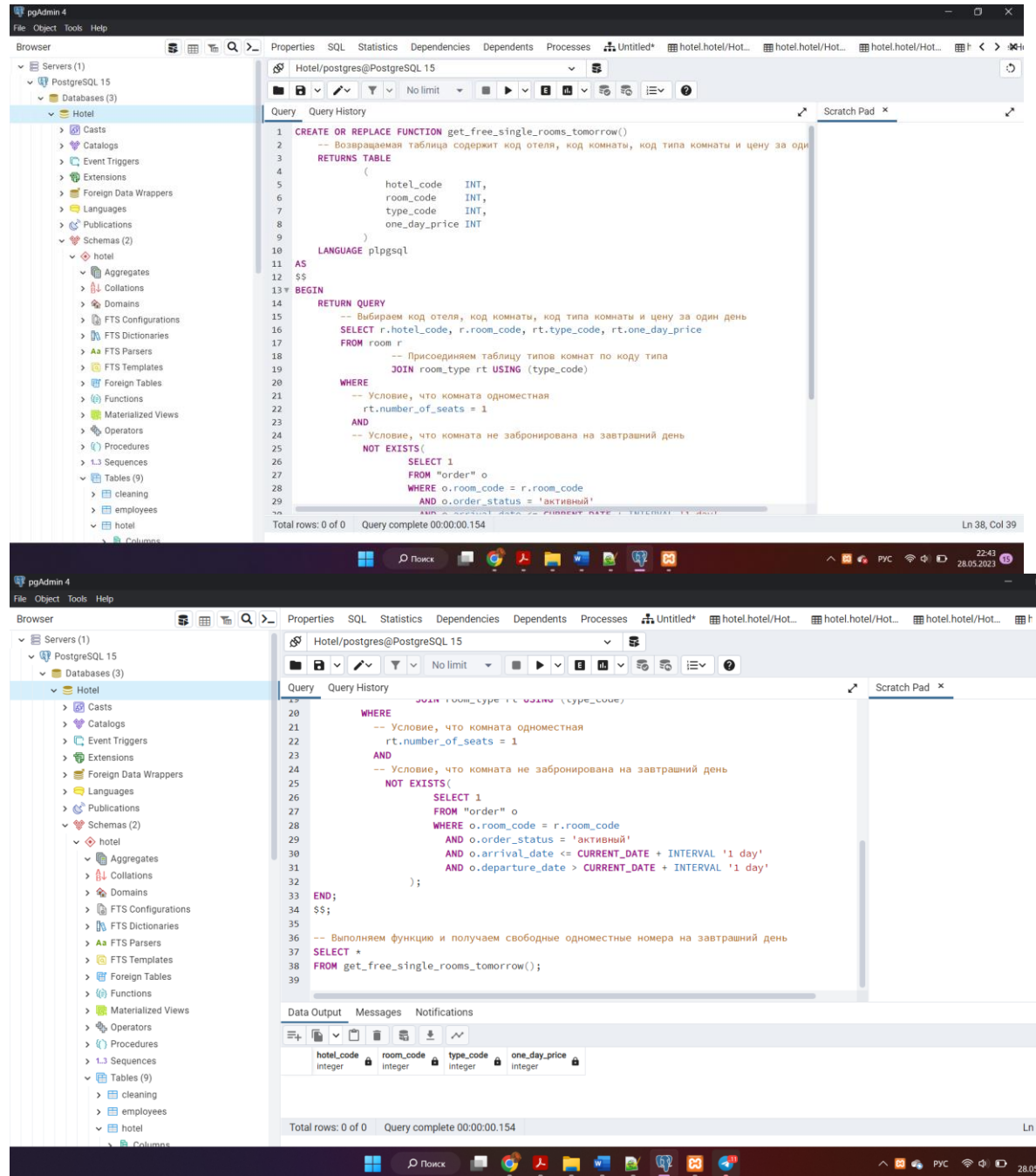
-

```
1 CREATE OR REPLACE PROCEDURE raise_filled_hotel_rooms_price()
2 LANGUAGE plpgsql
3 AS
4 $$
5 BEGIN
6     -- Увеличиваем цену на 5% для типов номеров, которые находятся в отелях без свободных номеров
7     UPDATE room_type
8     SET one_day_price = one_day_price * 1.05
9     WHERE type_code IN (
10         -- Выбираем типы номеров из тех отелей, где все номера заняты
11         SELECT type_code
12         FROM room
13         WHERE hotel_code IN (
14             -- Выбираем отели, где все номера заняты
15             SELECT hotel_code
16             FROM hotel h
17             JOIN room USING (hotel_code)
18             GROUP BY hotel_code
19             HAVING COUNT(CASE WHEN room_occupancy = 'Занято' THEN 1 END) = COUNT(room_code));
20     END;
21 $$;
22 CALL raise_filled_hotel_rooms_price();
```

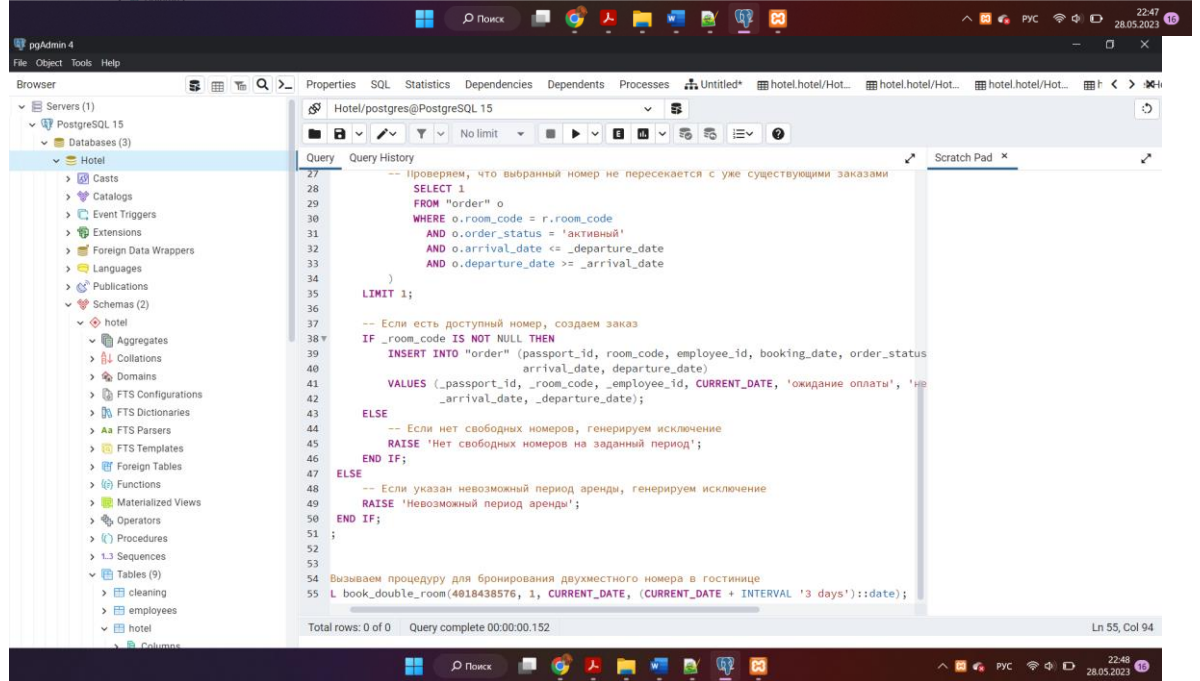
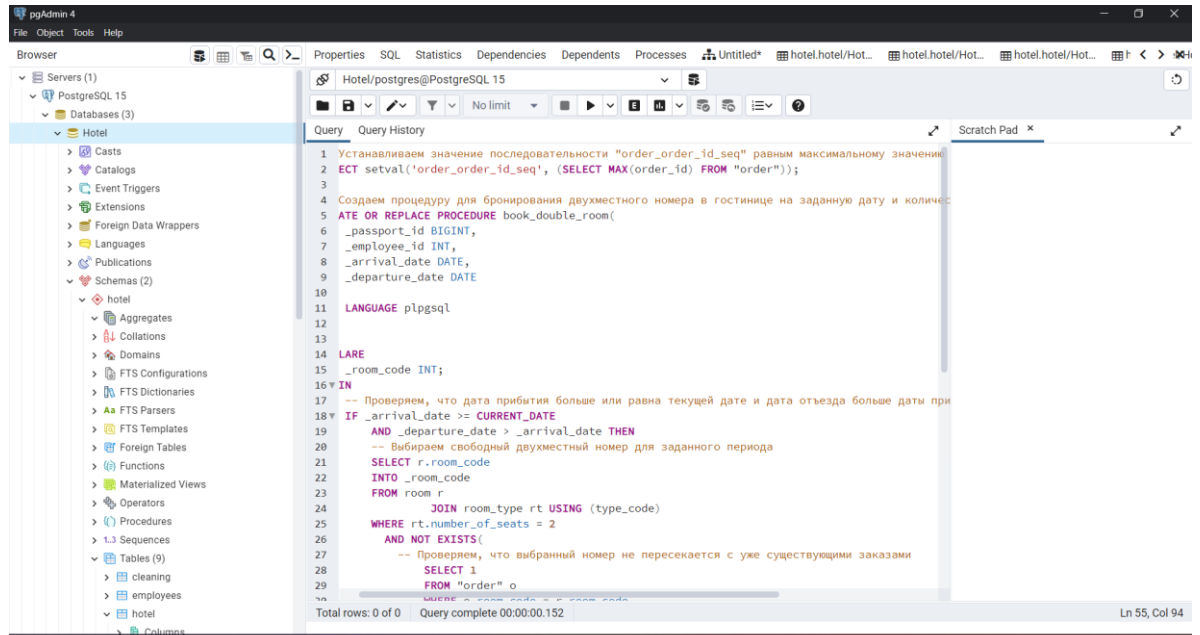
-

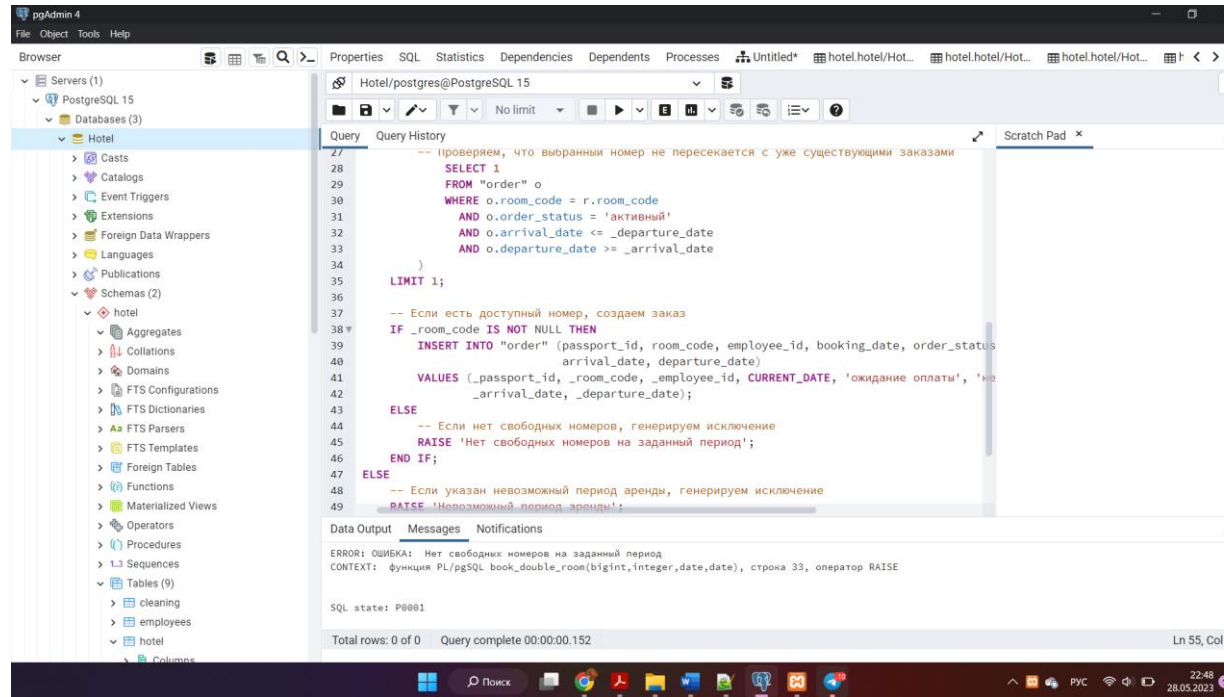
Data Output					
	type_code [PK] integer	space_type character varying (30)	comfort character varying (100)	number_of_seats integer	one_day_price integer
1	1	ванна	ванная	4	2500
2	2	спальня	телевизор	4	2700
3	3	детская	телевизор	4	2900

- для получения информации о свободных одноместных номерах отеля на завтрашний день;

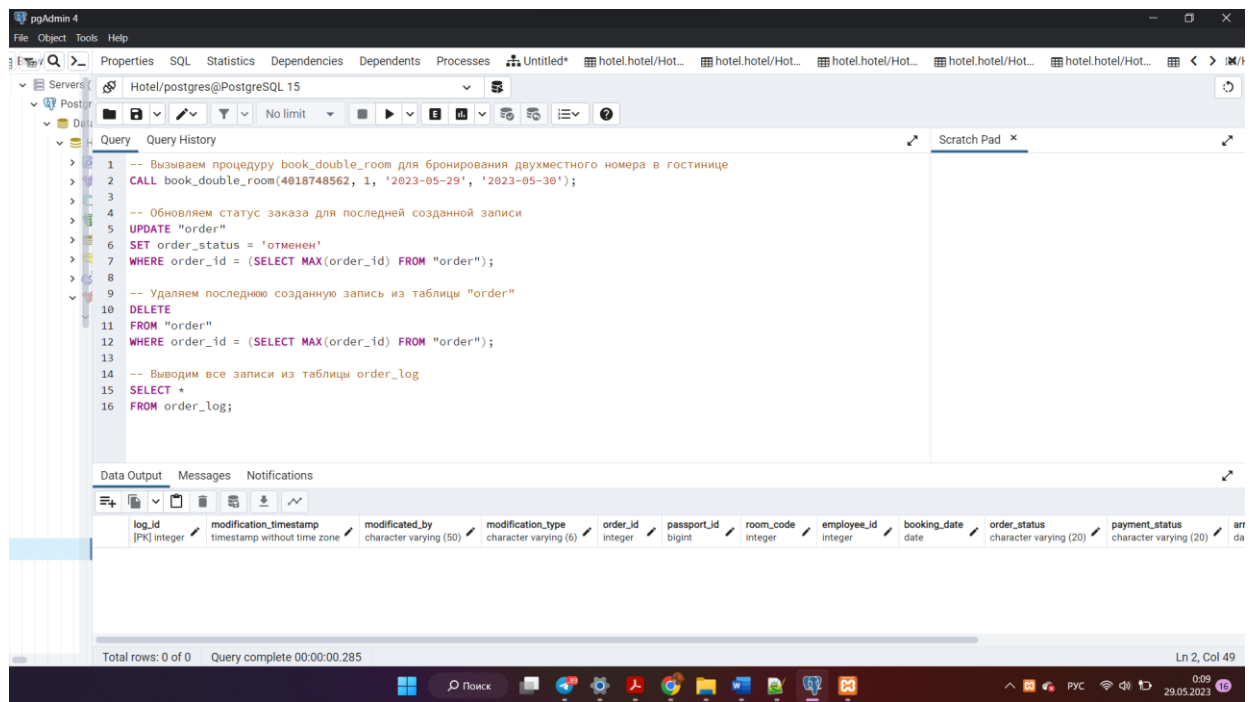


- бронирования двухместного номера в гостинице на заданную дату и количество дней проживания.





Создание триггеров:



pgAdmin 4

File Object Tools Help

Properties SQL Statistics Dependencies Dependents Processes Untitled* hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot...

Hotel/postgres@PostgreSQL 15

Query Query History

```
1 -- Вызываем процедуру book_double_room для бронирования двухместного номера в гостинице
2 CALL book_double_room(4018748562, 1, '2023-05-29', '2023-05-30');
3
4 -- Обновляем статус заказа для последней созданной записи
5 UPDATE "order"
6 SET order_status = 'отменен'
7 WHERE order_id = (SELECT MAX(order_id) FROM "order");
8
9 -- Удаляем последнюю созданную запись из таблицы "order"
10 DELETE
11 FROM "order"
12 WHERE order_id = (SELECT MAX(order_id) FROM "order");
13
14 -- Выводим все записи из таблицы order_log
15 SELECT *
16 FROM order_log;
```

Data Output Messages Notifications

modification_type	order_id	passport_id	room_code	employee_id	booking_date	order_status	payment_status	arrival_date	departure_date	costtt	payment_type
character varying (6)	integer	bigint	integer	integer	date	character varying (20)	character varying (20)	date	date	integer	character varying (30)

Total rows: 0 of 0 Query complete 00:00:00.285 Ln 2, Col 49

pgAdmin 4

File Object Tools Help

Properties SQL Statistics Dependencies Dependents Processes Untitled* hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot...

Hotel/postgres@PostgreSQL 15

Query Query History

```
1 -- Создаем функцию log_order для записи изменений в таблицу order_log
2 CREATE FUNCTION log_order()
3 RETURNS TRIGGER
4 LANGUAGE plpgsql
5 AS
6 $$
7 BEGIN
8 IF TG_OP = 'INSERT' THEN
9 -- Записываем информацию о новой записи
10 INSERT INTO order_log(modification_timestamp, modified_by, modification_type, order_id, passport_id,
11 room_code, employee_id, booking_date, order_status, payment_status, arrival_date,
12 departure_date, costtt, payment_type)
13 VALUES (NOW(), CURRENT_USER, TG_OP, NEW.order_id, NEW.passport_id,
14 NEW.room_code, NEW.employee_id, NEW.booking_date, NEW.order_status, NEW.payment_status,
15 NEW.arrival_date,
16 NEW.departure_date, NEW.costtt, NEW.payment_type);
17 RETURN NEW;
18 ELSEIF TG_OP = 'UPDATE' THEN
19 -- Записываем информацию об обновленной записи
20 INSERT INTO order_log(modification_timestamp, modified_by, modification_type, order_id, passport_id,
21 room_code, employee_id, booking_date, order_status, payment_status, arrival_date,
22 departure_date, costtt, payment_type)
23 VALUES (NOW(), CURRENT_USER, TG_OP, NEW.order_id, NEW.passport_id,
24 NEW.room_code, NEW.employee_id, NEW.booking_date, NEW.order_status, NEW.payment_status,
25 NEW.arrival_date,
26 NEW.departure_date, NEW.costtt, NEW.payment_type);
27 RETURN NEW;
28 ELSEIF TG_OP = 'DELETE' THEN
29 -- Записываем информацию об удаленной записи
30 INSERT INTO order_log(modification_timestamp, modified_by, modification_type, order_id, passport_id,
```

Total rows: 3 of 3 Query complete 00:00:00.129 Ln 49, Col 49

pgAdmin 4

File Object Tools Help

Properties SQL Statistics Dependencies Dependents Processes Untitled* hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot... hotel.hotel/Hot...

Hotel/postgres@PostgreSQL 15

Query Query History

```
30 INSERT INTO order_log(modification_timestamp, modified_by, modification_type, order_id, passport_id,
31 room_code, employee_id, booking_date, order_status, payment_status, arrival_date,
32 departure_date, costt, payment_type)
33 VALUES (NOW(), CURRENT_USER, TG_OP, OLD.order_id, OLD.passport_id,
34 OLD.room_code, OLD.employee_id, OLD.booking_date, OLD.order_status, OLD.payment_status,
35 OLD.arrival_date,
36 OLD.departure_date, OLD.costt, OLD.payment_type);
37 RETURN OLD;
38 END IF;
39 END;
40 $$;
41
42 -- Создаем триггер log_order_after_modification, который будет вызывать функцию log_order после изменения записей в
43 CREATE TRIGGER log_order_after_modification
44 AFTER INSERT OR UPDATE OR DELETE
45 ON "order"
46 FOR EACH ROW
47 EXECUTE FUNCTION log_order();
48 -- Вызываем процедуру book_double_room для бронирования двухместного номера в гостинице
49 CALL book_double_room(4018748562, 1, '2023-05-29', '2023-05-30');
50
51 -- Обновляем статус заказа для последней созданной записи
52 UPDATE "order"
53 SET order_status = 'отменен'
54 WHERE order_id = (SELECT MAX(order_id) FROM "order");
55
56 -- Удаляем последнюю созданную запись из таблицы "order"
57 DELETE
58 FROM "order"
59 WHERE order_id = (SELECT MAX(order_id) FROM "order");
60
61 -- Выводим все записи из таблицы order_log
62 SELECT *
63 FROM order_log;
```

Total rows: 3 of 3 Query complete 00:00:00.129 Ln 49, Col 49

pgAdmin 4

File Object Tools Help

Hotel/postgres@PostgreSQL 15

Query Query History

```
45 ON "order"
46 FOR EACH ROW
47 EXECUTE FUNCTION log_order();
48 -- Вызываем процедуру book_double_room для бронирования двухместного номера в гостинице
49 CALL book_double_room(4018748562, 1, '2023-05-29', '2023-05-30');
50
51 -- Обновляем статус заказа для последней созданной записи
52 UPDATE "order"
53 SET order_status = 'отменен'
54 WHERE order_id = (SELECT MAX(order_id) FROM "order");
55
56 -- Удаляем последнюю созданную запись из таблицы "order"
57 DELETE
58 FROM "order"
59 WHERE order_id = (SELECT MAX(order_id) FROM "order");
60
61 -- Выводим все записи из таблицы order_log
62 SELECT *
63 FROM order_log;
```

Data Output Messages Notifications

log_id	modification_timestamp	modified_by	modification_type	order_id	passport_id	room_code	employee_id	booking_date	order_status	payment_status	arr da
1	2023-05-29 00:11:05.908406	postgres	INSERT	5	4018748562	4	1	2023-05-29	ожидание оплаты	неоплачен	20
2	2023-05-29 00:11:05.908406	postgres	UPDATE	5	4018748562	4	1	2023-05-29	отменен	неоплачен	20
3	2023-05-29 00:11:05.908406	postgres	DELETE	5	4018748562	4	1	2023-05-29	отменен	неоплачен	20

Total rows: 3 of 3 Query complete 00:00:00.129 Ln 49, Col 49

The screenshot shows the pgAdmin 4 interface with a SQL query executed in the 'Query' tab. The query is as follows:

```
45 ON "order"
46 FOR EACH ROW
47 EXECUTE FUNCTION log_order();
48 -- Вызываем процедуру book_double_room для бронирования двухместного номера в гостинице
49 CALL book_double_room(4018748562, 1, '2023-05-29', '2023-05-30');
50
51 -- Обновляем статус заказа для последней созданной записи
52 UPDATE "order"
53 SET order_status = 'отменен'
54 WHERE order_id = (SELECT MAX(order_id) FROM "order");
55
56 -- Удаляем последнюю созданную запись из таблицы "order"
57 DELETE
58 FROM "order"
59 WHERE order_id = (SELECT MAX(order_id) FROM "order");
60
61 -- Выводим все записи из таблицы order_log
62 SELECT *
63 FROM order_log;
```

The 'Data Output' tab shows the results of the query, which are three rows of log data:

	modification_type	order_id	passport_id	room_code	employee_id	booking_date	order_status	payment_status	arrival_date	departure_date	cost	payment_type
1	INSERT	5	4018748562	4	1	2023-05-29	ожидание оплаты	неоплачен	2023-05-29	2023-05-30	[null]	[null]
2	UPDATE	5	4018748562	4	1	2023-05-29	отменен	неоплачен	2023-05-29	2023-05-30	[null]	[null]
3	DELETE	5	4018748562	4	1	2023-05-29	отменен	неоплачен	2023-05-29	2023-05-30	[null]	[null]

Total rows: 3 of 3 Query complete 00:00:00.129 Ln 49, Col 49

Выводы:

При выполнении данной лабораторной работы были приобретены практические навыки по работе с таблицами в базе данных PostgreSQL, созданию и выполнению процедур и функций, а также написанию необходимых для базы данных триггеров.