

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

**Отчет**  
**По Лабораторной работе 2**  
**ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В**  
**POSTGRESQL**  
**по дисциплине «Базы данных»**

Автор: Циминтия Н

Факультет: ИКТ

Группа: K32402

Преподаватель: Говорова М. М.

Дата: 22.03.2023

**Цель работы:** овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

**Программное обеспечение:** СУБД PostgreSQL 1X, pgAdmin 4.

**Практическое задание:**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

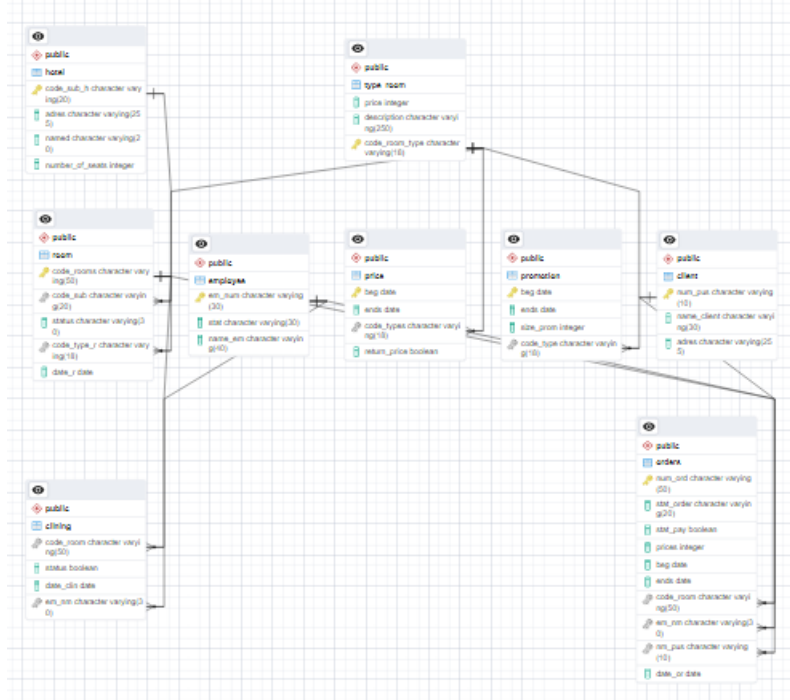
**Индивидуальное практическое задание:**

Предметная область «**Расписание занятий и распределение аудиторного фонда**».  
Составьте запросы на выборку:

**Задание 2.** Создайте запросы:

- Составить список всех 2-местных номеров отелей, с ценой менее 200 т.р., упорядочив данные в порядке уменьшения стоимости.
- Выбрать все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.
- Чему равен общий суточный доход каждого отеля за последний месяц?
- Составить список свободных номеров одного из отелей на текущий день.
- Найти общие потери от незанятых номеров за текущий день по всей сети.
- Определить, в каком отеле имеется наибольшее количество незанятых номеров на текущие сутки.
- Определить самый популярный тип номеров за последний год.

## II. схема логической модели базы данных, сгенерированная в Generate ERD



### SELECT

1 Составить список всех 2-местных номеров отелей, с ценой менее 200 т.р., упорядочив данные в порядке уменьшения стоимости.

```
1 SELECT code_room_type
2 FROM type_room
3 WHERE price < 200000 and number_of_rooms = 2
4 ORDER BY price DESC;
```

| Data Output |   | Сообщения | Notifications |
|-------------|---|-----------|---------------|
|             | code_room_type<br>[PK] character varying (18) |           |               |
| 1           | 12  |           |               |
| 2           | 15  |           |               |

- Выбрать все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.

```

1 SELECT *
2 FROM orders
3 WHERE ends >= CURRENT_DATE - INTERVAL '2 weeks';

```

| Data Output   Сообщения   Notifications |  |                                      |                     |                   |             |              |                                     |                                 |                                  |                 |  |
|---|--|--------------------------------------|---------------------|-------------------|-------------|--------------|-------------------------------------|---------------------------------|----------------------------------|-----------------|--|
|   | num_ord<br>[PK] character varying (50) | stat_order<br>character varying (20) | stat_pay<br>boolean | prices<br>integer | beg<br>date | ends<br>date | code_room<br>character varying (50) | em_nm<br>character varying (30) | nm_pus<br>character varying (10) | date_or<br>date |  |
| 1                                       | 21                                     | re                                   | true                | 323               | 2023-05-10  | 2023-06-01   | 12                                  | 23                              | 45                               | 2023-02-12      |  |

Чему равен общий суточный доход каждого отеля за последний месяц?

```

1 SELECT hotel.named, SUM(orders.prices) AS total_income
2 FROM orders
3 join hotel on orders.code_h = hotel.code_sub_h
4 WHERE date_or >= CURRENT_DATE - INTERVAL '1 month'
5 GROUP BY named;

```

| Data Output   Сообщения   Notifications |                                 |                        |
|---|---------------------------------|------------------------|
|   | named<br>character varying (20) | total_income<br>bigint |
| 1                                       | Op                              | 128734                 |

4 Составить список свободных номеров одного из отелей на текущий день.

```

1
2 SELECT r.code_rooms
3 FROM room r
4 LEFT JOIN orders o ON r.code_rooms = o.code_room
5 WHERE o.code_room IS NULL;
6

```

| Data Output   Сообщения   Notifications |   |
|---|---|
|   | code_rooms<br>[PK] character varying (50) |
| 1                                       | 13  |

- Найти общие потери от незанятых номеров за текущий день по всей сети.

```

1 SELECT code_type_r, COUNT(*) AS count
2 FROM room
3 WHERE date_r < '01-01-2023'
4 GROUP BY code_type_r
5 ORDER BY count DESC
6 LIMIT 1;

```

| Data Output  | Сообщения | Notifications |
|--|-----------|---------------|
| <div> <div>code_type_r</div> <div>character varying (10)</div> <div>count</div> <div>bigint</div> </div> |           |               |

- Определить, в каком отеле имеется наибольшее количество незанятых номеров на текущие сутки.

```

1 SELECT h.named, COUNT(*) AS available_rooms
2 FROM room r
3 JOIN hotel h on r.code_sub = h.code_sub_h
4 WHERE code_sub IN (
5     SELECT DISTINCT code_sub
6     FROM room
7 )
8 AND code_rooms NOT IN (
9     SELECT code_room
10    FROM orders
11   WHERE CURRENT_DATE BETWEEN beg AND ends
12 )
13 GROUP BY h.named
14 ORDER BY available_rooms DESC
15 LIMIT 1;

```

| Data Output  | Сообщения | Notifications |
|--|-----------|---------------|
| <div> <div>named</div> <div>character varying (20)</div> <div>available_rooms</div> <div>bigint</div> </div> |           |               |
| 1  | Op        | 1             |

- Определить самый популярный тип номеров за последний год.

```

1 SELECT code_room, COUNT(*) AS room_count
2 FROM orders
3 INNER JOIN room ON orders.code_room = room.code_rooms
4 WHERE ends >= CURRENT_DATE - INTERVAL '1 year'
5 GROUP BY code_room
6 ORDER BY room_count DESC
7 LIMIT 1;

```

Data Output   Сообщения   Notifications



|   | code_room<br>character varying (50) | room_count<br>bigint |
|---|-------------------------------------|----------------------|
| 1 | 12                                  | 2                    |

```

1 CREATE VIEW client_view AS
2 SELECT c.num_pus, c.name_client AS full_name, a.code_room
3 FROM client c
4 JOIN orders a ON c.num_pus = a.num_pus;

```

Data Output   Сообщения   Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 61 мсек.

Создаем запрос

```

1 EXPLAIN ANALYZE
2 select c.num_pus, o.beg from client c
3 join orders o ON c.num_pus=o.num_pus
4 group by c.num_pus, o.beg

```

Planning Time: 6.633 ms

Execution Time: 0.319 ms

ПОСЛЕ СОЗДАНИЯ ИНДЕКСА

```

1 CREATE INDEX pus_client ON client (num_pus);

```

Planning Time: 4.210 ms

Execution Time: 0.263 ms

```
EXPLAIN ANALYZE
select r.code_type_r, o.stat_pay from room r
join orders o ON r.code_rooms=o.code_room
group by r.code_type_r, o.stat_pay
```

Planning Time: 1.917 ms  
Execution Time: 0.258 ms

## ПОСЛЕ СОЗДАНИЯ ИНДЕКСА

```
1 CREATE INDEX type_rooms ON room (code_type_r, code_rooms);
```

Planning Time: 1.332 ms  
Execution Time: 0.213 ms

## INSERT, UPDATE, DELETE INSERT

```
1 INSERT INTO orders (code_room, ends)
2 VALUES ((SELECT code_type_r FROM room WHERE true), orders.ends);
```

```
1 UPDATE orders
2 SET code_room = 123
3 WHERE nm_pus = (
    SELECT num_pus
    FROM client
    WHERE name_client = 'Goni'
);
```

Data Output   Сообщения   Notifications

UPDATE 6

```
DELETE FROM client
WHERE EXISTS (
    SELECT 1
    FROM orders
    WHERE orders.nm_pus = '45' AND client.num_pus = orders.nm_pus
);
```

**Выводы:**

В процессе работы с pgAdmin я научился делать запросы на выборку и модификацию данных в базе данных PostgreSQL. Я освоил создание представлений, которые помогают упростить работу с данными, а также настройку индексов, что повышает производительность работы с базой данных. Эти знания помогут мне более эффективно работать с PostgreSQL и создавать более сложные запросы и представления для обработки большого объема данных. В целом, работа с pgAdmin позволила мне более полно использовать возможности PostgreSQL и повысить эффективность работы с базой данных.