

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки **09.03.03 Прикладная информатика**

**О Т Ч Е Т**

**Лабораторная работа 5**

Тема: Процедуры, функции, триггеры в PostgreSQL

Обучающийся: Файзулин Радмир Русланович, группы К3239

Преподаватель: Говорова Марина Михайловна

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| Цель работы:.....                            | 3  |
| Ход работы.....                              | 4  |
| Схема базы данных.....                       | 4  |
| Функции/процедуры согласно инд. заданию..... | 5  |
| Триггер согласно инд. заданию.....           | 10 |
| Вывод.....                                   | 11 |

**Цель работы:**

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

**Оборудование:** компьютерный класс.

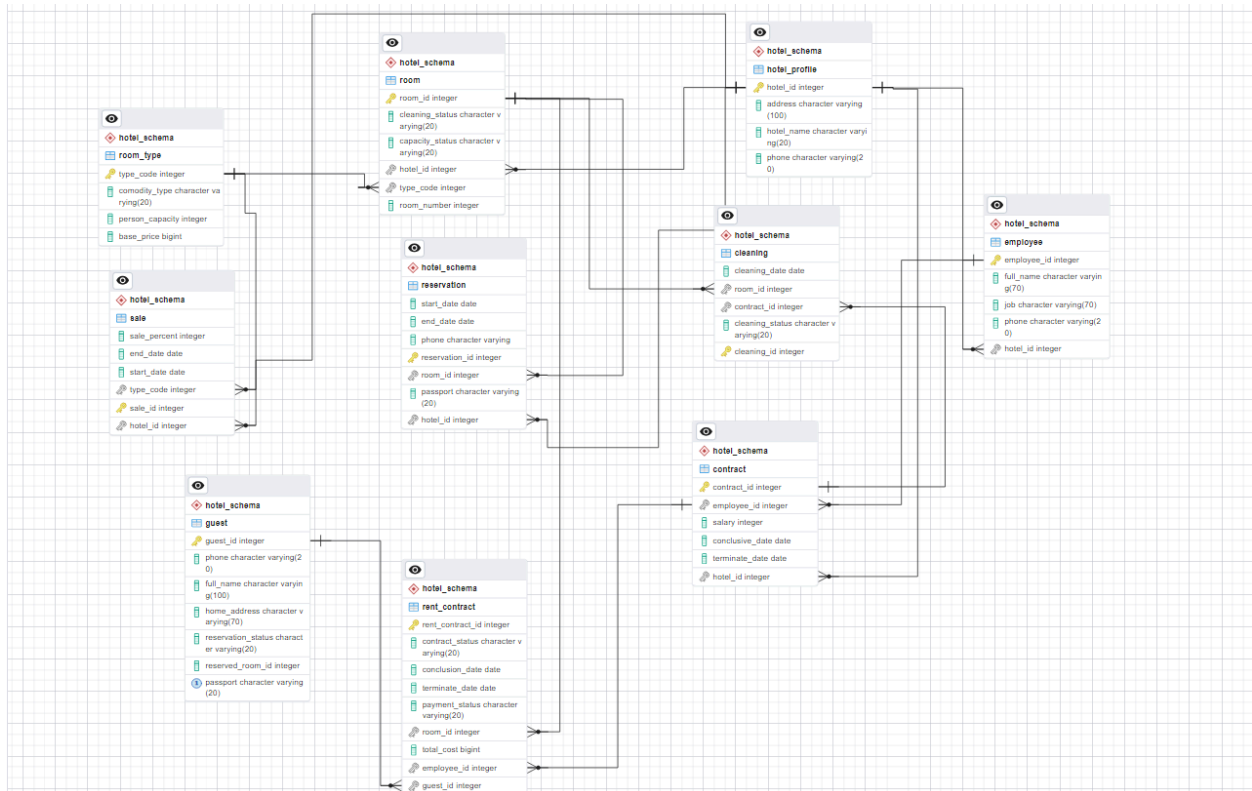
**Программное обеспечение:** СУБД PostgreSQL, pgAdmin 4, SQL Shell(psql).

**Практическое задание:**

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

## Ход работы

### Схема базы данных



### Hotel (Отель)







### Функции/процедуры согласно инд. заданию

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

```

CREATE OR REPLACE FUNCTION unsale (hotel_id_add INT)
RETURNS VOID AS $$
DECLARE
    i RECORD;
    j RECORD;
BEGIN
    FOR i IN SELECT DISTINCT type_code FROM hotel_schema.room_type
    LOOP
        FOR j IN SELECT DISTINCT hotel_id FROM hotel_schema.room WHERE
        (
            SELECT COUNT(*) FROM hotel_schema.room r
            WHERE r.hotel_id = hotel_id_add AND r.capacity_status = 'busy'
        ) = (
            SELECT COUNT(*) FROM hotel_schema.room r
            WHERE r.hotel_id = hotel_id_add
        ) LOOP
            IF (SELECT COUNT(*) FROM hotel_schema.room r
                WHERE r.hotel_id = j.hotel_id AND r.capacity_status = 'busy') =
                (SELECT COUNT(*) FROM hotel_schema.room r
                WHERE r.hotel_id = j.hotel_id) THEN
                INSERT INTO hotel_schema.sale (sale_percent, end_date, start_date,
                type_code, hotel_id)
                VALUES (-5, CURRENT_DATE + 1, CURRENT_DATE, i.type_code,
                j.hotel_id);
            END IF;
        END LOOP;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

```

|   |  sale_percent ^ 1 |  end_date |  start_date |  type_code |  sale_id |  hotel_id ^ 2 |
|---|--|--|--|---|---|--|
| 1 | -5   | 2024-04-05   | 2024-04-04   | 6   | 64  | 3  |
| 2 | -5   | 2024-04-05   | 2024-04-04   | 1   | 65  | 3  |
| 3 | -5   | 2024-04-05   | 2024-04-04   | 3   | 66  | 3  |
| 4 | -5   | 2024-04-05   | 2024-04-04   | 5   | 67  | 3  |
| 5 | -5   | 2024-04-05   | 2024-04-04   | 2   | 68  | 3  |
| 6 | -5   | 2024-04-05   | 2024-04-04   | 4   | 69  | 3  |

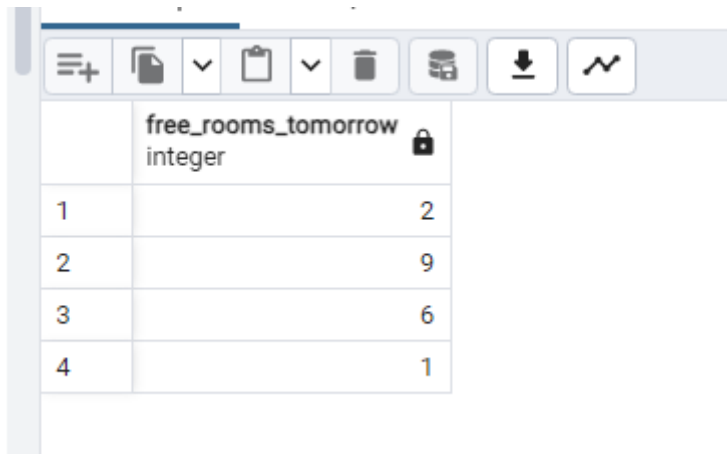
- для получения информации о свободных одноместных номерах отеля на завтрашний день;

```

CREATE OR REPLACE FUNCTION free_rooms_tomorrow (hotel_id_add INT)
RETURNS TABLE (room_number INT) AS $$
BEGIN
    RETURN QUERY
    SELECT room.room_number
    FROM hotel_schema.room
    WHERE room_id IN (
        SELECT room_id FROM hotel_schema.room
        WHERE hotel_id = hotel_id_add
        AND capacity_status = 'free'
        AND type_code IN (1, 2)
        AND room_id NOT IN (
            SELECT room_id FROM hotel_schema.reservation
            WHERE start_date <= CURRENT_DATE + 1
            AND end_date >= CURRENT_DATE + 1
        )
    )
    UNION
    SELECT room.room_number FROM hotel_schema.room
    WHERE room_id IN (
        SELECT room_id
        FROM hotel_schema.rent_contract
        WHERE terminate_date = CURRENT_DATE
        AND rent_contract.room_id NOT IN
            (SELECT room_id FROM hotel_schema.reservation
            WHERE reservation.start_date = CURRENT_DATE + 1)
    );

```

```
END;  
$$ LANGUAGE plpgsql;
```



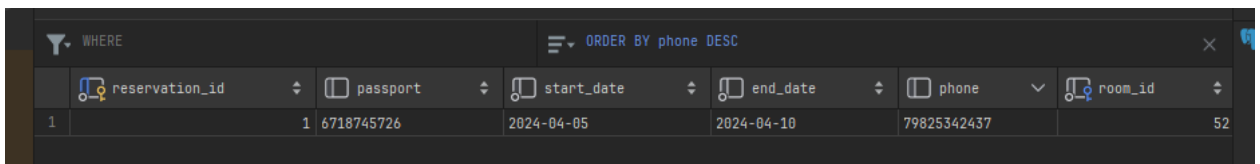
The image shows a screenshot of a database interface. At the top, there is a toolbar with icons for adding, deleting, and other table operations. Below the toolbar is a table with the name 'free\_rooms\_tomorrow' and a data type of 'integer'. The table has four rows of data.

|   | free_rooms_tomorrow<br>integer |
|---|--------------------------------|
| 1 | 2                              |
| 2 | 9                              |
| 3 | 6                              |
| 4 | 1                              |



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

```
CREATE OR REPLACE FUNCTION reserve (st_date DATE, day_count INT,
passp VARCHAR, phone_num VARCHAR, room_id_num INT)
RETURNS VOID AS $$
BEGIN
    INSERT INTO hotel_schema.reservation (passport, start_date, end_date,
phone, room_id)
    VALUES (passp, st_date, st_date + day_count, phone_num, room_id_num);
END;
$$ LANGUAGE plpgsql;
```



The screenshot shows a database query result with columns: reservation\_id, passport, start\_date, end\_date, phone, and room\_id. The result is ordered by phone number in descending order. The first row shows a reservation with passport 6718745726, start date 2024-04-05, end date 2024-04-10, phone 79825342437, and room\_id 52.

|   | reservation_id | passport   | start_date | end_date   | phone       | room_id |
|---|----------------|------------|------------|------------|-------------|---------|
| 1 | 1              | 6718745726 | 2024-04-05 | 2024-04-10 | 79825342437 | 52      |

Сначала написал вспомогательную функцию (ниже есть скрин для удобства чтения)

```
CREATE OR REPLACE FUNCTION free_rooms_by_period (hotel_id_add INT,
st_date DATE, fin_date DATE)
RETURNS TABLE (room_number INT) AS $$
BEGIN
    RETURN QUERY
    SELECT room.room_number
    FROM hotel_schema.room
    WHERE room_id IN (
        SELECT room_id FROM hotel_schema.room
        WHERE hotel_id = hotel_id_add
```

```

AND capacity_status = 'free'
AND type_code IN (3, 4, 5, 6)
AND room_id NOT IN (
    SELECT room_id FROM hotel_schema.reservation
    WHERE (start_date >= st_date AND start_date <= fin_date)
        AND (start_date <= st_date AND fin_date <=
end_date)
        AND (end_date >= st_date AND end_date <=
fin_date)
    )
)
UNION
SELECT room.room_number FROM hotel_schema.room
WHERE room_id IN (
    SELECT room_id
    FROM hotel_schema.rent_contract
    WHERE terminate_date <= st_date
    AND rent_contract.room_id NOT IN
        (SELECT room_id FROM hotel_schema.reservation
WHERE reservation.start_date = st_date + 1)
    );
END;
$$ LANGUAGE plpgsql;

```

```

CREATE OR REPLACE FUNCTION free_rooms_by_period (hotel_id_add INT, st_date DATE, fin_date DATE)
RETURNS TABLE (room_number INT) AS $$
BEGIN
    RETURN QUERY
    SELECT room.room_number
    FROM hotel_schema.room
    WHERE room_id IN (
        SELECT room_id FROM hotel_schema.room
        WHERE hotel_id = hotel_id_add
        AND capacity_status = 'free'
        AND type_code IN (3, 4, 5, 6)
        AND room_id NOT IN (
            SELECT room_id FROM hotel_schema.reservation
            WHERE (start_date >= st_date AND start_date <= fin_date)
            AND (start_date <= st_date AND fin_date <= end_date)
            AND (end_date >= st_date AND end_date <= fin_date)
        )
    )
    UNION
    SELECT room.room_number FROM hotel_schema.room
    WHERE room_id IN (
        SELECT room_id
        FROM hotel_schema.rent_contract
        WHERE terminate_date <= st_date
        AND rent_contract.room_id NOT IN
            (SELECT room_id FROM hotel_schema.reservation WHERE reservation.start_date = st_date + 1)
    );
END;
$$ LANGUAGE plpgsql;

```

Затем основную с использованием вспомогательной:

```

CREATE OR REPLACE FUNCTION reserve (st_date DATE, day_count INT,
passp VARCHAR, phone_num VARCHAR, hotel_id_num INT, room_id_num
INT)

```

```

RETURNS VOID AS $$

```

```

BEGIN

```

```

    IF room_id_num IN (SELECT free_rooms_by_period (hotel_id_add,
st_date, fin_date))

```

```

        THEN

```

```

            INSERT INTO hotel_schema.reservation (passport, start_date,
end_date, phone, hotel_id, room_id)

```

```

            VALUES (passp, st_date, st_date + day_count, phone_num,
hotel_id_num, room_id_num);

```

```

        END IF;

```

```

END;

```

```

$$ LANGUAGE plpgsql;

```

## Триггер согласно инд. заданию

1

- Логирование при любых изменениях в таблице guest

```
CREATE OR REPLACE FUNCTION guest_trigger_function()
RETURNS TRIGGER AS $$
BEGIN
    IF TG_OP = 'INSERT' THEN
        RAISE NOTICE 'Добавлен гость: %', NEW;
    ELSIF TG_OP = 'UPDATE' THEN
        RAISE NOTICE 'Изменен гость: % на %', OLD, NEW;
    ELSIF TG_OP = 'DELETE' THEN
        RAISE NOTICE 'Удален гость: %', OLD;
    END IF;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER guest_trigger
BEFORE INSERT OR UPDATE OR DELETE ON hotel_schema.guest
FOR EACH ROW
EXECUTE FUNCTION guest_trigger_function();
```

| Data Output  | Сообщения | Notifications |
|--|-----------|---------------|
| ЗАМЕЧАНИЕ: Добавлен гость: (64,79825342424,"Файзулин Ррадмир","Альпийский пер 15к2",reserves,24,6718747474)  |           |               |
| ERROR: конец триггерной процедуры достигнут без RETURN<br>CONTEXT: функция PL/pgSQL guest_trigger_function() |           |               |
| ОШИБКА: конец триггерной процедуры достигнут без RETURN<br>SQL-состояние: 2F005                              |           |               |

2

- Изменение статуса уборки на 'done' при выполнении уборки

```
CREATE OR REPLACE FUNCTION cleaning_trigger_function()
RETURNS TRIGGER AS $$
BEGIN
    IF TG_OP = 'INSERT' THEN
        UPDATE hotel_schema.room
        SET cleaning_status = 'done'
        WHERE room_id = NEW.room_id;
        RAISE NOTICE 'room_id % cleaning is done!', NEW.room_id;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER cleaning_trigger
BEFORE INSERT ON hotel_schema.cleaning
FOR EACH ROW
EXECUTE FUNCTION cleaning_trigger_function();
```

Запрос

история запросов

1

INSERT INTO hotel\_schema.cleaning (cleaning\_date, room\_id, contract\_id, cleaning\_status)

2

VALUES ('2024-04-04', 53, 9, 'done')

Data Output

Сообщения

Notifications

ЗАМЕЧАНИЕ: room\_id 53 cleaning is done!

INSERT 0 1

Запрос завершён успешно, время выполнения: 56 msec.

ДО

|   |    |       |      |   |   |   |
|---|----|-------|------|---|---|---|
| 4 | 53 | to do | free | 1 | 1 | 6 |
|---|----|-------|------|---|---|---|

ПОСЛЕ

|  |    |      |      |   |   |   |
|--|----|------|------|---|---|---|
|  | 53 | done | free | 1 | 1 | 6 |
|--|----|------|------|---|---|---|

## **Вывод**

Цели и задачи выполнены.

Овладел практическими навыками создания и использования процедур, функций и триггеров различных видов (добавление, удаление, обновление) в базе данных PostgreSQL с помощью СУБД PostgreSQL.

Было создано три различных функции согласно ИЗ, затем добавлено два триггера, первый для логирования событий в таблице guest, а второй для