

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИТМО»**

**Факультет инфокоммуникационных технологий**

**Дисциплина:**  
**«Проектирование и реализация баз данных»**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**  
**«Запросы на выборку и модификацию данных, представления и индексы в  
PostgreSQL»**

**Выполнил:**

студент группы К32392

Бочкарь Артём Артёмович

---

(подпись)

**Проверил(а):**

Говорова Марина Михайловна

---

(отметка о выполнении)

---

(подпись)

Санкт-Петербург  
2023 г.

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

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, 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.

## Выполнение

**Наименование БД:** hotel

**ERD диаграмма:**

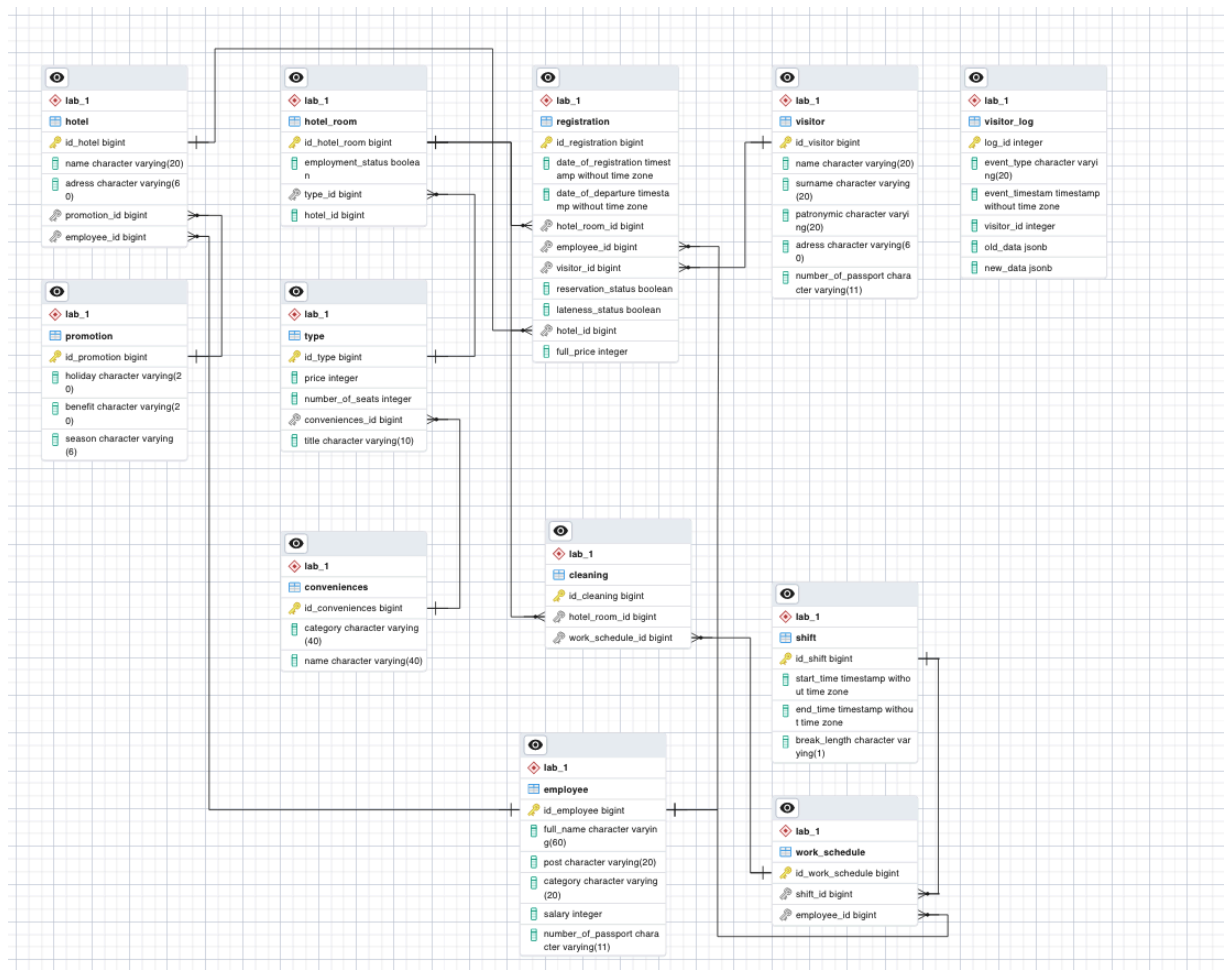


Рисунок 1 - ERD диаграмма

## Задание №1:

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

```
SELECT id_type
FROM lab_1.type
WHERE price < 200000 AND number_of_seats = 2
ORDER BY price DESC;
```

	id_type [PK] bigint
1	11
2	15
3	7

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

```
SELECT *
FROM lab_1.registration
WHERE date_of_departure >= CURRENT_DATE - INTERVAL '2 weeks';
```

	id_registration [PK] bigint	date_of_registration timestamp without time zone	date_of_departure timestamp without time zone	hotel_room_id bigint	employee_id bigint	visitor_id bigint	reservation_status boolean	lateness_status boolean
1	14	2023-05-16 20:15:12	2023-07-10 13:13:45	7	1	2	true	false
2	15	2023-05-17 11:10:09	2023-07-18 10:32:11	8	3	5	false	false


3. Чему равен общий суточный доход каждого отеля за последний месяц?

```
SELECT hotel.name, SUM(full_price) AS total_income
FROM lab_1.registration
JOIN lab_1.hotel ON registration.hotel_id = hotel.id_hotel
WHERE date_of_registration >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY hotel.name;
```

	name character varying (20)	total_income bigint
1	Шишка	24000
2	Гранд	259490
3	Пальма	12600
4	Леон	69800
5	Оазис	9250

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

```
SELECT COUNT(employment_status) AS free_rooms  
FROM lab_1.hotel_room  
WHERE employment_status = false AND hotel_id = 1
```

	<b>free_rooms</b> bigint 
1	3

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

```
SELECT SUM(price) AS total_loss  
FROM lab_1.hotel_room  
JOIN lab_1.type ON type.id_type = hotel_room.type_id  
WHERE employment_status = false
```

	<b>total_loss</b> bigint 
1	575500

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




```
SELECT hotel.name, COUNT(*) AS free_rooms  
FROM lab_1.hotel  
JOIN lab_1.hotel_room ON hotel.id_hotel = hotel_room.hotel_id  
WHERE employment_status = false  
GROUP BY hotel.name  
ORDER BY free_rooms DESC  
LIMIT 1;
```

	<b>name</b> character varying (20) 	<b>free_rooms</b> bigint 
1	Гранд	3

## 7. Определить самый популярный тип номеров за последний год.

```
WITH max_room_count AS (  
  SELECT MAX(room_count) AS max_count  
  FROM (  
    SELECT type.id_type, COUNT(*) AS room_count  
    FROM lab_1.type  
    JOIN lab_1.hotel_room ON type.id_type = hotel_room.type_id  
    JOIN lab_1.registration ON registration.hotel_room_id = hotel_room.id_hotel_room  
    WHERE date_of_registration >= CURRENT_DATE - INTERVAL '1 year'  
    GROUP BY type.id_type  
  ) AS subquery  
)
```

```
SELECT subquery.id_type, subquery.title, subquery.room_count  
FROM (  
  SELECT t.id_type, t.title, COUNT(*) AS room_count  
  FROM lab_1.type AS t  
  JOIN lab_1.hotel_room AS hr ON t.id_type = hr.type_id  
  JOIN lab_1.registration AS reg ON reg.hotel_room_id = hr.id_hotel_room  
  WHERE reg.date_of_registration >= CURRENT_DATE - INTERVAL '1 year'  
  GROUP BY t.id_type, t.title  
) AS subquery  
JOIN max_room_count AS mrc ON subquery.room_count = mrc.max_count;
```

	id_type [PK] bigint 	title character varying (10) 	room_count bigint 
1	8	Классик	4
2	6	Президент	4

## Задание №2:

### 1. Для турагентов (поиск свободных номеров в отелях).

```
CREATE VIEW search_available_rooms AS
SELECT id_hotel_room, title, hotel.name AS hotel_name, number_of_seats, conveniences.name
AS conveniences_name, price
FROM lab_1.hotel_room
JOIN lab_1.type ON hotel_room.type_id = type.id_type
JOIN lab_1.hotel ON hotel.id_hotel = hotel_room.hotel_id
JOIN lab_1.conveniences ON conveniences.id_conveniences = type.conveniences_id
WHERE employment_status = false;
```

CREATE VIEW

Запрос завершён успешно, время выполнения: 59 мсес.

### 2. Для владельца компании (информация о доходах каждого отеля в сети за прошедший месяц).

```
CREATE VIEW hotels_month_income AS
SELECT hotel.name, SUM(full_price) AS total_income
FROM lab_1.registration
JOIN lab_1.hotel ON registration.hotel_id = hotel.id_hotel
WHERE date_of_registration >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY hotel.name;
```

---

CREATE VIEW

Запрос завершён успешно, время выполнения: 63 мсес.

## Задание №3:

### 1. Регистрация нового посетителя

```
INSERT INTO lab_1.registration (date_of_registration, date_of_departure, hotel_room_id,
employee_id, visitor_id, reservation_status, lateness_status, hotel_id, full_price)
VALUES (
    check_in + INTERVAL '1 hour',
    check_in + duration - 1 + INTERVAL '1 hour',
    (
        SELECT id_hotel_room
        FROM lab_1.hotel_room
        JOIN lab_1.type ON hotel_room.type_id = type.id_type
        WHERE number_of_seats = 2
        AND employment_status = FALSE
```

```

        AND hotel_room.id_hotel_room NOT IN (
            SELECT registration.hotel_room_id
            FROM lab_1.registration
            WHERE date_of_registration <= check_in + duration - 1
            AND date_of_departure >= check_in
        )
    LIMIT 1
);

```

2. Обновление статуса занятости номера исходя из таблицы с регистрацией (если номер зарегистрирован на посетителя и посетитель ещё не выехал из него, то статус ставится true)

```

UPDATE lab_1.hotel_room
SET employment_status = true
WHERE EXISTS (
    SELECT *
    FROM lab_1.registration
    WHERE registration.hotel_room_id = hotel_room.id_hotel_room AND
    registration.date_of_departure IS NULL
)

```

До:

	id_hotel_room [PK] bigint	employment_status boolean	type_id bigint	hotel_id bigint
1	6	false	6	1
2	7	false	7	1
3	8	false	8	1
4	9	false	9	1
5	10	false	10	1
6	11	false	11	2
7	12	false	12	3
8	13	false	13	4

После:

	id_hotel_room [PK] bigint	employment_status boolean	type_id bigint	hotel_id bigint
1	6	true	6	1
2	7	true	7	1
3	8	false	8	1
4	9	false	9	1
5	10	false	10	1
6	11	false	11	2
7	12	true	12	3
8	13	false	13	4

### 3. Удаление сотрудника, который ничем не занимается

```
DELETE FROM lab_1.employee
WHERE NOT EXISTS (
    SELECT *
    FROM lab_1.registration, lab_1.hotel, lab_1.work_schedule
    WHERE registration.employee_id = employee.id_employee
    AND hotel.employee_id = employee.id_employee
    AND work_schedule.employee_id = employee.id_employee
)
```

До:

	id_employee [PK] bigint	full_name character varying (60)	post character varying (20)	category character varying (20)	salary integer	number_of_passport character varying (11)
1	1	Андреев Артем Сергеевич	Администратор	Первая	40000	4015 535678
2	2	Павлов Кирилл Андреевич	Администратор	Вторая	50000	4012 536587
3	3	Задохин Александр Антонович	Повар	Третья	80000	4013 125689
4	4	Уколов Георгий Владимирович	Беллбой	Первая	25000	4010 908045
5	5	Зайцев Олег Андреевич	Уборщик	Вторая	20000	4016 341223

После:

	id_employee [PK] bigint	full_name character varying (60)	post character varying (20)	category character varying (20)	salary integer	number_of_passport character varying (11)
1	1	Андреев Артем Сергеевич	Администратор	Первая	40000	4015 535678
2	2	Павлов Кирилл Андреевич	Администратор	Вторая	50000	4012 536587
3	3	Задохин Александр Антонович	Повар	Третья	80000	4013 125689
4	4	Уколов Георгий Владимирович	Беллбой	Первая	25000	4010 908045

## Задание №4:

```
1.
EXPLAIN ANALYZE
SELECT SUM(price) AS total_loss
FROM lab_1.hotel_room h
```



```
JOIN lab_1.type t ON t.id_type = h.type_id  
WHERE employment_status = false;
```

Planning Time: 0.206 ms

Execution Time: 0.139 ms

```
CREATE INDEX idx_price ON lab_1.type (price);
```

Planning Time: 0.125 ms

Execution Time: 0.070 ms

```
2.  
EXPLAIN ANALYZE  
SELECT *  
FROM lab_1.registration  
WHERE ((date_of_registration, date_of_departure) OVERLAPS ('2023-07-01'::DATE,  
'2023-07-31'::DATE))  
AND (full_price BETWEEN 1000 AND 10000);
```

Planning Time: 0.227 ms

Execution Time: 0.032 ms

```
CREATE INDEX dates_and_full_prices ON lab_1.registration (date_of_registration,  
date_of_departure, full_price);
```

Planning Time: 0.068 ms

Execution Time: 0.022 ms

## 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.