**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."IncreasePriceIfNoRoomsAvailable"()

LANGUAGE 'plpgsql'

AS $BODY$

DECLARE

cnt integer;

BEGIN

SELECT count(\*) INTO cnt

FROM public."Room"

WHERE occupancy\_status = 'occupied';

IF cnt = 0 THEN

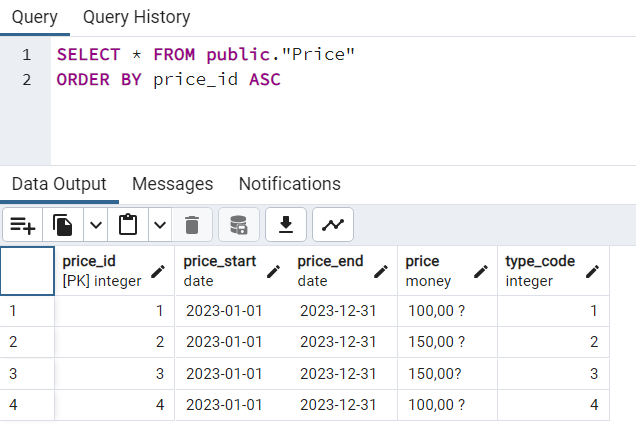
UPDATE public."Price" SET price = price \* 1.05;

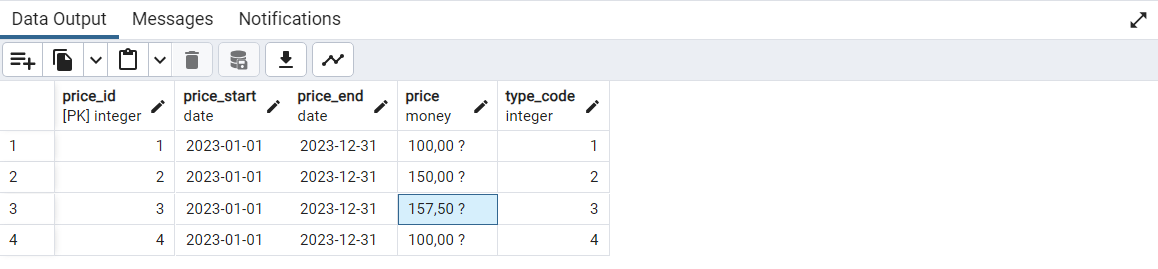
END IF;

END;

$BODY$;

****

****

****

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

CREATE OR REPLACE PROCEDURE public.GetAvailableSingleRoomsForTomorrow()

LANGUAGE plpgsql

AS $$

DECLARE

available\_room INTEGER;

BEGIN

FOR available\_room IN

SELECT r.room\_number

FROM public."Room" r

WHERE r.type\_code = 1

AND r.room\_number NOT IN (

SELECT o.id\_room

FROM public."Order" o

WHERE o.start\_of\_stay <= current\_date + interval '1 day'

AND o.end\_of\_stay > current\_date + interval '1 day'

)

LOOP

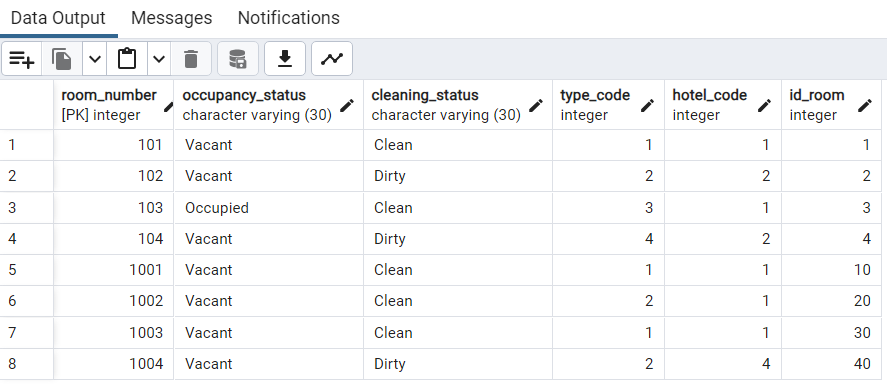
INSERT INTO available\_single\_rooms (room\_number) VALUES (available\_room);

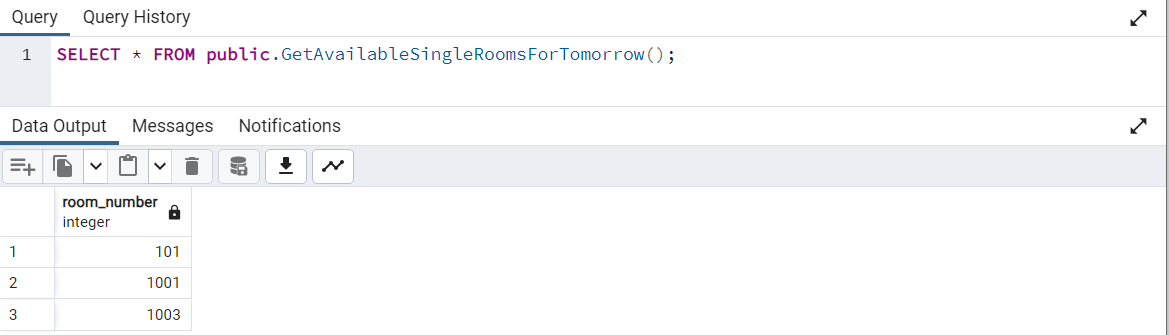
END LOOP;

END;

$$;





****

**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.