**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №5 «Процедуры, функции, триггеры в PostgreSQL**»**

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Евдокимова У.В.

Факультет: ИКТ

Группа: К3240

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

[Цель работы: 3](#_Toc152110467)

[Практическое задание 3](#_Toc152110468)

[Выполнение 3](#_Toc152110469)

[Вывод 10](#_Toc152110470)

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).

Практическое задание**:**

**Вариант 2 (max - 8 баллов)**

1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).

2**.** Создать авторский триггер по варианту индивидуального задания.

# Выполнение

**Создайте хранимые процедуры:**

1. Для увеличения цены всех номеров на 5 %, если в отеле нет свободных номеров

CREATE OR REPLACE PROCEDURE public.increase\_price\_if\_no\_vacancy()

LANGUAGE 'plpgsql'

AS $$

DECLARE

rooms\_occupied INT;

BEGIN

SELECT COUNT(\*) INTO rooms\_occupied

FROM room

WHERE occupancy\_status = 'Occupied';

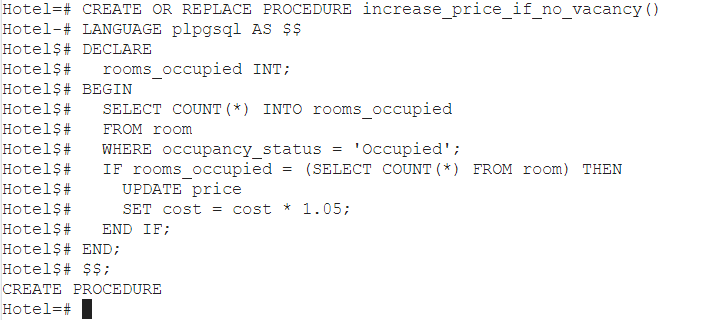
IF rooms\_occupied = (SELECT COUNT(\*) FROM room) THEN

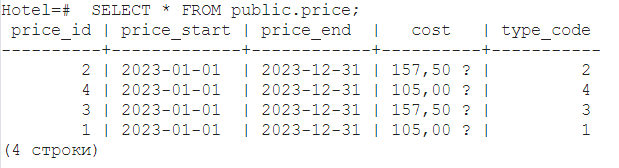
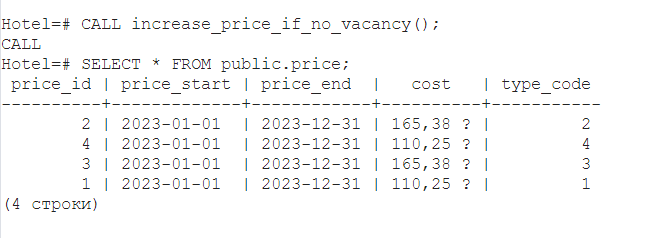
UPDATE price

SET cost = cost \* 1.05;

END IF;

END;

****

****

1. Для получения информации о свободных одноместных номерах отеля на завтрашний день

CREATE OR REPLACE PROCEDURE public.get\_available\_single\_rooms\_tomorrow()

LANGUAGE 'plpgsql'

AS $$

DECLARE

room\_record RECORD;

room\_cursor CURSOR FOR

SELECT

r.room\_number, rt.capacity, rt.name\_type\_code

FROM room r

JOIN "room type" rt ON r.type\_code = rt.type\_code

WHERE r.occupancy\_status = 'Free' AND rt.capacity = 1

AND NOT EXISTS (

SELECT 1

FROM "order" o

WHERE o.id\_room = r.id\_room

AND o.start\_of\_stay <= (CURRENT\_DATE + INTERVAL '1 day')

AND o.end\_of\_stay >= (CURRENT\_DATE + INTERVAL '1 day')

);

BEGIN

OPEN room\_cursor;

LOOP

FETCH room\_cursor INTO room\_record;

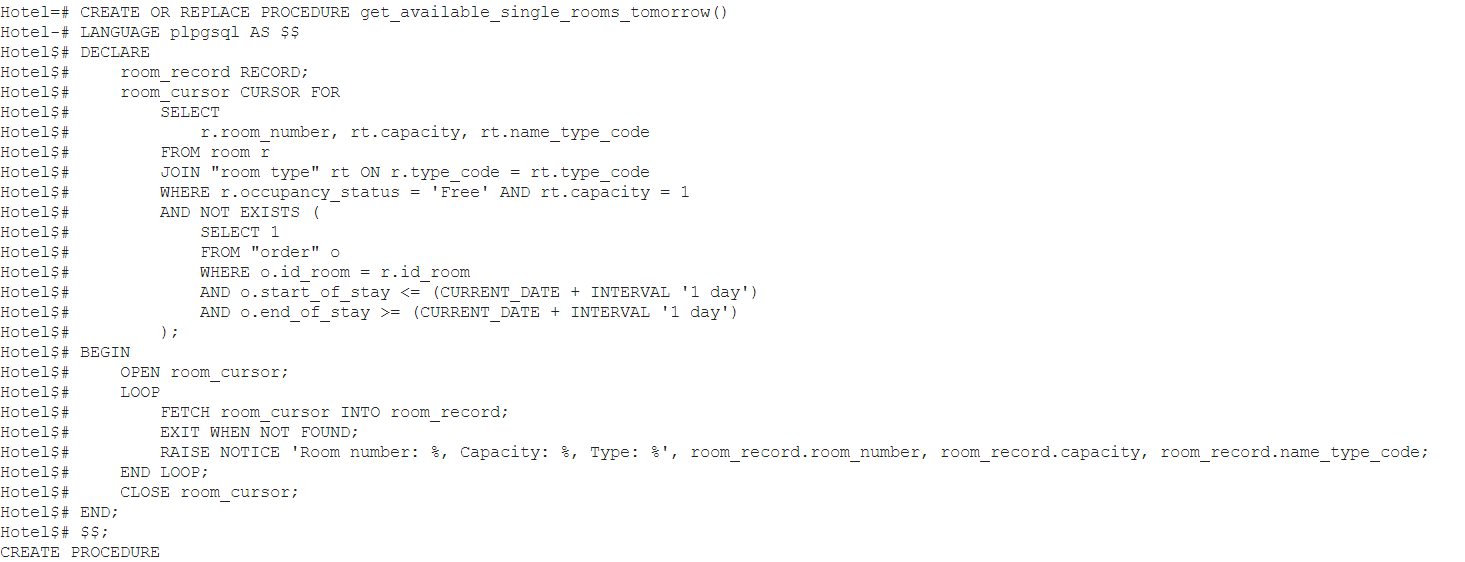
EXIT WHEN NOT FOUND;

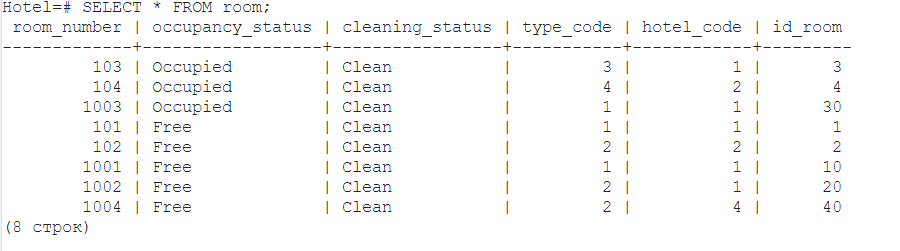
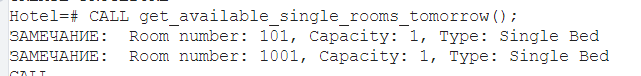
RAISE NOTICE 'Room number: %, Capacity: %, Type: %', room\_record.room\_number, room\_record.capacity, room\_record.name\_type\_code;

END LOOP;

CLOSE room\_cursor;

END;



Или

CREATE OR REPLACE PROCEDURE public.rooms\_tomorrow()

LANGUAGE 'plpgsql'

AS $$

BEGIN

DROP TABLE IF EXISTS available\_single\_rooms;

CREATE TEMP TABLE available\_single\_rooms AS

SELECT

r.room\_number, r.occupancy\_status, rt.capacity

FROM room r

JOIN "room type" rt ON r.type\_code = rt.type\_code

WHERE r.occupancy\_status = 'Free' AND rt.capacity = 1

AND NOT EXISTS (

SELECT 1

FROM "order" o

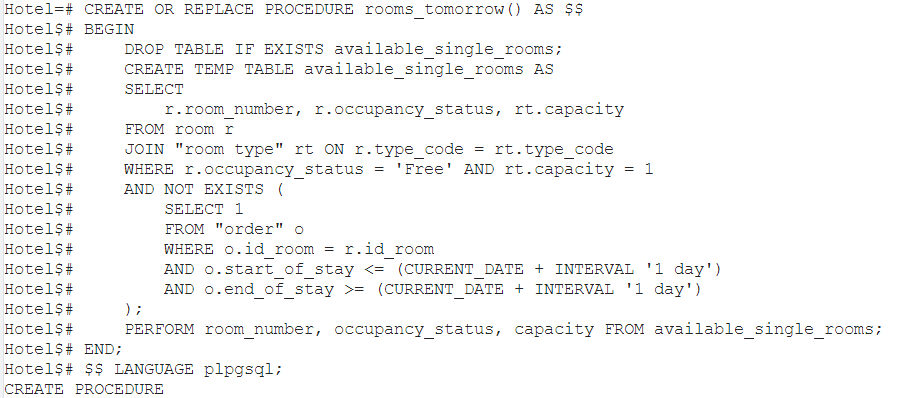
WHERE o.id\_room = r.id\_room

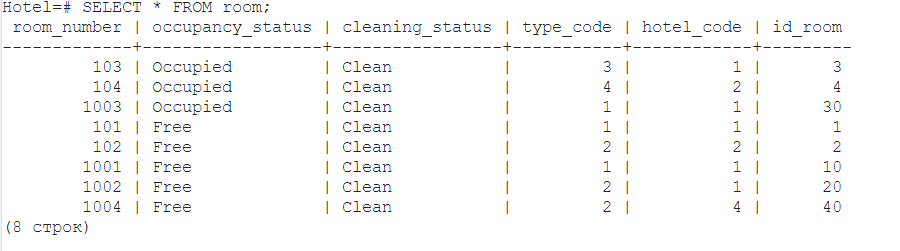
AND o.start\_of\_stay <= (CURRENT\_DATE + INTERVAL '1 day')

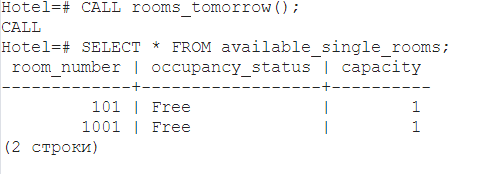
AND o.end\_of\_stay >= (CURRENT\_DATE + INTERVAL '1 day')

);

PERFORM room\_number, occupancy\_status, capacity FROM available\_single\_rooms;

END;  






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

CREATE OR REPLACE PROCEDURE public.to\_book\_double\_room(

IN guest\_pasport\_details character varying,

IN start\_of\_stay date,

IN number\_of\_days integer)

LANGUAGE 'plpgsql'

AS $BODY$

DECLARE

current\_booking\_date date;

room\_price money;

total\_cost money;

new\_order\_number integer;

BEGIN

current\_booking\_date := current\_date;

SELECT cost INTO room\_price

FROM "price"

WHERE type\_code = (SELECT type\_code FROM "room type" WHERE capacity = 2)

total\_cost := room\_price \* number\_of\_days;

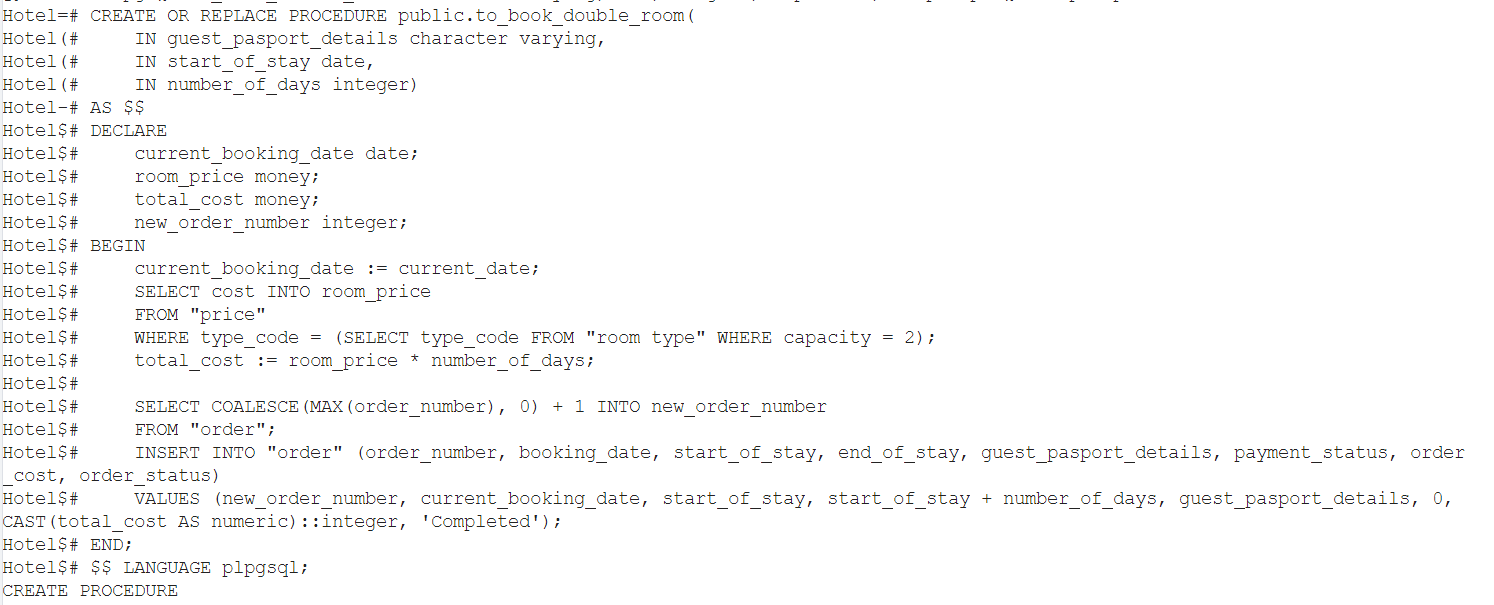
SELECT COALESCE(MAX(order\_number), 0) + 1 INTO new\_order\_number

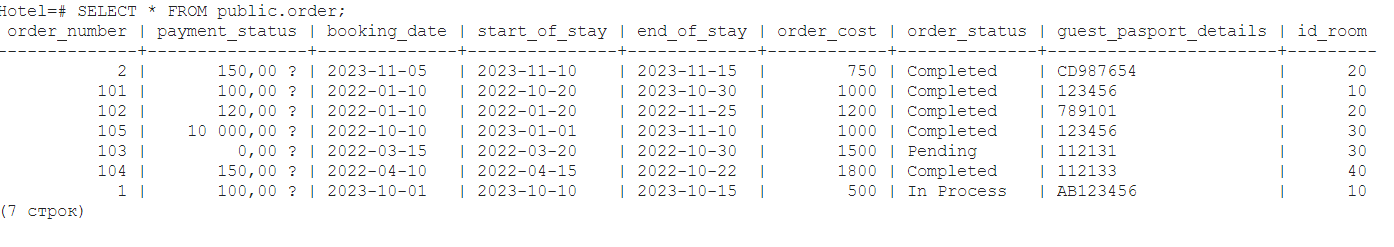
FROM "order";

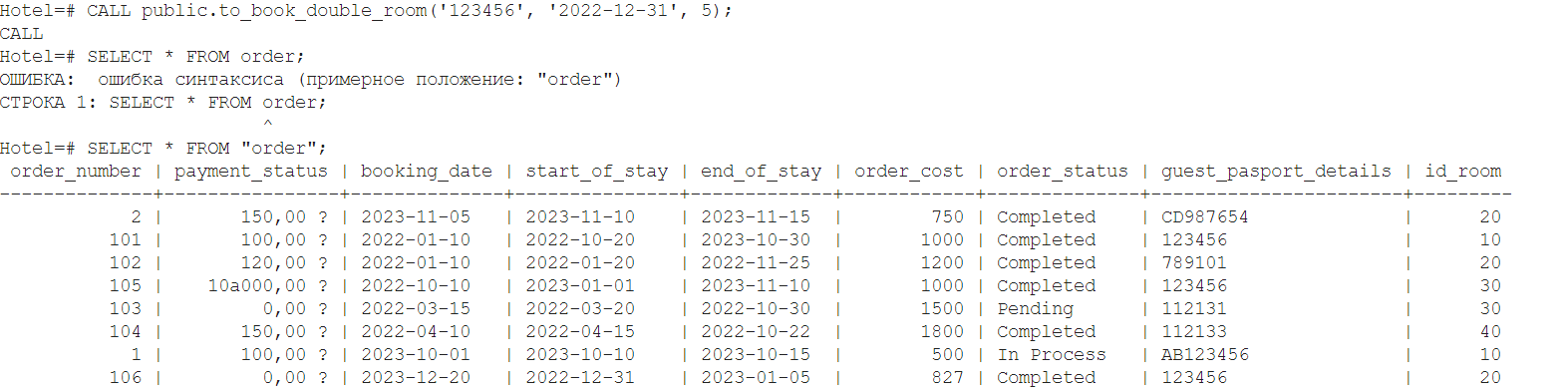
INSERT INTO "order" (order\_number, booking\_date, start\_of\_stay, end\_of\_stay, guest\_pasport\_details, payment\_status, order\_cost, order\_status)

VALUES (new\_order\_number, current\_booking\_date, start\_of\_stay, start\_of\_stay + number\_of\_days, guest\_pasport\_details, 0, CAST(total\_cost AS numeric)::integer, 'Completed');

END;

$BODY$; 



****

**Создайте необходимые триггеры:**

Триггер для обновления статуса заселения комнаты при бронировании.  
CREATE OR REPLACE FUNCTION set\_room\_to\_occupied()

RETURNS TRIGGER AS $$

BEGIN

IF (SELECT occupancy\_status FROM "Room" WHERE id\_room = NEW.id\_room) != 'Occupied' THEN

UPDATE "Room"

SET occupancy\_status = 'Occupied'

WHERE id\_room = NEW.id\_room;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_set\_room\_to\_occupied

AFTER INSERT ON "Order"

FOR EACH ROW

EXECUTE FUNCTION set\_room\_to\_occupied();

# Вывод

В ходе лабораторной работы была освоена работа с процедурами и триггерами.