

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

ОТЧЕТ

по Лабораторной работе № 4

**«Запросы на выборку и модификацию данных. Представления. Работа с
индексами»**

по дисциплине «Проектирование и реализация баз данных»

Обучающийся Корольков Артем Алексеевич

Факультет прикладной информатики

Группа K3240

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург

2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Выполнение	3
1.1 Название создаваемой базы данных	3
1.2 Схема логической модели базы данных, сгенерированная в Generate ERD	4
1.3 Запросы к базе данных	4
1.4 Запросы на модификацию данных.	11
1.5 Индексы	13
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

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

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, 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.

1 Выполнение

1.1 Название создаваемой базы данных

Название базы данных – «Отель».

Изучение предметной области:

База данных разрабатывается для сети отелей, расположенных в различных городах.

Основная задача системы — хранение данных об отелях и управление информацией о бронировании, регистрации и проживании гостей.

Система должна поддерживать актуальные данные о состоянии номеров, которые могут быть свободны, забронированы или заняты. При

заезде гости регистрируются, и информация о регистрации хранится в системе на протяжении года, после чего данные архивируются.

Каждому номеру ежедневно присваивается статус уборки: "убран" или "не убран". Горничные работают по графику, который обновляется ежедневно для обеспечения своевременной уборки всех номеров.

1.2 Схема логической модели базы данных, сгенерированная в Generate ERD

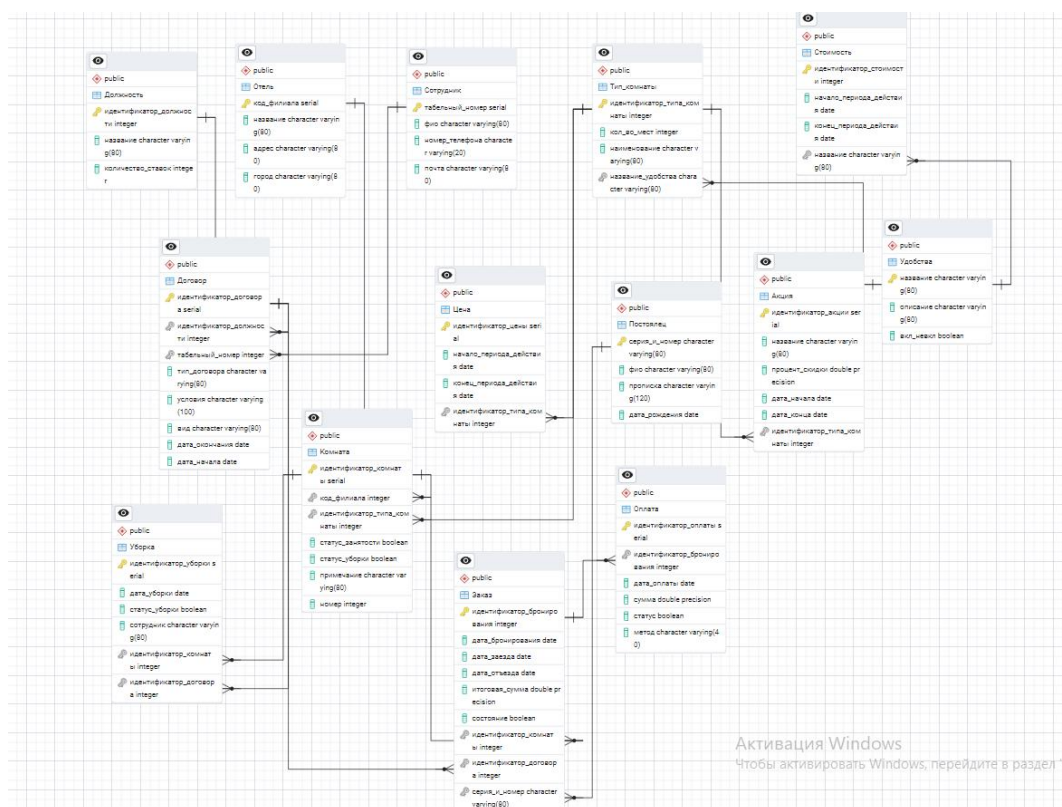


Рисунок 1 – Схема логической модели базы данных, сгенерированная в Generate ERD [2]

1.3 Запросы к базе данных

Задание 2. Создайте запросы:

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

```

SELECT
    r.room_id,
    h.name AS hotel_name,
    r.number AS room_number,
    rt.name AS room_type,
    rt.base_price AS price,
    (SELECT STRING_AGG(a.name, ', ')
     FROM Amenity a
     JOIN Room_type_amenity rta ON a.amenity_id = rta.amenity_id
     WHERE rta.room_type_id = rt.room_type_id) AS amenities
FROM
    Room r
JOIN Hotel h ON r.branch_code = h.branch_code
JOIN Room_type rt ON r.room_type_id = rt.room_type_id
WHERE
    rt.beds_count = 2
    AND rt.base_price < 200000
    AND r.occupied_status = false
ORDER BY
    rt.base_price DESC;

```

	room_id integer	hotel_name character varying (80)	room_number integer	room_type character varying (80)	price numeric (10,2)	amenities text
1	4	Гранд Отель	301	Люкс	12000.00	Wi-Fi, Мини-бар, Кондиционер, Телевизор, Сейф
2	6	Морской Бриз	202	Полулюкс	8000.00	Wi-Fi, Мини-бар, Кондиционер, Телевизор, Сейф
3	7	Лесная Сказка	103	Стандарт 2-местный	5000.00	Wi-Fi, Кондиционер, Телевизор
4	1	Гранд Отель	101	Стандарт 2-местный	5000.00	Wi-Fi, Кондиционер, Телевизор
5	2	Гранд Отель	102	Стандарт 2-местный	5000.00	Wi-Fi, Кондиционер, Телевизор
6	13	Лесная Сказка	107	Комфорт	4000.00	[null]
7	10	Гранд Отель	104	Комфорт	4000.00	[null]
8	9	Гранд Отель	103	Эконом	2500.00	[null]
9	11	Морской Бриз	105	Эконом	2500.00	[null]

Рисунок 2 - Скриншот выполнения запроса 1 с результатом

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

```

SELECT
    g.full_name AS guest_name,
    h.name AS hotel_name,
    r.number AS room_number,
    b.check_in_date,
    b.check_out_date,
    (b.check_out_date - b.check_in_date) AS nights_stayed
FROM
    Booking b
JOIN
    Guest g ON b.passport_number = g.passport_number
JOIN
    Room r ON b.room_id = r.room_id
JOIN
    Hotel h ON r.branch_code = h.branch_code
WHERE
    b.check_out_date BETWEEN CURRENT_DATE - INTERVAL '14 days' AND CURRENT_DATE
    AND b.status = true
ORDER BY
    b.check_out_date DESC;

```

	guest_name character varying (80)	hotel_name character varying (80)	room_number integer	check_in_date date	check_out_date date	nights_stayed integer
1	Федоров Дмитрий Сергеевич	Гранд Отель	301	2025-04-17	2025-04-24	7
2	Ковалева Ольга Игоревна	Гранд Отель	102	2025-04-13	2025-04-20	7
3	Смирнов Александр Петрович	Гранд Отель	101	2025-04-11	2025-04-17	6

Рисунок 3 - Скриншот выполнения запроса 2 с результатом

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

```

SELECT
    h.name AS hotel_name,
    h.city,
    SUM(p.amount) AS total_income
FROM
    Payment p
JOIN
    Booking b ON p.booking_id = b.booking_id
JOIN
    Room r ON b.room_id = r.room_id
JOIN
    Hotel h ON r.branch_code = h.branch_code
WHERE
    p.payment_date BETWEEN CURRENT_DATE - INTERVAL '1 month' AND CURRENT_DATE
    AND p.status = true
GROUP BY
    h.name, h.city
ORDER BY
    total_income DESC;

```

	hotel_name character varying (80)	city character varying (80)	total_income double precision
1	Гранд Отель	Москва	40000
2	Лесная Сказка	Казань	32000
3	Морской Бриз	Сочи	18000

Рисунок 4 - Скриншот выполнения запроса 3 с результатом

4. Составить список свободных номеров заданного отеля на текущий день.

```
-- 4. Свободные номера в заданном отеле на текущий день
SELECT
    r.room_id,
    r.number AS room_number,
    rt.name AS room_type
FROM
    Room r
JOIN Room_type rt ON r.room_type_id = rt.room_type_id
WHERE
    r.branch_code = 1 -- ID отеля
    AND r.room_id NOT IN (
        SELECT room_id
        FROM Booking
        WHERE CURRENT_DATE BETWEEN check_in_date AND check_out_date
        AND status = true
    )
    AND r.occupied_status = false;
```

	room_id integer	room_number integer	room_type character varying (80)
1	9	103	Эконом
2	10	104	Комфорт
3	1	101	Стандарт 2-местный
4	2	102	Стандарт 2-местный
5	4	301	Люкс

Рисунок 5 - Скриншот выполнения запроса 4 с результатом

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


```

SELECT
    SUM(p.amount) AS total_loss
FROM
    Room r
JOIN
    Room_type rt ON r.room_type_id = rt.room_type_id
JOIN
    Price p ON rt.room_type_id = p.room_type_id
WHERE
    r.occupied_status = false
    AND CURRENT_DATE BETWEEN p.period_start AND p.period_end;

```

	total_loss double precision
1	16500

Рисунок 6 - Скриншот выполнения запроса 5 с результатом

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

```

SELECT
    h.name AS hotel_name,
    h.city,
    COUNT(r.room_id) AS unoccupied_rooms_count
FROM
    Room r
JOIN
    Hotel h ON r.branch_code = h.branch_code
WHERE
    r.occupied_status = false
GROUP BY
    h.name, h.city
ORDER BY
    unoccupied_rooms_count DESC
LIMIT 1;

```

	hotel_name character varying (80)	city character varying (80)	unoccupied_rooms_count bigint
1	Гранд Отель	Москва	5

Рисунок 7 - Скриншот выполнения запроса 6 с результатом

7. Определить самый популярный тип номеров за последний год.


```

SELECT
    rt.name AS room_type,
    COUNT(b.booking_id) AS bookings_count
FROM
    Booking b
JOIN
    Room r ON b.room_id = r.room_id
JOIN
    Room_type rt ON r.room_type_id = rt.room_type_id
WHERE
    b.check_in_date BETWEEN CURRENT_DATE - INTERVAL '1 year' AND CURRENT_DATE
GROUP BY
    rt.name
ORDER BY
    bookings_count DESC
LIMIT 1;

```

	room_type character varying (80) 🔒	bookings_count bigint 🔒
1	Стандарт 2-местный	2

Рисунок 8 - Скриншот выполнения запроса 7 с результатом

Задание 3. Создайте представления:

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

```

CREATE VIEW available_rooms_view AS
SELECT
    h.name AS hotel_name,
    h.city,
    r.number AS room_number,
    rt.name AS room_type,
    rt.beds_count,
    p.amount AS price_per_night
FROM
    Room r
JOIN
    Hotel h ON r.branch_code = h.branch_code
JOIN
    Room_type rt ON r.room_type_id = rt.room_type_id
JOIN
    Price p ON rt.room_type_id = p.room_type_id
WHERE
    r.occupied_status = false
    AND r.cleaning_status = true
    AND CURRENT_DATE BETWEEN p.period_start AND p.period_end
ORDER BY
    h.city, h.name, p.amount;

```

	hotel_name character varying (80) 🔒	city character varying (80) 🔒	room_number integer 🔒	room_type character varying (80) 🔒	beds_count integer 🔒	price_per_night double precision 🔒
1	Лесная Сказка	Казань	107	Комфорт	2	4000
2	Гранд Отель	Москва	103	Эконом	2	2500
3	Гранд Отель	Москва	104	Комфорт	2	4000
4	Морской Бриз	Сочи	105	Эконом	2	2500
5	Морской Бриз	Сочи	106	Бизнес	1	3500

Рисунок 9 - Скриншот выполнения представления 1 с результатом

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

```

CREATE VIEW hotel_income_last_month_view AS
SELECT
    h.name AS hotel_name,
    h.city,
    SUM(p.amount) AS total_income,
    COUNT(b.booking_id) AS bookings_count,
    AVG(p.amount) AS average_booking_amount
FROM
    Payment p
JOIN
    Booking b ON p.booking_id = b.booking_id
JOIN
    Room r ON b.room_id = r.room_id
JOIN
    Hotel h ON r.branch_code = h.branch_code
WHERE
    p.payment_date BETWEEN CURRENT_DATE - INTERVAL '1 month' AND CURRENT_DATE
    AND p.status = true
GROUP BY
    h.name, h.city
ORDER BY
    total_income DESC;

```

	hotel_name character varying (80)	city character varying (80)	total_income double precision	bookings_count bigint	average_booking_amount double precision
1	Гранд Отель	Москва	40000	2	20000
2	Лесная Сказка	Казань	32000	1	32000
3	Морской Бриз	Сочи	18000	1	18000

Рисунок 10 - Скриншот выполнения представления 2 с результатом

1.4 Запросы на модификацию данных.

INSERT: Добавление новой комнаты в отель в Москве

```

INSERT INTO Room (branch_code, room_type_id, number, occupied_status, cleaning_status)
VALUES (
    (SELECT branch_code FROM Hotel WHERE city = 'Москва' LIMIT 1),
    (SELECT room_type_id FROM Room_type WHERE name = 'Стандарт' LIMIT 1),
    101, false, true
);

```

	room_id [PK] integer	branch_code integer	room_type_id integer	number integer	occupied_status boolean	cleaning_status boolean	note character varying (80)
1	1001	1	1	101	false	true	[null]

Рисунок 11 - Скриншот выполнения модифицированного запроса 1 с результатом

UPDATE: Увеличение цены номеров с Wi-Fi

```
UPDATE Room_type
SET base_price = base_price * 1.1
WHERE room_type_id IN (
    SELECT room_type_id FROM Room_type_amenity
    WHERE amenity_id = (SELECT amenity_id FROM Amenity WHERE name = 'Wi-Fi')
);
```

	room_type_id [PK] integer	name character varying (80)	base_price numeric (10,2)
1	1	Стандарт	3850.00
2	2	Стандарт 2-местный	5500.00
3	3	Полулюкс	8800.00
4	4	Люкс	13200.00
5	5	Апартаменты	22000.00
6	6	Эконом	2750.00
7	7	Комфорт	4400.00
8	8	Бизнес	3850.00

Рисунок 12 - Скриншот выполнения модифицированного запроса 2 с результатом

DELETE: Удаление неиспользуемых удобств

```
SELECT booking_id, passport_number
FROM Booking
WHERE passport_number IN (
    SELECT passport_number
    FROM Guest
    WHERE birth_date < '1960-01-01'
);
```

	booking_id [PK] integer	passport_number character varying (80)
1	20	5016 497834
2	21	6521 561720
3	27	9800 386779
4	28	6978 144542
5	42	3119 263964
6	52	7758 232708
7	76	8285 741708
8	80	8780 511269

Использование запроса

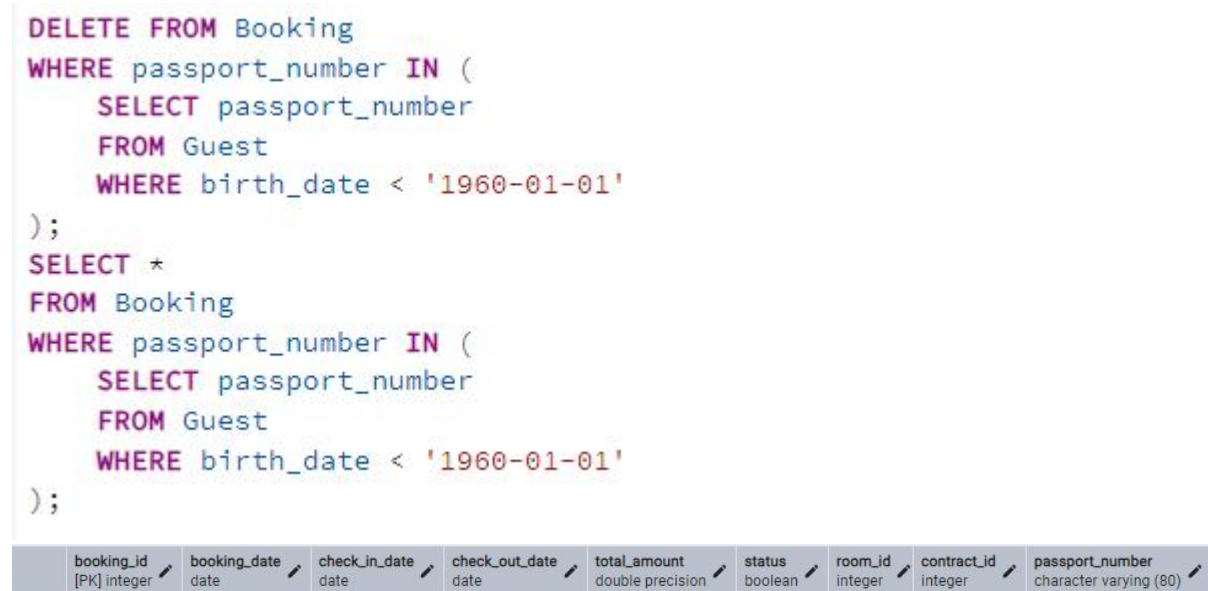


Рисунок 13 - Скриншот выполнения модифицированного запроса 3 с результатом

1.5 Индексы

Запросы

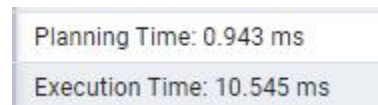
```
-- Запрос 1: Поиск бронирований по дате и статусу с соединением нескольких таблиц
EXPLAIN ANALYZE
SELECT b.booking_id, g.full_name, r.number, rt.name as room_type, h.name as hotel_name
FROM Booking b
JOIN Guest g ON b.passport_number = g.passport_number
JOIN Room r ON b.room_id = r.room_id
JOIN Room_type rt ON r.room_type_id = rt.room_type_id
JOIN Hotel h ON r.branch_code = h.branch_code
WHERE b.check_in_date BETWEEN '2023-01-01' AND '2023-12-31'
AND b.status = true;

-- Запрос 2: Агрегация данных по оплатам с группировкой
EXPLAIN ANALYZE
SELECT p.method, COUNT(*) as payment_count, SUM(p.amount) as total_amount
FROM Payment p
JOIN Booking b ON p.booking_id = b.booking_id
WHERE p.payment_date BETWEEN '2023-01-01' AND '2023-12-31'
GROUP BY p.method;

-- Запрос 3: Поиск комнат с определенными удобствами
EXPLAIN ANALYZE
SELECT r.number, rt.name as room_type, COUNT(a.amenity_id) as amenities_count
FROM Room r
JOIN Room_type rt ON r.room_type_id = rt.room_type_id
JOIN Room_type_amenity rta ON rt.room_type_id = rta.room_type_id
JOIN Amenity a ON rta.amenity_id = a.amenity_id
WHERE a.enabled = true
GROUP BY r.number, rt.name
HAVING COUNT(a.amenity_id) > 2;
```

Рисунок 14 - Скриншот выполнения запросов для проверки индексов

Время без индексов



Planning Time: 0.943 ms
Execution Time: 10.545 ms

Рисунок 15 - Скриншот времени выполнения запросов без индексов

Индексы

```
-- Индекс для ускорения поиска бронирований по дате заезда
CREATE INDEX idx_booking_check_in_date ON Booking(check_in_date);

-- Индекс для статуса бронирования
CREATE INDEX idx_booking_status ON Booking(status);

-- Составной индекс для оплат по дате и методу
CREATE INDEX idx_payment_date_method ON Payment(payment_date, method);

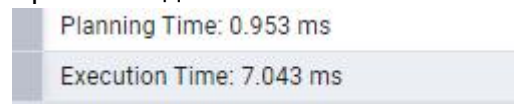
-- Индекс для соединения таблиц Payment и Booking
CREATE INDEX idx_payment_booking_id ON Payment(booking_id);

-- Индекс для удобств
CREATE INDEX idx_amenity_enabled ON Amenity(enabled);

-- Индекс для соединения Room_type и Room_type_amenity
CREATE INDEX idx_rta_room_type_id ON Room_type_amenity(room_type_id);
```

Рисунок 16 - Скриншот выполнения запросов для проверки индексов с индексами

Время с индексами



Planning Time: 0.953 ms
Execution Time: 7.043 ms

Рисунок 17 - Скриншот времени выполнения запросов с индексами

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была достигнута цель, заключающаяся в приобретении практических навыков работы с запросами на выборку и модификацию данных, созданием представлений, использованием подзапросов, а также применением индексов в базе данных PostgreSQL. Работа выполнялась в компьютерном классе с использованием программного обеспечения PostgreSQL и pgAdmin 4.

В рамках задания были созданы запросы и представления для выборки данных на основе индивидуального задания из предыдущей лабораторной работы. Были составлены три запроса на модификацию данных: один запрос на добавление новых записей с использованием подзапроса, один на обновление существующих данных с подзапросом и один на удаление записей с применением вложенного запроса. Также была изучена возможность построения графического представления запросов в pgAdmin 4 и просмотрена история выполненных операций для анализа их выполнения.

Дополнительно были созданы простой и составной индексы для двух различных запросов. Сравнение времени выполнения запросов проводилось с помощью команды EXPLAIN, что позволило наглядно увидеть разницу в производительности между запросами без индексов и с применением индексации. В результате индексирование показало значительное сокращение времени выполнения некоторых операций, особенно при выборке большого объема данных.

Лабораторная работа позволила закрепить навыки работы с подзапросами, представлениями и индексами, а также углубить понимание оптимизации запросов в реляционных базах данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация PostgreSQL [Электронный ресурс] // Официальный сайт PostgreSQL. 1996-2025. URL: <https://www.postgresql.org/docs/13/index.html> (дата обращения: 20.03.2025).
2. Документация pgAdmin 4 PostgreSQL [Электронный ресурс] // Официальный сайт pgAdmin. URL: <https://www.pgadmin.org/docs/pgadmin4/latest/> (дата обращения: 21.03.2025)