Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 3

по теме:

«Процедуры,функции,триггеры в PostgreSQL»

по дисциплине: Проектирование и реализация баз данных

Специальность:

09.03.03 Мобильная и сетевая разработка

Проверила:
Говорова М.М.
Дата: «..» … 2023 г.
Оценка _____

Выполнил:
студент группы К32392
Золотухин А.Н.

Санкт-Петербург 2022/2023

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

Практическое задание:

Вариант 1:

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4).Составить 3 запроса на модификацию данных (INSERT,UPDATE, DELETE) с использованием подзапросов.

2. 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_db

Схемы логической модели базы данных

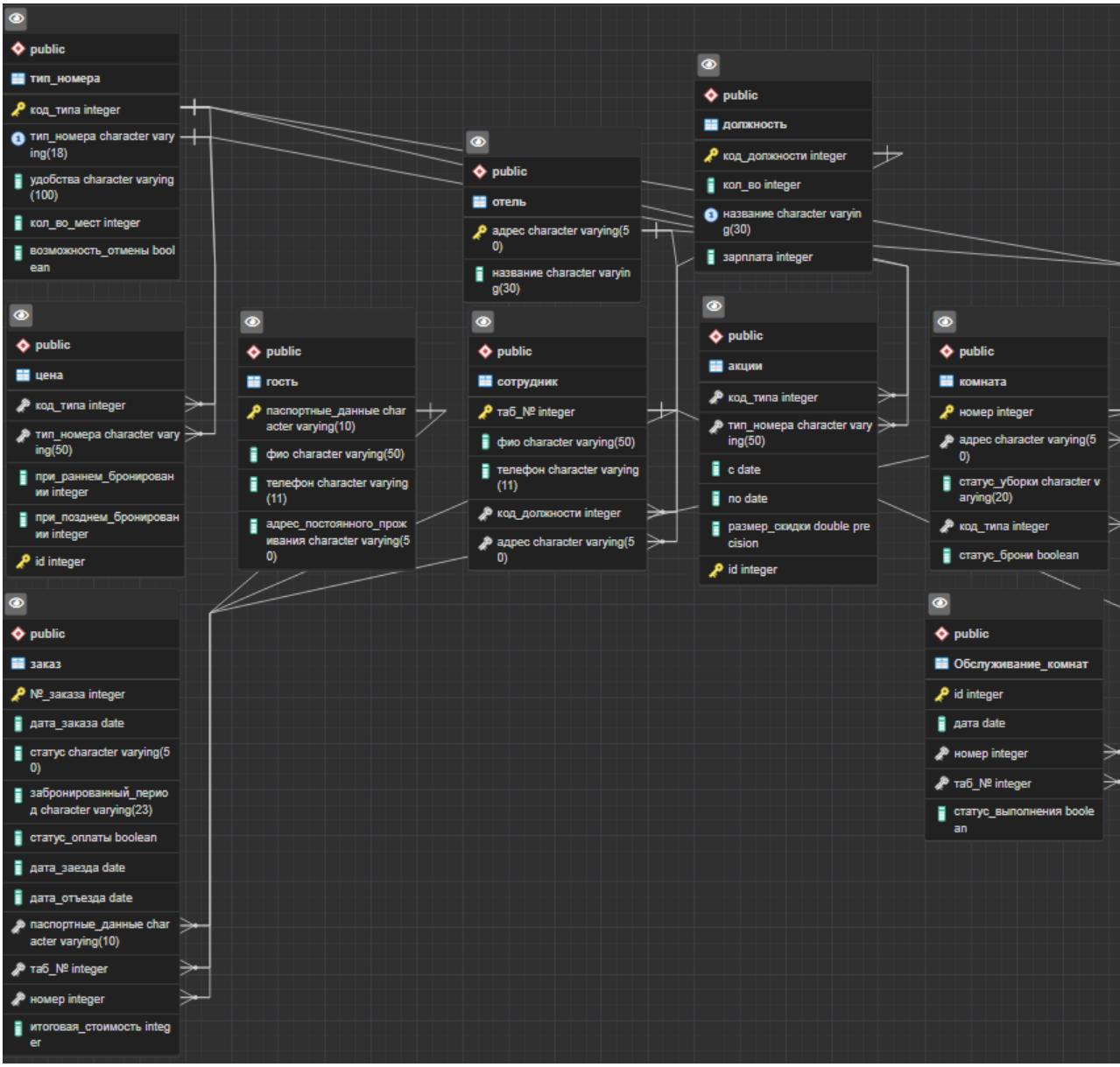


Рисунок 1 – ERD базы данных 1

Запросы на выборку.

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

Запрос:

```
CREATE OR REPLACE FUNCTION increase_price_if_no_vacancies()
RETURNS VOID AS $$
BEGIN
   UPDATE цена
   SET при_раннем_бронировании = при_раннем_бронировании * 1.05,
      при_позднем_бронировании = при_позднем_бронировании * 1.05
   WHERE (SELECT COUNT(*)
         FROM комната AS k
         WHERE k.статус_брони = false
         AND k.адрес IN (SELECT o.адрес FROM отель AS o)) = 0;
END;
$$ LANGUAGE plpgsql;
```

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

Запрос:

```
CREATE OR REPLACE FUNCTION
get_single_room_vacancies_tomorrow_with_address
(адрес_отеля VARCHAR, номера_тип VARCHAR)
RETURNS TABLE(номер INTEGER, тип_номера VARCHAR, стоимость
INTEGER, дата DATE) AS $$
BEGIN
   RETURN QUERY
   SELECT k.номер, tn.тип_номера, c.при_раннем_бронировании,
z.дата_заезда
   FROM комната AS k
   LEFT JOIN заказ AS z ON k.номер = z.номер
   JOIN тип_номера AS tn ON k.код_типа = tn.код_типа
   JOIN цена AS c ON tn.код_типа = c.код_типа
   WHERE tn.тип_номера = номера_тип AND k.адрес = адрес_отеля
   AND (z.дата_заезда != CURRENT_DATE + INTERVAL '1 day' OR
z.дата_заезда IS NULL);
END;
$$ LANGUAGE plpgsql;
```

Вывод:

	номер integer	тип_номера character varying	стоимость integer	дата date
1	228	Артем	8000	[null]
2	229	Артем	8000	2023-06-01
3	229	Артем	8000	2023-04-10
4	230	Артем	8000	[null]
5	231	Артем	8000	2023-07-01

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

Запрос:

```
CREATE OR REPLACE FUNCTION book_double_room(
    p_passport_data VARCHAR(10),
    p_start_date DATE,
    p_days INTEGER,
    p_employee_id INTEGER
)
RETURNS VOID AS $$
DECLARE
    v_room RECORD;
    v_end_date DATE;
    v_total_cost NUMERIC;
BEGIN
    SELECT INTO v_room k.номер, k.адрес, k.статус_уборки, k.код_типа
    FROM комната AS k
    JOIN тип_номера AS tn ON k.код_типа = tn.код_типа
    WHERE k.статус_брони = false
    AND tn.кол_во_мест = 2
    AND k.статус_уборки = 'убрано'
    LIMIT 1;

    IF FOUND THEN
        v_end_date := p_start_date + p_days;
        SELECT INTO v_total_cost c.при_раннем_бронировании * p_days
        FROM цена AS c
        WHERE c.код_типа = v_room.код_типа;

        INSERT INTO заказ ("№_заказа", "дата_заказа", "статус",
"забронированный_период", "статус_оплаты", "дата_заезда", "дата_отъезда",
"паспортные_данные", "таб_№", "номер", "итоговая_стоимость")
        VALUES (DEFAULT, CURRENT_DATE, 'забронировано', p_days, 'не
оплачено', p_start_date, v_end_date, p_passport_data, p_employee_id,
```

v_room.номер, v_total_cost);

UPDATE комната
SET статус_брони = true
WHERE номер = v_room.номер;
ELSE
RAISE NOTICE 'Не найдено свободных двухместных номеров';
END IF;
END;
$$ LANGUAGE plpgsql;

Создание Триггеров.

1. Триггер, который не позволяет добавлять нового сотрудника определенной должности, если количество таких сотрудников превысит значения поля "кол_во" их должности в таблице "должность".

Запрос: (Почему то в этот раз он скопировался с форматированием – ничего не понимаю)

```
CREATE OR REPLACE FUNCTION check_position_limit()
RETURNS TRIGGER AS $$
DECLARE
    position_count INTEGER;
    position_limit INTEGER;
BEGIN
    SELECT COUNT(*) INTO position_count
    FROM сотрудник
    WHERE код_должности = NEW.код_должности;

    SELECT кол_во INTO position_limit
    FROM должность
    WHERE код_должности = NEW.код_должности;

    IF position_count >= position_limit THEN
        RAISE EXCEPTION 'Превышен лимит сотрудников для должности с кодом
%', NEW.код_должности;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_position_limit_trigger ON сотрудник;

CREATE TRIGGER check_position_limit_trigger
BEFORE INSERT ON сотрудник
FOR EACH ROW
EXECUTE FUNCTION check_position_limit();
```

Проверка работы: У нас в отелях может быть всего 5 уборщиков, я сделал

так, что их уже 5, при попытке добавления нового сотрудника вот что будет:

```
1  SELECT COUNT(*) FROM "сотрудник" WHERE "код_должности" = 1;
```

Loading..

Data output | Messages | Notifications

	count bigint
1	5

```
1  INSERT INTO сотрудник (таб_№, фио, телефон, код_должности, адрес)
2  VALUES (203, 'Как же хочется Пицы', '9003335678', 1, 'ул. Артем, д.1');
```

Data output | Messages | Notifications

```
ERROR: ОШИБКА:  Превышен лимит сотрудников для должности с кодом 1
CONTEXT:  функция PL/pgSQL check_position_limit(), строка 15, оператор RAISE


SQL state: P0001
```

2. Триггер, который не позволяет бронировать комнату, которая на данный момент уже занята!

Запрос:

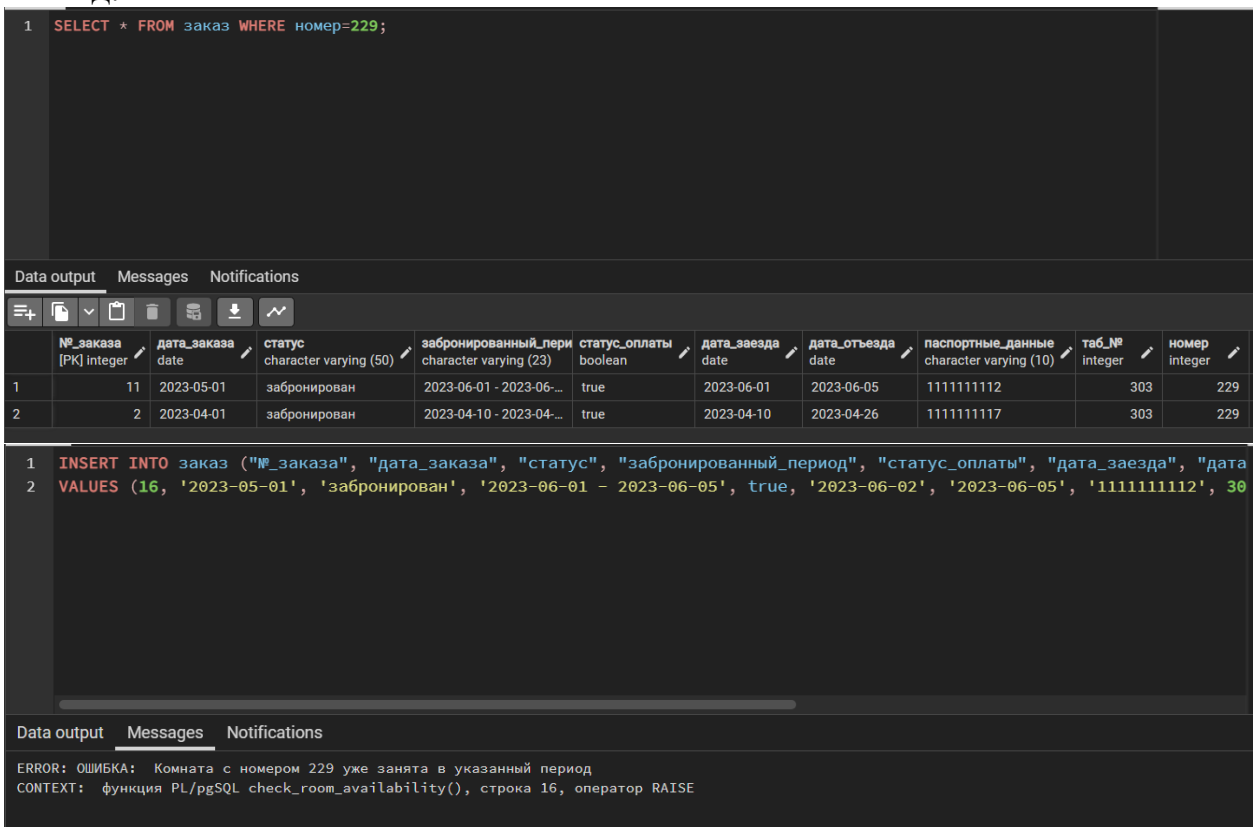
CREATE OR REPLACE FUNCTION check_room_availability()
RETURNS TRIGGER AS $$

```
DECLARE
   room_availability BOOLEAN;
BEGIN
   SELECT NOT EXISTS (
      SELECT 1
      FROM заказ
      WHERE номер = NEW.номер
      AND ((NEW.дата_заезда BETWEEN дата_заезда AND дата_отъезда)
         OR (NEW.дата_отъезда BETWEEN дата_заезда AND дата_отъезда)
         OR (дата_заезда BETWEEN NEW.дата_заезда AND
NEW.дата_отъезда)
         OR (дата_отъезда BETWEEN NEW.дата_заезда AND
NEW.дата_отъезда))
   ) INTO room_availability;

   IF NOT room_availability THEN
      RAISE EXCEPTION 'Комната с номером % уже занята в указанный
период', NEW.номер;
   END IF;

   RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Вывод:



Вывод: При выполнении данной лабораторной работы были приобретены практические навыки по работе с таблицами в базе данных PostgreSQL, выполнением процедур и функций, а также использованием триггеров.