**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 K32392

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:**

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

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;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

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';



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;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

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

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

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;

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

7. Определить самый популярный тип номеров за последний год.

SELECT type.title, 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

GROUP BY type.title

ORDER BY room\_count DESC

LIMIT 1;

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

**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;

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

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;

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

**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:

Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

После:

Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

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:

Изображение выглядит как текст, чек, снимок экрана, Шрифт

Автоматически созданное описание

После:

Изображение выглядит как текст, Шрифт, снимок экрана, чек

Автоматически созданное описание

**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;

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

CREATE INDEX idx\_price ON lab\_1.type (price);

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

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);

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

CREATE INDEX dates\_and\_full\_prices ON lab\_1.registration (date\_of\_registration, date\_of\_departure, full\_price);

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

**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.